SIAM J. DISCRETE MATH.
Vol. 10, No. 1, pp. 1–17, February 1997

001

STEINER TREES FOR TERMINALS CONSTRAINED TO CURVES*

J. H. RUBINSTEIN[†], D. A. THOMAS[‡], AND N. C. WORMALD[†]

Abstract. We give a polynomial time algorithm for solving the Euclidean Steiner tree problem when the terminals are constrained to lie on a fixed finite set of disjoint finite-length compact simple smooth curves. The problem is known to be NP-hard in general. We also show it to be NP-hard if the terminals lie on two parallel infinite lines or on a bent line segment provided the bend has an angle of less than 120°.

Key words. Steiner trees, polynomial algorithms

AMS subject classifications. 05C05, 90B85, 68R10

PII. S0895480192241190

1. Introduction.

Suppose we are given a finite set of n points, called *terminals*, in the plane. A *Steiner tree* is a tree connecting all the given points, and the *Steiner problem* is to find a Steiner tree of shortest total length, called a *minimal Steiner tree.* This has applications in areas such as the design of pipelines, drainage systems, and wiring. The tree can have extra (nonterminal) vertices, called *Steiner* vertices. In solving the Steiner problem, one has to determine not only the underlying graph of the tree (called its topology) but also the precise positions of the Steiner vertices. As shown by Hwang [4], the latter problem does not cause great difficulties, since a minimal Steiner tree can be determined in linear time given its topology.

The following is well known (see Gilbert and Pollak [3]).

LEMMA 1. *In a minimal Steiner tree, Steiner vertices all have degree* 3*, and the three incident edges meet at* 120° *angles.*

We define a *Steiner component* of a tree to be a maximal subtree, all of whose nonleaves are Steiner vertices. We may deduce that in a given Steiner component, the edges are oriented in only three different directions.

However, making use of the number and complexity of the possible topologies, Garey, Graham, and Johnson [1] showed that the Steiner problem is NP-complete. Therefore it is expected that there is no polynomial time algorithm to find a minimal Steiner tree.

In 1987, R. Graham suggested trying to solve the Steiner problem in the case where the terminals all lie on a circle. We have been considering a more general version of this question: for any problem whose input is a set of points in the plane, we can restrict ourselves to the case where all given points lie on a fixed set of smooth curves of finite total length in the plane. We call this a *G-constrained* problem. So, for the G-constrained Steiner problem, the terminals must lie on G. In this paper we prove the following theorem. (Note that a compact curve must have finite length.)

THEOREM 1. *If G is a fixed finite set of disjoint compact simple smooth curves in the plane, then there is a polynomial time algorithm for the G-constrained Steiner tree problem.*

* Received by the editors December 7, 1992; accepted for publication (in revised form) June 30, 1995.

http://www.siam.org/journals/sidma/10-1/24119.html

[†] Department of Mathematics, University of Melbourne, Parkville, VIC 3052, Australia (rubin@mundoe.maths.mu.oz.au, nick@mundoe.maths.mu.oz.au).

[‡] Department of Electrical and Electronic Engineering, University of Melbourne, Parkville, VIC 3052, Australia (doreen@mullian.ee.mu.oz.au).

Note. We ignore the complexity of the presentation of G and the terminals and assume that elementary geometric constructions can be performed in bounded time. So, in particular, by "polynomial time" we mean time polynomial in the number n of terminals.

For example, the restriction of the Steiner problem to terminals which lie on the smooth curves in Figure 1 has a polynomial time algorithm.

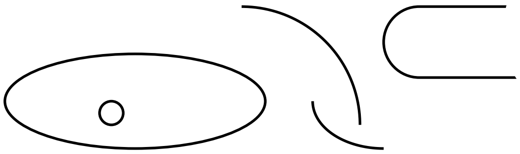


FIG. 1. *Disjoint smooth curves.*

We prove Theorem 1 in the following section. The degree of the polynomial involved in our proof is in general quite high, and so we do not attempt to determine it precisely. It is highly dependent on the geometry of G. In fact the running time of the algorithm depends on the length and the maximum curvature of G as well as the proximity of the curves or sections of a curve.

We also show in section 3 that the smoothness condition cannot be dropped totally from Theorem 1, by showing that the special case of the Steiner problem, in which the terminals are constrained to lie on two line segments meeting at an angle which is less than 120°, is NP-hard. This is much stronger than the main result of [1] that the Steiner problem is NP-hard, but we use an argument different from and in many ways simpler than the argument there. In particular, our result implies that the Steiner problem remains NP-hard even when restricted to sets of terminals lying in convex position. As we show, the problem is even NP-hard when the terminals are restricted to lying on two parallel lines. Our first proof assumes that infinite precision real arithmetic takes finite time. Due to the intricacies of handling infinite precision and the fact that our proof does not really require infinite precision, we also prove that the corresponding discretized versions of these problems are NP-complete. From the argument in [1], it follows that the Steiner problem is NP-hard in the strong sense, whereas the argument we give here for the restricted problems does not provide this conclusion. However, in both [1] and here, the discretized problems have not been shown NP-complete in the strong sense. In fact, Provan [5] gave a full polynomial time approximation scheme for the case where terminals are in convex position. It follows that the discretization of this problem cannot be NP-complete in the strong sense unless P = NP.

It follows from our results that the G-constrained Steiner problem is still NP-hard when G is the curve in Figure 2. However, the proof of Theorem 1 can be adapted to give a polynomial time approximation scheme when G is a fixed finite set of smooth curves of finite total length (see Figure 3); i.e., we get an algorithm constructing a tree whose length is within δ of that of the minimal Steiner tree for any prescribed $\delta > 0$.

The topologies occurring in the NP-hardness proofs in section 3 cannot cause a problem if all angles in G are greater than 120°, so we do not hesitate to conjecture that Theorem 1 can be strengthened as follows.

Conjecture. If G is a fixed collection of compact smooth curves of finite total

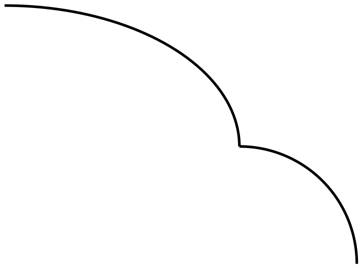


FIG. 2. *Curves meeting at an angle.*

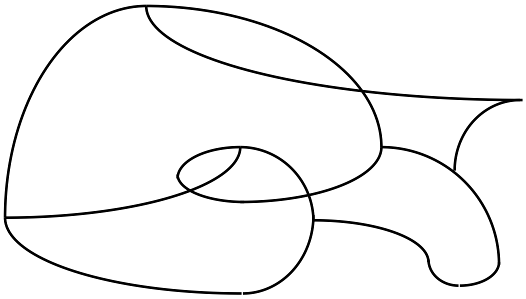


FIG. 3. *Smooth curves meeting and crossing.*

length, then there is a polynomial time algorithm for the G-constrained minimal Steiner tree problem if the minimum angle formed by the meeting of any two curves is strictly greater than 120°.

In view of the results in section 3, the truth of the conjecture would imply that only those G with a minimum angle of exactly 120° would be of undetermined complexity. For these, the second-order behavior of the curves near such angles would undoubtedly be the determining factor.

2. Proof of Theorem 1: An algorithm. Suppose G is a fixed finite set of disjoint smooth (i.e., continuously differentiable) compact curves of finite total length. Let L denote an upper bound on the length of a curve containing G, for instance the total length of the curves in G plus the sum of the distances between the components of G. Then L is an upper bound on the length of a minimal Steiner tree in this problem.

In view of the smoothness of the curves of G, we can choose, for any $\delta > 0$, a covering of G made up of a finite set $\mathcal{Q}(\delta)$ of simply connected compact curves which overlap only on their endpoints with the properties that

$$G = \bigcup_{Q \in \mathcal{Q}(\delta)} Q$$

and such that all tangents to the curve of G at points in any fixed $Q \in \mathcal{Q}$ and in the two neighboring curves in $\mathcal{Q}$ have direction within δ of each other. (See Figure 4(a).)

Choosing δ small enough allows us to consider the curves of G to be approximately straight in each $Q \in \mathcal{Q}$ and the two neighboring curves. For some of our statements we assume that δ is sufficiently small, without explicitly stating so.

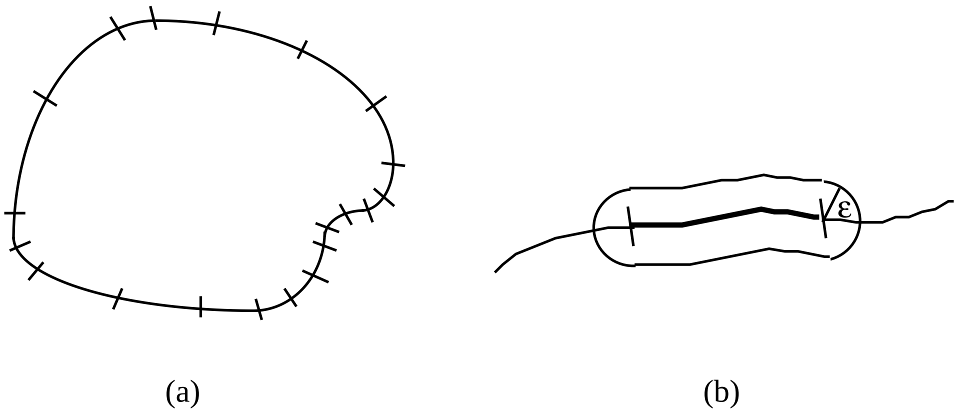


FIG. 4. *Capsules.*

Also, there is some minimal $\epsilon > 0$ for which the set of points at distance exactly ϵ from the curves is self-intersecting. We will fix ϵ to have a smaller positive value than this, as determined below.

Given $Q \in \mathcal{Q}(\delta)$, define the *capsule* $C(Q)$ to be

$$\{x \,:\, d(x, Q) \leq \epsilon\},$$

where d denotes Euclidean distance. (See Figure 4(b).) Note that $G \cap C(Q)$ is a simply connected curve by the choice of ϵ.

We define the *direction* of G in a capsule to be the direction of a tangent to G at some point in the capsule. We choose ϵ so small that the direction of G is within δ of the direction of all tangents to the curve of G at points within the capsule. This can be done by choosing it small enough to make sure that the parts of G contained in the capsule are within the neighboring elements of $\mathcal{Q}$. Despite this condition on ϵ, we are at liberty in our argument to make δ as small as we please and (if necessary by subdividing elements in $\mathcal{Q}$) to ensure that for each $Q \in \mathcal{Q}$ the length of $G \cap C(Q)$ is at most 100ϵ. (The constant 100 is chosen for convenience: any number greater than 3 would do.) We may henceforth regard the set of capsules as fixed. The determination of the capsules is a step of our algorithm which we will not describe explicitly. It needs to be done once only for any given set of curves, and so takes constant time independent of n. Thus, we assume the capsules have been determined before starting the algorithm.

Let S be a minimal Steiner tree for a given set of n terminals lying on G. For a capsule $C = C(Q)$, the set of edges of S which intersect C and are incident with Steiner vertices of S not necessarily in C induce a forest F. We next analyze this forest in detail. Note that we consider terminal-to-terminal edges last.

2.1. Bounding the number of paths to the boundary of the capsule. We define an *alternating path*, or *zigzag*, to be a path in S such that all internal vertices of the path are Steiner vertices, and such that at most two directions are used in the edges of the path. By Lemma 1, every zigzag is contained in a maximal zigzag with a terminal at each end. Zigzags are defined in F analogously, but in that case the maximal zigzags will terminate at vertices outside C.

If an edge of F is nearly parallel to G in C, an "alternating topology" can occur as in Figure 5. To be precise, we define an *alternating branch* of F to be a branch

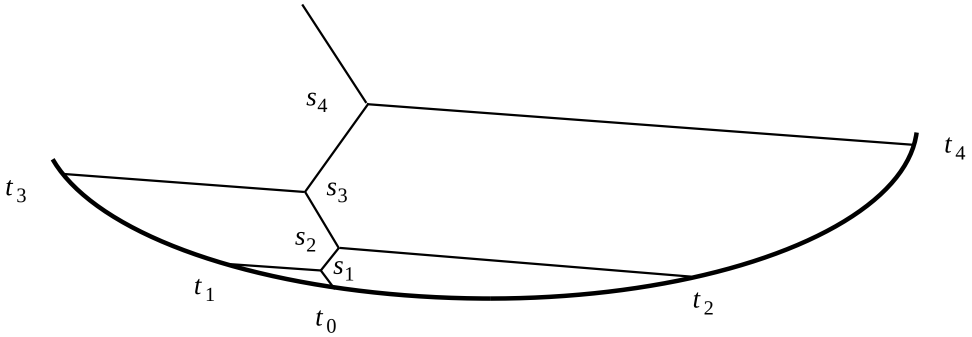


FIG. 5. *Alternating branch.*

of a Steiner component S_0 of S (i.e., a connected component of $S_0 - e$ for some edge e of S_0) contained in F, containing at least two Steiner vertices, and in which some zigzag using the direction of e contains all the Steiner vertices in the branch. Since an alternating branch of F is a subtree of a Steiner component of S, it has three directions occurring in its edges. It is easy to verify that in an alternating branch, one of the three directions is arbitrarily close to the direction of G in C. (In Figure 5, the angles between s_2t_2 and the tangent to G at t_2 and between s_1t_1 and the tangent to G at t_1 add to give the angle between the tangents to G at t_1 and t_2. This is less than δ.) The other two directions determine a maximal zigzag in the alternating branch containing the Steiner vertices, called the *zigzag of the branch.* This is the same zigzag as referred to in the definition if there are more than two Steiner vertices. For example, the zigzag of the branch in Figure 5 is t_0, s_1, s_2, s_3, s_4. A *cherry* of F is a branch of a Steiner component of S contained in F and containing just one Steiner vertex and two terminals. Each alternating branch contains a unique cherry.

We shall now choose a maximal zigzag from each terminal in Q in F but not in an alternating branch or cherry of F. First choose any edge incident with such a terminal u. Then the other end of this edge is a Steiner vertex v, because all edges in F are incident with at least one such vertex. For the second edge of the zigzag, choose another edge incident with v leading away from $G \cap C$, and then extend these two edges to a maximal zigzag in F. Note that this zigzag will be adjacent to some other Steiner vertex in F, as the branch containing the zigzag is not alternating. Next, from each cherry of F containing a terminal in Q, choose a maximal zigzag in F originating in the cherry and using no direction close to (i.e., within δ of) the direction of G in C. Note that for each alternating branch B of F containing such a cherry, this determines the maximal zigzag in F containing the zigzag of B. Denote the set of all the maximal zigzags chosen in either of these two ways by Z. Also denote the set of terminals on Q which are adjacent in S to terminals outside $C(Q)$ by Z'.

Clearly, no edge is in more than two maximal zigzags since once a choice of two directions is made then the zigzag is determined. A zigzag in Z with both ends in $G \cap C$ would necessarily use a direction close to that of G in C. This is because if no direction is close to G then the zigzag is chosen leading away from G at the first Steiner vertex so it clearly cannot end on G. Thus, by construction, such a zigzag must have been formed from a terminal not in an alternating branch or a cherry, as in the latter cases one end is not in the capsule C. But in view of the directions of edges in this zigzag, the terminal must be contained in an alternating branch or in a cherry, and so this is impossible.

We now consider the zigzags in Z. A zigzag in F with no end on Q must originate in a cherry spanning one of the ends of Q to one side of $G \cap C - Q$. There can be at most four of these zigzags, because no two cherries on the same side of $G \cap C$ can enclose intersecting segments of $G \cap C$. Thus in Z we have at least $|Z| - 4$ paths from Q to the boundary of C. Clearly no edge is in more than two maximal zigzags. Hence the total length of edges in F is at least $(|Z| - 4)\epsilon/2$. Each terminal in Z' also contributes at least ϵ to the length of $S - F$. Since S is a minimal Steiner tree, we now have

$$|Z \cup Z'| \leq \frac{2L}{\epsilon} + 4,$$

which is a constant independent of n, depending on the length and maximum curvature of G.

2.2. The algorithm. We need to consider the structure of F inside the capsules. With this aim, define $Y(C)$ to be the set whose elements are

(i) the maximal alternating branches in F which contain at least one terminal in Q;

(ii) the cherries in F not in alternating branches in F, which contain at least one terminal in Q;

(iii) the terminals in Q which are in F but not in any alternating branch or any cherry of F;

(iv) the terminals in Z'.

For elements of $Y(C)$ in (i)–(iii), we can associate a unique maximal zigzag. Since each alternating branch contains precisely one cherry, each element of $Y(C)$ of type (i) contains the beginning of just one maximal zigzag in $Z(C)$. For (ii), for each cherry we chose a unique maximal zigzag. For (iii) we associated a unique maximal zigzag with such a terminal, leading away from $G \cap C$. Thus $|Y(C)| = |Z \cup Z'|$ and is hence bounded. Thus, we consider the following algorithm for finding a minimal Steiner tree S on the given terminals. Let M denote the length of the shortest tree found so far in the algorithm. The algorithm grows the Steiner tree inward starting from the curve but leaves terminal-to-terminal edges until last. Recall that we are assuming that the capsules have already been determined.

0. Put $M = \infty$.

1. In each capsule C choose the terminals in $Y(C)$ and select which of these are in Z'. Also choose the sets of terminals in the alternating branches and cherries in $Y(C)$, and the remaining terminals adjacent to Steiner vertices, and the adjacency between the vertices within the alternating branches (which includes the arrangement of Steiner vertices in the alternating branches).

2. Choose the adjacencies of the subforest of S induced by all remaining edges incident with Steiner vertices. These edges may start at terminals and connect to Steiner vertices outside the capsules or may be part of S outside the capsules.

Repeat steps 3–5 for each possible choice made in steps 1 and 2.

3. Compute a minimal length set of edges to add to the forest in step 2 in order to complete the choice of adjacencies in S. These edges must be terminal to terminal.

4. Find a minimal tree S with these adjacencies.

5. If the length of S is less than M, set M equal to the length of S and put $S_0 = S$.

6. Output S_0 as the minimal Steiner tree.

2.3. Each step can be determined in polynomial time. Since there is a fixed number of capsules, the sum of $|Y(C)|$ over all C is bounded. Step 2 can be carried out in a bounded number of ways and in bounded time because it amounts to a choice of a forest whose leaves are elements in $Y(C)$, over all capsules C. Step 3 can be done in polynomial time by computing the distance of any Steiner component containing a Steiner vertex from any other such component by a direct edge from terminal to terminal. This gives a weighted graph whose vertices are these Steiner components, and all we need is a minimum weight spanning tree in this graph, which is computable in polynomial time using standard methods. Step 4 can be carried out in polynomial time by Hwang's result mentioned in the introduction.

The proof is completed by showing that, for sufficiently small δ, the alternatives for step 1 can be determined in time polynomial in n. A crucial part of this is to show that they are polynomial in number.

Note that the choice of terminals in cherries and alternating branches in a capsule will affect the valid alternatives in an adjacent capsule, as the capsules overlap. Nevertheless, all we have to do is show that there is a polynomial number of alternatives for each capsule C, since a choice for a capsule and its adjacent ones can be checked in polynomial time by first checking that the graph is a tree and then using Hwang [4].

Elements of $Y(C)$ of the second and third kind (cherries and isolated terminals) contain either two terminals or one terminal and so each can be chosen in time n^2. Thus it is only the maximal alternating branches that cause trouble. All we need to show is that each can be chosen in a polynomial number of ways. The difficulty is that they can contain an unbounded number of terminals.

The zigzag of a maximal alternating branch B in $Y(C)$ determines an ordering of the terminals $t_0, \dots, t_j$ and of the Steiner vertices $s_1, \dots, s_j$ in B, as in Figure 5. For $1 \leq i \leq j-2$, there is a constant c_1 such that

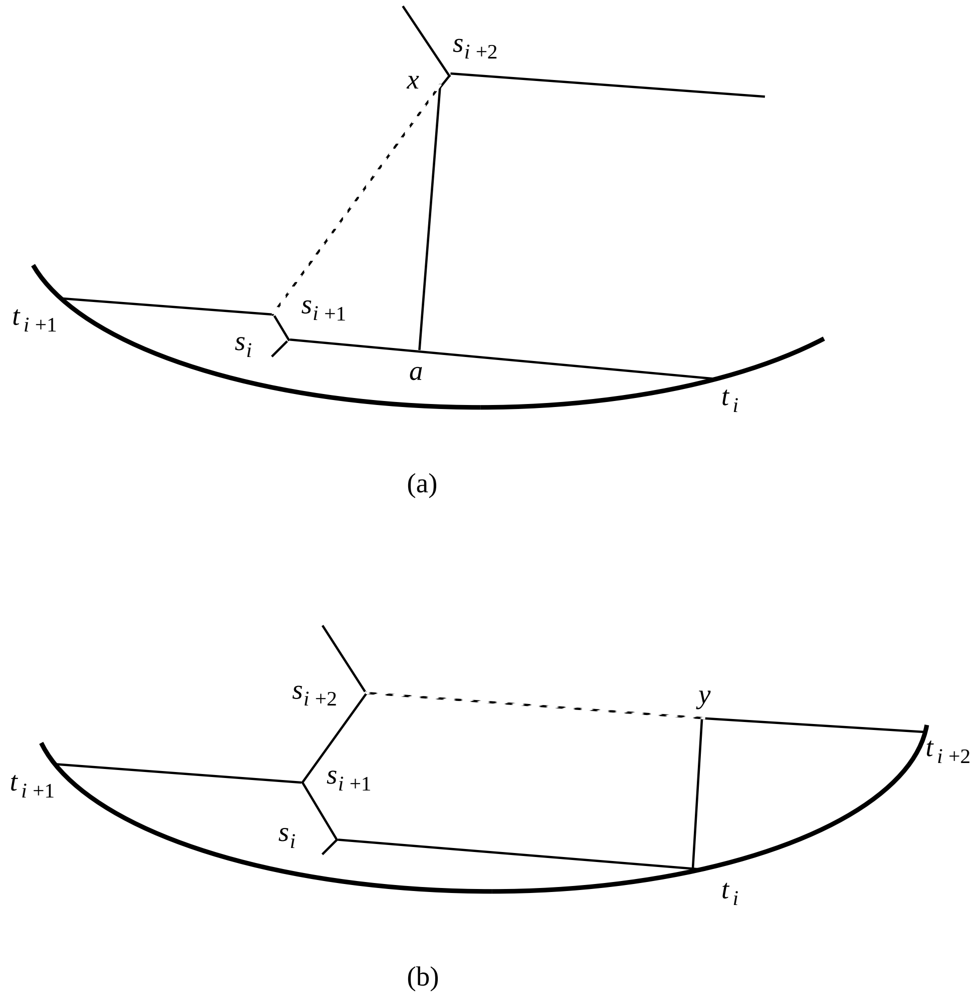
$$\frac{d(s_i, s_{i+1})}{d(s_i, t_i)} > c_1. \tag{1}$$

If this ratio is arbitrarily small there are two cases. In the first, if $d(s_{i+1}, s_{i+2})/d(s_i, s_{i+1})$ is large enough, then the tree S can be shortened by replacing the line from x to s_{i+1} by the line from x to a where x and a are appropriately chosen points between s_{i+1} and s_{i+2} and between s_i and t_i, respectively (see Figure 6(a)). In the second case if $d(s_{i+1}, s_{i+2})/d(s_i, s_{i+1})$ is small, S can be shortened by replacing the line from y to s_{i+2} by the line from t_i to y where y is the closest point to t_i on the line from s_{i+2} to t_{i+2} (see Figure 6(b)). A similar argument applies even if s_{i+1} is the first Steiner vertex s_1 in the alternating branch. In this case s_i is replaced by t_0.

Furthermore, since the line from s_i to s_{i+1} makes an angle close to 60° with the direction of G in C, we have for $1 \leq i \leq j-1$

$$\frac{d(s_{i+1}, t_{i+1})}{d(s_i, s_{i+1})} > \omega_1, \tag{2}$$

where ω_1 can be made as large as we like, without affecting c_1, by our choice of δ. In fact, in the triangle $s_{i+1}\ t_{i+1}\ s_i$, the angle at t_{i+1} can be made as small as necessary by the selection of δ.



(a)

(b)

FIG. 6. *Nonoptimal trees.*

This equation is also valid for $i = 0$ if we take $s_0 = t_0$. From (1) and (2),

$$\frac{d(s_{i+1}, t_{i+1})}{d(s_i, t_i)} > \omega_2, \qquad \frac{d(s_{i+1}, s_i)}{d(s_i, s_{i-1})} > \omega_2 \tag{3}$$

for an arbitrarily large constant ω_2, $1 \leq i \leq j - 2$.

These inequalities will be used to deduce two things: first, the terminals of two different alternating branches (on opposite sides of G) cannot intermingle along G very much; and second, when the section of G containing the terminals of the branch is decided, the choice of which terminals are to be included in the branch has a polynomial number of alternatives.

2.4. Intermingling does not occur. The outermost terminals along the curve of $G \cap C$ are t_j and t_{j-1}. We refer to the terminals $t_0, \ldots, t_{j-3}$ as the *inner* terminals of B. Note that t_{j-2} is *not* inner.

Suppose that some inner terminal t' of some other alternating branch B' lies between two inner terminals of B along the curve of $G \cap C$. Let d_0 denote $d(s_{j-3}, s_{j-2})$.

By (3), this is approximately equal to $d(t_0, s_{j-2})$, since the distances $d(t_0, s_0), \ldots, d(s_{j-4}, s_{j-3})$ are all negligible compared with $d(s_{j-3}, s_{j-2})$. So using (1) we have that

$$\frac{d(t_0, s_{j-2})}{d(t_0, t_{j-3})} > c_1'$$

for some constant c_1'. Thus

$$\frac{d_0}{d(t_0, t_{j-3})} > c_2$$

for some constant c_2. But t' lies between the inner terminals of B, and so if t' lies on the same side of t_0 as t_{j-3} we have $d(t_0, t_{j-3}) > d(t_0, t')$. The same conclusion follows if t' lies on the other side of t_0 because t_{j-3} is much further away from t_0 than t_{j-4}. Hence $d(t_0, t') < c_2' d_0 < d(t_{j-2}, s_{j-2})$ by (2), where c_2' denotes some constant. We next show by the minimality of S that no point in B', except perhaps t', can have distance to t_{j-2} less than $d(t_{j-2}, s_{j-2})$. Because if b' is such a point, then we can consider joining t_{j-2} to b' and erasing the edge $t_{j-2}s_{j-2}$ (in the case where the path in S from B' to t_{j-2} is via s_{j-2}) or joining t_0 to t' and erasing $t_{j-2}s_{j-2}$ (otherwise). This gives a tree shorter than S. Similarly, no point in B' can have distance to t_{j-1} less than $d(t_{j-1}, s_{j-1})$ or distance to s_{j-1} less than $d(s_{j-1}, s_{j-2})$. This gives three circles A_1, A_2, and A_3 from the interior of which B' is excluded (see Figure 7). A short interval I of $G \cap C$ lies between these excluded regions. Lines l_1 and l_2 can be drawn from the ends of I at angles of $60 + \delta°$ to the direction of G in C, as shown in Figure 7. It follows from (2) that the radius of A_2 divided by the distance from A_2 to the far end of I can be made arbitrarily large. Hence l_2 intersects A_2. Similarly, it follows that l_1 intersects A_1. Thus there are bounded regions R_1, R_2, and R_3 as shown in Figure 7.

Assume now that some terminal (not necessarily inner) of B' lies outside I. Referring to the terminals of B' as $t_0', t_1', \ldots$ as for B, let i be maximized such that t_i' lies in I. Either t_i' is the leftmost terminal of B' in I, in which case the line from s_i' to s_{i+1}' must lie in regions R_1 and R_2, or it is the rightmost terminal of B' in I, in which case the line from s_i' to s_{i+1}' must lie in regions R_1 and R_3. This forces the line from t_{i+1}' to s_{i+1}' to enter A_1 or A_2, which is a contradiction.

Thus all terminals of B' lie inside I. Since B was arbitrary, we now get a contradiction, by reversing the roles of B and B', unless all the inner terminals of B' lie between two adjacent inner terminals of B. In this case we can join all the terminals of B' along G to the closest terminal of B and delete B'. Since B' has length at least ϵ (it leaves the capsule C), and we can assume that I has length less than ϵ, this shortens S. If this operation disconnects S, we can reconnect it either to t_{j-2} or to t_{j-1} to obtain a tree shorter than S since the edge of B' leaving the capsule slants to the left or to the right at approximately 60° to the direction of G in the capsule. (In verifying this, note that by (2) and (3), in the first case, $d(s_{j-2}, t_0)$ is arbitrarily small compared with $d(t_{j-2}, t_0)$ and hence with ϵ also by the assumption that the length of $G \cap C(Q)$ is at most 100ϵ. The second case is similar.) The conclusion is that no inner terminal of any alternating branch in $Y(C)$ lies between two inner terminals of another alternating branch.

2.5. Assigning terminals to branches. For the alternating branch B as before, we can choose the terminals $t_{j-4}, t_{j-3}, \ldots, t_j$ and t_0 in at most n^6 ways. Do this

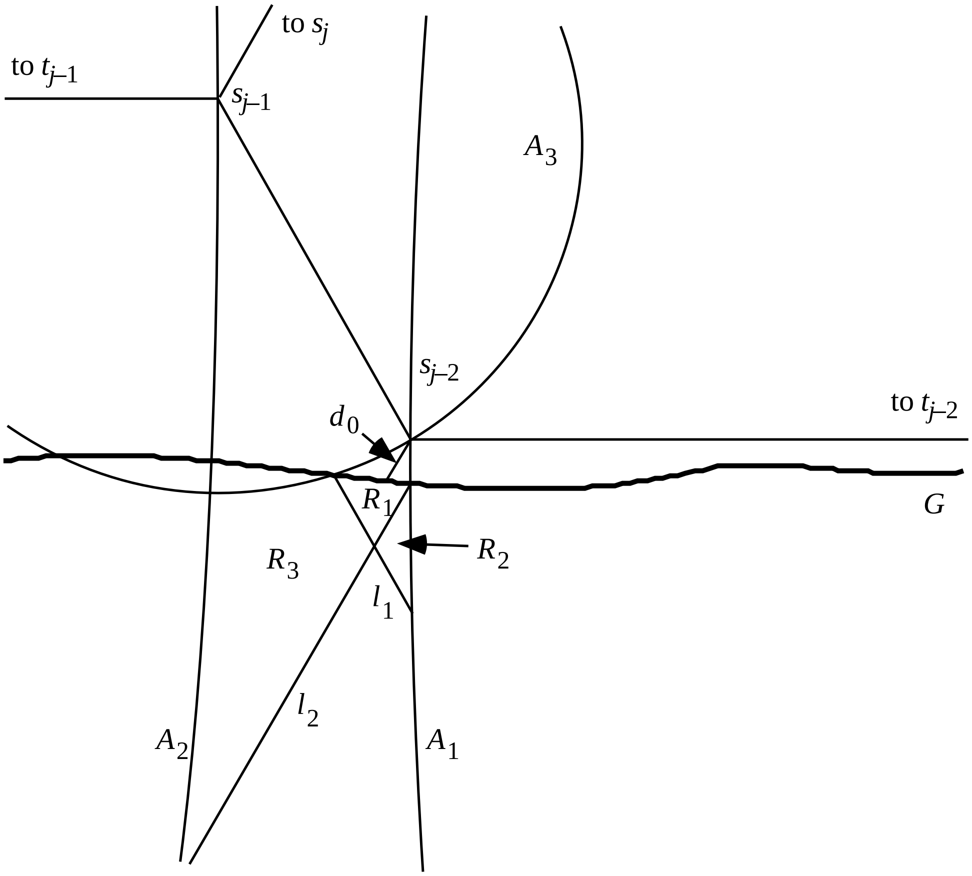


FIG. 7. *Excluded regions.*

for each alternating branch. Let I_0 denote the portion of $G \cap C$ between t_{j-4} and t_{j-3}. We can assume from above that no terminals on I_0 are in other alternating branches, and we can ignore the ones in cherries (assuming that they have been chosen already). Thus terminals on I_0 are the terminals of B and terminals adjacent in S only to other terminals.

Suppose that an edge of S is incident with both a terminal t in I_0 and a terminal t' outside I_0. Then either t is in Z', and is already chosen, or t' is in $G \cap C$, in which case S is clearly not optimal due to the edges of B being close to, nearly parallel to, and nearly overlapping the edge tt'. Similarly, no edge connecting two terminals in I_0 can "overlap" another such edge, in the sense that the end of one edge cannot lie between the two ends of the other edge (betweenness is measured along G).

Thus we can assume that the terminals in I_0 other than $t_0, t_1, \ldots, t_{j-3}$ are all connected in S along G to the terminals of B. Any edge from a terminal in B to one not in B must be directed away from t_0 since otherwise an angle of less than 120° is created (see Figure 8). We next show that it is enough in this situation to choose t_0 and the direction from t_0 to t_1.

First, no terminals can lie in between t_0 and t_1, so t_1 is now determined. For the

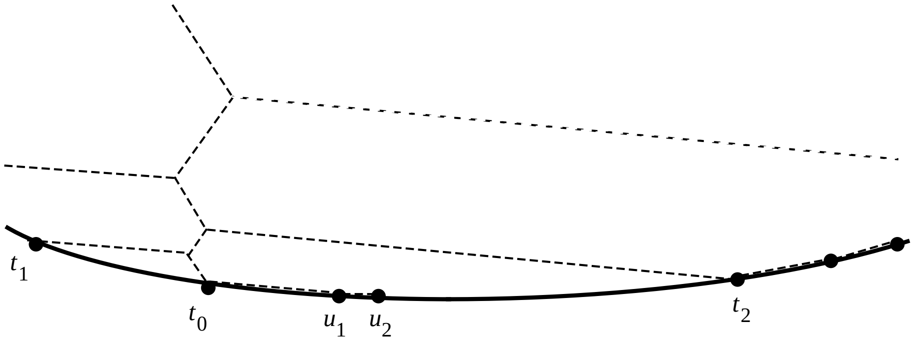


FIG. 8. *Terminals near an alternating branch.*

same reason that produces (2) we have

$$\frac{d(s_1,t_1)}{d(t_0,s_1)} > \omega_1.$$

Also, for the same reason that produces (1), the terminals u_1 and u_2 in Figure 8 must have distance at most $c_3 d(t_0, s_1)$ from t_0 for some constant c_3. On the other hand, from (3) and (4),

$$\frac{d(s_2,t_2)}{d(t_0,s_1)} > \omega_1$$

and thus t_2 is uniquely determined as the first terminal around G from t_0 in the direction away from t_1 and of distance at least $c_3 d(t_0, s_1)$ from t_0. In a similar fashion the terminals t_3, t_4, and so on are determined uniquely, one after another. This completes the proof, since t_0 and the direction from t_0 to t_1 can be chosen in at most $2n$ ways.

3. NP-hardness of angles. We present the following NP-complete problem in the format given by Garey and Johnson [2].

SUBSET SUM.

INSTANCE: A set $S = \{d_1, \ldots, d_n\}$ of integers and an integer s.

QUESTION: Is there a subset J of S such that $\sum_{i \in J} d_i = s$?

We will use the fact that SUBSET SUM is NP-complete to deduce that the following problem is NP-hard.

PALIMEST (parallel line minimal Euclidean Steiner tree).

INSTANCE: A set T of points in the plane contained in two parallel lines and a number l.

QUESTION: Is there a Steiner tree S with terminals T and length at most l?

In [1] a thorough discussion can be found of the difficulties of describing the complexity of Steiner tree problems due to the existence of square roots in the Euclidean metric. Instead of repeating that discussion, we state our results in two forms. Theorems 2 and 3 assume that infinite precision computation such as calculating the distance between points in the plane precisely can be done in finite time. Then Theorem 4 states that discretized versions of these problems, which are more realistic from the point of view of practical computing, are NP-complete. In view of Provan's result

[5], these discretized versions cannot be NP-complete in the strong sense unless P = NP.

THEOREM 2. *PALIMEST is NP-hard.*

The proof of Theorem 2 is by reduction from SUBSET SUM. Given an instance $S = \{d_1, \ldots, d_n\}$ and s of SUBSET SUM, we construct an instance $X(S)$ of PALIMEST as follows. First choose four parallel lines l_1 and l_2, l_1' and l_2' such that l_1' and l_2' are between l_1 and l_2. Then construct a tree with all leaves on l_1 and l_2, with all edges at 30° or 90° to the direction of l_1 and l_2, and with structure as shown in Figure 9.

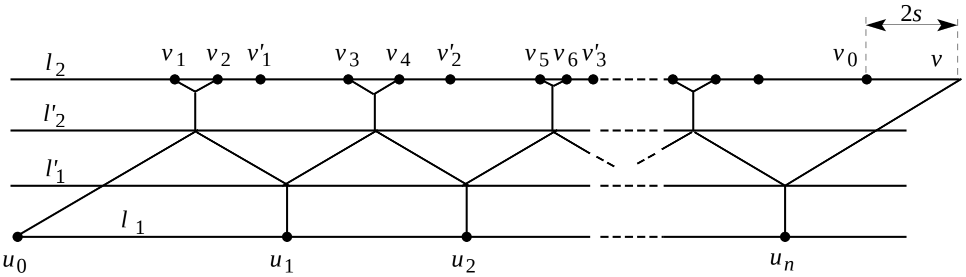


FIG. 9. *A particularly troublesome set of terminals.*

The leaves of the tree on l_1 are $u_0, \ldots, u_n$, and those on l_2 are $v_1, \ldots, v_{2n}$ and v. Note that the symmetry of Figure 9 means that once the distances between the lines are chosen, the only flexibility is where the Steiner points between v_{2i-1} and v_{2i} are located. In particular the locations of the points v, $u_0, \ldots, u_n$ are determined. The points $v_1, \ldots, v_{2n}$ are chosen so that $d(v_{2i-1}, v_{2i}) = d_i$ for each i. Note that this can be achieved by adjusting the height of the Steiner points. For PALIMEST, let T be the set $\{v_1, \ldots, v_{2n}\} \cup \{u_0, \ldots, u_n\}$ together with the n points v_i' on l_2 of distance d_i to the right of v_{2i} for each i and the point v_0 on l_2 of distance $2s$ to the left of v. Choose

$$d(l_1, l_2) >> d(l_1, l_1') = d(l_2, l_2') >> D = \sum_{i=1}^{n} d_i$$

so that, seen from a long way off, the tree looks like Figure 10. Clearly, on this scale the tree describes a minimal Steiner tree for the terminals in T. Thus, such a minimal Steiner tree must have the form of the tree shown in Figure 9, except that the rightmost edge is incident with v_0 rather than v, and that each connection up to l_2 except that at v_0 contains a Steiner vertex adjacent to both v_{2i} and one of v_{2i-1} and v_i'. Whichever vertex is missed is adjacent directly to v_{2i} in T. The two options are shown in Figure 11. We refer to them as the left and right options of the ith upper attachment. In any tree which uses one of these two options for each i and has the property that all edges meet at Steiner vertices at 120°, the angle α between l_1 and the edge to u_0 is determined. We call such a tree, whether minimal or not, an *α-degree tree.*

PROPOSITION 1. *There exists a* 30*-degree tree if and only if there is a subset J of S such that $\sum_{i \in J} d_i = s$.*

Proof. This follows from the simple observation that in a 30-degree tree, use of the right option at the ith attachment for each i in a set J has the effect of translating the

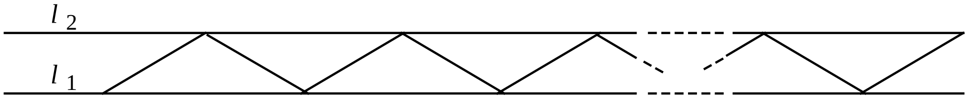


FIG. 10. *A minimal tree.*

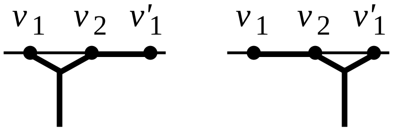


FIG. 11. *Two modes of attachment.*

rightmost leaf of the tree, which lies on l_2 near v, along l_2 by a distance of $2\sum_{i\in J} d_i$ to the left. □

The main fact remaining to be shown is that 30-degree trees, if they occur, are minimal. To facilitate this we define a generalization of an α-degree tree called an *α-degree configuration.* First add an extra line l_3 parallel to l_2 and intersecting all the upper vertical lines of an α-degree tree A, such that $d(l_2, l'_2) >> d(l_3, l_2) >> D$. Every upper vertical edge of A is sliced into two edges by l_3, and the ends of these edges at l_3 are then permitted to move freely up and down l_3 as if moving along a curtain rail on runners, as in Figure 12. The length of the configuration is computed as the sum of the lengths of its edges; distances along l_3 are ignored. The directions of the edges in the configuration are still restricted to three directions at 120° to each other, and the angle of the edge at u_0 to l_1 is α. By a *configuration* we mean an α-degree configuration for some α.

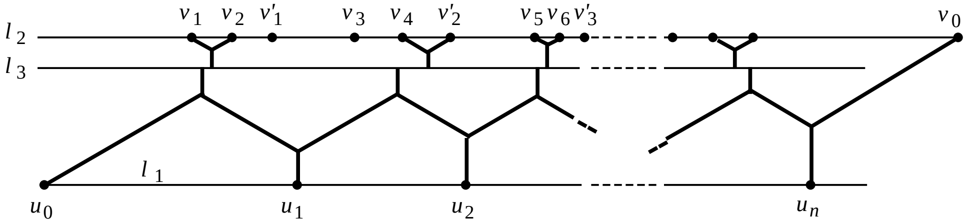


FIG. 12. *An α-degree configuration.*

PROPOSITION 2. *All shortest configurations are* 30*-degree configurations.*

Proof. This follows from the observations that in a shortest configuration each edge meeting l_3 must do so at 90°, and that each component of a shortest configuration is a minimal Steiner tree and hence, by Lemma 1, has edges at Steiner vertices meeting at 120°. □

PROPOSITION 3. *All* 30*-degree configurations have equal length.*

Proof. The sum of the lengths of the edges above l_3 is clearly constant, since $d(v_{2i-1}, v_{2i}) = d_i = d(v_{2i}, v'_i)$. Let $w_1, \ldots, w_n$ denote the points on l_3 touched by the configuration in left-to-right order. For $0 \leq i < n$, let a_{2i} denote the horizontal distance from u_i to w_{i+1} and a_{2i+1} the horizontal distance from w_{i+1} to u_{i+1}, and let a_{2n} denote the horizontal distance from u_n to v_0. Also for $1 \leq i \leq n$ let x_{2i-1} denote the length of the edge below l_3 incident with w_i, and let x_{2i} denote the length of the edge incident with u_i. We have for $1 \leq i < 2n$,

$$x_1 + a_0/\sqrt{3} = x_i + x_{i+1} + a_i/\sqrt{3} = x_{2n} + a_{2n}/\sqrt{3} = d(l_1, l_3)$$

and thus

$$\sum_{i=1}^{2n} 2x_i + \sum_{i=0}^{2n} a_i/\sqrt{3} = (2n+1)d(l_1, l_3).$$

Since $\sum_{i=0}^{2n} a_i$ is fixed equal to the horizontal distance from u_0 to v_0, it follows that the length of the configuration is constant. □

In view of Proposition 3, we define L to denote the length of all 30-degree configurations.

Proposition 4. *There exists a* 30*-degree tree if and only if the minimal Steiner trees have length* L.

Proof. A 30-degree tree is a 30-degree configuration which, by Propositions 2 and 3, is a shortest configuration of length L, and is hence a minimal Steiner tree, because all α-degree Steiner trees are configurations. Any non-30-degree tree is not a minimal configuration by Proposition 2, and hence has length greater than L. □

For PALIMEST use T as constructed above and set $l = L$. From Proposition 4 it follows that the answer to the given instance of SUBSET SUM is YES if and only if the answer to this instance of PALIMEST is YES. Since this instance of PALIMEST is computable in time polynomial in n, this completes the proof of Theorem 2.

We now consider graphs with angles in the form of the following problem.

β-INSEMEST (intersecting segment minimal Euclidean Steiner tree).
INSTANCE: A set T of points in the plane contained in two line segments emanating from a point at angle β and a number l.
QUESTION: Is there a Steiner tree S with terminals T and length at most l?

Theorem 3. *For* $\beta < 120°$*,* β*-INSEMEST is NP-hard.*

Proof. This follows the proof of Theorem 2, using the same instance of SUBSET SUM but with different figures. Choose two line segments m_1 and m_2 meeting at an angle of $\beta°$ and bisected by a line l_0 at 30° to horizontal. As in the proof of Theorem 2, we use a tree with all edges in three directions 120° to each other, one direction being horizontal. Seen from a distance, the tree looks like Figure 13(a). When expanded, the interior of circle C_1 is as seen in Figure 13(b) and that of the circle C'_1 is as seen in Figure 13(c), while the interior of C_2 looks like that of C_1 (but shows u_{n-1} instead of u_n, and C_3 and C'_2 instead of C_2 and C'_1). After repeating this pattern n times, we get to C_n, which is as shown in Figure 13(d). The distance between u_0 and v_1 is chosen much larger than D, but the precise relative placement of v_{2i-1} and v_{2i} with respect to v'_i, and of v with respect to v_0, is determined below.

For INSEMEST, let T be the set $\{v_1, \ldots, v_{2n}\} \cup \{u_0, \ldots, u_n\} \cup \{v'_1, \ldots, v'_n\}$ together with a point v_0 on m_2 to the left of v. By an adjustment of the structure, the vertical edge leading into C'_i and the distances $d(v_{2i-1}, v_{2i})$, $d(v_{2i}, v'_i)$, and $d(v, v_0)$ can be made small enough so that a minimal Steiner tree for the terminals in T has the same adjacency structure as the α-degree trees in the proof of Theorem 2. We use this name in the present situation for such a tree in which the edge incident with u_0 makes an angle α with the horizontal. The constraint that β is less than 120° is required to ensure that α-degree trees exist. It is fairly easy to see that a minimal Steiner tree for T must be such a tree, provided the Steiner vertices inside C'_i are caused to be sufficiently close to m_2.

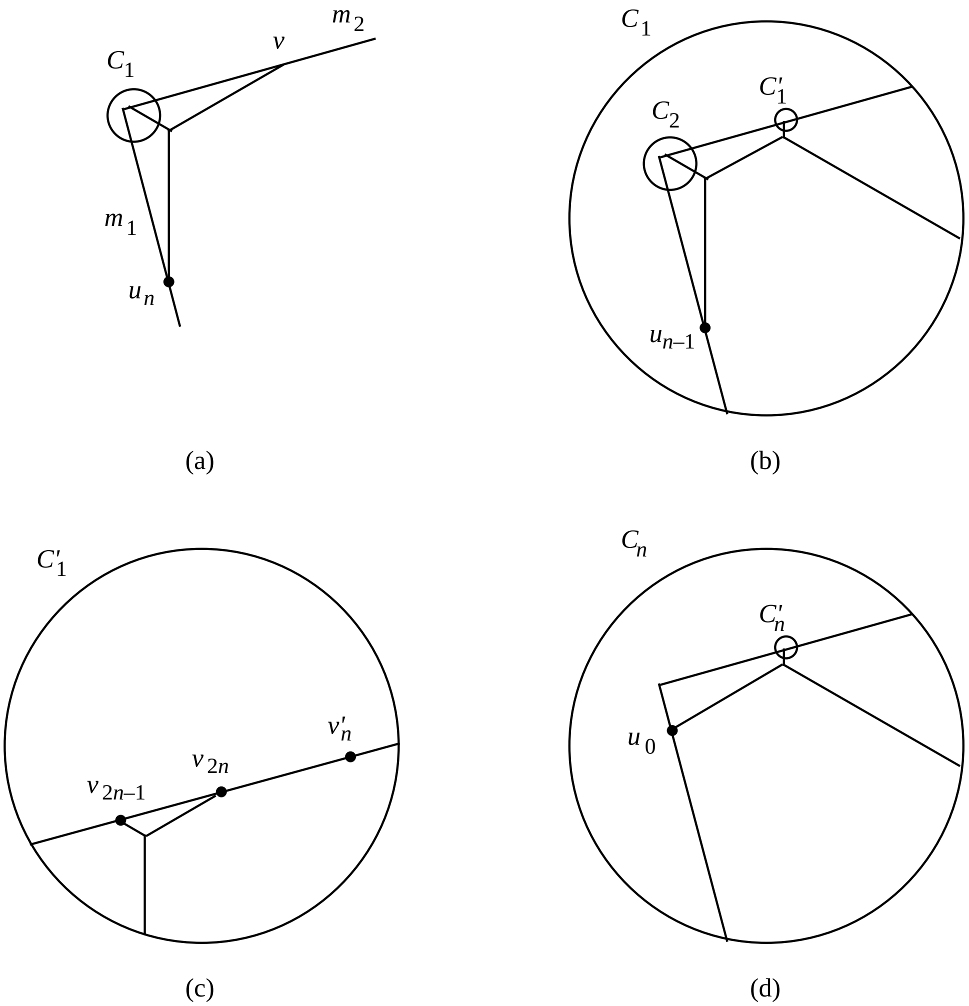


FIG. 13. *Terminals on two intersecting lines.*

We must still specify the exact placement of the v_i, v_i', and of v. The distances $d(v_{2i-1}, v_{2i})$ and $d(v_{2i}, v_i')$ are chosen so that in the two options shown in Figure 14, the vertical line in the left option is of distance exactly d_i to the left of the one in the right option, and such that the length of this branch of the tree above the horizontal line l_i is the same in both cases. Note that since the bisector of the angle between m_1 and m_2 is at 30° to horizontal and β is less than 120°, the angle of m_2 to horizontal is less than 30°. Thus it can be seen that the desired placement of these vertices is always possible (by Melzak's algorithm). As in Figure 9, we choose a "leftmost" 30-degree tree to define the positions of $v_1, v_3, \ldots, v_{2n-1}$ and v. Then v_0 is chosen by displacing the edge of the 30-degree tree incident with v horizontally at a distance $2s$ to the left. v_0 is then the intersection of this displaced edge with m_2.

The rest of the proof now parallels the proof of Theorem 2 very closely, except that in place of the line l_3 we have individual lines l_i, one for each i, providing curtain rails for each of the upper attachments of an α-degree configuration. Propositions 1–4

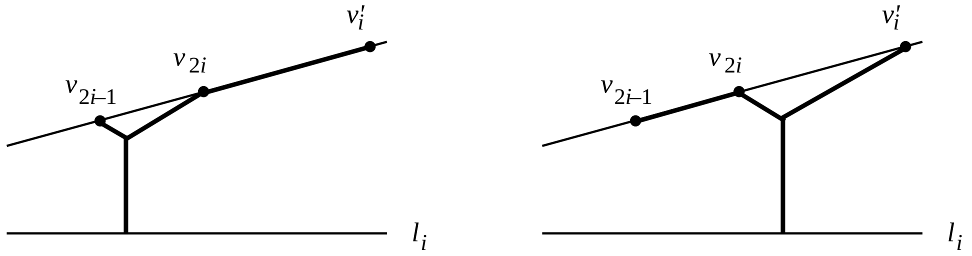


FIG. 14. *Two equally long options.*

remain true in the new context, and we obtain Theorem 3.

For the discretized problems, we introduce as in [1] the discretized (and more computationally realistic) Euclidean metric d' in which the length of an edge joining two points x and y is taken as $d'(x, y) = \lceil d(x, y) \rceil$. The discretized Euclidean length of a tree is then the sum of the discrete Euclidean lengths of its edges. For the discretized problems DPALIMEST and DINSEMEST, we restrict the input points to points with integer coordinates, the condition that they are contained in a line becomes the condition that they are at distance less than 1 on that line, and the lengths of the trees are replaced by their discretized Euclidean lengths.

THEOREM 4. *DPALIMEST, and β-DINSEMEST for $\beta < 120°$, are NP-complete.*

Proof. This depends on the fact that we can scale up the instances in the proofs of Theorems 2 and 3 so that even when restricted to integer coordinates, all the optimal trees for the discretized versions correspond to optimal trees for the nondiscretized versions. The scale up can be chosen of the form $C^{P(n)}$ for a constant C and $P(n)$ any polynomial in n. (Another way to look at this is to observe that by scaling down the integers, we can alternatively regard the discretized problems as restrictions to rationals in which the number of places in their decimal representations is bounded above by a polynomial in n.)

First consider PALIMEST. It has to be verified that an appropriate instance of this problem can be computed in polynomial time from the instance of SUBSET SUM. There are several observations that ensure that nothing can go wrong here. First, the diameter of the set of terminals described can be chosen, for example, so that it is bounded above by $W = P(D)$ for some polynomial P.

Second, consider an α-degree tree with a choice of right-hand upper connections corresponding to a subset of $\{d_1, \ldots, d_n\}$ whose sum is not s. It is easily seen that the difference between α and $30°$ is then at least C/W for a constant C. But for such α the difference in length between an α-degree configuration and a 30-degree configuration will be at least C/W^2. This can be seen by noting that the part of an α-degree configuration near a Steiner vertex, which has size W/n, can be replaced, using $120°$ angles, by a network that is shorter by at least $C/(nW)$. (The second variation can be shown to be nonzero using the techniques in [6, section 1].) Hence appropriate scaling will make this difference arbitrarily large, even using discretized Euclidean lengths of edges. In this way we can arrange that the total discretized length of the α-degree tree is longer than that of the 30-degree tree. It follows that PALIMEST is NP-complete.

Now consider β-DINSEMEST. The difference in scale between C_i and C_{i+1} is a constant, and so the width of the set of terminals can be bounded above by $C^n P(D)$ for some constant C. The rest of the proof goes as for PALIMEST. □

REFERENCES

[1] 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.

[2] M. R. Garey and D. S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.

[3] E. Gilbert and H. Pollak, *Steiner minimal trees*, SIAM J. Appl. Math., 16 (1968), pp. 1–29.

[4] F. K. Hwang, *A linear time algorithm for full Steiner trees*, Oper. Res. Lett., 5 (1986), pp. 235–237.

[5] J. S. Provan, *Convexity and the Steiner tree problem*, Networks, 18 (1988), pp. 55–72.

[6] J. H. Rubinstein and D. A. Thomas, *Graham's problem on shortest networks for points on a circle*, Algorithmica, 7 (1992), pp. 193–218.

SIAM J. DISCRETE MATH.
Vol. 10, No. 1, pp. 18–29, February 1997

PARTITIONS WITH RESTRICTED BLOCK SIZES, MÖBIUS FUNCTIONS, AND THE k-OF-EACH PROBLEM*

SVANTE LINUSSON[†]

Abstract. Given a list of n real numbers, one wants to decide whether every number in the list occurs at least k times. It will be shown that $\Omega(n \log n)$ is a sharp lower bound for the depth of an algebraic decision or computation tree solving this problem for a fixed k. For linear decision trees, the coefficient can be taken to be arbitrarily close to 1 (using the ternary logarithm). This is done by using the Björner–Lovász–Yao method, which turns the problem into one of estimating the Möbius function for a certain partition lattice. The method will work also for the more general T-multiplicity problem when T is additive and cofinite. A formula for the exponential generating function for the Möbius function of a partition poset with restricted block sizes in general will also be given.

Key words. decision tree, Möbius function, partitions

AMS subject classifications. Primary, 68Q25; Secondary, 05A15, 05A18, 06A07, 68R05

PII. S089548019426855X

Introduction. The membership problem is the problem of determining whether a given point belongs to a certain prescribed region in $\mathbb{R}^n$. In a sequence of papers, [BLY], [Y1], [BL], and [Y2], Björner, Lovász, and Yao have developed a technique for determining lower bounds for the depth of decision and computation trees in terms of the Betti numbers for the region. In the case of subspace arrangements, the lower bounds specialize to expressions involving the Möbius function for the corresponding intersection lattice. Björner, Lovász, and Yao were originally motivated by the k-equal problem, i.e., to determine whether there exists k equal numbers among a list of n given numbers. For fixed k they were able to show the sharp lower bound $\Omega(n \log n)$ for that problem.

In this paper we will give a more general method for estimation of the Möbius function which will lead to the same lower bound for the k-of-each problem stated in the abstract. We will also consider the following more general problem: given a set T of positive integers and a list of n real numbers, determine whether every number in the list occurs with a multiplicity m such that $m \in T$. We will call this the *T-multiplicity problem.* It will be shown that if T is additive, cofinite, and does not contain 1, then the same lower bound is again valid. It will also be shown that in the case of linear decision trees, the coefficient can be taken arbitrarily close to 1 (using the ternary logarithm). The method will also reprove the results for the k-equal problem.

In section 3 we will calculate the exponential generating function for the Möbius function of a partition poset with an arbitrary set of forbidden block sizes. In section 5 this will be used to get a lower bound on the absolute value of the Möbius functions corresponding to the computational problems, leading to the complexity-theoretic lower bound. In section 2 an algorithm is given to show that this lower bound is sharp.

* Received by the editors May 27, 1994; accepted for publication (in revised form) December 21, 1995.
http://www.siam.org/journals/sidma/10-1/26855.html

[†] Department of Mathematics, Royal Institute of Technology, S-100 44 Stockholm, Sweden (linusson@math.kth.se).

1. Preliminaries. By identifying the list of numbers with a point $\mathbf{x} \in \mathbb{R}^n$, the k-of-each problem can be viewed as deciding whether $\mathbf{x}$ belongs to a certain subset $V_{n,k}$ of $\mathbb{R}^n$. $V_{n,k}$ can be described as the union of a set of linear subspaces, a so called *subspace arrangement.* Given n and k, let $\mathcal{A}_{n,k}$ denote the set of all linear subspaces of $\mathbb{R}^n$ defined by some equations of type $x_{i_1} = x_{i_2} = \cdots = x_{i_r}$ where $r \geq k$ such that every coordinate occurs in one of the equations. Then

$$V_{n,k} = \cup_{A \in \mathcal{A}_{n,k}} A.$$

Now the problem is to decide whether $\mathbf{x}$ is in $V_{n,k}$ or not. Partially order the elements of $\mathcal{A}_{n,k}$ by reverse inclusion. Adding $\mathbb{R}^n$ as $\hat{0}$ we get the *intersection lattice* denoted $L_{n,k}$. (For a discussion of lattices and subspace arrangements, see [B1] and [OT].)

We will consider three slightly different models for deciding if a point $\mathbf{x}$ belongs to the subspace arrangement or not. A *linear decision tree* is a rooted ternary tree where at every interior node a linear function is evaluated at $\mathbf{x}$, and the three edges leaving the node are labeled "<," "=," and ">," corresponding to whether the outcome of the linear test is less than, equal to, or greater than zero. The leaves of the tree are marked YES or NO, thus giving the answer to whether or not $\mathbf{x}$ belongs to the subspace arrangement. We will use Theorem 3.7 in [BL], which gives a lower bound on the number of leaves in any linear decision tree for the k-of-each problem. Taking the ternary logarithm gives a lower bound on the depth of a linear decision tree $C_1(V_{n,k})$, namely Theorem A.

Theorem A (Björner and Lovász). *The depth of a linear decision tree determining the k-of-each problem is bounded below by the following inequality:*

$$C_1(V_{n,k}) \geq \log_3 \left(\sum_{x \in \Pi_{n,k}} |\mu(\hat{0}, x)| \right). \qquad \square$$

The second model will be a *degree-d algebraic decision tree*, which differs from a linear decision tree by having polynomial tests of degree at most d at each node instead of linear ones. The third model is an *algebraic computation tree* where a node can perform a binary arithmetic calculation or test whether a previously calculated number is less than, equal to, or greater than zero. For a detailed description of such trees, see [Y1]. Let $C_d(V_{n,k})$ and $C(V_{n,k})$ denote the minimal depth for an algebraic decision tree using polynomials of degree $\leq d$ and an algebraic computation tree, respectively. The following lower bounds follow from recent work of Yao; see the proof of Theorem 3 in [Y2].

Theorem B (Yao). *The depth of a degree-d algebraic decision tree and of an algebraic computation tree is bounded below by*

$$C_d(V_{n,k}) \geq \alpha_d \log \left(\sum_{x \in \Pi_{n,k}} |\mu(\hat{0}, x)| \right) - \beta_d n$$

and

$$C(V_{n,k}) \geq \alpha \log \left(\sum_{x \in \Pi_{n,k}} |\mu(\hat{0}, x)| \right) - \beta n,$$

respectively, for some constants $\alpha, \alpha_d, \beta, \beta_d > 0$. $\square$

For a survey of these topological methods in complexity theory, see [B2]. For the definitions of lattice, Möbius function, and other combinatorial terminology, the reader is referred to basic books in combinatorics, e.g., [S1].

We will consider the partition lattice $\Pi_{n,k}$ consisting of partitions of $\{1, 2, \dots, n\}$, where block sizes $1, 2, \dots, k-1$ are forbidden, with the discrete partition $(1)(2)\cdots(n)$ added as zero. Observe that $\Pi_{n,k}$ is a lattice with the same join-operator as Π_n. The meet-operation is that of Π_n (coarsest common refinement) unless one gets some block of size less than k; then the meet will be $\hat{0}$. Our interest in $\Pi_{n,k}$ comes from the following proposition.

PROPOSITION 1. *$L_{n,k}$ is isomorphic to $\Pi_{n,k}$.*

Proof. If $\sigma \in \Pi_{n,k}$, let

$$B_\sigma = \{x \in \mathbb{R}^n | x_i = x_j \text{ if } i \text{ and } j \text{ are in the same block in } \sigma\}.$$

We get that $\dim B_\sigma =$ number of blocks in σ.

It is immediate that $B_\sigma \vee B_\pi = B_\sigma \cap B_\pi = B_{\sigma\vee\pi}$ and from this follows that $L_{n,k} \cong \Pi_{n,k}$. □

2. Algorithm. The problem posed is the following: given a list of n real numbers, one wants to decide whether every number in the list occurs at least k times.

The following algorithm shows that the k-of-each problem can be solved using a linear tree with depth $n \log_2(n/k) + 3n$.

ALGORITHM.

1. Divide the numbers into k separate lists with (approximately) n/k elements in each and sort each list completely. This takes $k(n/k \log_2 n/k) = n \log_2 n/k$ comparisons.
2. Find the smallest number a among the smallest elements in each list. This takes $k-1$ comparisons.
3. Remove all elements equal to a. This takes at most (number of elements equal to a)$+k$ comparisons. If the number of elements equal to a is less than k then the answer is NO, if not then repeat from 2 until all elements are removed.

Steps 2 and 3 can be repeated at most n/k times, so the total number of comparisons performed is at most

$$n \log_2 \frac{n}{k} + \frac{n}{k}(k - 1 + k) + n \leq n \log_2 \frac{n}{k} + 3n. \qquad \square$$

3. The Möbius function. The intersection lattices we will be interested in have a combinatorial description in terms of set partitions. We will derive the exponential generating function for such partition posets. This is done also in [BL, section 4] but only in the case when singleton blocks are allowed. Here we will need the case when singleton blocks are forbidden. We will treat both cases simultaneously with a method different from the one used in [BL].

Given any set $T \subseteq \mathbb{Z}_+ = \{1, 2, 3, \dots\}$, we consider the set $\Pi_{n,T}$ of partitions of $[n] := \{1, 2, \dots, n\}$ into blocks whose sizes are in T. Ordering the elements by refinement we get a poset, which is not a lattice in general. If $1 \notin T$ then we have to add the discrete partition $(1)(2)\cdots(n)$ to $\Pi_{n,T}$ as $\hat{0}$. We denote by $\mu_{n,T}$ the Möbius function of the poset $\Pi_{n,T}$, where the subscript n often will be suppressed. Let also $\mu_T(n) := \mu_{n,T}(\hat{0}, \hat{1})$ if $n \in T$. It will be convenient to extend the definition of $\mu_{n,T}(\pi, \sigma)$ by setting it to 0 if either π or σ is not in $\Pi_{n,T}$. In particular, $\mu_T(n) = 0$ if $n \notin T$.

When doing the calculations we will use a well-known (see, e.g., [S1]) property of the Möbius function for posets. Given a poset P and $a, b, c \in P$, then

$$[\hat{0}, a] \cong [\hat{0}, b] \times [\hat{0}, c] \implies \mu_P(\hat{0}, a) = \mu_P(\hat{0}, b)\mu_P(\hat{0}, c).$$

We also need a not-so-well-known fact about the Möbius function for which we have not found any reference so we include a proof.

Lemma 1. *For any poset P and $a, b, c \in P$ we have*

$$[\hat{0}, a]\backslash\{\hat{0}\} \cong ([\hat{0}, b]\backslash\{\hat{0}\}) \times ([\hat{0}, c]\backslash\{\hat{0}\}) \implies \mu_P(\hat{0}, a) = -\mu_P(\hat{0}, b)\mu_P(\hat{0}, c).$$

Proof. Using the definition of Möbius function we get

$$\begin{aligned} \mu_P(\hat{0}, a) &= -\sum_{\hat{0} < \pi \le a} \mu_P(\pi, a) \\ &= -\sum_{\substack{\hat{0} < \pi_1 \le b \\ \hat{0} < \pi_2 \le c}} \mu_P(\pi_1 \times \pi_2, c) = -\sum_{\substack{\hat{0} < \pi_1 \le b \\ \hat{0} < \pi_2 \le c}} \mu_P(\pi_1, b)\mu_P(\pi_2, c) \\ &= -(-\mu_P(\hat{0}, b))(-\mu_P(\hat{0}, c)) = -\mu_P(\hat{0}, b)\mu_P(\hat{0}, c). \qquad \square \end{aligned}$$

We can now prove the basic recurrence formula.

Lemma 2. *If $n \in T\backslash\{1\}$ we have*

$$(1) \qquad \mu_T(n) = -\sum_{\sum_{i \in T\backslash\{n\}} ic_i = n} {\mu_T}^{c_1}(1) \cdots {\mu_T}^{c_{n-1}}(n-1) \frac{n!}{\Pi (j!)^{c_j} c_j!} \qquad \textit{if } 1 \in T,$$

(2)

$$\mu_T(n) = -1 + \sum_{\sum_{i \in T\backslash\{n\}} ic_i = n} (-1)^{\sum c_i} {\mu_T}^{c_2}(2) \cdots {\mu_T}^{c_{n-2}}(n-2) \frac{n!}{\Pi (j!)^{c_j} c_j!} \qquad \textit{if } 1 \notin T.$$

Proof. When $1 \in T$ we have

$$[\hat{0}, (1, 2, 3, \ldots, l)(l+1, \ldots, n)] = [\hat{0}, (1, 2, \ldots, l)] \times [\hat{0}, (l+1, \ldots, n)].$$

From the properties of the Möbius function stated above we get $\mu_T(\hat{0}, (1, 2, \ldots, l)(l + 1, \ldots, n)) = \mu_T(l)\mu_T(n-l)$.

We also know that there are $\frac{n!}{\prod_{j=1}^{n} (j!)^{c_j} c_j!}$ partitions of $[n]$ of type $c_1, \ldots, c_n$. By definition of the Möbius function we get the first formula.

If $1 \notin T$ we have instead

$$[\hat{0}, (1, 2, \ldots, l)(l+1, \ldots, n)]\backslash\{\hat{0}\} = ([\hat{0}, (1, 2, \ldots, l)]\backslash\{\hat{0}\}) \times ([\hat{0}, (l+1, \ldots, n)]\backslash\{\hat{0}\}),$$

and hence Lemma 1 shows that

$$\mu_T(\hat{0}, (1, 2, \ldots, l)(l+1, \ldots, n)) = -\mu_T(l)\mu_T(n-l).$$

This gives the second equation, where the -1 term is for $\hat{0} = (1)(2)\cdots(n)$, which is not included in the sum. $\square$

The second step is to calculate the exponential generating function for each specific T. Define

$$F_T(x) := \sum_{n=1}^{\infty} \mu_T(n) \frac{x^n}{n!},$$

remembering that $\mu_T(n) = 0$ if $n \notin T$. Define also for every positive integer n

$$s_T(n) := \sum_{\sigma \in \Pi_{n,T}} \mu_T(\hat{0}, \sigma).$$

By the definition of the Möbius function, we have $s_T(n) = 0$ if $n \in T\backslash\{1\}$. However, $s_T(1) = 1$ for any T. We also define

$$p_T(x) := \sum_{n \in \mathbb{Z}_+} s_T(n) \frac{x^n}{n!}.$$

Remark. Note that if $n \notin T \cup \{1\}$, $s_T(n)$ gets the value that $-\mu_T(n)$ would have had if n had belonged to T. Hence if $n \notin T \cup \{1\}$, one can replace $\mu_T(n)$ by $-s_T(n)$ in Lemma 2.

Proposition 2. *The exponential generating function for* $\Pi_{n,T}$ *is given by*

$$F_T(x) = \ln(1 + p_T(x)) \qquad \textit{if } 1 \in T, \tag{3}$$

$$F_T(x) = -\ln(e^x - p_T(x)) \qquad \textit{if } 1 \notin T. \tag{4}$$

Proof. In the first case, $1 \in T$.

Using the recurrence formula in Lemma 2, we get

$$\begin{aligned}
0 &= \sum_{n \in T\backslash\{1\}} 0 \frac{x^n}{n!} = \sum_{n \in T\backslash\{1\}} \left(\sum_{\sum_{i \in T} ic_i = n} {\mu_T}^{c_1}(1) \cdots {\mu_T}^{c_n}(n) \frac{n!}{\Pi (j!)^{c_j} c_j!} \right) \frac{x^n}{n!} \\
&= \prod_{j \in T} \left(1 + \frac{\mu_T(j) x^j}{j!} + \frac{{\mu_T}^2(j) x^{2j}}{(j!)^2 2!} + \cdots \right) - 1 \\
&\quad - \sum_{n \in \mathbb{Z}_+ \backslash (T\backslash\{1\})} \left(\sum_{\sum_{i \in T} ic_i = n} {\mu_T}^{c_1}(1) \cdots {\mu_T}^{c_n}(n) \frac{n!}{\Pi (j!)^{c_j} c_j!} \right) \frac{x^n}{n!} \\
&\overset{*}{=} \prod_{j \in T} e^{\mu_T(j) \frac{x^j}{j!}} - 1 - \sum_{n \in \mathbb{Z}_+ \backslash (T\backslash\{1\})} s_T(n) \frac{x^n}{n!} \\
&= e^{F_T(x)} - 1 - \sum_{n \in \mathbb{Z}_+} s_T(n) \frac{x^n}{n!},
\end{aligned}$$

and the equation follows. The $*$ equality (above and below) follows from the above remark.

In the second case, $1 \notin T$.

$$\begin{aligned}
0 &= \sum_{n \in T} \left(\sum_{\sum_{i \in T} ic_i = n} (-1)^{\sum c_i} {\mu_T}^{c_2}(2) \cdots {\mu_T}^{c_n}(n) \frac{n!}{\Pi (j!)^{c_j} c_j!} - 1 \right) \frac{x^n}{n!} \\
&= \sum_{n \in T} \left(\sum_{\sum_{i \in T} ic_i = n} \frac{(-\mu_T(2) x^2)^{c_2}}{(2!)^{c_2} c_2!} \frac{(-\mu_T(3) x^3)^{c_3}}{(3!)^{c_3} c_3!} \cdots \frac{(-\mu_T(n) x^n)^{c_n}}{(n!)^{c_n} c_n!} \right) - \sum_{n \in T} \frac{x^n}{n!} \\
&= \prod_{j \in T} \left(1 - \frac{\mu_T(j) x^j}{j!} + \frac{{\mu_T}^2(j) x^{2j}}{(j!)^2 2!} - \cdots \right) - 1 - \sum_{n \in T} \frac{x^n}{n!}
\end{aligned}$$

$$-\sum_{n\notin T}\left(\sum_{\sum_{i\in T} ic_i=n} \frac{(-\mu_T(2)x^2)^{c_2}}{(2!)^{c_2}c_2!}\frac{(-\mu_T(3)x^3)^{c_3}}{(3!)^{c_3}c_3!}\cdots\frac{(-\mu_T(n-2)x^{n-2})^{c_{n-2}}}{((n-2)!)^{c_{n-2}}c_{n-2}!}\right)$$

$$\overset{*}{=}\prod_{j\in T} e^{-\mu_T(j)\frac{x^j}{j!}} - 1 - \sum_{n\in T}\frac{x^n}{n!} + \sum_{n\notin T}(s_T(n)-1)\frac{x^n}{n!}$$

$$= e^{-F_T(x)} - e^x + \sum_{n\notin T} s_T(n)\frac{x^n}{n!}$$

and the proposition follows. □

The reader well acquainted with the exponential formula (see, e.g., [S2]) might wonder whether it is possible to use it to prove Proposition 2. Indeed, that is the case, but we have preferred to give a more direct proof to avoid unnecessary terminology.

4. Three lemmas. This section consists of three lemmas with proofs of a rather technical nature. The reader is advised to just read the statements of the lemmas and then go on to the proofs of the main theorems in the next section. The interested reader can then come back to sort out the technicalities.

To prove the main theorems, we will need an upper bound for the radius of convergence for $\ln(e^z - p(z))$, considered as a function on $\mathbb{C}$, for certain polynomials $p(z)$. To do this we will need the following lemma for which I'm indebted to Daniel Bertilsson.

MODULUS LEMMA. *Let $p(z)$ be a polynomial of degree $k-1$ such that $p(0) = 0$ and $p'(0) = 1$. Then there is a zero of $e^z - p(z)$ with modulus less than $9k$.*

Proof. The main ingredient in the argument is the following version of Landau's theorem (see [J] and [H]): *suppose $f : D_1 := \{z \in \mathbb{C} \mid |z| < 1\} \to \mathbb{C}\backslash\{0,1\}$ is an analytic function. Then $|f'(0)| \leq 2|f(0)|\,(|\ln|f(0)|| + A)$ where $A = \frac{\Gamma(1/4)^4}{4\pi^2} \approx 4.45$.*

Suppose now $e^z - p(z) \neq 0$ for all $z, |z| < R$. Define an analytic function $g : D_R \to \mathbb{C}$ by

$$g(z)^k := 1 - e^{-z}p(z).$$

This is possible since $1 - e^{-z}p(z)$ is assumed to be nonzero in the simply connected region D_R. There is a number $\omega, \omega^k = 1$ such that $g(z) \neq \omega$ for all $z, |z| < R$. (Otherwise g would assume all k-roots of unity as values, and hence $p(z) = 0$ for k different $z \in \mathbb{C}$.) Define $f(z) := \frac{g(Rz)}{\omega}$ for all $z \in D_1$. The function f does not take the values 0 and 1. Landau's theorem says

$$\left|\frac{g'(0)}{\omega}R\right| \leq 2\left|\frac{g(0)}{\omega}\right|\left(\left|\ln\left|\frac{g(0)}{\omega}\right|\right| + A\right),$$

but $kg'(0)g(0)^{k-1} = \frac{d}{dz}(1 - e^{-z}p(z))\mid_{z=0} = -p'(0) = -1$ and $g(0) = 1$, so $\frac{R}{k} \leq 2A$; i.e., $R \leq 2Ak \approx 8.9k$. We can now conclude that in the disc $|z| < 9k$ there is a zero to $e^z - p(z)$. □

LEMMA OF COSINES. *Let $\theta_1, \theta_2, \ldots, \theta_m$ be real numbers. Then there is an integer N such that in any set of N consecutive integers, there is an integer n such that*

$$\left|\sum_{i=1}^{m} \cos n\theta_i\right| > \frac{\sqrt{m}}{2}.$$

This should be a known lemma but we have not been able to locate it in the literature, so we include a proof due to Mats Boij.

Proof. For all integers n we define $f(n) := \sum_{i=1}^{m} \cos n\theta_i$, and for all integers n and N we define $g_N(n) := \sum_{k=n}^{n+N-1} f(k)^2/N$. We can compute $f(n)^2$ as

$$\begin{aligned} f(n)^2 &= \sum_{i=1}^{m} \cos^2 n\theta_i + \sum_{i<j} 2\cos n\theta_i \cos n\theta_j \\ &= \frac{m}{2} + \frac{1}{2}\sum_{i=1}^{m} \cos 2n\theta_i + \sum_{i<j} \cos n(\theta_i+\theta_j) + \cos n(\theta_i-\theta_j). \end{aligned}$$

We now use the following well-known formula for cosines:

$$\sum_{k=n}^{n+N-1} \cos k\varphi = \frac{\sin\frac{N\varphi}{2}\cos\frac{N+2n-1}{2}\varphi}{\sin\frac{\varphi}{2}}.$$

This shows that either $\sin\varphi/2 = 0$ and $\sum_{k=n}^{n+N-1} \cos k\varphi = N$ or

$$\left|\sum_{k=n}^{n+N-1} \cos k\varphi\right| \le \frac{1}{\sin\frac{\varphi}{2}}. \tag{5}$$

We have that

$$\begin{aligned} g_N(n) = \sum_{k=n}^{n+N-1} f(k)^2/N &= \frac{m}{2} + \frac{1}{N}\sum_{k=n}^{n+N-1}\frac{1}{2}\sum_{i=1}^{m}\cos 2k\theta_i \\ &+ \frac{1}{N}\sum_{k=n}^{n+N-1}\sum_{i<j}(\cos k(\theta_i+\theta_j) + \cos k(\theta_i-\theta_j)). \end{aligned}$$

Changing the order of summation, together with (5), shows that there is an integer N such that $g_N(n) > m/4$ for all integers n. But then there is an integer n in every set of N consecutive integers such that $f(n)^2 > m/4$; that is, $|f(n)| > \sqrt{m}/2$, which proves the lemma. □

It might seem natural to assume that the k-of-each problem for a fixed k will become more difficult to solve when n increases, i.e., that the depth of an optimal tree increases monotonically with n. The following lemma gives almost monotonicity in a certain sense, enough for our purpose.

Monotonicity lemma. *The depth of an optimal linear decision tree (degree-d algebraic decision tree, algebraic computation tree) for $V_{n,k}$ is at most $2n$ more than the depth of a linear tree (degree-d algebraic decision tree, algebraic computation tree) for $V_{n+r,k}$ for all $r \geq k$.*

Proof. Case 1. Linear decision tree or degree-d algebraic decision tree.

Given $\mathbf{x} \in \mathbb{R}^n$, it suffices with $n-1$ comparisons to find the largest coordinate of $\mathbf{x}$, and hence there is a linear tree S of depth $n-1$ that can find the position i of the largest coordinate x_i. At every leaf of S we place a modified version of an optimal tree for $V_{n+r,k}$, where we have done the substitution $x_{n+1} = x_{n+2} = \cdots = x_{n+r} = x_i + 1$ with i being the position of the largest coordinate corresponding to that leaf. This substitution does not alter the degree of the tree and is hence legal. And since $x_i + 1$ is larger than all the coordinates in $\mathbf{x}$, the tree will give the correct answer.

Case 2. Algebraic computation tree.

To test if $x_i - x_j$ is less than, equal to, or larger than zero, we first do the subtraction in an arithmetic node and then the test in the next node. Hence a tree with depth $2n-2$ is sufficient to find the largest coordinate i. We also need an extra node to do the calculation $x_i + 1$. Then we proceed as in Case 1. Altogether we get a tree with depth $C(V_{n+r,k}) + 2n - 2 + 1 < C(V_{n+r,k}) + 2n$. $\square$

5. Main theorems. We will start by proving the lower bounds for the k-of-each problem. When $T = \mathbb{Z}_+ \backslash \{1, 2, \ldots, k-1\}$, let $\mu_{n,k}$ denote $\mu_{n,T}$, $\mu_k(n)$ denote $\mu_T(n)$, and so on. In $\Pi_{n,k}$ we have that $s_k(n) = 1$ for all $n < k$, so we get from Proposition 2 that

$$F_k(x) = -\ln(e^x - p_k(x)),$$

where $p_k(x) = \sum_{n=1}^{k-1} \frac{x^n}{n!}$. Now we have come to the main theorem. It says that the algorithm in section 2 is (up to a constant) the fastest possible in the worst case.

THEOREM 1. *The depth of a degree-d algebraic decision tree or of an algebraic computation tree for the k-of-each problem will be at least*

$$\Omega\left(n \log n\right).$$

Given $\epsilon > 0$ and k, there is a number $N_{k,\epsilon}$ such that the depth of a linear tree solving the k-of-each problem for $n > N_{k,\epsilon}$ is bounded below by

$$(1 - \epsilon) n \log_3 n.$$

Proof. We will use the results on $F_k(z)$, the exponential generating function for $\mu_k(n)$ found in section 3, to estimate $|\mu_k(n)|$. The theorem will then follow from Theorems B and A.

Let R_k denote the radius of convergence for $F_k(z)$ considered as a function on $\mathbb{C}$. It is well known from analysis that $\frac{1}{R_k} = \overline{\lim}\left(\frac{|\mu_k(n)|}{n!}\right)^{1/n}$. Let $z_1, \bar{z}_1, \ldots, z_t, \bar{z}_t$ denote the nonreal zeros of $e^z - p_k(z)$ with modulus R_k. Since $e^z - p_k(z) = 1 + z^k/k! + z^{k+1}/(k+1)! + \cdots$ there are no positive real zeros, but $-R_k$ might be a zero. Let $\delta = 1$ if this is the case; otherwise, let $\delta = 0$. Write $z_j = R_k e^{i\theta_j}$ with $0 < \theta_j < \pi$ for $j = 1, \ldots, t$. Observe that there cannot be other zeros with modulus arbitrarily close to R_k, since an entire function with an accumulation point of zeros has to be identically zero. So we can speak of the next zero which will have strictly larger modulus than R_k. Let R' denote this value (it might be infinity), which will be the radius of convergence of

$$\ln\left(\frac{e^z - p_k(z)}{(z - (-R_k))^{\delta}\Pi_{j=1}^{t}(z - z_j)(z - \bar{z}_j)}\right) =: \sum_n b_n z^n.$$

As long as we are only dealing with a real power series with a nonzero constant there is no problem using the laws of logarithm. But when it comes to separating $(z - z_j)$ from $(z - \bar{z}_j)$ we have to take care. However, with the usual branchcut along the negative real axis the following calculations are valid when z is a real number $0 < z < R_k$.

$$\begin{aligned}
&\ln(z^2 - 2\mathrm{Re}(z_j)z + R_k^2) = \ln((z + R_k e^{i(\theta_j - \pi)})(z + R_k e^{-i(\theta_j - \pi)}))\\
&= \ln(z + R_k e^{i(\theta_j - \pi)}) + \ln(z + R_k e^{-i(\theta_j - \pi)})\\
&= \ln R_k e^{i(\theta_j - \pi)} + \ln\left(\frac{z}{R_k e^{i(\theta_j - \pi)}} + 1\right)\\
&+ \ln R_k e^{-i(\theta_j - \pi)} + \ln\left(\frac{z}{R_k e^{-i(\theta_j - \pi)}} + 1\right)\\
&= \ln R_k^2 + \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n}\left(-\frac{z}{z_j}\right)^n + \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n}\left(-\frac{z}{\bar{z}_j}\right)^n\\
&= \ln R_k^2 - \sum_{n=1}^{\infty} 2\mathrm{Re}\left(\frac{1}{z_j}\right)^n \frac{z^n}{n}.
\end{aligned}$$

The calculations for a real zero are easier and all together we get

$$\begin{aligned}
&\ln(e^z - p_k(z)) = \sum_{j=1}^{t} \ln(z^2 - 2\mathrm{Re}(z_j)z + R_k^2) + \delta \ln(z - (-R_k))\\
&+ \ln\left(\frac{e^z - p_k(z)}{(z - (-R_k))^{\delta}\Pi_{j=1}^{t}(z - z_j)(z - \bar{z}_j)}\right)\\
&= \sum_{j=1}^{t}\left(\ln R_k^2 - \sum_{n=1}^{\infty} 2\mathrm{Re}\left(\frac{1}{z_j}\right)^n \frac{z^n}{n}\right) + \delta\left(\ln R_k - \sum_{n=1}^{\infty}\left(-\frac{1}{R_k}\right)^n \frac{z^n}{n}\right) + \sum_n b_n z^n.
\end{aligned}$$

Note that since $\frac{1}{R'} = \overline{\lim}\,(|b_n|)^{1/n}$ we get for any c with $1 < c < \frac{R'}{R_k}$ that there exists a number c_k such that $n > c_k \implies |b_n| \le \frac{1}{c^n R_k^n}$. By comparing coefficients we get the following bound for sufficiently large n:

$$\begin{aligned}
\frac{|\mu_k(n)|}{n!} &\ge \frac{|2\sum_{j=1}^{t}\cos(-n\theta_j) + \delta\cos(-n\pi)|}{nR_k^n} - |b_n|\\
&\ge \frac{1}{R_k^n}\left(\frac{|2\sum_{j=1}^{t}\cos(-n\theta_j) + \delta\cos(-n\pi)|}{n} - \frac{1}{c^n}\right).
\end{aligned}$$

Now we need to estimate the sum of cosines from below and R_k from above. The Modulus Lemma shows that $R_k < 9k$. The Lemma of Cosines with $m = 2t$ or $2t+1$ shows that there exists a number M_k such that in any set of M_k consecutive numbers there is an integer n such that $|2\sum_{j=1}^{t}\cos(-n\theta_j) + \delta\cos(-n\pi)| > 1/2$.

Let c'_k be such that $\frac{1}{c^n} < \frac{1}{4n}$ whenever $n > c'_k$. Using the lemmas above we get that for every integer $n > \max\{c'_k, c_k, M_k + k + 2\}$ there is an integer m such that $k < n - m < M_k + k$ and

$$|\mu_k(m)| \ge \frac{m!}{4m(9k)^m} \ge \left(\frac{m}{3}\right)^m \frac{1}{4m(9k)^m}$$

$$\implies \log_3|\mu_k(m)| > m\log_3\frac{m}{27k} - m = m\log_3 m - (4 + \log_3 k)m.$$

The last tool we need is the Monotonicity Lemma which says that the depth of an algebraic decision or algebraic computation tree is almost monotone with respect to

n. Since $|\mu_k(n)|$ is one of the terms in Theorem B and the other terms in the sum are all positive, we get

$$\begin{aligned} C(V_{n,k}) \geq C(V_{m,k}) - 2m \geq \alpha \log_3 |\mu_k(m)| - (\beta+2)m \geq \alpha m \log_3 m - (\beta+6+\log_3 k)m \\ \geq \alpha(n-k-M_k)\log_3(n-k-M_k) - (\beta+6+\log_3 k)n \\ \geq \alpha n \log_3 n - \beta' n \end{aligned}$$

for some constants $0 < \alpha \leq 1$ and β'. The last step is using $\log(n-k-M_k) \geq \log n - \log(k+M_k)$, which is true since $n > M_k + k + 2 > 4$. The above estimation is valid also for $C_d(V_{n,k})$, so the first part of the theorem follows.

To prove the second part let $n \geq \max\{2(M_k+k)/\epsilon, c'_k, c_k\}$ and let m be as above. From Theorem A and the Monotonicity Lemma we get

$$\begin{aligned} C_1(V_{n,k}) &> \log_3 |\mu_k(m)| - 2m > m \log_3 m - (6+\log_3 k)m \\ &> \left(1-\frac{\epsilon}{2}\right) n \log_3\left(\left(1-\frac{\epsilon}{2}\right) n\right) - (6+\log_3 k)n \\ &> \left(1-\frac{\epsilon}{2}\right) n \log_3 n - \left(6+\log_3 k - \log_3\left(1-\frac{\epsilon}{2}\right)\right)n. \end{aligned}$$

Choose $N_{k,\epsilon} \geq \max\{2(M_k+k)/\epsilon, c'_k, c_k\}$ and also large enough for $\frac{\epsilon}{2}\log_3 n > 6 + \log_3(1+\frac{\epsilon}{2})k$ to be true for all $n \geq N_{k,\epsilon}$. The second part of the theorem follows. □

Given a set $T \subseteq \mathbb{Z}_+$ and a list of n numbers, we can consider the problem of deciding whether every number in the list occurs with a multiplicity m such that $m \in T$. We will call this the *T-multiplicity problem.* This is a more general problem containing the k-of-each problem as the special case when $T = \{k, k+1, k+2, \dots\}$. We will say that T is *additive* if $a, b \in T$ implies $a + b \in T$. Without any extra effort we can now get the following theorem.

THEOREM 2. *Let $T \subseteq \mathbb{Z}_+$ be an additive and cofinite set not containing* 1. *Then the depth of a degree-d algebraic decision or algebraic computation tree for the T-multiplicity problem will be at least*

$$\Omega\left(n \log n\right).$$

Given $\epsilon > 0$ there is a number $N_{T,\epsilon}$ such that the depth of a linear tree solving the T-multiplicity problem is bounded below by

$$(1-\epsilon) n \log_3 n.$$

Proof. It is not difficult to verify that since T is additive it corresponds to a subspace arrangement with $\Pi_{n,T}$ as an intersection lattice, i.e., a generalization of Proposition 1. Since T is cofinite, we get that $\max \mathbb{Z}\backslash T$ is finite. It is the degree of the polynomial $p_T(x)$ in $F_T(x) = -\ln(e^x - p_T(x))$; hence we can use the Modulus Lemma. The Monotonicity Lemma is also still valid with k replaced by $\max \mathbb{Z}\backslash T$. The theorems of Björner, Lovász, and Yao give suitable generalizations of Theorems A and B. Hence we can apply the same proof as for Theorem 1. □

If $1 \in T$ then the only case when the T-multiplicity problem gives a subspace arrangement is when $T = \{1, k, k+1, k+2, \dots\}$ which is the k-equal problem. The above proof is valid also in this case. Note that the algorithm of section 2 will (with obvious modifications) still work for the T-multiplicity problem for arbitrary T, but here k is the smallest number in T.

6. Remarks and open problems.

Remark 1. The Modulus Lemma is not needed for odd values of k since then there is a real root. This means that one could prove the theorem for odd k more easily. And for even k one can, by doubling the input, turn it into the $(2k-1)$-of-each problem. This way one could prove Theorem 1 without the Modulus Lemma but get a constant one half in front of the lower bound. But this proof would not work for Theorem 2.

Remark 2. Another interesting invariant for the partition poset $\Pi_{n,T}$ is the characteristic polynomial $\phi_T(n;t) = \sum_{\pi\in\Pi_n} \mu_T(\hat{0},\pi)t^{b(\pi)}$, where $b(\pi)$ denotes the number of blocks in π. Also let $\phi_T(0;t) = 1$. The method of section 3 can be used to calculate its exponential generating function $G_T(x,t) = \sum_{n=0}^{\infty} \phi_T(n;t)\frac{x^n}{n!}$.

$$G_T(x,t) = \left(1 + \sum_{n\in\mathbb{Z}_+\setminus(T\setminus\{1\})} s_T(n)\frac{x^n}{n!}\right)^t \qquad \text{if } 1 \in T,$$

$$G_T(x,t) = e^{tx} + 1 - \left(e^x - \sum_{n\in\mathbb{Z}_+\setminus T} s_T(n)\frac{x^n}{n!}\right)^t \qquad \text{if } 1 \notin T.$$

The first formula can also be found in [BL].

Problem 1. As noted, the algorithm works for any T. What about the lower bound for general T? The so-called k-divisibility problem is a T-multiplicity problem with $T = \{k, 2k, 3k, \dots\}$. In [BL] it is shown to have $\Omega(n\log n)$ as lower bound. Here T is additive but not cofinite. When is cofiniteness a necessary condition? Does there exist any nontrivial set T such that the T-multiplicity problem can be solved faster than $n\log n$?

Problem 2. A referee asked if it is possible to make the lower bound for linear trees solving the k-of-each problem uniform in k, i.e., if $\forall\epsilon > 0, \exists N_\epsilon$ such that $\forall n > N_\epsilon \forall k$ $C_1(V_{n,k}) \geq (1-\epsilon)n\log_3(n/k)$. This would indeed be an interesting sharpening of the result. To this end one would need a more detailed analysis of the zeros of $e^z - p_k(z)$ to determine if estimates of the constants c_k, c'_k, and M_k that are independent of k exist. For small values of k numerical tests suggest that there are $k-1$ zeros with the same modulus and no other zeros. This would imply that c_k and c'_k are small and independent of k.

Problem 3. The algorithm solves the k-of-each problem in $n\log_2 n + (3-\log_2 k)n$ steps, and the theorem gives $(1-\epsilon)n\log_3 n$ as a lower bound for linear decision trees. Is it possible to sharpen the lower bound to binary logarithm?

Acknowledgments. I thank my research adviser Anders Björner who suggested the problem and helped me all the way through. I am also very grateful to Michael Benedicks, Johan Håstad, Mattias Jonsson, Pavel Valter, and anonymous referees for invaluable suggestions and discussions.

REFERENCES

[B1] A. Björner, *Subspace arrangements*, in Proc. 1st European Congress of Mathematics, Paris, 1992, A. Joseph et al., eds., Birkhäuser Boston, Cambridge, MA, 1994, pp. 321–370.

[B2] A. Björner, *Nonpure shelling, f-vectors, subspace arrangements and complexity*, Discrete Math., in Proc. 6th Formal Power Series and Algebraic Combinatorics (Dimacs 1994), to appear.

[BL] A. Björner and L. Lovász, *Linear decision trees, subspace arrangements and Möbius functions*, J. Amer. Math. Soc., 7 (1994), pp. 677–706.

[BLY] A. Björner, L. Lovász, and A. Yao, *Linear decision trees: Volume estimates and topological bounds*, in Proc. 24th Annual ACM Symp. on Theory of Computing, ACM Press, New York, 1992, pp. 170–177.

[H] J. A. Hempel, *The Poincaré metric on the twice punctured plane and the theorems of Landau and Schottky*, J. London Math. Soc., 20 (1979), pp. 435–445.

[J] J. A. Jenkins, *On explicit bounds in Landau's theorem* 2, Canad. J. Math., 33 (1981), pp. 559–562.

[OT] P. Orlik and H. Terao, *Arrangements of Hyperplanes*, Springer-Verlag, Berlin, New York, 1992.

[S1] R. Stanley, *Enumerative Combinatorics Vol.* 1, Wadsworth & Brooks/Cole, Pacific Grove, CA, 1986.

[S2] R. Stanley, *Generating functions*, in Studies in Combinatorics, G.-C. Rota, ed., MAA Studies in Mathematics, 1978, pp. 100–141.

[Y1] A. Yao, *Algebraic decision trees and Euler characteristics*, in Proc. 33rd Annual IEEE Symposium on Foundations of Computer Science, October, 1992, pp. 268–277.

[Y2] A. Yao, *Decision tree complexity and Betti numbers*, in Proc. 26th Annual ACM Symposium on Theory of Computing, ACM Press, New York, 1994, pp. 615–624.

SIAM J. Discrete Math.
Vol. 10, No. 1, pp. 30–40, February 1997

ISOPERIMETRIC INEQUALITIES AND EIGENVALUES*

NABIL KAHALE†

Abstract. An upper bound is given on the minimum distance between i subsets of same size of a regular graph in terms of the ith largest eigenvalue in absolute value. This yields a bound on the diameter in terms of the ith largest eigenvalue for any integer i. Our bounds are shown to be asymptotically tight for explicit families of graphs having an asymptotically optimal ith largest eigenvalue. A result by Quenell [*Adv. Math.*, 106 (1994), pp. 122–148] relating the diameter, the second eigenvalue, and the girth of a regular graph is obtained as a by-product.

Key words. eigenvalues, diameter, Chebychev polynomials, expanders

AMS subject classification. 05C35

PII. S0895480193259825

1. Introduction.

Many combinatorial properties of a graph are related to the spectrum of its adjacency matrix [2, 3, 4, 17]. The adjacency matrix A of an undirected graph is the 0–1 matrix indexed by the vertices and such that the entry (u, v) is equal to 1 if and only if (u, v) is an edge. Since the adjacency matrix of any graph H on n vertices is symmetric and real, its eigenvalues are real and will be denoted by $\lambda_0(H) \geq \lambda_1(H) \geq \cdots \geq \lambda_{n-1}(H)$. In this paper, we explore the relation between the spectrum of a graph and its isoperimetric properties. We focus our attention on the diameter, which is defined to be the maximum distance in H between any pair of vertices, that will be denoted by $D(H)$. The diameter plays an important role in network design in parallel and distributed computing.

Let $\lambda = \lambda(H) = \max(\lambda_1, |\lambda_{n-1}|)$. It is known that if a graph is k-regular, then $\lambda_0 = k$ and $\lambda \leq k$, with equality if and only if the graph is disconnected or bipartite. Moreover, the graph is an expander if and only if [2] there exists a gap between k and λ_1. Thus, the existence of an upper bound on the diameter in terms of the eigenvalue gap is not surprising. Such a bound first appeared in [3], where it was shown that, when G is k-regular,

$$D(G) \leq 2\sqrt{\frac{2k}{(k-\lambda_1)}}\log_2 n. \tag{1}$$

Chung [5] (see also [12]) established that

$$D(G) \leq \left\lfloor \frac{\log(n-1)}{\log(k/\lambda)} \right\rfloor + 1, \tag{2}$$

* Received by the editors December 9, 1993; accepted for publication (in revised form) February 1, 1996. This research was partly conducted when the author was at the Massachusetts Institute of Technology and partly when he was at the Center for Discrete Mathematics and Theoretical Computer Sciences. This research was supported in part by the Defense Advanced Research Projects Agency contracts N00014-92-J-1799 and N00014-91-J-1698, the Air Force contract F49620-92-J-0125, and the Army contract DAAL-03-86-K-0171. This paper is partly based on *Better expansion for Ramanujan graphs*, by Nabil Kahale, which appeared in the 32nd Annual Symposium on Foundations of Computer Science, IEEE, San Juan, Puerto Rico, October 1–4, 1991, pp. 398–404, and on *On the second eigenvalue and linear expansion of regular graphs*, by Nabil Kahale, which appeared in DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 10, American Mathematical Society, Providence, RI, 1993, pp. 49–62.

http://www.siam.org/journals/sidma/10-1/25982.html

† AT&T Bell Laboratories, Murray Hill, NJ 07974 (kahale@research.att.com).

which beats (1) when λ is small. Equation (2) was further improved in [6, 16], where it was shown that

$$D(G) \leq \left\lfloor \frac{\cosh^{-1}(n-1)}{\cosh^{-1}(k/\lambda)} \right\rfloor + 1. \tag{3}$$

In this paper, we establish isoperimetric bounds that are a function of the subsequent eigenvalues and do not depend on the second eigenvalue. More precisely, we have Theorem 1.1.

Theorem 1.1. *Let $G = (V, E)$ be an undirected k-regular graph and $\delta_0, \delta_1, \ldots, \delta_{n-1}$ be the eigenvalues of its adjacency matrix, with $|\delta_0| \geq |\delta_1| \geq \cdots \geq |\delta_{n-1}|$. Let $d(X, Y)$ denote the distance between two subsets X and Y. If $|\delta_i| < k$ and $X_1, X_2, \ldots, X_{i+1}$ are $i+1$ subsets of V of same cardinality xn, then*

$$\min_{1 \leq j < h \leq i+1} d(X_j, X_h) \leq \left\lceil \frac{\cosh^{-1}(x^{-1}-1)}{\cosh^{-1}(k/|\delta_i|)} \right\rceil + 1. \qquad \square$$

Equation (2) is (up to an additive constant 1) a particular case of Theorem 1.1. We will use Theorem 1.1 to derive upper bounds on the diameter of G in terms of δ_i. In section 3, we establish a lower bound on the size of $N^t(X)$, where X is a set of vertices and $N^t(X)$ is the set of nodes that can be reached from X by a path of length t; that is,

$$\underbrace{N(N(\ldots N(}_{t \text{ times}}X))\ldots).$$

The lower bound on $|N^t(X)|$ is a function of the size of X and of the second eigenvalue in absolute value λ of the graph. As a first corollary, we obtain an upper bound on the distance between two subsets of a given size. As a second corollary, we get a simple proof of a recent result [15] relating the diameter, the girth, and λ. Section 3 combines ideas in [3, 12, 17]. In section 4, we prove Theorem 1.1 and derive a relation between the diameter of a graph and its subsequent eigenvalues.

In section 5, we study the tightness of the aforementioned bounds. For fixed n and k, the right-hand side of (3) is small when λ is small. It is known, however, that for any sequence $G_{n,k}$ of k-regular graphs on n vertices, $\liminf \lambda(G_{n,k}) \geq 2\sqrt{k-1}$ as n goes to infinity [2, 12, 14]. A Ramanujan graph is a k-regular graph where all eigenvalues not equal to $\pm k$ are at most $2\sqrt{k-1}$ in absolute value. Ramanujan graphs have been constructed explicitly in [12, 13]. It is known [12] (and in the non-bipartite case follows from (3)) that the diameter of a k-regular Ramanujan graph on n vertices is at most $(2+o(1))\log_{k-1} n$. On the other hand, it is easy to see that it is at least $(1+o(1))\log_{k-1} n$. To the best of our knowledge, these are the best-known asymptotic bounds on the diameter of the known explicit families of Ramanujan graphs [12, 13]. In section 5, for many integers k, we construct explicitly a family of k-regular graphs with $\lambda = (2+o(1))\sqrt{k-1}$ and diameter $(2+o(1))\log_{k-1} n$. We generalize our construction to show that our bound on the diameter in terms of δ_i is asymptotically tight for explicit families of graphs having an asymptotically optimal ith largest eigenvalue.

Section 3 is based on [9]. A longer version of the paper appears in [10].

2. Notation and background. Let $G = (V, E)$ be an undirected k-regular graph on n vertices. Denote by $L^2(V)$ the set of real-valued functions on V and

$L_0^2(V) = \{f \in L^2(V);\ \sum_{v\in V} f(v) = 0\}$. As usual, we define the scalar product of two vectors f and g of $L^2(V)$ by

$$f \cdot g = \sum_{v \in V} f(v) g(v)$$

and the Euclidean norm of a vector f by $||f|| = \sqrt{f \cdot f}$. The adjacency matrix A of G defines a linear operator in $L^2(V)$ that maps every vector $f \in L^2(V)$ to the vector Af defined by

$$(Af)(v) = \sum_{(v,w)\in E} f(w). \tag{4}$$

This operator is self-adjoint since $\forall f, g \in L^2(V)$, we have

$$(Af) \cdot g = f \cdot (Ag) = \sum_{(v,w)\in E} f(v) g(w). \tag{5}$$

For any subset W of V, we denote by χ_W the characteristic vector of W:

$$\chi_W(v) = \begin{cases} 1 & \text{if } v \in W, \\ 0 & \text{otherwise.} \end{cases}$$

The support of a vector $f \in L^2(V)$ is defined to be the set of nodes v for which $f(v) \neq 0$. We sometimes order the eigenvalues of a graph H according to their absolute values and denote them by $\delta_i(H)$, so that $|\delta_0(H)| \geq |\delta_1(H)| \geq \cdots \geq |\delta_{n-1}(H)|$. Denote by $\lambda_i(B)$ the $(i+1)$st largest eigenvalue of a matrix B with real eigenvalues. The l_1-norm $||h||_1$ of a vector h is defined to be $\sum_{v\in V} |h(v)|$.

The Chebychev polynomial of degree t is the unique polynomial P_t satisfying the equation

$$P_t(\cosh z) = \cosh(tz) \tag{6}$$

for any complex number z. Chebychev polynomials have been used in [12] in the study of expanders. The following facts easily follow from (6).

Fact 1. *For any complex number z, we have $P_t(-z) = (-1)^t P_t(z)$.* □

Fact 2. *For any real number s between -1 and 1, we have $|P_t(s)| \leq 1$.* □

3. Bounds on the distance between two subsets. The following theorem generalizes a result of Tanner [17]. We use a similar proof technique.

Theorem 3.1. *Let $G = (V, E)$ be a k-regular graph on n vertices and λ its second largest eigenvalue in absolute value. For any subset X of V and any integer $t \geq 1$, we have*

$$|N^t(X)| \geq \frac{{P_t}^2(k/\lambda)|X|}{1 + ({P_t}^2(k/\lambda) - 1)|X|/n}. \tag{7}$$

If G is a nonbipartite Ramanujan graph of degree k, then

$$\frac{|N^t(X)|}{|X|} \geq \frac{((k-1)^t + 1)^2}{4(k-1)^t + ((k-1)^t - 1)^2 |X|/n} \geq \frac{(k-1)^t}{4 + (k-1)^t |X|/n}.$$

In particular, if $|X|/n$ is at most $4(k-1)^{-t-1}$, then $|N^t(X)| \geq (k-2)(k-1)^{t-1}|X|/4$.

Proof. Denote by f the characteristic vector of X. Let $f = \bar{f} + f_0$, where $\bar{f}$ is a constant vector and $f_0 \in L_0^2(V)$. It follows from Fact 1 that P_t is of the form $P_t(s) = c_t s^t + c_{t-2} s^{t-2} + \cdots$, and so $P_t(\lambda^{-1}A)\ f = c_t \lambda^{-t} A^t f + c_{t-2}\lambda^{-(t-2)} A^{t-2} f + \cdots$. This implies that the support of the vector $g = P_t(\lambda^{-1}A)\ f$ is a subset of $N^t(X)$. This is because the support of the vector $A^l f$ is $N^l(X)$, and $N^t(X) \supseteq N^{t-2}(X) \supseteq \cdots$. We will obtain a lower bound on the size of $N^t(X)$ by comparing the norm of g with its sum of coordinates. Since $A\bar{f} = k\bar{f}$, we have

$$g = P_t(\lambda^{-1}A)\ \bar{f} + P_t(\lambda^{-1}A)\ f_0 = P_t(k/\lambda)\ \bar{f} + P_t(\lambda^{-1}A)\ f_0.$$

Equation (4) shows that $L_0^2(V)$ is invariant under A, and so $P_t(\lambda^{-1}A)\ f_0 \in L_0^2(V)$. The eigenvalues of the restriction $A_{|L_0^2(V)}$ of A to $L_0^2(V)$ are λ_i for $1 \le i \le n-1$. By the Pythagorean theorem, we have

$$||g||^2 = {P_t}^2(k/\lambda)||\bar{f}||^2 + ||P_t(\lambda^{-1}A)\ f_0||^2 \le {P_t}^2(k/\lambda)||\bar{f}||^2 + ||f_0||^2.$$

The second inequality follows from the fact that the operator $P_t(\lambda^{-1}A_{|L_0^2(V)})$ is self-adjoint and its eigenvalues $P_t(\lambda_i/\lambda)$, $1 \le i \le n-1$, are at most 1 in absolute value (Fact 2). It follows from the Cauchy–Schwarz inequality that

$$(8) \qquad \begin{aligned} |\chi_{N^t(X)} \cdot g|^2 &\le |N^t(X)|\ ({P_t}^2(k/\lambda)||\bar{f}||^2 + ||f_0||^2) \\ &= |N^t(X)|\ ({P_t}^2(k/\lambda)||\bar{f}||^2 + |X| - ||\bar{f}||^2). \end{aligned}$$

The sum of coordinates $\chi_{N^t(X)} \cdot g$ of g is equal to $P_t(k/\lambda)|X|$. This is because the sum of coordinates of Ah is equal to k times the sum of coordinates of h, as follows immediately from (4). By replacing the terms $\chi_{N^t(X)} \cdot g$ and $||\bar{f}||$ by their values in (8), we get

$$(9) \qquad \frac{|X|}{|N^t(X)|} \le \frac{|X|}{n} + \frac{1 - |X|/n}{{P_t}^2(k/\lambda)},$$

which implies (7).

If G is a nonbipartite Ramanujan graph, we can replace λ by $2\sqrt{k-1}$ in (7). We have

$$\begin{aligned} P_t\left(\frac{k}{2\sqrt{k-1}}\right) &= P_t\left(\cosh\left(\frac{\ln(k-1)}{2}\right)\right) \\ &= \cosh\left(t\frac{\ln(k-1)}{2}\right) \\ &= \frac{(k-1)^{t/2} + (k-1)^{-t/2}}{2}. \end{aligned}$$

The rest of the theorem follows by an easy calculation. □

COROLLARY 3.2. *If G is nonbipartite and X and Y are two subsets of V of cardinality xn and yn, respectively,*

$$d(X,Y) \le \left\lfloor \frac{\cosh^{-1}\sqrt{(x^{-1}-1)(y^{-1}-1)}}{\cosh^{-1}(k/\lambda)} \right\rfloor + 1.$$

Proof. If t is an integer such that the right-hand side of (9) is less than $|X|/(n-|Y|)$, then $|N^t(X)| > n-|Y|$, and so the distance between X and Y is at most t. Let $\theta = \cosh^{-1}(k/\lambda)$, so that $P_t(k/\lambda) = \cosh(t\theta)$. We want t to be such that

$$x + \frac{1-x}{\cosh^2(t\theta)} < \frac{x}{1-y}.$$

Solving for t yields the desired bound on $d(X, Y)$. □

By applying Corollary 3.2 to any pair of subsets consisting of single vertices, we obtain (3), which has already been established in [6, 16].

COROLLARY 3.3 (see [15]). *If G is nonbipartite,*

$$D(G) \leq \left\lfloor \frac{\cosh^{-1}\left(\frac{n}{k(k-1)^{r-1}} - 1\right)}{\cosh^{-1}(k/\lambda)} \right\rfloor + 2r + 1,$$

where $r = \lfloor (c(G)-1)/2 \rfloor$ is the injectivity radius *of G.*

Proof. We remind the reader that $c(G)$ is the girth of G. Consider any pair of vertices u and v. The subsets $N^r(\{u\})$ and $N^r(\{v\})$ have size $k(k-1)^{r-1}$. By applying Corollary 3.2 to these subsets, we get

$$d(N^r(\{u\}), N^r(\{v\})) \leq \left\lfloor \frac{\cosh^{-1}\left(\frac{n}{k(k-1)^{r-1}} - 1\right)}{\cosh^{-1}(k/\lambda)} \right\rfloor + 1.$$

Corollary 3.3 follows immediately. □

Corollary 3.3 had first been established by Quenell [15].

4. Relation with subsequent eigenvalues. We now show the following special case of Theorem 1.1.

THEOREM 4.1. *If $G = (V, E)$ is a k-regular graph and $|\delta_i| < k$, then for any set S of $i+1$ vertices of V,*

$$\min_{\{u,v\}\subset S, u\neq v} d(u,v) \leq \left\lceil \frac{\cosh^{-1}(n-1)}{\cosh^{-1}(k/|\delta_i|)} \right\rceil + 1.$$

Proof. Let e_j be an eigenvector of A corresponding to the eigenvalue δ_j, and let $f \in L^2(V)$ be a nonzero function null on $V - S$ such that f belongs to the vector space E_i spanned by $e_i, e_{i+1}, \ldots, e_{n-1}$. The existence of f follows from the fact that $\dim L^2(S) = i+1$ and $\dim E_i = n-i$. Given an integer t, let $g = P_t(|\delta_i|^{-1}A)f$. The vector space E_i is invariant under A, and the eigenvalues of the restriction of A to E_i are δ_h for $i \leq h \leq n-1$. By a reasoning similar to the proof of Theorem 3.1 the eigenvalues $P_t(\delta_h/|\delta_i|)$, for $i \leq h \leq n-1$, of the restriction of the operator $P_t(|\delta_i|^{-1}A)$ to E_i are at most 1 in absolute value, and so $||g|| \leq ||f||$. Assume now that $\min_{\{u,v\}\subset S, u\neq v} d(u,v) > 2t$. Then the vectors $P_t(|\delta_i|^{-1}A)\chi_{\{u\}}$, for $u \in S$, have disjoint supports, and so

$$\begin{aligned} ||g||_1 &= \left\| \sum_{u\in S} f(u)\ P_t(|\delta_i|^{-1}A)\chi_{\{u\}} \right\|_1 \\ &= \sum_{u\in S} |f(u)|\ ||P_t(|\delta_i|^{-1}A)\chi_{\{u\}}||_1 \end{aligned} \tag{10}$$

$$\begin{aligned} &\geq \sum_{u\in S} |f(u)| P_t(k/|\delta_i|) \\ &= P_t(k/|\delta_i|)||f||_1. \end{aligned}$$

The third equation follows from the fact that the sum of coordinates of the vector $P_t(|\delta_i|^{-1}A)\chi_{\{u\}}$ is $P_t(k/|\delta_i|)$.

On the other hand,

$$\begin{aligned} ||g||_1 &\leq \sqrt{n}||g|| \\ &\leq \sqrt{n}||f|| \\ &\leq \sqrt{n/2}||f||_1. \end{aligned} \tag{11}$$

The first inequality is a consequence of the Cauchy–Schwarz inequality. The last inequality is valid because $f \in L_0^2(V)$. Indeed,

$$\begin{aligned} ||f||^2 &= ||f^+||^2 + ||f^-||^2 \\ &\leq ||f^+||_1^2 + ||f^-||_1^2 \\ &= \frac{||f||_1^2}{2}, \end{aligned}$$

where $f^+ = \max(f, 0)$ and $f^- = \min(f, 0)$.

Combining (10) and (11) shows that

$$P_t(k/|\delta_i|) \leq \sqrt{n/2}. \tag{12}$$

Equation (12) does not hold for $t = \lfloor l \rfloor + 1$, where

$$l = \frac{\cosh^{-1}\sqrt{n/2}}{\cosh^{-1}(k/|\delta_i|)},$$

and so $\min_{\{u,v\}\subset S, u\neq v} d(u,v) \leq 2\lfloor l \rfloor + 2$.

This bound can be slightly improved when l is an integer. Indeed, let $t = l$ and assume as before that $\min_{\{u,v\}\subset S, u\neq v} d(u,v) > 2l$. Since $P_l(k/|\delta_i|) = \sqrt{n/2}$, all terms of (11) are equal (otherwise, (12) would be a strict inequality for $t = l$). This implies that the support of g is equal to V. It follows that every point in G is at distance at most l from some point in S, and so $\min_{\{u,v\}\subset S, u\neq v} d(u,v) \leq 2l + 1$. We conclude (whether l is an integer or not) that $\min_{\{u,v\}\subset S, u\neq v} d(u,v) \leq \lceil 2l \rceil + 1$. The lemma follows by noting that $2\cosh^{-1}\sqrt{n/2} = \cosh^{-1}(n-1)$. □

Theorem 1.1 can be shown in a similar way to Theorem 4.1. The main difference is that we consider a function f in $L^2(\cup_{j=1}^{i+1} X_j)$ which is constant on each X_j.

Corollary 4.2. *If G is a k-regular connected graph and $|\delta_i| < k$, where i is an integer between 1 and $n-1$,*

$$D(G) \leq i \left\lceil \frac{\cosh^{-1}(n-1)}{\cosh^{-1}(k/|\delta_i|)} \right\rceil + 2i - 1. \tag{13}$$

If r is the injectivity radius of G then

$$D(G) \leq i \left\lceil \frac{\cosh^{-1}(\frac{n}{k(k-1)^{r-1}} - 1)}{\cosh^{-1}(k/|\delta_i|)} \right\rceil + 2ir + 2i - 1.$$

Proof. Let u and v be two vertices at maximal distance in G. Consider a shortest path between u and v. There exists a sequence of $i+1$ vertices $u_0 = u, u_1, \ldots, u_i = v$ on this path at distance at least $\lfloor D(G)/i \rfloor$ from each other. By applying Theorem 4.1 to the set $\{u_0, u_1, \ldots, u_i\}$, we get the first bound on $D(G) = d(u,v)$. The second bound can be established in a similar fashion by applying Theorem 1.1 to the subsets $N^r(\{u_j\})$. □

5. Tightness of bounds. We show that, for any fixed i, (13) is asymptotically tight for certain families of k-regular graphs having asymptotically optimal $|\delta_i|$. We use techniques similar to [11]. We start with the case $i = 1$.

THEOREM 5.1. *For any integer k such that $k-1$ is prime congruent to* 1 *modulo* 4*, there exists an infinite explicit family of k-regular graphs G_n on n vertices with $\lambda(G_n) = (2+o(1))\sqrt{k-1}$ and diameter $(2+o(1))\log_{k-1} n$.*

Proof. Let H be a nonbipartite k-regular Ramanujan graph on n' vertices of girth at least $(2/3+o(1))\log_{k-1} n'$. Such a graph has been explicitly constructed in [12]. Consider two identical trees T and T' of depth $l = \lfloor \log_{k-1} m - 2 \rfloor$, where $m = \lfloor n'/\log n' \rfloor$ and whose internal nodes have degree k. All leaves in T and T' have the same depth, and H, T, and T' are disjoint. Let F be a set of edges in H at distance at least $r = \Omega(\log_{k-1}(n'/m))$ from each other and such that the number of edges in F is equal to the number of leaves in T (F can be found greedily). Identify one endpoint of each edge in F to a leaf in T and the other endpoint to a leaf in T' in such a way that all leaves of T and T' are identified to distinct vertices in H. By deleting the edges in F, we obtain a k-regular graph G on n vertices. The diameter of G is at least twice the depth of T, which is $(1+o(1))\log_{k-1} n$. We show that $\lambda(G) = (2+o(1))\sqrt{k-1}$. Equation (3) then implies that the diameter of G is equal to $(2+o(1))\log_{k-1} n$. We only need to show the upper bound on $\lambda' = \lambda(G)$, since $\lambda' \geq (2+o(1))\sqrt{k-1}$ for any family of k-regular graphs as the number of vertices goes to infinity [2, 12, 14]. Let A be the adjacency matrix of H and A' the adjacency matrix of G. We assume that $\lambda' > 2\sqrt{k-1}$ (otherwise we are done), and let $\lambda' = 2\sqrt{k-1}\cosh\theta'$, with $\theta' > 0$. We also assume that $\lambda' = \lambda_1(G)$. The case $\lambda' = -\lambda_{n-1}(G)$ can be treated similarly. Denote by $V(G)$, $V(H)$, $V(T)$, and $V(T')$ the vertex sets of G, H, T, and T', respectively.

Let $g \in L^2(V(G))$ be an eigenvector of A' corresponding to λ', and let $f \in L^2(V(H))$ be the vector of $L^2(V(H))$ that coincides with g on $V(H)$. By (5), we have

$$\begin{aligned} (14)\quad \lambda'||g||^2 &= g \cdot A'g \\ &= f \cdot Af - \sum_{(u,v)\in F} g(u)g(v) + \sum_{(u,v)\in E(T)} g(u)g(v) + \sum_{(u,v)\in E(T')} g(u)g(v) \\ &\leq f \cdot Af + (2\sqrt{k-1}+1) \sum_{v \in V(T)\cup V(T')} g(v)^2. \end{aligned}$$

The third equation follows from the fact that the largest eigenvalue of T is at most $2\sqrt{k-1}$. Since $g \in L_0^2(V(G))$,

$$\sum_{w\in V(H)} f(w) = - \sum_{w \in (V(T)\cup V(T'))-V(H)} g(w).$$

We need the following lemma whose proof can be found in [11].

LEMMA 5.2. *If $H = (V,E)$ is k-regular on n vertices, then for any $f \in L^2(V)$,* *we have*

$$f \cdot Af \leq \lambda_1(H)||f||^2 + \frac{k - \lambda_1(H)}{n} \left(\sum_{v \in V} f(v) \right)^2 . \qquad \square$$

Using Lemma 5.2 and the Cauchy–Schwarz inequality, we get

$$\begin{aligned} f \cdot Af &\leq \lambda_1(H)||f||^2 + \frac{k}{n} \left(\sum_{w \in (V(T) \cup V(T')) - V(H)} g(w) \right)^2 \\ &\leq 2\sqrt{k-1} \left(||g||^2 - \sum_{w \in (V(T) \cup V(T')) - V(H)} g(w)^2 \right) \\ &\quad + \frac{2km}{n} \sum_{w \in (V(T) \cup V(T')) - V(H)} g(w)^2 \\ &\leq 2\sqrt{k-1}||g||^2 \end{aligned}$$

for sufficiently large n. Combining this with (14) yields

$$\lambda'||g||^2 \leq 2\sqrt{k-1} \left(||g||^2 + 2 \sum_{v \in V(T) \cup V(T')} g(v)^2 \right) . \tag{15}$$

Next, we show that $\sum_{v \in V(T) \cup V(T')} g(v)^2$ is small compared with $||g||^2$. We use the following lemma whose proof is implicit in [11] and is given in detail in [10, sec. 5.3].

Lemma 5.3. *Let $G = (V, E)$ be a k-regular graph and g an eigenvector of G corresponding to the eigenvalue $2\sqrt{k-1}\cosh\theta$, with $\theta > 0$. If l and l' are two nonnegative integers with $l < l'$, and u is a node of G such that the subgraph induced on the set of nodes at distance at most l' from u is a tree, then*

$$\sum_{v \in V : d(u,v) = l} g(v)^2 \leq e^{-2(l'-l)\theta}||g||^2. \qquad \square$$

By applying Lemma 5.3 in the case where u is the root of T and $l' = l + r$, we see that

$$||g||^2 \geq e^{2r\theta'} \sum_{v \in V(T) : d(u,v) = l} g(v)^2.$$

By applying the lemma to $l-1, l-2, \ldots, 0$, we obtain

$$||g||^2 \geq e^{2r\theta'}(1 - e^{-2\theta'}) \sum_{v \in V(T)} g(v)^2.$$

Combining this with (15) yields

$$\cosh\theta' \leq 1 + 4\frac{e^{-2r\theta'}}{1 - e^{-2\theta'}}.$$

As a consequence, $\theta' \leq 2(\log r)/r$, for large n, and so $\lambda' \leq (2 + o(1))\sqrt{k-1}$. $\square$

It is known (see, e.g., [8]) that $|\delta_i(G_n)| \geq (2+o(1))\sqrt{k-1}$ for any family of k-regular graphs as the number of vertices goes to infinity. For graphs such that $|\delta_i(G_n)| = (2+o(1))\sqrt{k-1}$, (13) implies that $D(G) \leq (2+o(1))i\log_{k-1} n$. The following theorem, obtained jointly with N. Alon [1], shows that this bound is tight for some families of graphs. The proof uses the max-min characterization of the eigenvalues.

Fact 3. *If B is a self-adjoint operator in a vector space L and $\lambda_i(B)$ its $(i+1)$st largest eigenvalue, then*

$$\lambda_i(B) = \max_{H} \min_{g \in H - \{0\}} \frac{g \cdot Bg}{||g||^2},$$

where H ranges over the vector subspaces of L of dimension $i+1$. □

Theorem 5.4. *If $k-1$ is a prime congruent to 1 modulo 4 and i is a positive integer, there exists an infinite explicit family of k-regular graphs G_n on n vertices of diameter $(2+o(1))i\log_{k-1} n$ and such that $\delta_j(G_n) = k - O(1/n)$, for $0 \leq j \leq i-1$, and $|\delta_i(G_n)| = (2+o(1))\sqrt{k-1}$.*

Proof. Consider a family (F_n) of k-regular graphs satisfying the conditions of Theorem 5.1 and whose girth goes to infinity as n goes to infinity. Such a family can be constructed explicitly, as shown in the proof of Theorem 5.1. We construct the graphs G_n (for n multiple of i and such that $F_{n/i}$ exists) as follows: consider i distinct copies of $F_{n/i}$, denoted by $F^j_{n/i}$, for $1 \leq j \leq i$. Let (u_j, v_j) be a pair of vertices in $F^j_{n/i}$ at maximal distance from each other, and let u'_j (respectively, v'_j) be a vertex of $F^j_{n/i}$ adjacent to u_j (respectively, v_j). We form the graph G_n by connecting v_j (respectively, v'_j) to u_{j+1} (respectively, u'_{j+1}), for $1 \leq j \leq i-1$, and deleting the edge between u_j and u'_j, for $2 \leq j \leq i$ and the edge between v_j and v'_j, for $1 \leq j \leq i-1$. Clearly, the diameter of G_n is $(2+o(1))i\log_{k-1} n$.

The eigenvalues of the union of the $F_{n/i}$ satisfy the conditions of Theorem 5.4. Indeed, the first i eigenvalues are equal to k, and the $(i+1)$st largest eigenvalue in absolute value is, in absolute value, equal to $\lambda(F_{n/i}) = (2+o(1))\sqrt{k-1}$. This is because the eigenvalues of a graph are the union of the eigenvalues of its connected components. We now show that the $i+1$ largest eigenvalues of G_n are close to the $i+1$ largest eigenvalues of the union of the $F_{n/i}$.

Let A (respectively, A') be the adjacency matrix of the union of the $F_{n/i}$ (respectively, G_n). Let V_j be the vertex set of $F^j_{n/i}$ and V the vertex set of G_n. It follows from (5) that for any $f \in L^2(V)$,

(16)
$$\begin{aligned} &|A'f \cdot f - Af \cdot f| \\ &\quad = 2\left|\sum_{j=1}^{i-1} (f(v_j)f(u_{j+1}) + f(v'_j)f(u'_{j+1}) - f(u_{j+1})f(u'_{j+1}) - f(v_j)f(v'_j))\right| \\ &\quad \leq 2\sum_{j=1}^{i} \left(f(u_j)^2 + f(u'_j)^2 + f(v_j)^2 + f(v'_j)^2\right). \end{aligned}$$

Denote by H the subspace of $L^2(V)$ of vectors which are constant on each V_j. Equation (16) shows that, for each $f \in H$, we have $A'f \cdot f \geq (k - 8i/n)||f||^2$. Since $\dim H = i$, it follows from Fact 3 that $\lambda_{i-1}(G)$ is lower bounded by $k - 8i/n$, and so are $|\delta_0|, |\delta_1|, \ldots, |\delta_{i-1}|$.

We now show that $|\delta_i(G_n)| = (2+o(1))\sqrt{k-1}$. We only need to show the upper bound, as the lower bound holds for any family of k-regular graphs [8]. Let r be the injectivity radius of G_n. It is at least the injectivity radius of $F_{n/i}$. If $|\delta_i| \leq 2\sqrt{k-1}$, we are done, so we will assume in the rest of the proof that $\delta_i > 2\sqrt{k-1}$. (The case where $\delta_i < -2\sqrt{k-1}$ can be treated similarly.) Let $\delta_i = 2\sqrt{k-1}\cosh\theta$, with $\theta > 0$, and e_h an eigenvector of A' corresponding to δ_h for $0 \leq h \leq i$. Since $|\delta_h| \geq \delta_i$, for $0 \leq j \leq i$, it follows from Lemma 5.3 that $|e_h(u_j)| \leq e^{-r\theta}||e_h||$. Using the Cauchy–Schwarz inequality and the orthogonality of the vectors e_h, this implies that for any vector $f \in \operatorname{Vect}(e_0, e_1, \ldots, e_i)$,

$$|f(u_j)| \leq \sqrt{i+1}e^{-r\theta}||f||. \tag{17}$$

Indeed, if $f = \sum_{h=0}^{i} c_h e_h$, then

$$\begin{aligned} f(u_j)^2 &\leq (i+1)\sum_{h=0}^{i} {c_h}^2 e_h(u_j)^2 \\ &\leq (i+1)e^{-2r\theta}\sum_{h=0}^{i} {c_h}^2 ||e_h||^2 \\ &= (i+1)e^{-2r\theta}||f||^2. \end{aligned}$$

Equation (17) remains valid if u_j is replaced by u'_j, v_j, or v'_j. Since the vector space $\operatorname{Vect}(e_0, e_1, \ldots, e_i)$ is of dimension greater than H, it intersects $H^\perp - \{0\}$. Let g be an element of this intersection. Since the restriction of g to each V_j belongs to $L_0^2(V_j)$,

$$||Ag||^2 \leq \lambda(F_{n/i})^2||g||^2 \leq (4+o(1))(k-1)||g||^2.$$

Combining this with (17) applied to the vector $f = A'g$ yields

$$\begin{aligned} ||A'g||^2 &\leq ||Ag||^2 + \sum_{j=1}^{i}(A'g)(u_j)^2 + (A'g)(u'_j)^2 + (A'g)(v_j)^2 + (A'g)(v'_j)^2 \\ &\leq (4+o(1))(k-1)||g||^2 + 4i(i+1)e^{-2r\theta}||A'g||^2. \end{aligned} \tag{18}$$

But since $g \in \operatorname{Vect}(e_0, e_1, \ldots, e_i)$, we have $||A'g||^2 \geq {\delta_i}^2||g||^2$. Combining this with (18) shows that

$$\cosh^2\theta \leq 1 + o(1) + 4i(i+1)e^{-2r\theta}\cosh^2\theta,$$

which implies that $\theta = o(1)$, since $r = \omega(1)$. We conclude that $|\delta_i| \leq (2+o(1))\sqrt{k-1}$ holds. □

Concluding remark. The results in sections 3–4 (including Theorem 1.1) can be easily extended to general graphs using the techniques in [6]. This yields bounds in terms of the eigenvalues of the Laplacian. Recently, other generalizations and extensions of our results, notably to continuous spaces, have been accomplished in [7].

Acknowledgment. We are grateful to Noga Alon for many helpful discussions and his contribution to Theorem 5.4.

REFERENCES

[1] N. Alon, Private Communication, 1992.

[2] N. Alon, *Eigenvalues and expanders*, Combinatorica, 6 (1986), pp. 83–96.

[3] N. Alon and V. D. Milman, λ_1*, isoperimetric inequalities for graphs and superconcentrators*, J. Combin. Theory Ser. B, 38 (1985), pp. 73–88.

[4] N. Biggs, *Algebraic Graph Theory*, Cambridge University Press, London, 1974.

[5] F. R. K. Chung, *Diameters and eigenvalues*, J. Amer. Math. Soc., 2 (1989), pp. 187–196.

[6] 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.

[7] F. R. K. Chung, A. Grigor'yan, and S.-T. Yau, *Upper bounds for eigenvalues for the discrete and continuous Laplace operators*, Adv. Math., (1996), pp. 165–178.

[8] J. Friedman, *Some geometric aspects of graphs and their eigenfunctions*, Duke Math. J., 69 (1993), pp. 487–525.

[9] N. Kahale, *Better expansion for Ramanujan graphs*, in Proceedings of the 32nd Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, Piscataway, NJ, 1991, pp. 296–303.

[10] N. Kahale, *Expander Graphs*, Technical Report MIT/LCS/TR-591, MIT Laboratory for Computer Science, Cambridge, MA, 1993.

[11] N. Kahale, *Eigenvalues and expansion of regular graphs*, J. Assoc. Comput. Mach., 42 (1995), pp. 1091–1106.

[12] A. Lubotzky, R. Phillips, and P. Sarnak, *Ramanujan graphs*, Combinatorica, 8 (1988), pp. 261–277.

[13] G. A. Margulis, *Explicit group-theoretical constructions of combinatorial schemes and their applications to the design of expanders and concentrators*, Problemy Peredachi Informatsii, 24 (1988), pp. 51–60.

[14] A. Nilli, *On the second eigenvalue of a graph*, Discrete Math., 91 (1991), pp. 207–210.

[15] G. Quenell, *Spectral diameter estimates for k-regular graphs*, Adv. Math., 106 (1994), pp. 122–148.

[16] P. Sarnak, *Some Applications of Modular Forms*, Cambridge University Press, London, 1990.

[17] R. M. Tanner, *Explicit construction of concentrators from generalized n-gons*, SIAM J. Alg. Disc. Meth., 5 (1984), pp. 287–294.

SIAM J. Discrete Math.
Vol. 10, No. 1, pp. 41–56, February 1997

q-SERIES ARISING FROM THE STUDY OF RANDOM GRAPHS*

GEORGE E. ANDREWS[†], DAVIDE CRIPPA[‡], AND KLAUS SIMON[‡]

Abstract. This paper deals with q-series arising from the study of the transitive closure problem in random acyclic digraphs. In particular, it presents an identity involving divisor generating functions which allows us to determine the asymptotic behavior of polynomials defined by a general class of recursive equations, including the polynomials for the mean and the variance of the size of the transitive closure in random acyclic digraphs.

Key words. q-series, divisor generating functions, polynomials, random graphs, transitive closure, probability distributions

AMS subject classifications. 05C80, 11P81, 33D15, 60E10, 60F99

PII. S0895480194262497

1. Introduction.

In [7] Simon, Crippa, and Collenberg studied the distribution of the transitive closure in the $G_{n,p}$-model of a random acyclic digraph. By interpreting the random variable describing the size of the transitive closure of a node as a discrete-time, pure-birth process, they succeeded in finding closed expressions for its distribution, mean, and variance. Developing these expressions as polynomials in $q \overset{\text{def}}{=} 1 - p$ (where p is the probability of existence of an edge), they formulated some conjectures regarding their asymptotic behavior as n, the number of nodes, tends to ∞.

This is the departure point of this paper, whose main result is the identity formulated in section 2. This identity not only allows us to prove in section 4 the conjectures formulated in [7], but also provides a generalization of an identity proposed by Uchimura involving divisor generating functions. Moreover, we will show in section 3 that it allows us to determine the asymptotic behavior of polynomials defined by a general class of recursive equations.

These results are not just of theoretical interest: in section 5 we will show that they can be easily implemented in one of the current symbolic computation systems and finally that they provide a bridge among the fields of discrete mathematics, probability theory, and number theory.

2. The main identity.

There are several definitions and identities in the literature that we require. First of all, we need the definition of q-hypergeometric series[1]

$$(1)\qquad {}_r\phi_s\left(\begin{matrix} a_1, a_2, \ldots, a_r \\ b_1, b_2, \ldots, b_s \end{matrix}\ ; q, t\right) = \sum_{n\geq 0} \frac{(a_1, a_2, \ldots, a_r; q)_n\, t^n}{(q, b_1, b_2, \ldots, b_s; q)_n},$$

where

$$(2)\qquad (A_1, A_2, \ldots, A_r; q)_n = \prod_{i=1}^{r} \prod_{j=0}^{n-1} (1 - A_i\, q^j)$$

* Received by the editors January 31, 1994; accepted for publication (in revised form) February 1, 1996.
http://www.siam.org/journals/sidma/10-1/26249.html

[†] Department of Mathematics, Pennsylvania State University, University Park, PA 16802.
[‡] Institute for Theoretical Computer Science, ETH–Zurich, CH–8092 Zurich, Switzerland.

[1] An actual introduction to the theory of hypergeometric series is [4].

and

$$(A_1, A_2, \ldots, A_r; q)_\infty = \prod_{i=1}^{r} \prod_{j=0}^{\infty} (1 - A_i\, q^j). \tag{3}$$

Throughout this paper we will assume that q is a variable with $0 < q < 1$, and therefore (1) converges absolutely provided $|t| < 1$. Whenever no misunderstanding can arise, we will denote the q-shifted factorial by $(A_1, A_2, \ldots, A_r)_n$, resp., $(A_1, A_2, \ldots, A_r)_\infty$, instead of by (2), resp., (3). We also need the identity for the q-exponential function

$$\sum_{n\geq 0} \frac{z^n}{(q)_n} = \frac{1}{(z)_\infty}, \tag{4}$$

the q-Gauss sum

$${}_2\phi_1\left(\begin{matrix} a, b \\ c \end{matrix}; q, c/a\,b\right) = \frac{(c/a, c/b; q)_\infty}{(c, c/a\,b; q)_\infty}, \tag{5}$$

and the Chu–Vandermonde convolution (see [3])

$$\sum_{r=1}^{k} \binom{j}{r}\binom{k-1}{k-r} = \binom{k+j-1}{k}. \tag{6}$$

Further, following the notation of [5], we will write $\sigma_i(n)$ for the sum of ith powers of the divisors of n; i.e.,

$$\sigma_i(n) \overset{\text{def}}{=} \sum_{d|n} d^i.$$

In particular, then, $\sigma_0(n)$ will denote the number of divisors of n. The generating function of $\sigma_i(n)$ will be denoted by

$$S_i(q) \overset{\text{def}}{=} \sum_{n\geq 1} \sigma_i(n)\, q^n. \tag{7}$$

At this point we can state the major identity of this paper.

THEOREM 2.1. *Let $S_i(q)$ be defined as above. Then for any integer $k \geq 1$ there is a polynomial $M_k(x_1, \ldots, x_k)$ with rational coefficients such that*

$$\sum_{n\geq 1} \frac{(-1)^{n-1}\, q^{\binom{n+1}{2}}}{(q)_n\,(1-q^n)^k} = M_k(S_0(q), \ldots, S_{k-1}(q)). \tag{8}$$

In order to prove this theorem we need several lemmas.

LEMMA 2.2. *For any integer $k \geq 1$ the following identity holds:*

$$r_k \overset{\text{def}}{=} \sum_{n\geq 1} \frac{(-1)^{n-1}\, q^{\binom{n+1}{2}}}{(q)_n\,(1-q^n)^k} = (q)_\infty \sum_{j\geq 0} \frac{q^j}{(q)_j}\binom{k+j-1}{k}. \tag{9}$$

Proof. Let

$$R(z) \overset{\text{def}}{=} \sum_{k\geq 1} r_k\, z^k;$$

then we have

$$\begin{aligned}
R(z) &= \sum_{n\geq 1} \frac{(-1)^{n-1}\, q^{\binom{n+1}{2}}}{(q)_n} \sum_{k\geq 1} \left(\frac{z}{1-q^n}\right)^k \\
&= \sum_{n\geq 1} \frac{(-1)^{n-1}\, q^{\binom{n+1}{2}}}{(q)_n} \frac{\frac{z}{1-q^n}}{1-\frac{z}{1-q^n}} \\
&= -\frac{z}{1-z} \sum_{n\geq 1} \frac{(-1)^n\, q^{\binom{n+1}{2}}}{(q)_n \left(1-\frac{q^n}{1-z}\right)} \\
&= \sum_{n\geq 1} \frac{\left(\frac{1}{1-z}\right)_n (-1)^n\, q^{\binom{n+1}{2}}}{(q)_n \left(\frac{q}{1-z}\right)_n} \\
&= -1 + \lim_{\tau\to 0} {}_2\phi_1 \left(\begin{matrix} \frac{1}{1-z}, \frac{q}{\tau} \\ \frac{q}{1-z} \end{matrix} ; q, \tau \right) \\
&\overset{(5)}{=} -1 + \lim_{\tau\to 0} \frac{(q, \frac{\tau}{1-z})_\infty}{(\frac{q}{1-z}, \tau)_\infty} \\
&= -1 + \frac{(q)_\infty}{\left(\frac{q}{1-z}\right)_\infty} \\
&\overset{(4)}{=} -1 + (q)_\infty \sum_{j\geq 0} \frac{\left(\frac{q}{1-z}\right)^j}{(q)_j} \\
&= -1 + (q)_\infty \sum_{j\geq 0} \frac{q^j}{(q)_j} \sum_{k\geq 0} \binom{k+j-1}{k} z^k.
\end{aligned}$$

Therefore, by comparing the coefficients, for $k \geq 1$ we obtain

$$r_k = (q)_\infty \sum_{j\geq 0} \frac{q^j}{(q)_j} \binom{k+j-1}{k}. \qquad \square$$

Lemma 2.3.

$$\frac{1}{r!} \left[\frac{d^r}{d\epsilon^r} \frac{(q)_\infty}{(\epsilon\, q)_\infty} \right]_{\epsilon=1} = (q)_\infty \sum_{n\geq 0} \frac{q^n}{(q)_n} \binom{n}{r}. \tag{10}$$

Proof.

$$\begin{aligned}
\frac{1}{r!} \left[\frac{d^r}{d\epsilon^r} \frac{(q)_\infty}{(\epsilon\, q)_\infty} \right]_{\epsilon=1} &\overset{(4)}{=} \frac{1}{r!} \left[\frac{d^r}{d\epsilon^r} (q)_\infty \sum_{n\geq 0} \frac{\epsilon^n\, q^n}{(q)_n} \right]_{\epsilon=1} \\
&= \frac{(q)_\infty}{r!} \sum_{n\geq 0} \frac{n\,(n-1)\cdots(n-r+1)\, \epsilon^{n-r}\, q^n}{(q)_n} \Bigg|_{\epsilon=1} \\
&= (q)_\infty \sum_{n\geq 0} \frac{q^n}{(q)_n} \binom{n}{r}. \qquad \square
\end{aligned}$$

Lemma 2.4. *Let*

$$T_r \equiv T_r(\epsilon) \equiv T_r(\epsilon, q) = \sum_{n\geq 1} \frac{q^{r\,n}}{(1-\epsilon\, q^n)^r}.$$

Then there exists a polynomial $N_k(x_1, \dots, x_k)$ with rational coefficients such that for any integer $k \geq 1$ the following holds:

$$\frac{d^k}{d\epsilon^k} \frac{(q)_\infty}{(\epsilon\, q)_\infty} = \frac{(q)_\infty}{(\epsilon\, q)_\infty} N_k(T_1, \dots, T_k). \tag{11}$$

Proof. We will proceed by induction on k. For $k = 1$ we have

$$\begin{aligned} \frac{d}{d\epsilon} \frac{(q)_\infty}{(\epsilon\, q)_\infty} &= (q)_\infty \frac{d}{d\epsilon} \prod_{i\geq 1} (1-\epsilon\, q^i)^{-1} \\ &= (q)_\infty \sum_{j\geq 1} \left(\prod_{\substack{i\geq 1 \\ i\neq j}} (1-\epsilon\, q^i)^{-1} \right) \left(q^j\, (1-\epsilon\, q^j)^{-2} \right) \\ &= (q)_\infty \sum_{j\geq 1} \prod_{i\geq 1} (1-\epsilon\, q^i)^{-1} \frac{q^j}{1-\epsilon\, q^j} \\ &= \frac{(q)_\infty}{(\epsilon\, q)_\infty} \sum_{j\geq 1} \frac{q^j}{1-\epsilon\, q^j} \\ &= \frac{(q)_\infty}{(\epsilon\, q)_\infty} T_1, \end{aligned} \tag{12}$$

so choose $N_1(x_1) = x_1$. Further, notice that

$$\begin{aligned} \frac{d}{d\epsilon} T_r(\epsilon) &= \frac{d}{d\epsilon} \sum_{n\geq 1} q^{r\,n} (1-\epsilon\, q^n)^{-r} \\ &= \sum_{n\geq 1} q^{r\,n} (1-\epsilon\, q^n)^{-r-1}\, r\, q^n \\ &= r \sum_{n\geq 1} q^{(r+1)\,n} (1-\epsilon\, q^n)^{-(r+1)} \\ &= r\, T_{r+1}(\epsilon). \end{aligned} \tag{13}$$

Now assume that our result is established up to a k; then we have

$$\begin{aligned} \frac{d^{k+1}}{d\epsilon^{k+1}} \frac{(q)_\infty}{(\epsilon\, q)_\infty} &= \frac{d}{d\epsilon} \left(\frac{(q)_\infty}{(\epsilon\, q)_\infty} N_k(T_1, \dots, T_k) \right) \\ &= \frac{(q)_\infty}{(\epsilon\, q)_\infty} T_1\, N_k(T_1, \dots, T_k) + \frac{(q)_\infty}{(\epsilon\, q)_\infty} \frac{d}{d\epsilon} N_k(T_1, \dots, T_k) \\ &= \frac{(q)_\infty}{(\epsilon\, q)_\infty} N_{k+1}(T_1, \dots, T_{k+1}), \end{aligned}$$

where

$$N_{k+1}(T_1, \dots, T_{k+1}) = T_1\, N_k(T_1, \dots, T_k) + \frac{d}{d\epsilon} N_k(T_1, \dots, T_k). \tag{14}$$

It remains to show that (14) is a polynomial in $T_1, \ldots, T_{k+1}$ with rational coefficients. Clearly this holds for the first term. Further, $N_k(T_1, \ldots, T_k)$ is a sum of terms of the form

$$c\, T_1^{j_1} \cdots T_k^{j_k},$$

and thus, according to (13), its derivative with respect to ϵ will be a polynomial in $T_1, \ldots, T_{k+1}$ with rational coefficients. □

LEMMA 2.5. *For any integer $k \geq 1$ there exist rational numbers $c_{k,j}$, $0 \leq j \leq k-1$, such that*

$$T_k(1,q) = \sum_{j=0}^{k-1} c_{k,j}\, S_j(q). \tag{15}$$

Proof. Recall the definition of Stirling numbers of the first kind:

$$x\,(x-1)\cdots(x-n+1) = \sum_{k=0}^{n} s(n,k)\, x^k.$$

Now we have

$$\begin{aligned}
&T_k(1,q)\\
&= \sum_{n\geq 1} \frac{q^{k\,n}}{(1-q^n)^k}\\
&= \sum_{n\geq 1} \frac{q^n\,(1-(1-q^n))^{k-1}}{(1-q^n)^k}\\
&= \sum_{n\geq 1} \frac{q^n}{(1-q^n)^k} \sum_{j=0}^{k-1} \binom{k-1}{j} (-1)^j\,(1-q^n)^j\\
&= \sum_{j=0}^{k-1} \binom{k-1}{j} (-1)^j \sum_{n\geq 1} \sum_{m\geq 0} \binom{k-j+m-1}{k-j-1} q^{n\,(1+m)}\\
&= \sum_{j=0}^{k-1} \binom{k-1}{j} (-1)^j \sum_{n\geq 1} \sum_{m\geq 0} \frac{q^{n\,(1+m)}}{(k-j-1)!}\,(m+1)\,(m+2)\cdots(m+k-j-1)\\
&= \sum_{j=0}^{k-1} \binom{k-1}{j} \frac{(-1)^{k-1}}{(k-j-1)!} \sum_{m,n\geq 1} q^{n\,m}\,(-m)\,(-m-1)\cdots(-m-k+j+2)\\
&= \sum_{j=0}^{k-1} \binom{k-1}{j} \frac{(-1)^{k-1}}{(k-j-1)!} \sum_{h=0}^{k-j-1} s(k-j-1,h)\,(-1)^h \sum_{m,n\geq 1} m^h\, q^{n\,m}\\
&= \sum_{j=0}^{k-1} \binom{k-1}{j} \frac{(-1)^{k-1}}{(k-j-1)!} \sum_{h=0}^{k-j-1} s(k-j-1,h)\,(-1)^h\, S_h(q)\\
&= \sum_{h=0}^{k-1} S_h(q) \sum_{j=0}^{k-h-1} \binom{k-1}{j} \frac{(-1)^{h+k-1}}{(k-j-1)!}\, s(k-j-1,h)\\
&= \sum_{h=0}^{k-1} c_{k,h}\, S_h(q). \qquad \square
\end{aligned}$$

At this point we have all the lemmas needed to prove Theorem 2.1.

Proof.

$$\begin{aligned}
\sum_{n\geq 1} \frac{(-1)^{n-1}\, q^{\binom{n+1}{2}}}{(q)_n\,(1-q^n)^k} &\overset{(9)}{=} (q)_\infty \sum_{j\geq 0} \frac{q^j}{(q)_j} \binom{k+j-1}{k} \\
&\overset{(6)}{=} (q)_\infty \sum_{j\geq 0} \frac{q^j}{(q)_j} \sum_{r=1}^{k} \binom{j}{r}\binom{k-1}{k-r} \\
&= \sum_{r=1}^{k} \binom{k-1}{k-r} (q)_\infty \sum_{j\geq 0} \frac{q^j}{(q)_j} \binom{j}{r} \\
&\overset{(10)}{=} \sum_{r=1}^{k} \binom{k-1}{k-r} \frac{1}{r!} \left[\frac{d^r}{d\epsilon^r}\,\frac{(q)_\infty}{(\epsilon\, q)_\infty}\right]_{\epsilon=1} \\
&\overset{(11)}{=} \sum_{r=1}^{k} \binom{k-1}{k-r} \frac{1}{r!}\, N_r(T_1(1,q),\ldots,T_r(1,q)) \\
&\overset{(15)}{=} \sum_{r=1}^{k} \binom{k-1}{k-r} \frac{1}{r!}\, N_r(c_{1,0}\, S_0(q), c_{2,0}\, S_0(q) \\
&\quad +c_{2,1}\, S_1(q),\ldots, c_{r,0}\, S_0(q)+\cdots+c_{r,r-1}\, S_{r-1}(q)) \\
&= M_k(S_0(q),\ldots,S_{k-1}(q)),
\end{aligned}$$

where $M_k(x_1,\ldots,x_k)$ is a polynomial in $x_1,\ldots,x_k$ with rational coefficients. □

Remark. We note from the construction of N_r in the proof of Lemma 2.4 that we have

$$N_r(T_1,\ldots,T_r) = \sum_{\pi\vdash r} c(r,\pi)\, x_1^{m_1(\pi)}\cdots x_r^{m_r(\pi)}, \tag{16}$$

where $\pi\vdash r$ means $\pi = (1^{m_1(\pi)}\, 2^{m_2(\pi)}\, 3^{m_3(\pi)}\ldots)$ is a partition of r ($\sum_{j\geq 1} m_j(\pi)\, j = r$). Therefore, according to the transformation of the N_r into M_k, we also have

$$M_k(x_1,\ldots,x_k) = \sum_{r=1}^{k}\sum_{\pi\vdash r} c(r,\pi)\, x_1^{m_1(\pi)}\cdots x_r^{m_r(\pi)}. \tag{17}$$

Next we will compute the cases $k = 1, 2, 3$. In section 5 we will show that this computation can be carried out by means of a current symbolic computation system, so that the polynomial $M_k(x_1,\ldots,x_k)$ can be found for any $k\geq 1$.

$k = 1$: Following the proof of Theorem 2.1 we have

$$\begin{aligned}
\sum_{n\geq 1} \frac{(-1)^{n-1}\, q^{\binom{n+1}{2}}}{(q)_n\,(1-q^n)} \overset{(9)}{=} (q)_\infty \sum_{j\geq 0} \frac{j\, q^j}{(q)_j} &\overset{(10)}{=} \left[\frac{d}{d\epsilon}\,\frac{(q)_\infty}{(\epsilon\, q)_\infty}\right]_{\epsilon=1} \\
\overset{(12)}{=} T_1(1,q) \;=\; \sum_{n\geq 1} \frac{q^n}{1-q^n} &\;=\; S_0(q),
\end{aligned}$$

and therefore

$$M_1(x_1) = x_1. \tag{18}$$

This identity was already proven by Uchimura in [8] and thus our theorem provides a generalization of it.

$k=2$: In this case we obtain in a similar way

$$\begin{aligned}
\sum_{n\geq 1}\frac{(-1)^{n-1}\,q^{\binom{n+1}{2}}}{(q)_n\,(1-q^n)^2} &\overset{(9)}{=} (q)_\infty \sum_{j\geq 0}\frac{\binom{j+1}{2}\,q^j}{(q)_j}\\
&= (q)_\infty \sum_{j\geq 0}\left(\binom{j}{2}+\binom{j}{1}\right)\frac{q^j}{(q)_j}\\
&\overset{(10)}{=} \frac{1}{2}\left[\frac{d^2}{d\epsilon^2}\,\frac{(q)_\infty}{(\epsilon\,q)_\infty}\right]_{\epsilon=1}+\left[\frac{d}{d\epsilon}\,\frac{(q)_\infty}{(\epsilon\,q)_\infty}\right]_{\epsilon=1}\\
&\overset{(12)}{=} \frac{1}{2}\left[\frac{d}{d\epsilon}\,\frac{(q)_\infty}{(\epsilon\,q)_\infty}\,T_1(\epsilon)\right]_{\epsilon=1}+S_0(q)\\
&\overset{(12,13)}{=} \frac{1}{2}\left[\frac{(q)_\infty}{(\epsilon\,q)_\infty}\,T_2(\epsilon)+\frac{(q)_\infty}{(\epsilon\,q)_\infty}\,T_1(\epsilon)^2\right]_{\epsilon=1}+S_0(q)\\
&= \frac{1}{2}\,T_2(1,q)+\frac{1}{2}\,S_0(q)^2+S_0(q).
\end{aligned}$$

Following Lemma 2.5 through for $k=2$, we find further that

$$T_2(1,q)=c_{2,0}\,S_0(q)+c_{2,1}\,S_1(q)=-S_0(q)+S_1(q).$$

Hence

$$\sum_{n\geq 1}\frac{(-1)^{n-1}\,q^{\binom{n+1}{2}}}{(q)_n\,(1-q^n)^2}=\frac{1}{2}\,\left(S_1(q)+S_0(q)+S_0(q)^2\right),$$

and so

$$M_2(x_1,x_2)=\frac{1}{2}\,x_2+\frac{1}{2}\,x_1+\frac{1}{2}\,x_1^2. \tag{19}$$

$k=3$: First of all, by analyzing Lemma 2.5 we get

$$\begin{aligned}
S_2(q) &= \sum_{m\geq 1}\sum_{n\geq 1} n^2\,q^{n\,m}\\
&= \sum_{m\geq 1}\sum_{n\geq 1}\left(2\binom{n+1}{2}-n\right)\,q^{n\,m}\\
&= 2\sum_{m\geq 1}\frac{q^m}{(1-q^m)^3}-S_1(q) \qquad (20)
\end{aligned}$$

and

$$\begin{aligned}
T_3(1,q) &= \sum_{n\geq 1}\frac{q^n\,(1-(1-q^n))^2}{(1-q^n)^3}\\
&= \sum_{n\geq 1}\frac{q^n}{(1-q^n)^3}-2\sum_{n\geq 1}\frac{q^n}{(1-q^n)^2}+\sum_{n\geq 1}\frac{q^n}{1-q^n}\\
&\overset{(20)}{=} \frac{1}{2}\,(S_2(q)+S_1(q))-2\,S_1(q)+S_0(q)\\
&= \frac{1}{2}\,S_2(q)-\frac{3}{2}\,S_1(q)+S_0(q).
\end{aligned}$$

Hence

$$\begin{aligned}
&\sum_{n\geq 1} \frac{(-1)^{n-1}\, q^{\binom{n+1}{2}}}{(q)_n\,(1-q^n)^3} \\
&\overset{(9)}{=} \quad (q)_\infty \sum_{j\geq 0} \frac{\binom{j+2}{3}\, q^j}{(q)_j} \\
&= \quad (q)_\infty \sum_{j\geq 0} \left(\binom{j}{3} + 2\binom{j}{2} + \binom{j}{1} \right) \frac{q^j}{(q)_j} \\
&\overset{(10)}{=} \quad \frac{1}{3!} \left[\frac{d^3}{d\epsilon^3}\, \frac{(q)_\infty}{(\epsilon\, q)_\infty} \right]_{\epsilon=1} + \left[\frac{d^2}{d\epsilon^2}\, \frac{(q)_\infty}{(\epsilon\, q)_\infty} \right]_{\epsilon=1} + \left[\frac{d}{d\epsilon}\, \frac{(q)_\infty}{(\epsilon\, q)_\infty} \right]_{\epsilon=1} \\
&\overset{(k\,=\,2)}{=} \quad \frac{1}{3!} \left[\frac{d^2}{d\epsilon^2}\, \frac{(q)_\infty}{(\epsilon\, q)_\infty}\, T_1(\epsilon) \right]_{\epsilon=1} + T_2(1,q) + S_0(q)^2 + S_0(q) \\
&= \quad \frac{1}{3!} \left[\frac{d}{d\epsilon} \left(\frac{(q)_\infty}{(\epsilon\, q)_\infty}\, T_2(\epsilon) + \frac{(q)_\infty}{(\epsilon\, q)_\infty}\, T_1(\epsilon)^2 \right) \right]_{\epsilon=1} + S_0(q)^2 + S_1(q) \\
&= \quad \frac{1}{3!} \left[\frac{(q)_\infty}{(\epsilon\, q)_\infty}\, 2\, T_3(\epsilon) + \frac{(q)_\infty}{(\epsilon\, q)_\infty}\, T_1(\epsilon)\, T_2(\epsilon) + \frac{(q)_\infty}{(\epsilon\, q)_\infty}\, T_1(\epsilon)^3 \right. \\
&\qquad\qquad \left. + \frac{(q)_\infty}{(\epsilon\, q)_\infty}\, 2\, T_1(\epsilon)\, T_2(\epsilon) \right]_{\epsilon=1} + S_0(q)^2 + S_1(q) \\
&= \quad \frac{1}{3}\, T_3(1,q) + \frac{1}{2}\, T_1(1,q)\, T_2(1,q) + \frac{1}{6}\, T_1(1,q)^3 + S_0(q)^2 + S_1(q) \\
&= \quad \frac{1}{6}\, S_2(q) + \frac{1}{2}\, S_1(q) + \frac{1}{3}\, S_0(q) + \frac{1}{2}\, S_0(q)^2 + \frac{1}{2}\, S_0(q)\, S_1(q) + \frac{1}{6}\, S_0(q)^3,
\end{aligned}$$

and so

$$M_3(x_1, x_2, x_3) = \frac{1}{6}\, x_3 + \frac{1}{2}\, x_2 + \frac{1}{3}\, x_1 + \frac{1}{2}\, x_1^2 + \frac{1}{2}\, x_1\, x_2 + \frac{1}{6}\, x_1^3. \tag{21}$$

To conclude this section we will show that Theorem 2.1 constitutes the basis for finding and proving new identities involving q-series. In [8] Uchimura proved that

$$(x)_\infty \sum_{m\geq 1} m\, \frac{x^m}{(x)_m} \equiv \sum_{m\geq 1} m\, x^m \prod_{j\geq m+1} (1-x^j) = S_0(x); \tag{22}$$

we can generalize this identity to the following.

Theorem 2.6. *For any integer $k \geq 1$ there exists a polynomial $H_k(x_1, \ldots, x_k)$ with rational coefficients such that*

$$(x)_\infty \sum_{m\geq 1} m^k\, \frac{x^m}{(x)_m} = H_k(S_0(x), \ldots, S_{k-1}(x)). \tag{23}$$

Proof. Let $k \geq 1$ be an integer and consider first the expression

$$B_\alpha(x) \overset{\text{def}}{=} \sum_{n\geq 1} \frac{(-1)^{n-1}\, \alpha^n\, x^{\binom{n+1}{2}}}{(x)_n\, (1-x^n)^k}. \tag{24}$$

Clearly then, by Theorem 2.1, $B_1(x) = M_k(S_0(x), \dots, S_{k-1}(x))$. On the other hand, we have

$$\begin{aligned}
B_\alpha(x) &= \sum_{n \geq 1} \frac{(-1)^{n-1}\,\alpha^n\,x^{\binom{n}{2}}}{(x)_{n-1}}\;\frac{x^n}{(1-x^n)^{k+1}} \\
&= \sum_{n \geq 1} \frac{(-1)^{n-1}\,\alpha^n\,x^{\binom{n}{2}}}{(x)_{n-1}}\;\sum_{m \geq 1}\binom{m+k-1}{k} x^{n\,m} \\
&= \sum_{m \geq 1} \alpha\,x^m \binom{m+k-1}{k} \sum_{n \geq 1} \frac{(-\alpha)^{n-1}\,x^{(n-1)\,m}\,x^{\binom{n}{2}}}{(x)_{n-1}} \\
&= \alpha\,(\alpha\,x)_\infty \sum_{m \geq 1} \binom{m+k-1}{k} \frac{x^m}{(\alpha\,x)_m},
\end{aligned}$$

where the last identity follows from [5, Theorem 348] by letting $j \to \infty$ and $a = -\alpha\,x^n$. At this point we can prove our theorem by induction on k. As mentioned before, the case $k = 1$ has been treated by Uchimura. So let $k \geq 2$ and let our theorem hold up to $k - 1$. Then we have

$$\begin{aligned}
k!\,M_k(S_0(x), \dots, S_{k-1}(x)) &= k!\,B_1(x) \\
&= (x)_\infty \sum_{m \geq 1} k! \binom{m+k-1}{k} \frac{x^m}{(x)_m} \\
&= (x)_\infty \sum_{m \geq 1} \sum_{i=1}^{k} g_{i,k}\,m^i\,\frac{x^m}{(x)_m} \\
&= \sum_{i=1}^{k} g_{i,k}\,(x)_\infty \sum_{m \geq 1} m^i\,\frac{x^m}{(x)_m},
\end{aligned}$$

where $g_{k,k} = 1$, and thus

$$\begin{aligned}
(x)_\infty \sum_{m \geq 1} m^k\,\frac{x^m}{(x)_m} &= \left[k!\,M_k(S_0(x), \dots, S_{k-1}(x)) - \sum_{i=1}^{k-1} g_{i,k}\,(x)_\infty \sum_{m \geq 1} m^i\,\frac{x^m}{(x)_m} \right] \\
&= \left[k!\,B_1(x) - \sum_{i=1}^{k-1} g_{i,k}\,H_i(S_0(x), \dots, S_{i-1}(x)) \right] \\
&\overset{\text{def}}{=} H_k(S_0(x), \dots, S_{k-1}(x)). \qquad \square
\end{aligned}$$

In section 5 we will show that the polynomial $H_k(x_1, \dots, x_k)$ can also be determined by a symbolic computation program.

3. A set of recursive equations. Theorem 2.1 allows us to determine the asymptotic behavior of polynomials defined by a general class of recursive equations. Let

$$f(n) = \sum_{k \geq 0} c_k\,n^k$$

be a nonzero polynomial in n with rational coefficients and $M_j \equiv M_j(S_0(q), \dots, S_{j-1}(q))$ be the same polynomials defined in Theorem 2.1. Then the following theorem holds.

Theorem 3.1. *Let $a_n(q)$ be a polynomial in q defined by the recursive equation*

$$a_n(q) = f(n) + (1 - q^{n-1})\, a_{n-1}(q), \qquad n \geq 1, \tag{25}$$

initialized with $a_0(q) = 0$. Then there are rational coefficients h_j such that

$$\lim_{n\to\infty} \left(\sum_{i=1}^{n} f(i) - a_n(q) \right) = \sum_{j\geq 1} h_j\, M_j; \tag{26}$$

for the coefficients h_j the following hold:

$$h_1 = c_0 \tag{27}$$

and

$$h_j = \sum_{i\geq j-1} (-1)^{i-j+1} \binom{i-1}{j-2}\, i! \sum_{k\geq i} c_k\, \tilde{s}(k,i), \tag{28}$$

where $\tilde{s}(k,i)$ represents the Stirling number of the second kind.

Proof. For the generating function of $f(n)$ we get

$$\begin{aligned} F(z) &= \sum_{n\geq 1} f(n)\, z^n \\ &= \sum_{m\geq 0} \underbrace{\sum_{k\geq m} c_k\, \tilde{s}(k,m)\, m!}_{d_m}\, \frac{z^m}{(1-z)^{m+1}} - c_0, \end{aligned}$$

and therefore for $z = q^n$ we obtain

$$\begin{aligned} F(q^n) &= \sum_{m\geq 0} d_m\, \frac{q^{n\,m}}{(1-q^n)^{m+1}} - c_0 \\ &= \frac{d_0}{1-q^n} + \sum_{m\geq 1} d_m\, \frac{q^n\,(1-(1-q^n))^{m-1}}{(1-q^n)^{m+1}} - c_0 \\ &= \sum_{m\geq 1} d_m \sum_{j=0}^{m-1} \underbrace{\binom{m-1}{j} (-1)^j}_{e_{m,j}}\, \frac{q^n}{(1-q^n)^{m+1-j}} + c_0\, \frac{q^n}{1-q^n}. \end{aligned}$$

Now let

$$A(z) \overset{\text{def}}{=} \sum_{n\geq 1} a_n(q)\, z^n; \tag{29}$$

then by substituting (25) in (29) we get

$$\begin{aligned} A(z) &= \sum_{n\geq 1} \left[f(n) + (1-q^{n-1})\, a_{n-1}(q) \right]\, z^n \\ &= \sum_{n\geq 1} f(n)\, z^n + z\, A(z) - z\, A(q\, z) \\ &= F(z) + z\, A(z) - z\, A(q\, z), \end{aligned}$$

thus obtaining

$$A(z) = \frac{F(z)}{1-z} - \frac{z}{1-z}\,A(q\,z). \tag{30}$$

Finally, iterative substitution results in

$$A(z) = \sum_{n\geq 0} \frac{(-1)^n\,F(q^n z)\,z^n\,q^{\binom{n}{2}}}{(z)_{n+1}}. \tag{31}$$

At this point we substitute $z = q$, use the expression for $F(q^n)$, and apply Theorem 2.1 to obtain

$$\begin{aligned}
A(q) &= \sum_{n\geq 1} \frac{(-1)^{n-1}\,F(q^n)\,q^{\binom{n}{2}}}{(q)_n} \\
&= \sum_{m\geq 1} d_m \sum_{j=0}^{m-1} e_{m,j} \sum_{n\geq 1} \frac{(-1)^{n-1}\,q^{\binom{n+1}{2}}}{(q)_n\,(1-q^n)^{m+1-j}} + c_0 \sum_{n\geq 1} \frac{(-1)^{n-1}\,q^{\binom{n+1}{2}}}{(q)_n\,(1-q^n)} \\
&= \sum_{m\geq 1} d_m \sum_{j=0}^{m-1} e_{m,j}\,M_{m+1-j} + c_0\,M_1 \\
&= c_0\,M_1 + \sum_{j\geq 2} M_j \sum_{i\geq j-1} d_i\,e_{i,i-j+1} \\
&= \sum_{j\geq 1} h_j\,M_j,
\end{aligned}$$

where $h_1 = c_0$, and for $j \geq 2$ we have

$$h_j = \sum_{i\geq j-1} d_i\,e_{i,i-j+1} = \sum_{i\geq j-1} (-1)^{i-j+1} \binom{i-1}{j-2}\, i! \sum_{k\geq i} c_k\,\tilde{s}(k,i).$$

To complete the proof it is sufficient to notice that $a_n(q)$ can also be written as

$$a_n(q) = \sum_{i=1}^{n} f(i) - \sum_{i=1}^{n-1} q^i\,a_i(q),$$

and therefore we get

$$\lim_{n\to\infty} \left(\sum_{i=1}^{n} f(i) - a_n(q) \right) = A(q). \qquad \square$$

4. Applications to probability theory. As mentioned in the introduction, we will show in this section that our theory allows us to prove in a very concise way two results formulated in [7]. For this reason we will briefly restate some of the results contained in that paper.

Let the $G_{n,p}$-model of a random acyclic digraph[2] be defined by the vertex set $V = \{1,\ldots,n\}$ and the set of edges (i,j) with $i < j$, where every edge occurs independently

[2] An actual introduction to the theory of random graphs is [6].

with probability p, $0 < p < 1$. In this model the size γ_n^* of the reflexive, transitive closure of node 1, i.e., the cardinality of the set of nodes reachable through a directed path starting in 1 (including node 1 itself), is a random variable with the following distribution:

$$Pr(\gamma_n^* = h) = q^{n-h} \prod_{i=1}^{h-1} (1 - q^{n-i}), \tag{32}$$

where $q \overset{\text{def}}{=} 1 - p$. This can be proven by interpreting γ_n^* as a discrete-time, pure-birth process with time $t = n$. Such a process can be described by a sequence of random variables X_t, $t \in \mathbb{N}$, assuming the states $\ell = 1, 2, 3, \ldots$ with probabilities $P_{t,\ell}$, and by a sequence of transition probabilities λ_ℓ, $0 \le \lambda_\ell \le 1$, such that $P_{t,\ell} = 0$ for $t \le 0$ or $\ell \notin \{1, \ldots, t\}$, $P_{1,1} = 1$, and $P_{t,\ell} = (1 - \lambda_\ell)\, P_{t-1,\ell} + \lambda_{\ell-1}\, P_{t-1,\ell-1}$ otherwise. For γ_n^* holds $\lambda_\ell = 1 - q^\ell$ and (32) can be verified by induction on n. At this point we are able to give a very concise proof of the following theorem, already proved in [7].

THEOREM 4.1. *For all q, $0 < q < 1$, we have*

$$\lim_{n\to\infty} (n - E(\gamma_n^*)) = \sum_{i=1}^{\infty} \frac{q^i}{1 - q^i} \equiv S_0(q). \tag{33}$$

Proof. Let us set

$$e_n(q) \overset{\text{def}}{=} E(\gamma_n^*); \tag{34}$$

then using (32) it can be proven that $e_n(q)$ satisfies the recursion

$$e_n(q) = 1 + (1 - q^{n-1})\, e_{n-1}(q), \tag{35}$$

which is of type (25) with the polynomial $f(n) = 1$. According to Theorem 3.1 there are coefficients h_j such that

$$\lim_{n\to\infty} \left(\sum_{i=1}^{n} f(i) - e_n(q) \right) = \lim_{n\to\infty} (n - E(\gamma_n^*)) = \sum_{j\ge1} h_j\, M_j.$$

As for $f(n) = 1$, we have $c_0 = 1$ and $c_k = 0$ for all $k \ge 1$; we get $h_1 = 1$ and $h_j = 0$ for all $j \ge 2$. Finally, we saw in (18) that $M_1(x) = x$ and so we have proven

$$\lim_{n\to\infty} (n - E(\gamma_n^*)) = M_1(S_0(q)) = S_0(q) \equiv \sum_{i=1}^{\infty} \frac{q^i}{1 - q^i}. \qquad \square$$

The next theorem concerns the asymptotic for the variance of γ_n^* and it is not as straightforward as the one for the mean.

THEOREM 4.2. *For all q, $0 < q < 1$, we have*

$$\lim_{n\to\infty} (\,Var(\gamma_n^*)) = \sum_{i=1}^{\infty} \frac{i\, q^i}{1 - q^i} \equiv S_1(q). \tag{36}$$

Proof. Let us consider first the second moment of the distribution

$$x_n(q) \overset{\text{def}}{=} E((\gamma_n^*)^2). \tag{37}$$

Again using (32), it can be proven that $x_n(q)$ satisfies the recursion

$$x_n(q) = 2\,e_n(q) - 1 + (1 - q^{n-1})\,x_{n-1}(q); \tag{38}$$

this equation is not of type (25), but by induction on n we can show that

$$x_n(q) = 2\,n\,e_n(q) - \sum_{i=1}^{n}(2\,i-1)\prod_{j=1}^{n-1}(1-q^j). \tag{39}$$

If we now define

$$z_n(q) \overset{\mathrm{def}}{=} \sum_{i=1}^{n}(2\,i-1)\prod_{j=1}^{n-1}(1-q^j), \tag{40}$$

it is easy to show that $z_n(q)$ satisfies the following recursion:

$$z_n(q) = (2\,n-1) + (1-q^{n-1})\,z_{n-1}(q), \tag{41}$$

which allows us to apply Theorem 3.1. For the polynomial $f(n) = 2\,n-1$ we have $c_0 = -1$, $c_1 = 2$, and $c_k = 0$ for all $k \geq 2$, and these values result in $h_1 = -1$, $h_2 = 2$, and $h_j = 0$ for all $j \geq 3$. By (18) and (19) we know further that $M_1(x) = x$ and $M_2(x,y) = 1/2\,(x+y+x^2)$, and therefore we get

$$\begin{aligned}
\lim_{n\to\infty}\left(\sum_{i=1}^{n} f(i) - z_n(q)\right) &= \lim_{n\to\infty}\left(\sum_{i=1}^{n}(2\,i-1) - z_n(q)\right)\\
&= \lim_{n\to\infty}\big(n^2 - z_n(q)\big)\\
&= -M_1(S_0(q)) + 2\,M_2(S_0(q), S_1(q))\\
&= S_0(q)^2 + S_1(q).
\end{aligned}$$

Finally we get

$$\begin{aligned}
\lim_{n\to\infty}\big(\,Var(\gamma_n^*)\big) &= \lim_{n\to\infty}\big(x_n(q) - e_n(q)^2\big)\\
&= \lim_{n\to\infty}\big(2\,n\,e_n(q) - z_n(q) - e_n(q)^2\big)\\
&= \lim_{n\to\infty}\big((n^2 - z_n(q)) - (n - e_n(q))^2\big)\\
&= S_0(q)^2 + S_1(q) - S_0(q)^2\\
&= S_1(q). \qquad \square
\end{aligned}$$

5. Programming the results. The results we have obtained can be easily programmed on one of the current symbolic computation systems. We have chosen Mathematica and now present two short listings implementing the results of Theorems 2.1 and 2.6.

Accordingly, `c[k,h]` correspond to the coefficients $c_{k,h}$ of Lemma 2.5, by means of which we build $T_k(1,q)$, represented by `T[k]` and by `Tt[k]`, depending on the step of the computation. The generating functions $S_i(q)$ are not explicitly computed and they will figure in the result as `S[i]`. Further, in order to build the polynomials $N_k(T_1,\ldots,T_k) \equiv$ `Nn[k]`, we had to define a functional `De[ ]` expressing the derivative with respect to ϵ as defined by (13) and later used in (14). The polynomial

$M_k(S_0(q), \ldots, S_{k-1}(q))$ of Theorem 2.1 is finally expressed by `M[k]`.

```
In[1]:=  c[k_,h_] := Sum[Binomial[k-1,j] (-1)^(h+k-1)
                    StirlingS1[k-j-1,h]/(k-j-1)!,{j,0,k-h+1}]
In[2]:=  T[k_] := Sum[S[h] c[k,h],{h,0,k-1}]
In[3]:=  De[Tt[r_]] := r Tt[r+1]
In[4]:=  De[a_ b_] := a De[b] + b De[a]
In[5]:=  De[a_^b_] := b a^(b-1) De[a]
In[6]:=  De[a_+b_] := De[a] + De[b]
In[7]:=  De[n_] := If[IntegerQ[n],0,De[n]]
In[8]:=  Nn[1] := Tt[1]
In[9]:=  Nn[k_] := Tt[1] Nn[k-1] + De[Nn[k-1]]
In[10]:=  M[k_] := Expand[Sum[Binomial[k-1,k-r] Nn[r]/r!,{r,1,k}]
                    /.{Tt->T}]
In[11]:=  M[4]
                         2       3       4
           S[0]    11 S[0]    S[0]    S[0]    11 S[1]    3 S[0] S[1]
Out[11]:=  ---- + -------- + ----- + ----- + ------- + ----------- +
            4        24        4      24       24            4
                 2            2
             S[0]  S[1]   S[1]    S[2]   S[0] S[2]   S[3]
             ---------- + ----- + ---- + --------- + ----
                 4          8      4         6        24

```

Similarly, we use `g[i,k]` to denote $g_{i,k}$ in the proof of Theorem 2.6 and we express the result by `H[k]`.

```
In[12]:=  g[i_,k_] := CoefficientList[Product[m+j,{j,0,k-1}],m][[i+1]]
In[13]:=  H[1] := S[0]
In[14]:=  H[k_] := Expand[k! M[k] - Sum[g[i,k] H[i],{i,1,k-1}]]
In[15]:=  H[2]
              2
Out[15]:= S[0]  + S[1]
In[16]:=  H[3]
              3
Out[16]:= S[0]  + 3 S[0]  S[1] + S[2]
In[17]:=  H[4]
              4         2         2
Out[17]:= S[0]  + 6 S[0]  S[1] + 3 S[1]  + 4 S[0] S[2] + S[3]
```

6. Conclusion and open questions. In this paper we have presented some q-series arising from the study of random graphs, in particular from the distribution of the transitive closure in random acyclic digraphs.

The main identity, expressed in Theorem 2.1, seems to play a key role in proving other results, like those in Theorems 2.6 and 3.1. As the corresponding proofs are constructive, the results we have obtained can be implemented in symbolic computation systems, and this could allow the automatic generation of new identities.

In [2] Bressoud and Subbarao showed that

$$\sigma_0(n) = -\sum_{\pi \vdash n}{}'(-1)^{\#(\pi)} \lambda(\pi), \tag{42}$$

where $\pi \vdash n$ means that π is a partition of n, the prime on the summation restricts the sum to those partitions which have distinct parts, $\#(\pi)$ is the number of parts in π, and $\lambda(\pi)$ is the smallest part in π.

Using Theorem 2.6, we are now able to derive many more identities of the same type. In fact, for any $k \geq 1$ the coefficient of x^N in the left-hand side of (23) is given by

$$N^k + \sum_{j=1}^{N-1} j^k \sum_{\pi \vdash (N-j)}{}'(-1)^{\#(\pi)} [\lambda(\pi) \geq j+1], \tag{43}$$

where the expression $[P]$ is evaluated to 1 if P is true and to 0 if P is false. On the other hand, we can also determine the coefficient of x^N in $H_k(S_0(x), \ldots, S_{k-1}(x))$; for $k = 2$ we get, for instance, from section 5

$$H_2(S_0(x), S_1(x)) = S_0(x)^2 + S_1(x),$$

and therefore we obtain the following identity:

$$\sigma_1(N) + \sum_{i=1}^{N-1} \sigma_0(i)\, \sigma_0(N-i) = N^2 + \sum_{j=1}^{N-1} j^2 \sum_{\pi \vdash (N-j)}{}'(-1)^{\#(\pi)} [\lambda(\pi) \geq j+1]. \tag{44}$$

As far as further research is concerned, we would like to mention some open problems.

Van Hamme [10] showed a finite analogue of identity (8) for $k = 1$:

$$\sum_{k=1}^{n} \frac{(-1)^{k-1}\, q^{\binom{k+1}{2}}}{1-q^k} \begin{bmatrix} n \\ k \end{bmatrix} = \sum_{k=1}^{n} \frac{q^k}{1-q^k}, \tag{45}$$

where $\left[\begin{smallmatrix} n \\ k \end{smallmatrix}\right]$ is the Gaussian polynomial defined by

$$\begin{bmatrix} n \\ k \end{bmatrix} \stackrel{\text{def}}{=} \frac{(q)_n}{(q)_k\, (q)_{n-k}}.$$

Later this was generalized by Uchimura [9] for any nonnegative integer m to

$$\sum_{k=1}^{n} \frac{(-1)^{k-1}\, q^{\binom{k+1}{2}}}{1-q^{k+m}} \begin{bmatrix} n \\ k \end{bmatrix} = \sum_{k=1}^{n} \frac{q^k}{1-q^k} \Bigg/ \begin{bmatrix} k+m \\ k \end{bmatrix}. \tag{46}$$

This result has also been proven utilizing the differentiation techniques we have applied here (see [1]). It is therefore natural to ask if it is possible to find for $k > 1$ a finite analogue to (8) and a generalization of type (46).

Another open question concerns the polynomial $H_k(S_0(x), \ldots, S_{k-1}(x))$ of Theorem 2.6: by letting the program presented in section 5 compute H_k for higher k we conjecture that

$$H_k(S_0(x), \ldots, S_{k-1}(x)) = \sum_{\pi \vdash k} c(k, \pi)\, S_0(x)^{m_1(\pi)} \cdots S_{k-1}(x)^{m_r(\pi)}, \tag{47}$$

analogous to the remark we made for the polynomial M_k.

Finally, it would be interesting to generalize Theorem 3.1 to the case where $f(n)$ is a periodical sequence. For instance, if we consider

$$f(n) = (-1)^n,$$

we obtain

$$\lim_{n\to\infty}\left(\sum_{i=1}^{n} f(i) - a_n(q)\right) = \sum_{j\geq 1}(-q)^{j^2}. \tag{48}$$

Note added in proof. K. Dilcher (Discrete Math., 145 (1995), pp. 83–93) has found an ingenious alternative proof for Theorem 2.1 in addition to many related results.

REFERENCES

[1] G.E. Andrews and K. Uchimura, *Identities in combinatorics* IV: *Differentiation and harmonic numbers*, Utilitas Math., 28 (1985), pp. 265–269.

[2] D. Bressoud and M. Subbarao, *On Uchimura's connection between partitions and the number of divisors*, Canad. Math. Bull., 27 (1984), pp. 143–145.

[3] R.L. Graham, D.E. Knuth, and O. Patashnik, *Concrete Mathematics*, Addison–Wesley, Reading, MA, 1990.

[4] G. Gasper and M. Rahman, *Basic Hypergeometric Series*, Cambridge University Press, Cambridge, 1990.

[5] G. Hardy and E. Wright, *An Introduction to the Theory of Numbers*, Oxford Science Publications, New York, 1989.

[6] E. Palmer, *Graphical Evolution*, Wiley Interscience Publishers, New York, 1985.

[7] K. Simon, D. Crippa, and F. Collenberg, *On the distribution of the transitive closure in random acyclic digraphs*, in Algorithms - ESA '93, Springer-Verlag, Berlin, Lecture Notes in Computer Science, 726 (1993), pp. 345–356.

[8] K. Uchimura, *An identity for the divisor generating function arising from sorting theory*, J. Combin. Theory Ser. B, 31 (1981), pp. 131–135.

[9] K. Uchimura, *A generalization of identities for the divisor generating function*, Utilitas Math., 25 (1984), pp. 377–379.

[10] L. Van Hamme, *Advanced problem* 6407, Amer. Math. Monthly, 9 (1982), pp. 703–704.

SIAM J. DISCRETE MATH.
Vol. 10, No. 1, pp. 57–72, February 1997

OBSTRUCTIONS FOR 2-MÖBIUS BAND EMBEDDING EXTENSION PROBLEM*

MARTIN JUVAN† AND BOJAN MOHAR†

Abstract. Let $K = C \cup e_1 \cup e_2$ be a subgraph of G consisting of a cycle C and disjoint paths e_1 and e_2 connecting two interlacing pairs of vertices in C. Suppose that K is embedded in the Möbius band in such a way that C lies on its boundary. An algorithm is presented which in linear time extends the embedding of K to an embedding of G, if such an extension is possible, or finds a "nice" obstruction for such embedding extensions. The structure of obtained obstructions is also analyzed in detail.

Key words. surface embedding, obstruction, Möbius band, algorithm

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

PII. S0895480194264150

1. Introduction.

Let K be a subgraph of a graph G. A *K-bridge* (or a *K-component*) in G is a subgraph of G which is either an edge $e \in E(G) \setminus E(K)$ (together with its endpoints) which has 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. A vertex of K of degree in K different from two is a *main vertex* of K. For convenience, if a connected component of K is a cycle, then we choose an arbitrary vertex of it and declare it to be a main vertex of K as well. A *branch* of K is any path (possibly a closed path) in K whose endpoints are main vertices, but no internal vertex on this path is a main vertex. If a K-bridge has all vertices of attachment on a single branch of K, it is said to be *local*.

This paper is part of a larger project [JMM, M4] which shows that there is a linear time algorithm to construct embeddings of graphs in an arbitrary (fixed) surface, generalizing the well-known Hopcroft–Tarjan algorithm [HT] for testing planarity in linear time. Our algorithms rely on the theory of bridges: a subgraph K of G is embedded in the surface and then either this embedding is extended to an embedding of G or an obstruction for such extensions is found. In this paper we solve and analyze a particular case of this problem where the underlying surface is the Möbius band dissected by K into two faces. It is shown that obstructions for extending the embedding of K either are small or have a very special (millipede) structure. Moreover, finding an embedding extension or such an obstruction requires only linear time (see Theorem 5.3).

These results are used and extended in [JM] and [M1]. Related results are also obtained in [M1, M2].

In our algorithms, we consider embeddings of graphs. In case of orientable surfaces, embeddings can be described combinatorially [GT] by specifying a *rotation system*: for each vertex v of the graph G we have cyclic permutation π_v of its neighbors, representing their circular order around v on the surface. Although the Möbius

* Received by the editors March 7, 1994; accepted for publication (in revised form) February 1, 1996. 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/10-1/26415.html

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

band is nonorientable, such a presentation suffices in our case since it is enough to specify rotation system in each of the faces of the chosen embedding of K. In order to make a clear presentation of our algorithm, we have decided to use this description only implicitly. Whenever we say that we have an embedding, we mean such a combinatorial description.

Concerning the time complexity of our algorithms, we assume a random-access machine (RAM) model with unit cost for basic operations. This model was introduced by Cook and Reckhow [CR]. More precisely, our model is the *unit-cost* RAM, where operations on integers whose value is $O(n)$ need only constant time (n is the size of the given graph).

2. Parallel computations with constant time overhead. We will need the following simulation of parallelism performed on a unit-cost RAM. At certain steps of our algorithm we will not be able to decide in advance between two possible choices. In such a case we will continue computations simultaneously in both directions. This will enable us to efficiently choose between the two alternatives. During such parallel computations no new parallelism will be introduced.

Denote by $\mathcal{P}_1$ and $\mathcal{P}_2$ both parallel processes. During the parallel computation exactly one of the following three cases will occur:

(i) The process $\mathcal{P}_1$ terminates *successfully*. This means that at the beginning of the parallelism the decision for $\mathcal{P}_1$ would be the right one. In this case, we say that the parallel computation terminates *successfully*. We also stop $\mathcal{P}_2$ (if still active) and restore the memory to the state before starting parallelism, choose the alternative $\mathcal{P}_1$ as the proper one, and continue with (nonparallel) computation from this point on.

(ii) If $\mathcal{P}_2$ terminates successfully, then we act as in the previous case, except that we stop $\mathcal{P}_1$ and choose the second alternative as the right one.

(iii) If neither $\mathcal{P}_1$ nor $\mathcal{P}_2$ terminates successfully, then the parallel computation is said to terminate *nonsuccessfully*.

If one of the processes fails, we still continue to run the remaining one. If it succeeds, case (i) or (ii) occurs; if the other process also fails, we have case (iii).

In our application of parallelism, the processes $\mathcal{P}_1$ and $\mathcal{P}_2$ will try to extend a partial embedding of a graph in two different ways. If appropriate embedding extension is found by one of them, this process will be termed as successful. Otherwise an obstruction for a particular type of embedding extension problem will be found. In case (iii) the "union" of both obstructions will give rise to a more general obstruction.

We want to ensure that the amount of time spent by both processes is proportional to the work done by either of them. To reach this goal, the actual implementation proceeds as follows. Each parallel process will have only read access to the memory of the main process (*global memory*) and also its own "copy" of this memory (*local memory*). Because of the restrictions on the time spent by the parallel computations we do not copy the data from the global memory to the process' local memory. Otherwise it might happen that the process performs only a small amount of work and then terminates successfully; therefore, the amount of work done at this parallel session is small, while copying the whole graph and auxiliary structures to the local memory could take time proportional to the size of the input. To avoid these time-consuming operations we propose the following simple memory management for local memory. Each cell in the local memory is either *empty* or *occupied*. If it is empty, this means that its corresponding cell in the global memory would still have the initial contents if the current parallel process would be performed on the global memory. If it is occupied, its new contents are stored in the local memory, so that the global memory remains unchanged. When requiring contents of a cell, the current process first checks in the local memory whether the cell is empty or occupied. If it is empty, it reads the contents from the corresponding cell in the global memory. Otherwise, it takes data from the local memory. New cell contents are always stored in the process' local memory.

To be able to efficiently delete the contents of the parallel process' local memory after the termination of the process (and so prepare it for another parallel session) each parallel process is associated with a list of occupied cells in its local memory. When deleting the contents of the local memory, only these cells need to be considered. (Only the very first "cleaning" is done by the main process in the initialization phase of the algorithm.) Initially, at the start of the parallel process, all cells in the local memory are empty. Moreover, the list of occupied cells is also empty. When during the computation an empty cell becomes occupied, the list is updated accordingly.

It is obvious that the above memory management adds only constant time overhead to every operation performed by the parallel process. Moreover, the final "cleaning" of the local memory needs at most time proportional to the amount of work performed by the process.

It can be shown that parallelism can be realized on the standard RAM although we do not have access to the program counter. The time complexity increases by a constant factor (depending on the length of the program) in order to maintain parallelism.

Let us mention that the above method of choosing between alternatives by testing them in parallel could also be (equally efficiently) implemented when the number of alternatives is constant (but possibly greater than two).

3. Obstructions. Let K be a fixed graph embedded in some surface. *Embedding extension problem* asks if for a given graph $G \supseteq K$ it is possible to extend the chosen embedding of K to an embedding of G. A subgraph Ω of $G - E(K)$ is an *obstruction* (for embedding extensions of K to G) if there is no embedding of $K \cup \Omega$ extending the chosen embedding of K. Because of Lemma 4.1 we will be able to assume that all obstructions we will work with contain only entire K-bridges. Moreover, we will be only interested in *minimal* obstructions, i.e., obstructions in which no bridge is redundant. It will turn out that for our particular case of embedding extension problem, minimal obstructions can be precisely characterized. They are either small, i.e., composed of a small number of bridges, or (although arbitrarily large) of a very special form which will be introduced in what follows.

Let $K = C \cup e_1 \cup e_2$ be a graph homeomorphic to K_4, where C is a cycle and e_1, e_2 are disjoint paths connecting pairs of interlacing vertices in C. Suppose that K is 2-cell embedded in the Möbius band in such a way that C lies on its boundary. Denote by F_1 and F_2 the faces of K under this embedding (cf. Figure 1). We say that K-bridges B and B' *overlap* in a face of K if they cannot be simultaneously embedded in that face.

For the purpose of the following definitions we will assume that all bridges of K in G are small (Lemma 4.1). If this were not the case, the bridges B_i° appearing in the definitions should be replaced by their H-subgraphs (cf. [M2, M3]).

A *thin millipede* in G *based* on e_1 and with *apex* $x \in V(e_2)$ is a subgraph M of $G - E(K)$ which can be expressed as $M = B_1^\circ \cup \cdots \cup B_m^\circ$ ($m \geq 7$), where we have the following:

(M1) Each of B_1° and B_m° is a K-bridge in G. Moreover, $B_1^\circ \cup B_2^\circ \cup B_3^\circ$ is uniquely

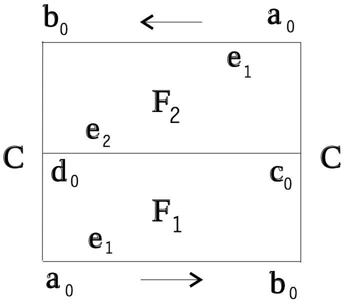


FIG. 1. *Embedding of K in the Möbius band.*

embeddable in $F_1 \cup F_2$. Let F_α be the face containing B_1° under this embedding. Similarly, $B_{m-2}^\circ \cup B_{m-1}^\circ \cup B_m^\circ$ is uniquely embeddable, and let F_β be the face containing B_m°. If m is even, then $\alpha = \beta$. If m is odd, then $\alpha \neq \beta$.

(M2) $B_2^\circ, \dots, B_{m-1}^\circ$ are distinct K-bridges that are attached to e_1 and to x and are not attached to K elsewhere.

(M3) For each $i = 1, 2, \dots, m-1$, B_i° and B_{i+1}° overlap in F_1 and in F_2.

(M4) For $i > 1$ and $i + 2 \le j < m$, B_i° and B_j° can be simultaneously embedded in F_1 and in F_2. The same holds when $i = 1$ and $3 \le j < m$ for the face F_α. Similarly, B_i° $(1 < i \le m-2)$ and B_m° can be simultaneously embedded in F_β. Additionally, $B_1^\circ \cup B_m^\circ$ can be embedded in $F_\alpha \cup F_\beta$.

It is clear by (M1) and (M3) that a thin millipede M obstructs embedding extensions of K to G.

Our notion of millipedes differs slightly from the concept of millipedes introduced in [M2]. The millipedes in [M2] can be shorter (i.e., $m < 7$ is allowed), and their subgraphs B_i° are allowed to be proper subgraphs of bridges in order that millipedes become minimal obstruction (with respect to the graph inclusion). On the other hand, after eliminating redundant branches in bridges B_i°, we can get from our thin millipedes a millipede in the sense of [M2].

We will also need *skew millipedes* based on e_1. They are defined similarly as thin millipedes. The apex of a thin millipede is replaced by a pair of vertices $x, y \in V(e_2)$ where no K-bridge is attached to e_2 on the (open) segment between x and y. The bridges $B_1^\circ, B_2^\circ, \dots, B_m^\circ$ satisfy (M1) and (M3), while (M2) and (M4) are replaced by the following.

(M2′) $B_2^\circ, \dots, B_{m-1}^\circ$ are distinct K-bridges. If i is even $(1 < i < m)$, then B_i° is attached to e_1 and to x (and not elsewhere). If i is odd $(1 < i < m)$, then B_i° is attached to e_1 and to y (and not elsewhere).

(M4′) For $i > 1$ and $i + 2 \le j < m$, B_i° and B_j° can be simultaneously embedded in F_α if either $i \not\equiv \alpha \pmod 2$ or $j \equiv \alpha \pmod 2$ (or both). They can be simultaneously embedded in $F_{3-\alpha}$ if either $i \equiv \alpha \pmod 2$ or $j \not\equiv \alpha \pmod 2$ (or both). For $3 \le j < m$, $B_1^\circ \cup B_j^\circ$ can be embedded in F_α. For $1 < i \le m-2$, $B_i^\circ \cup B_m^\circ$ can be embedded in F_β. Additionally, $B_1^\circ \cup B_2^\circ \cup B_3^\circ \cup B_{m-2}^\circ \cup B_{m-1}^\circ \cup B_m^\circ$ can be embedded in $F_1 \cup F_2$.

An equivalent definition of a skew millipede is that (M2′) together with the last condition in (M4′) holds and after contracting the (closed) segment on e_2 between x and y, we get a thin millipede.

In referring to a *millipede*, we mean either a thin or a skew millipede. It is clear from the description that millipedes are obstructions for embedding extensions.

It follows from (M4) ((M4′), respectively) that they are also minimal (no bridge is redundant).

An obstruction will be called *nice* if it is either composed of a small number of bridges (at most 13) or is a millipede. Millipedes based on e_2 and with apex $x \in V(e_1)$ ($\{x, y\} \subseteq V(e_1)$, respectively) are defined analogously. If the numbering of bridges in a millipede is reversed (i.e., $B'_i = B^\circ_{m-i+1}$), then $B'_1, \ldots, B'_m$ also satisfy (M1)–(M4) (or (M1)–(M4′)).

4. 2-Möbius band algorithm. Let G be a connected graph and $K = C \cup e_1 \cup e_2$ a subgraph of G homeomorphic to K_4, where C is a cycle and e_1, e_2 are disjoint paths connecting interlacing pairs a_0, b_0 and c_0, d_0 (respectively) of vertices in C. Suppose that K is embedded in the Möbius band with C on its boundary and that F_1 and F_2 are the faces of this embedding (cf. Figure 1). The problem of extending the embedding of K to an embedding of G will be referred to as the 2-*Möbius band embedding extension problem* [M1].

In this section we will outline a linear time algorithm for the 2-Möbius band embedding extension problem which finds an embedding extension whenever possible. We will show in section 5 how to extend this algorithm in order to construct a nice obstruction in case embedding extensions do not exist.

The next result will enable us to replace every K-bridge B in G by a small subgraph $\tilde{B} \subseteq B$ such that the embedding extension problem for the new graph is equivalent to the original one.

If B is a bridge of K in G, denote by $b(B)$ the number of branches of $B \cup K$ that are contained in B. The number $b(B)$ is called the *size* of B.

LEMMA 4.1 (see [M3]). *Let G, K be as above. Every K-bridge B in G contains a subgraph $\tilde{B}$ with size at most* 13 *such that for an arbitrary set of nonlocal K-bridges $B_1, \ldots, B_k$, any embedding of $K \cup \tilde{B}_1 \cup \cdots \cup \tilde{B}_k$ in the Möbius band with C on the boundary can be extended to an embedding of $K \cup B_1 \cup \cdots \cup B_k$. Moreover, the replacement of all K-bridges B by their subgraphs $\tilde{B}$ can be done in linear time.*

Let $\mathcal{B}$ be the set of K-bridges in G. We assume that no bridge in $\mathcal{B}$ is local on e_1 or on e_2. Denote by $\mathcal{B}_0$ the subset of $\mathcal{B}$ containing exactly those bridges which have no vertex of attachment in $C - e_1 - e_2$. These bridges are candidates to be embedded either in F_1 or in F_2. From now on we will also assume that the replacement of all K-bridges B by their small subgraphs $\tilde{B}$ (Lemma 4.1) has already been made. Moreover, we assume that every bridge can be embedded in at least one of the faces F_1, F_2. Otherwise we get a small obstruction and stop immediately. In particular, if some bridge is attached only to two vertices of K, the above replacement changes it into a branch. Moreover, we will assume that multiple branches between the same vertices of K have been replaced by a single one.

Suppose that $B \in \mathcal{B}_0$. For $\mathrm{y} \in \{a, b\}$, let y_B be the vertex of attachment of B on e_1 as close to y_0 as possible. Define similarly c_B and d_B as "extreme" attachments of B on e_2. Since there are no local bridges, the quantities x_B ($\mathrm{x} \in \{a, b, c, d\}$) are well defined for every $B \in \mathcal{B}_0$. We define $\bar{a} = d$, $\bar{b} = c$, $\bar{c} = b$, and $\bar{d} = a$ and $\tilde{a} = c$, $\tilde{b} = d$, $\tilde{c} = a$, and $\tilde{d} = b$. Note that $\bar{\mathrm{x}}_0$ and x_0 are in the same side (left or right) of F_1 and that $\tilde{x}_0$ and x_0 lie in the opposite corners of F_1.

We will first construct four lists of bridges in $\mathcal{B}_0$. They will be denoted by S_{x}, where x stands for either a, b, c, or d. The list S_{x} corresponds to the (oriented) branch e_1 or e_2 of K containing the vertex x_0 oriented from x_0 towards the other endpoint (e.g., S_c corresponds to e_2 oriented from c_0 towards d_0). Every list S_{x} will link all bridges from $\mathcal{B}_0$. Their order in S_{x} will be consistent with the following requirements.

(S1) If x_Q is closer to x_0 than x_R, then the bridge Q precedes R in S_x.

(S2) If $\mathrm{x}_Q = \mathrm{x}_R$ and $\bar{\mathrm{x}}_Q$ is closer to $\bar{\mathrm{x}}_0$ than $\bar{\mathrm{x}}_R$, then Q precedes R in S_x.

(S3) If $\mathrm{x}_Q = \mathrm{x}_R$, $\bar{\mathrm{x}}_Q = \bar{\mathrm{x}}_R$, and Q is attached only to x_Q and $\bar{\mathrm{x}}_Q$ and R has at least three vertices of attachment, then Q precedes R in the list S_x.

If a pair of bridges from $\mathcal{B}_0$ does not fit any of (S1), (S2), or (S3), then their order in S_x is irrelevant. If a set of bridges from $\mathcal{B}_0$ is embedded in F_1, then their order in F_1 from left to right is consistent with S_a and S_d and inverse to their order in S_b or S_c.

Suppose that e_j is the branch containing x_0. Let $v_1, v_2, \ldots, v_k$ be the vertices of e_j in direction from x_0 towards the other end. The list S_x is the concatenation of lists S_x^s, $s = 1, \ldots, k$, where each S_x^s links all bridges $B \in \mathcal{B}_0$ with $\mathrm{x}_B = v_s$ (in order respecting (S2) and (S3)). The lists S_x^s are constructed simultaneously as follows:

```
$S_\mathrm{x}^s := \emptyset,\ s = 1, \ldots, k$
Label all bridges in $\mathcal{B}_0$.
for all $u \in V(e_{3-j})$ do
  {The vertices $u$ are taken in order as they appear on
    $e_{3-j}$ from $\bar{\mathrm{x}}_0$ towards the other end.}
  for all edges $f$ incident with $u$ do
    if $f$ is a foot of a labeled bridge $B$ then
      if $B$ is attached only to two vertices then
        add $B$ at the end of $S_\mathrm{x}^s$, where $s$ is such that $v_s = \mathrm{x}_B$
        unlabel $B$
      endif
  endfor
  for all edges $f$ incident with $u$ do
    if $f$ is a foot of a labeled bridge $B$ then
      if $B$ is attached to three or more vertices then
        add $B$ at the end of $S_\mathrm{x}^s$, where $s$ is such that $v_s = \mathrm{x}_B$
        unlabel $B$
      endif
  endfor
endfor
Link $S_\mathrm{x}^1, \ldots, S_\mathrm{x}^k$ into $S_\mathrm{x}$.
```

It is easy to realize the traversals in the above algorithm so that the overall time spent by the algorithm is linear. Note that the double traversal of bridges with $\bar{\mathrm{x}}_B = u$ assures that condition (S3) will be fulfilled. Condition (S2) is satisfied at the end since the traversal of the “opposite” branch e_{3-j} is performed in the direction from $\bar{\mathrm{x}}_0$ towards the other end. Clearly, (S1) is guaranteed by the use of sublists S_x^s and their appropriate linking at the end.

We are now ready to discuss the main part of the algorithm. Roughly speaking, it is based on the following idea. Suppose that a subset of bridges $\mathcal{B}' \subseteq \mathcal{B}$ is *already embedded* in $F_1 \cup F_2$. Their presence in $F_1 \cup F_2$ blocks some embeddings of the remaining bridges. Some of the bridges thus need to be embedded in F_1; some others can only be embedded in F_2. We say that these bridges are *forced* in F_1 (or F_2, respectively). By adding these blocked bridges to $\mathcal{B}'$, we obtain additional bridges with only one face left for their embeddings. By repeating this procedure, either we get stuck (which proves that no embedding extension exists with the initial $\mathcal{B}'$ embedded as given) or no more bridges are blocked by the chosen embedding of $\mathcal{B}'$. In the latter case, it is clear that the bridges in $\mathcal{B}'$ can be left embedded as they are without obstructing any possible embeddings of the remaining bridges. The procedure described above is called FORCING.

At the very beginning, the bridges from $\mathcal{B}\setminus\mathcal{B}_0$ are uniquely embeddable, and they are used as the starting set $\mathcal{B}'$. If FORCING does not embed all of the bridges (and does not get stuck), then the problem is how to restart. (This cannot be avoided if, for example, $\mathcal{B}_0 = \mathcal{B}$.) This problem will be solved by using parallel computations. We choose a bridge B and start two parallel processes: the first one corresponds to embedding B in F_1, the other to the case where we embed B in F_2. The details of how to perform such parallel computations without increasing the overall time complexity are described in section 2. Each of the two parallel processes either finds an embedding for a set of bridges which does not interfere with any embedding of the remaining bridges (*successful* termination) or gets stuck (*nonsuccessful* termination). It has been described in section 2 how the two parallel processes react if one or the other stops successfully. To ensure linear time complexity we have to choose the starting bridge B appropriately: it must be the initial bridge in one of the lists S_{x}. Of course, these lists are updated during the algorithm by removing the already embedded bridges.

For x being any of a, b, c, or d, we will use three vertices $\mathrm{x}, \mathrm{x}_1, \mathrm{x}_2$ on the branch e_j ($j \in \{1, 2\}$) containing the vertex x_0. For $u, v \in V(e_j)$, denote by $[u, v)$ the segment of e_j from u to v (including u but not including v) and similarly by $[u, v]$ the closed segment of e_j from u to v (including both u and v). During the algorithm, all bridges attached to $[\mathrm{x}_0, \mathrm{x})$ are already embedded and all remaining bridges attached to $[\mathrm{x}, \mathrm{x}_i)$ ($i = 1, 2$) are blocked in F_{3-i} by already embedded bridges, so they will need to be put in F_i. (In particular, if a bridge that has not yet been embedded is attached to $[\mathrm{x}, \mathrm{x}_1) \cap [\mathrm{x}, \mathrm{x}_2)$, then we are in trouble.) In the algorithm we also use bridges $B_{\mathrm{x},1}$, $B_{\mathrm{x},2}$. They are needed only for the efficient construction of obstructions, and their use is described in more detail in section 5.

The main part of the algorithm is the following.

```
Determine lists $S_{\mathrm{x}}$, $\mathrm{x} \in \{a, b, c, d\}$, as explained above.
Determine $\mathcal{B}_0$. Let $\mathcal{B}' := \mathcal{B} \setminus \mathcal{B}_0$.
Embed $\mathcal{B}'$.
**if** no embedding exists **then** OBSTRUCTION
Determine initial values of $\mathrm{x}, \mathrm{x}_1, \mathrm{x}_2, B_{\mathrm{x},1}, B_{\mathrm{x},2}$ for $\mathrm{x} \in \{a, b, c, d\}$.
FORCING
**if not** successful **then** OBSTRUCTION
Initialize auxiliary variables for parallel computations.
**while** $\mathcal{B}_0 \neq \emptyset$ **do**
  $B :=$ the first bridge in $S_a$
  **for** every embedding of $B$ in $F_1 \cup F_2$ **do in parallel**
    Determine initial values of $\mathrm{x}, \mathrm{x}_1, \mathrm{x}_2, B_{\mathrm{x},1}, B_{\mathrm{x},2}$ for $\mathrm{x} \in \{a, b, c, d\}$.
    FORCING
  **end parallel for**
  **if not** successful **then** OBSTRUCTION
**endwhile**
{If we reach this point, all the bridges have been embedded.}
Return the obtained embedding extension.
```

Procedure OBSTRUCTION reports that no embedding extension exists and terminates. We will show in section 5 that by extending this procedure, one can also construct nice obstructions (cf. section 3) for embedding extensions in linear time. Procedure FORCING is described below.

```
**procedure** FORCING
{Some bridges are already embedded. They block some embeddings of
 the remaining bridges. A bridge $B \in \mathcal{B}_0$ is blocked exactly when it is
```

```
      attached to a segment $[\mathrm{x},\mathrm{x}_i)$ for some $\mathrm{x} \in \{a,b,c,d\}$, $i \in \{1,2\}$.
      In that case, it must be embedded in $F_i$.}
        while $\exists \mathrm{x} \in \{a,b,c,d\}$ such that $\mathrm{x} \neq \mathrm{x}_1$ or $\mathrm{x} \neq \mathrm{x}_2$ do
        if $\mathrm{x} \neq \mathrm{x}_1$ and $\mathrm{x} \neq \mathrm{x}_2$ then
          $y := \min\{\mathrm{x}_1,\mathrm{x}_2\}$ (closer to x)
          if $\exists B \in \mathcal{B}_0$ attached to $[x,y)$ then Stop(not successful)
          $\mathrm{x} := y$
        endif
        if $\mathrm{x} \neq \mathrm{x}_1$ then $i := 1$ else $i := 2$ endif
        $\mathcal{B}_i :=$ all bridges in $\mathcal{B}_0$ attached to $[\mathrm{x},\mathrm{x}_i)$
        Embed $\mathcal{B}_i$ in $F_i$.
        if no embedding exists then Stop(not successful)
        $\mathrm{x} := \mathrm{x}_i$
        Update $a_{3-i}$, $b_{3-i}$, $c_{3-i}$, $d_{3-i}$, and $S_\mathrm{x}$.
        $\mathcal{B}_0 := \mathcal{B}_0 \setminus \mathcal{B}_i$
        $B_{x,i} :=$ extreme bridge in $\mathcal{B}_i$
        Let $B_{x,i}$ point to $B_{x,3-i}$.
        {This will be needed in the construction of obstructions.}
    endwhile
    Return(successful)
  end {Forcing}
```

The search for $B \in \mathcal{B}_0$ that is attached to $[\mathrm{x}, y)$ in the above procedure can be easily implemented by advancing through the list S_x. Similarly, the embeddability of $\mathcal{B}_i$ in F_i is checked by moving along the list S_x and comparing extreme vertices of attachment of bridges with already blocked segments on e_1 and e_2. More precisely, this is achieved as follows. Let $B_1, \ldots, B_t$ be the bridges in $\mathcal{B}_i$ listed in the order as they appear in S_x. Suppose that $\mathrm{x}_0 \in V(e_j)$ and denote by y_0 the other endpoint of e_j. Obviously, each bridge B_k $(1 \leq k \leq t)$ must have an embedding in F_i. Suppose first that $i = 1$. Each B_k must also be attached to e_j (entirely on the segment $[\mathrm{x}, \mathrm{y}_{3-i}]$) and to e_{3-j} (entirely to the segment $[\bar{\mathrm{x}}_{3-i}, \bar{\mathrm{y}}_{3-i}]$; otherwise it overlaps with the already embedded bridges). Moreover, for $k = 1, \ldots, t-1$ the bridge B_{k+1} must be entirely attached to the segment $[\mathrm{y}_{B_k}, \mathrm{y}_0]$ of e_j and to the segment $[\tilde{\mathrm{x}}_{B_k}, \tilde{\mathrm{x}}_0]$ of e_{3-j}; otherwise it overlaps with B_k in F_i. If none of these tests fails, then the bridges in $\mathcal{B}_i$ can be simultaneously embedded in F_i. When $i = 2$, some details in the above tests have to be modified appropriately since the list S_x is constructed with respect to embeddings in the face F_1. In particular, $\bar{\mathrm{x}}$ and $\bar{\mathrm{y}}$ have to be replaced by $\tilde{\mathrm{x}}$ and $\tilde{\mathrm{y}}$, respectively, and vice versa. Moreover, bridges $B \in \mathcal{B}_i$ with the same extreme attachment x_B have to be considered in the order that is opposite to their order in S_x. During the above test we also change $\mathcal{B}_0$.

Initial values of a, a_1, a_2 (and similarly for other $\mathrm{x}, \mathrm{x}_1, \mathrm{x}_2$, $\mathrm{x} \in \{b, c, d\}$) are determined at the very beginning as follows. We take $a = a_0$. The vertex a_1 is equal to the vertex of attachment on e_1 closest to b_0 of bridges in $\mathcal{B} \setminus \mathcal{B}_0$ that are attached to the open segment from c_0 to a_0 on C. The corresponding bridge is taken as $B_{a,2}$. If there are no such bridges, then $a_1 = a_0$ (and $B_{a,2}$ is undefined). Similarly, a_2 is the attachment on e_1 closest to b_0 of bridges in $\mathcal{B} \setminus \mathcal{B}_0$ attached to the open segment on C from a_0 to d_0. The corresponding bridge is then $B_{a,1}$.

There is a slight difference in determining the initial values of $\mathrm{x}, \mathrm{x}_1, \mathrm{x}_2$ in the parallel part. The values x remain unchanged. If B (the initial bridge in S_a) is embedded in F_1, then $\mathrm{x}_1 = \mathrm{x}$ for $\mathrm{x} \in \{a, b, c, d\}$, $a_2 = b_B$, $b_2 = b$, $c_2 = c$, and $d_2 = c_B$.

We take $B_{a,1} = B_{d,1} = B$. Other $B_{\mathrm{x},j}$ are undefined.

If B is embedded in F_2, the situation is more complex. In this case the process of determining the initial values of $\mathrm{x}, \mathrm{x}_1, \mathrm{x}_2, B_{\mathrm{x},1}, B_{\mathrm{x},2}$ ($\mathrm{x} \in \{a, b, c, d\}$) will require some additional preprocessing in order to decide between two possible choices:

(a) If all bridges attached to e_1 only at a can be simultaneously embedded in F_2 (together with B), then they can go in F_2 without loss of generality. All other bridges attached to e_2 on $(d_B, c]$ must be in F_1, and, after fixing these embeddings, we change c to become the vertex d_B and proceed in the same way as in the above case when B was in F_1. (We set $a_1 = b_B$, $c = c_1 = d_B$, $c_2 = d_R$, $b_2 = a_R$, where R is the "leftmost" bridge among those which were embedded in F_1; if R is not attached to the segment $[d, d_B)$, we take $c_2 = c$; if there are no such bridges, then $c_2 = c$, $b_2 = b$. Also, $B_{a,2} = B$, $B_{b,1} = B_{c,1} = R$, or undefined; other $B_{\mathrm{x},j}$ are always undefined.) If the above bridges cannot be simultaneously embedded in F_1, we terminate nonsuccessfully.

(b) Two bridges B', B'' attached to e_1 only at a overlap in F_2 or such a bridge B' overlaps with B in F_2. Hence, one of B', B'' should be embedded in F_1. Then all bridges attached to e_2 on $[d, d_B)$ must be in F_2. Similarly, all bridges attached to e_2 only at d_B and attached to $(a, b]$ on e_1 will necessarily go into F_2. After fixing these embeddings we let $d = d_B$ and change other values $\mathrm{x}, \mathrm{x}_1, \mathrm{x}_2$ ($\mathrm{x} \in \{a, b, c, d\}$) as described below.

We need to make the decision about (a) or (b) in such a way that the time spent on this is proportional to the number of bridges whose embedding is determined during this process. (Otherwise, we can lose linearity.) This is achieved by traversing the list S_c. Let B' be the current bridge in the traversal. If $b_{B'} \neq a$, then we must embed B' in F_1; if it overlaps with already embedded bridges, call Obstruction. If not, embed B' in F_1 and proceed with the next bridge in the list S_c. Otherwise ($b_{B'} = a_{B'} = a$) we try to embed B' in F_2. If successful, we proceed with the next bridge in the list. If B' overlaps in F_2 with some already embedded bridge, we have (b). If B' overlaps in F_2 with an already embedded bridge $B'' \neq B$, then B'' is unique. Therefore, all other bridges that have been embedded during our traversal can retain their embeddings without loss of generality. The same is true in the other case when B' overlaps with B. In the first case we set $R = B''$ while in the latter case we take $R = B'$. In both cases we will consider R as a nonembedded bridge in what follows. Let Q be the last bridge embedded in F_1 during the traversal of S_c which has an attachment on $[c, c_R]$ (possibly undefined). Next we embed in F_2 all bridges attached to e_2 on $[d, d_B)$ and all bridges attached to d_B and to $(a, b]$. (If this is not possible, call Obstruction.) After these changes, the values $\mathrm{x}, \mathrm{x}_1, \mathrm{x}_2$ are determined as follows: a, b remain unchanged, $a_1 = a_2 = a$, $b_1 = a$, $b_2 = a_Q$ (or b if Q is undefined or attached to e_2 only at $(c_R, c]$), $c = c_1 = c_R$, $c_2 = d_Q$ (or c if Q is undefined), $d = d_B$, $d_1 = c_B$, $d_2 = d_B$. Bridges $B_{\mathrm{x},i}$ are defined accordingly.

If none of the above stop cases occurs, we stop when reaching d_B and then we have case (a).

5. 2-Möbius band obstructions.

Our algorithm can be extended in a relatively simple way so that when no embedding extension exists, it returns a nice obstruction. Procedure Obstruction takes care of this task if we modify it as explained in what follows.

There are three places where the presence of an obstruction is discovered:

(i) when embedding bridges of $\mathcal{B}'$,

(ii) in procedure Forcing,

(iii) when determining the initial values in the parallel part.

In case (i), we either get a K-bridge $B \in \mathcal{B}'$ that cannot be embedded in any of the faces, or get two bridges $B_1, B_2 \in \mathcal{B}'$ that are both embeddable only in F_i ($i \in \{1,2\}$), where they overlap. It is clear that this case leads to a small obstruction which can be determined efficiently by applying the results of [M2].

Consider now (ii). In Forcing, there are two obstruction stops. The first possibility is when a bridge $B \in \mathcal{B}$ is attached to $[\mathrm{x}, y)$. This means that possible embedding of B in F_1 is blocked by $B_{\mathrm{x},1}$ and its embedding in F_2 is blocked by $B_{\mathrm{x},2}$. When $B_{\mathrm{x},1}$ was embedded, we remembered which bridge forced it to be in F_1. It is similar for all other embedded bridges. Thus we can reconstruct a chain

$$(B_1, F_{i_1}) \to (B_2, F_{i_2}) \to \cdots \to (B_p, F_{i_p}), \tag{1}$$

where the notation $(Q, F) \to (R, F')$ means that Q and R cannot be simultaneously embedded in F (Q being embedded in F forces R being embedded in F') and where B_1 is one of the initial bridges with fixed embedding, and $(B_p, F_{i_p}) = (B, F_1)$. Let us note that $i_1, \dots, i_p \in \{1,2\}$ and that any two consecutive i_r, i_{r+1} are distinct. Also, $B_{p-1} = B_{\mathrm{x},1}$. Similarly, we have a chain forcing B to be in F_2:

$$(B'_1, F_{j_1}) \to (B'_2, F_{j_2}) \to \cdots \to (B'_q, F_{j_q}), \tag{2}$$

where $(B'_q, F_{j_q}) = (B, F_2)$. It is clear that $(Q, F_i) \to (R, F_{3-i})$ is equivalent to $(R, F_i) \to (Q, F_{3-i})$. Therefore (2) is equivalent to

$$(B'_q, F_{3-j_q}) \to (B'_{q-1}, F_{3-j_{q-1}}) \to \cdots \to (B'_1, F_{3-j_1}). \tag{3}$$

Note that $(B'_q, F_{3-j_q}) = (B_p, F_{i_p}) = (B, F_1)$. Now, (1) and (3) can be concatenated and rewritten in the form

$$(R_1, F_{s_1}) \to (R_2, F_{s_2}) \to \cdots \to (R_r, F_{s_r}), \tag{4}$$

where $(R_1, F_{s_1}) = (B_1, F_{i_1})$ and $(R_r, F_{s_r}) = (B'_1, F_{3-j_1})$.

The second stop in Forcing occurs when $\mathcal{B}_i$ cannot be simultaneously embedded in F_i. If $B \in \mathcal{B}_i$ overlaps in F_i with some of the already embedded bridges, we have exactly the same situation as above: we get (4). (As explained, this can be discovered by a simple comparison of the extreme attachments of bridges in $\mathcal{B}_i$ with $a_{3-i}, b_{3-i}, c_{3-i}, d_{3-i}$.)

The next possibility is that a bridge $B \in \mathcal{B}_i$ cannot be embedded in F_i (i.e., its only embedding is in F_{3-i}). Then we have

$$(R_1, F_{s_1}) \to (R_2, F_{s_2}) \to \cdots \to (R_r, F_{s_r}), \tag{5}$$

where $(R_r, F_{s_r}) = (B, F_i)$. This chain is not only of the same form as (4) but also obeys the same condition that will be used in producing nice obstructions: R_1 is embeddable only in F_{s_1}, and R_r is embeddable only in F_{3-s_r}, the opposite face of F_{s_r}.

It is similar if two bridges from $\mathcal{B}_i$ overlap in F_i. We easily get a chain of form (4) having the same properties as in the other cases.

If procedure Obstruction is reached because of unsuccessful termination of the parallel computation, we get two chains of the form (4), one from each parallel process. The first one starts with (B, F_1) and it is discovered in (ii). It satisfies the chain condition (the first bridge uniquely embeddable, the last one assigned to the wrong face) under the assumption that B is embeddable only in F_1. The second process gives rise to a similar chain. However, in this case the situation is slightly different. We either get a chain of the form (4) that is obtained in (ii) and starts with (B, F_2) or get a small obstruction from (iii) which itself gives rise to a chain of the form (4). More precisely, there are two possible calls to Obstruction in (iii). If there are two bridges R', R'' that overlap in F_2 and are forced in F_2 by B, then

$$(B, F_2) \to (R', F_1) \to (R'', F_2) \to (B, F_1)$$

is the required chain of the form (4). The second possibility is when the set of bridges

$$\mathcal{B}'' = \{R \in \mathcal{B} \mid d_R \in [d, d_B) \text{ or } (d_R = d_B \text{ and } b_R \in (a, b])\}$$

cannot be simultaneously embedded in F_2. In this case, there is also a pair B', $\tilde{B}$ of bridges (where $\tilde{B} = B''$ or B) attached to e_1 only at a and attached to e_2 entirely on $[c, d_B]$ that overlap in (F_1 and) F_2. Suppose first that there are two bridges $R', R'' \in \mathcal{B}''$ that overlap in F_2. Then the bridges B', $\tilde{B}$, R', R'' form a small obstruction for the whole embedding extension problem. The remaining possibility why the bridges from $\mathcal{B}''$ cannot be simultaneously embedded in F_1 is that there is a uniquely embeddable bridge $R' \in \mathcal{B}''$ that has no embedding in F_2. Then B', $\tilde{B}$, and R' form a small obstruction and we are done.

If the chain of the first parallel process starts with (B, F_1) and the chain of the other process starts with (B, F_2), we can concatenate one chain with the "inverse" of the other to get a chain of the form

$$(R_1, F_{s_1}) \to (R_2, F_{s_2}) \to \cdots \to (R_r, F_{s_r}). \tag{6}$$

In general, there are three possibilities why the chain of form (6) (or (4)) leads to an obstruction.

(A) As described before, R_1 is embeddable only in F_{s_1} and R_r is embeddable only in F_{3-s_r}. We allow that $R_1 = R_r$.

(B) $(R_1, F_{s_1}) = (R_r, F_{s_r}) = (B, F_1)$ and (B, F_2) appears somewhere in the chain.

(C) R_1 is embeddable only in F_{s_1} and $(B, F_1), (B, F_2)$ both appear somewhere in the chain.

The last case (C) can be transformed into a chain of type (A) as follows. If $(B, F_1) = (R_i, F_{s_i})$, $(B, F_2) = (R_j, F_{s_j})$, $i < j$, then we get

$$(R_1, F_{s_1}) \to \cdots \to (R_j, F_{s_j}) \to (R_{i-1}, F_{3-s_{i-1}}) \to \cdots \to (R_1, F_{3-s_1}).$$

We will show that the obstruction formed by the chain (6) (viewing (4) as case (A) of (6)) can be efficiently transformed into either a small obstruction or a (thin or skew) millipede. This will be achieved through a series of successive reductions of the chain (6). We will assume that $r \geq 14$. Otherwise we have a small obstruction formed by at most 13 bridges from our chain. If during the following reductions the length of the chain drops below 14, we automatically stop because we have obtained a small obstruction.

We say that bridges R and R' are *parallel* in F_i ($i \in \{1, 2\}$) if they cannot be simultaneously embedded in F_{3-i}, i.e., $(R, F_{3-i}) \to (R', F_i)$.

Lemma 5.1. *Let bridges R_i and R_{i+2} from (6) be parallel in F_{s_i}. Then in every embedding of $R_i \cup R_{i+1} \cup R_{i+2}$, the bridge R_{i+2} is embedded in F_{s_i}.*

Proof. Assume that there is an embedding of $R_i \cup R_{i+1} \cup R_{i+2}$ such that R_{i+2} is embedded in F_{3-s_i}. Since R_i is parallel with R_{i+2} in F_{s_i}, it is embedded in F_{s_i}. By (6), R_{i+1} is embedded in F_{3-s_i} and R_{i+2} should be embedded in F_{s_i}, which is a contradiction. □

Similar arguments also show that if R_i and R_{i+2j} are parallel in F_{s_i}, then in every embedding of $R_i \cup \cdots \cup R_{i+2j}$, the bridge R_{i+2j} is embedded in F_{s_i}. In such a case the bridge R_{i+2j} can be regarded as uniquely embeddable under the condition that the final obstruction also contains the bridges $R_i, \ldots, R_{i+2j-1}$. In what follows, we will need the above claim for $j = 1$ and $j = 2$.

If there is a pair (R_i, F_{s_i}) $(1 < i < r)$ in the chain (6) such that R_i can be embedded only in one face, then we act as follows. We may assume that R_i can be embedded in F_{s_i}, since otherwise we could look at the reversed chain

$$(6') \qquad (R_r, F_{3-s_r}) \to \cdots \to (R_2, F_{3-s_2}) \to (R_1, F_{3-s_1}),$$

where R_i appears in the right face. If the chain is of type (A), then we can shorten the obstruction by taking $(R_i, F_{s_i}) \to \cdots \to (R_r, F_{s_r})$. If the chain is of type (B), then we can change it into type (C). Let j $(1 < j < r)$ be an index such that $(R_j, F_{s_j}) = (R_r, F_{3-s_r})$. We take the chain $(R_i, F_{s_i}) \to \cdots \to (R_r, F_{s_r})$ if $i < j$. It is similar if $i > j$ when we take $(R_i, F_{s_i}) \to \cdots \to (R_r, F_{s_r}) \to (R_2, F_{s_2}) \to \cdots \to (R_j, F_{s_j})$. The obtained chain can be further reduced to type (A) as shown previously. It is easy to see how to implement the above tests and reductions in linear time.

From now on we will assume that every bridge participating in the chain (6), except the first and the last one when we have a chain of type (A), has (allowed) embeddings in faces F_1 and F_2. If there is a pair (R, F) which appears twice in the chain of type (A) we leave out pairs between the two appearances. In chains of type (B) this is performed only when the two appearances lie in the same segment of the chain between R_1 and its appearance in the other face. Again, this task can be easily performed in linear time.

Suppose that we have a chain of type (B). Then we perform another checking which will be needed in the proof of Lemma 5.2. Let (R_j, F_{s_j}) be the occurrence of R_1 in the other face. If $(R_{j-3}, F_{s_{j-3}}) \to (R_j, F_{s_j})$ or $(R_j, F_{s_j}) \to (R_{j+3}, F_{s_{j+3}})$, then we can shorten our chain by leaving out the two superfluous pairs. We repeat this change as long as possible. Under every embedding of $R_1 \cup \cdots \cup R_{j-1}$ in $F_1 \cup F_2$, the bridge $R_1 = R_j$ is embedded in F_{s_j}. Therefore we may assume that $j \geq 6$ since otherwise we can transform our chain of type (B) into a chain of type (A) (with at most four additional bridges which guarantee unique embeddability of R_1 in F_{s_j}). In this case we also repeat previous reductions on the new chain. Similarly, we may assume that $j \leq r - 5$. Note that all these changes can be done in linear time.

Next we check if there are pairs of parallel bridges which appear not far apart in the chain. Suppose that we have a chain of type (B) with bridges R_i and R_{i+2} being parallel in F_k. By reversing the chain, if necessary, we may assume that $F_k = F_{s_i}$. There exists an index j, $1 < j < r$, such that $(R_j, F_{s_j}) = (R_1, F_{3-s_1})$. We will regard R_{i+2} as uniquely embeddable in F_{s_i} (Lemma 5.1). We will actually achieve this property at the end by adding bridges R_i and R_{i+1} into the final obstruction. If $i+2 \leq j$, our chain can be shortened and transformed into type (C) by taking $(R_{i+2}, F_{s_{i+2}}) \to \cdots \to (R_r, F_{s_r})$. If $i + 2 > j$, we transform our chain into $(R_{i+2}, F_{s_{i+2}}) \to \cdots \to (R_r, F_{s_r}) = (R_1, F_{s_1}) \to \cdots \to (R_j, F_{s_j})$ which can be viewed as a chain of type (C). In both cases, our chain of type (C) can be further changed into type (A). If we have a pair of parallel bridges R_i, R_{i+4}, we take the same steps, except that in this case the final obstruction will have to contain not only bridges R_i, R_{i+1} but also bridges R_{i+2}, R_{i+3}. Obtaining the chain of type (A) we again perform the above reductions (no intermediate bridge uniquely embeddable, no repetitions). Note that this additional work can occur only once—when changing type (B) into type (A).

Let us now explain how to react regarding parallel bridges if we have a chain of type (A). For $j = r, r-1, \ldots, 3$ we check whether R_j is parallel with R_{j-2} and whether R_{j+2} is parallel with R_{j-2} (when $j \leq r-2$). If R_j and R_{j-2} are parallel in F_{s_j}, we shorten the chain by removing the initial part $(R_1, F_{s_1}) \to \cdots \to (R_{j-1}, F_{s_{j-1}})$ and stop. If they are parallel in F_{3-s_j}, then we remove the tail $(R_{j+1}, F_{s_{j+1}}) \to \cdots \to (R_r, F_{s_r})$ and continue with work. It is similar when R_{j-2} and R_{j+2} are parallel.

Let us remark that if R_1 is not embeddable in F_{3-s_1}, then R_1 and R_3 are parallel in $F_{s_1} = F_{s_3}$. Similarly, R_{r-2} and R_r are usually parallel in F_{3-s_r}. It is obvious how to perform the above tasks in linear time. By Lemma 5.1 the chain obtained after this reduction (together with at most $4 + 4 = 8$ additional bridges which guarantee the unique embeddability of the first and the last bridge in the chain of type (A)) still determines an obstruction. By the above remark, each bridge R_i $(1 \leq i \leq r)$ can be embedded in F_1 and in F_2, and no two bridges R_i, R_{i+2} $(1 \leq i \leq r-2)$ or R_i, R_{i+4} $(1 \leq i \leq r-4)$ are parallel in any of the faces.

Let $\mathcal{R}_1 = \{R_{2i-1} \mid 1 \leq i \leq \lceil r/2 \rceil\}$ and $\mathcal{R}_2 = \{R_{2i} \mid 1 \leq i \leq \lfloor r/2 \rfloor\}$.

Lemma 5.2. *There exists $j \in \{1, 2\}$ and a vertex $x \in V(e_j)$ such that every bridge from $\mathcal{R}_1$ is attached to e_j only at x. Similarly, there exists $k \in \{1, 2\}$ and a vertex $y \in V(e_k)$ such that every bridge from $\mathcal{R}_2$ is attached to e_k only at y.*

Proof. Since we have decided to stop whenever our obstructing family of bridges contains 13 or fewer members, we have $r \geq 6$. Consider the bridges $R_i, R_{i+2}, R_{i+4} \in \mathcal{R}_1$. Since they are pairwise nonparallel in F_1 and in F_2, they can be simultaneously embedded in any of the faces. Therefore their union cannot contain two disjoint paths connecting branches e_1 and e_2. Note that not all three bridges can be equal to each other. Hence there exists a vertex x in one of the branches, say e_j, such that x is the only vertex of attachment of $R_i \cup R_{i+2} \cup R_{i+4}$ to e_j. Moreover, $R_i \cup R_{i+2} \cup R_{i+4}$ is attached to at least two vertices on the branch e_{3-j}. Similarly, there is a vertex x' in the branch $e_{j'}$ such that $R_{i+2} \cup R_{i+4} \cup R_{i+6}$ is attached to $e_{j'}$ only at x'. If $R_{i+2} \neq R_{i+4}$, then it easily follows that $e_{j'} = e_j$ and $x' = x$. On the other hand, $R_{i+2} = R_{i+4}$ can only happen if our chain is of type (B) and $(R_{i+3}, F_{s_{i+3}}) = (R_1, F_{3-s_1})$. Since in this case $(R_{i+2}, F_{s_{i+2}}) \to (R_{i+3}, F_{s_{i+3}})$ and $(R_{i+3}, F_{s_{i+3}}) \to (R_{i+4}, F_{s_{i+4}}) = (R_{i+2}, F_{s_{i+2}})$, bridges R_{i+2} and R_{i+3} must overlap on e_1 or e_2. If they overlap on e_j, then $(R_i, F_{s_i}) \to (R_{i+3}, F_{s_{i+3}})$ which is not possible because of previous reductions. Therefore R_{i+2} and R_{i+3} overlap on e_{3-j}. Suppose that $x' \neq x$. Then also $e_{j'} = e_{3-j}$. Since R_{i+2} overlaps on $e_{3-j} = e_{j'}$ with R_{i+3} and since R_{i+6} is attached on $e_{j'}$ to the same vertex x' as R_{i+2}, we have $(R_{i+3}, F_{s_{i+3}}) \to (R_{i+6}, F_{s_{i+6}})$. But this is a contradiction, since we have reduced such forcing at previous steps. Consequently, $x' = x$. By increasing i, we easily derive the claimed result.

The proof of the second part is almost identical. □

Additionally, we claim that either x and y lie on the same branch, or there is a small obstruction. For vertices $u, v \in V(e_1)$ we say that u is to the *left* of v (or v is to the *right* of u) if u is closer to a_0 than v is. Similarly if $u, v \in V(e_2)$ we say that u is to the *left* of v if it is closer to d_0. Suppose now that $r > 5$ and that $x \in V(e_1)$, $y \in V(e_2)$. We will distinguish between two possibilities.

(i) If there is a bridge $R_i \in \mathcal{R}_1$ which is attached on e_2 to the left and to the right of y, then a small obstruction is obtained as follows. When the chain is of type (A), pairs $(R_1, F_{s_1}) \to (R_2, F_{s_2}) \to (R_i, F_{s_i})$ together with (R_r, F_{s_r}) if r is even or together with $(R_{r-1}, F_{s_{r-1}}) \to (R_r, F_{s_r})$ if r is odd form the desired obstruction. If the chain is of type (B), then R_1 is just the branch xy since $R_1 \in \mathcal{R}_1 \cap \mathcal{R}_2$. Let $(R_j, F_{s_j}) = (R_1, F_{3-s_1})$ be the occurrence of R_1 in the other face. Since R_2 overlaps with R_1 in F_{s_1} and R_1, R_2 are attached to e_2 only at y, R_2 is attached to e_1 to the left and to the right of x. Similarly, R_{j+1} is attached to the left and to the right of y on e_2. Then $R_1 \cup R_2 \cup R_{j+1}$ is a small obstruction. The case when there is a bridge $R \in \mathcal{R}_2$ attached on e_1 to the left and to the right of x is similar.

(ii) There is no bridge attached on e_1 to the left and to the right of x, and also there is no bridge attached on e_2 to both sides of y. In this case the chain must be of type (A) since otherwise $R_1 \in \mathcal{R}_1 \cap \mathcal{R}_2$ would be just the branch xy and would not be obstructed by any of the bridges. It is easy to see that under every embedding of $R_1 \cup R_2 \cup R_3 \cup R_4 \cup R_r$, the bridge R_r is embedded in F_{s_r}. Since this is the wrong face for R_r, we have an obstruction.

In both cases the obtained small obstruction (together with bridges which assure the unique embeddability of R_1 and R_r) contains at most 13 bridges.

So far we have been able to restrict attachments of the bridges from the chain at one of the branches to at most two vertices. It remains to find a millipede (or a small obstruction) composed of some of these bridges. First we examine the case when $x = y$. By a planarity testing we try to embed $\mathcal{R}_1 \cup \mathcal{R}_2$ in $F_1 \cup F_2$. (Planarity testing can be used because $\mathcal{R}_1 \cup \mathcal{R}_2$ is attached to one of e_1, e_2 just at a point.) If the test fails, there will be a small obstruction composed of three mutually overlapping bridges. Such bridges can be discovered in linear time by a traversal of the corresponding branch e_i $(i \in \{1, 2\})$ since bridges R_j, R_k overlap if and only if the interiors of their attachment intervals on e_i are not disjoint. This fact can also be used to prove that we always get exactly three such bridges. The other case is when $\mathcal{R}_1 \cup \mathcal{R}_2$ admits an embedding in $F_1 \cup F_2$. Then the chain is of type (A). In this case we must also consider the additional bridges that assure the unique embeddability of R_1 and R_r. Either they give rise to a small obstruction (together with R_1, R_2, R_{r-1}, R_r) or we get a thin millipede after eliminating possible superfluous additional bridges (cf. Claims 2 and 3 below).

Suppose now that $x \neq y$. Then our chain is of type (A). Note that in this case $\mathcal{R}_1 \cap \mathcal{R}_2 = \emptyset$. Without loss of generality we may assume that $x, y \in V(e_1)$ so that x is to the left of y and $F_1 = F_{s_1}$. The main idea of the algorithm is to traverse e_2 from left to right and at each step embed those bridges from $\mathcal{R}_1 \cup \mathcal{R}_2$ which are forced in one of the faces by previously embedded bridges.

Bridges forming a millipede will be denoted by $Q_1, Q_2, \ldots$. For $i = 1, 2, \ldots$, we will denote by l_i and r_i the leftmost and the rightmost vertex of attachment of Q_i on e_2, respectively. Let $Q_1 = R_1$. Since Q_1 has to be embedded in F_1, every bridge from $\mathcal{R}_2$ with vertex of attachment (strictly) to the left of r_1 should go in F_2. Therefore we embed these bridges in F_2. (If they cannot be simultaneously embedded, then we get a small obstruction and stop.) Denote by Q_2 the rightmost (with respect to attachments on e_2) of these bridges. If r_2 lies to the left of r_1 (or $r_2 = r_1$), we can find a small obstruction (for details see case (iii) below). Hence, every bridge from $\mathcal{R}_1$ with vertex of attachment to the left of r_1 is forced in F_1 by Q_2. We may assume that all these bridges can be simultaneously embedded in F_1. Otherwise a small obstruction can be found. Continuing this process we obtain a sequence of bridges $Q_1, Q_2, Q_3, \ldots$ such that for every i, bridge Q_i overlaps on e_2 with Q_{i+1}. There are several possibilities when we terminate this construction. Throughout the discussion of each possibility we will assume that the last embedded bridge in the above sequence is Q_s and that it is embedded in F_1. Note that in this case $Q_1, Q_3, \dots, Q_s \in \mathcal{R}_1$ and $Q_2, Q_4, \dots, Q_{s-1} \in \mathcal{R}_2$. Let $\mathcal{B}$ be the set of bridges from $\mathcal{R}_2$ that have an attachment on $[r_{s-1}, r_s)$.

(i) When trying to simultaneously embed in F_2 all bridges from $\mathcal{B}$, we encounter a pair of overlapping bridges Q, Q'. Since Q_s, Q, Q' pairwise overlap on e_2, they form a small obstruction.

(ii) If $R_r \in \mathcal{B}$, then we set $Q_{s+1} = R_r$ and stop.

(iii) Embed in F_2 all bridges from $\mathcal{B}$ and let Q_{s+1} be the rightmost among these bridges. Assume that r_{s+1} is not strictly to the right of r_s. If among the remaining bridges there is no bridge attached to e_2 entirely on the segment $[r_s, c_0]$, then $R_r \in \mathcal{R}_1$. Moreover, since $(R_{r-1}, F_2) \to (R_r, F_1)$ and since R_{r-1} is already embedded, (Q_{s+1}, F_2) also forces (R_r, F_1). Hence Q_s, Q_{s+1} and R_r (together with additional bridges guaranteeing unique embeddability of R_r) form a small obstruction. Otherwise, let R_i be the first bridge from the chain that is attached to e_2 only at $[r_s, c_0]$. By minimality of i and since $(R_{i-1}, F_{s_{i-1}}) \to (R_i, F_{s_i})$, the bridge R_{i-1} must be attached to the left and right of r_s (and also $F_{s_{i-1}} = F_1$). Then $R_{i-2} \in \mathcal{R}_2$ must be attached on e_2 entirely to the left of r_s. Since Q_{s+1} is the rightmost among bridges embedded in F_2, Q_{s+1} and R_{i-1} overlap on e_2. Therefore, Q_s, Q_{s+1}, and R_{i-1} form a small obstruction.

(iv) Now we have r_{s+1} strictly to the right of r_s. Next we check if there is a nonembedded bridge $Q \in \mathcal{R}_1$ attached to $[r_{s-1}, r_s)$. If it exists, then Q_{s-1}, Q_s, Q_{s+1}, and Q form a small obstruction. Otherwise, every bridge attached to the left of r_s has been embedded, another member Q_{s+1} of a possible millipede has been obtained, and we can proceed with the next iteration.

If in the above steps a small obstruction has not been encountered, then we have stopped in (ii) and the bridges $Q_1 = R_1, Q_2, \dots, Q_s, Q_{s+1} = R_r$ taken as $B_2^\circ, \dots, B_{m-1}^\circ$ ($m = s + 3$), respectively, satisfy (M2′), (M3), and (M4′) from the definition of skew millipedes. We will obtain B_1° and B_m° from the additional bridges (which guarantee the unique embeddability of R_1 and R_r, respectively) and either prove that the obtained sequence $B_1^\circ, B_2^\circ, \dots, B_m^\circ$ satisfies (M1)–(M4′) or obtain a small obstruction from these bridges.

Denote by Q_0 the additional bridges that guarantee the unique embeddability of Q_1. Define similarly Q_{s+2} (the corresponding bridges for Q_{s+1}). Recall that each of Q_0 and Q_{s+2} is composed of from one up to at most four bridges. In the following paragraphs we are going to show how to change Q_0 and Q_{s+2} to get a skew millipede. In each claim either we will prove the desired property or a small obstruction will be found.

Claim 0. *$\tilde{Q} = Q_0 \cup Q_1 \cup Q_2 \cup Q_s \cup Q_{s+1} \cup Q_{s+2}$ has an embedding in $F_1 \cup F_2$.* If there is no such embedding, this is a small obstruction, and we are done. Note that every embedding of $\tilde{Q}$ has Q_1 in F_1, Q_2 in F_2. Similarly, we know the faces where Q_s and Q_{s+1} are embedded.

Claim 1. *No bridge is attached to a vertex on $(x, y) \subset e_1$.* Suppose there is such a bridge B. If $\tilde{Q} \cup B$ is an obstruction, it contains at most 13 bridges, and we are done. Otherwise, B is attached only to $[x, y]$ and to $[r_1, l_{s+1}] \subseteq e_2$. Since B is not local, it has an attachment z on e_2. For some i, $2 \le i \le s$, $z \in (l_i, r_i)$. It is easy to see that $B \cup Q_{i-1} \cup Q_i \cup Q_{i+1}$ is an obstruction.

CLAIM 2. *Q_0 contains one bridge and l_1 is strictly to the left of l_2.* Consider an embedding of $Q_0 \cup Q_1$ induced by an embedding of $\tilde{Q}$. By the definition of Q_0, Q_1 cannot be re-embedded in F_2 under this embedding. Since our embedding is induced by $\tilde{Q}$, there is a bridge $B \subseteq Q_0$ which is attached on (l_1, r_1). If there are more candidates, we take the leftmost one. If B is attached out of e_2 to a vertex different from y, then $B \cup Q_1 \cup Q_2$ has unique embedding in $F_1 \cup F_2$, and we can replace Q_0 by the single bridge B and still retain the property (M1) (for $B_j^\circ = Q_{j-1}$, $j = 1, 2, 3$). It is also clear that in this case l_1 is to the left of l_2. The remaining case is when B is attached to e_1 only at y. In this case we extend the sequence $Q_1, \ldots, Q_{s+1}$ by adding B at its beginning and changing Q_0 into $Q_0 \setminus B$. Using similar arguments as above, one can prove that every embedding of the new Q_0 forces B to be embedded in F_2. Then we repeat the above reductions starting with Claim 0 (with the appropriate change of roles of x, y, F_1, F_2, etc.). Note that this extension occurs at most three times.

CLAIM 3. *Q_{s+2} contains one bridge and r_{s+1} is strictly to the right of r_s.* The proof of this claim is analogous to the proof of the previous claim.

Having all of the above properties, we define $m = s + 3$ and $B_j^\circ = Q_{j-1}$, $j = 1, \ldots, m$. Using the above claims and properties of the sequence $Q_1, \ldots, Q_{s+1}$, we see that the bridges B_j° $(1 \leq j \leq m)$ satisfy conditions (M1)–(M4$'$) from the definition of skew millipedes.

To summarize, we have proved the following result.

THEOREM 5.3. *Let $K = C \cup e_1 \cup e_2$ be a subgraph of a graph G for the 2-Möbius band embedding extension problem. Suppose that no K-bridge in G is local on one of the branches e_1, e_2. There is a linear time algorithm that either finds an embedding extension of K to G or returns an obstruction Ω for embedding extendibility. In the latter case, either Ω is small and contains at most 13 bridges or it is a millipede based on one of the branches e_1, e_2 and with apex on the other branch.*

Let us recall that large bridges in the original graph have been replaced by small bridges ($b(B) \leq 13$). Moreover, when we have a millipede, all bridges except B_1° and B_m° can be replaced by triads ($b(B) = 3$).

REFERENCES

[CR] S. A. COOK AND R. A. RECKHOW, *Time bounded random access machines*, J. Comput. System Sci., 7 (1976), pp. 354–375.

[GT] J. L. GROSS AND T. W. TUCKER, *Topological Graph Theory*, Wiley-Interscience, New York, 1987.

[HT] J. E. HOPCROFT AND R. E. TARJAN, *Efficient planarity testing*, J. ACM, 21 (1974), pp. 549–568.

[JM] M. JUVAN AND B. MOHAR, *2-restricted extensions of partial embeddings of graphs*, SIAM J. Discrete Math., submitted.

[JMM] M. JUVAN, J. MARINČEK, AND B. MOHAR, *A linear time algorithm for embedding graphs in the torus*, J. Algorithms, submitted.

[M1] B. MOHAR, *Projective plane and Möbius band obstructions*, Combinatorica, submitted.

[M2] B. MOHAR, *Obstructions for the disk and the cylinder embedding extension problems*, Comb. Probab. Comput., 3 (1994), pp. 375–406.

[M3] B. MOHAR, *Universal obstructions for embedding extension problems*, SIAM J. Discrete Math., submitted.

[M4] B. MOHAR, *A linear time algorithm for embedding graphs in an arbitrary surface*, SIAM J. Discrete Math., submitted.

SIAM J. Discrete Math.
Vol. 10, No. 1, pp. 73–82, February 1997

STIRLING NUMBERS FOR COMPLEX ARGUMENTS[*]

BRUCE RICHMOND[†] and DONATELLA MERLINI[‡]

Abstract. We define the Stirling numbers for complex values and obtain extensions of certain identities involving these numbers. We also show that the generalization is a natural one for proving unimodality and monotonicity results for these numbers. The definition is based on the Cauchy integral formula and can be used for many other combinatorial numbers.

Key words. Stirling numbers, asymptotic enumeration

AMS subject classifications. 5A05, 5A16

PII. S0895480195284329

1. Introduction.

In this note we propose a solution to the problem of Graham, Knuth, and Patashnik [3], which asks for a good generalization of the Stirling numbers of the first and second kinds ($\left[{n \atop k}\right]$ and $\left\{{n \atop k}\right\}$ in standard notation) to complex numbers n and k. We define these numbers as a contour integral which reduces to the Cauchy integral formula when n and k are integers. We show that when $n-k$ is an integer some identities involving these numbers generalize nicely to the complex case while others do not. In particular, the classical recurrences involving these numbers do generalize.

The first section gives definitions and some generalized identities. Our generalization seems suited for many numbers defined as coefficients of powers of a fixed function. A counting function with m parameters will become an analytic function of m complex variables.

In the second section we show that these generalized functions give natural proofs of the unimodality and log concavity of the original numbers for extensive ranges of n and k. The difference is that we study the derivatives of the generalized functions rather than the differences of the original discrete functions. The definition we use is implicit in the studies of the asymptotic behavior of various combinatorial numbers.

2. Definitions and easy consequences.

We begin with the classical definitions of the Stirling numbers in terms of their generating functions:

$$\left[{n \atop k}\right] = \frac{n!}{k!}[t^{n-k}]\left(\frac{1}{t}\ln\left(\frac{1}{1-t}\right)\right)^k,$$

$$\left\{{n \atop k}\right\} = \frac{n!}{k!}[t^{n-k}]\left(\frac{e^t-1}{t}\right)^k,$$

$[t^n]$ being the *coefficient of* operator. Using Cauchy's formula we have for $y, x \in \mathbf{N}$,

$$\left[{y \atop x}\right] = \frac{y!}{x!}\frac{1}{2\pi i}\oint_{|z|=r} z^{-y-1}\ln^x\left(\frac{1}{1-z}\right)dz,$$

[*] Received by the editors April 7, 1995; accepted for publication (in revised form) February 1, 1996.
http://www.siam.org/journals/sidma/10-1/28432.html

[†] Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario N2L3G1, Canada (lbrichmond@watdragon.uwaterloo.ca). The research of this author was supported by the Natural Sciences and Engineering Research Council of Canada.

[‡] Dipartimento di Sistemi e Informatica, Via Lombroso 6/17, I-50139 Firenze, Italy (donatella.merlini@dsi2.ing.unifi.it).

$$\left\{ {y \atop x} \right\} = \frac{y!}{x!} \frac{1}{2\pi i} \oint_{|z|=s} z^{-y-1} \left(e^z - 1\right)^x dz,$$

where $y! = \Gamma(y+1), x! = \Gamma(x+1), 0 < r < 1, 0 < s < \infty$, and the contours of integration are circles of radius r and s, respectively. We notice, however, that in these formulas x and y can be arbitrary complex numbers ($y \notin \mathbf{Z}^-$), and so we can use them to define Stirling numbers for complex variables. We first consider the case in which $x - y$ is an integer. In this case the integrands above are single valued so that the values of r and s are, subject to the constraints above, irrelevant.

Proposition 2.1. *If $x - y \in \mathbf{N}$, then*

$$\left[{y \atop x} \right] = \left[{y-1 \atop x-1} \right] + (y-1) \left[{y-1 \atop x} \right].$$

Proof. From the definition in the preceding paragraph we have

$$\left[{y-1 \atop x-1} \right] = \frac{(y-1)!}{(x-1)!} \frac{1}{2\pi i} \oint_{|z|=r} z^{-(y-1)-1} \ln^{x-1} \left(\frac{1}{1-z} \right) dz.$$

If we integrate by parts the expression

$$\left[{y-1 \atop x-1} \right] = \frac{(y-1)!}{(x-1)!} \frac{1}{2\pi i} \oint_{|z|=r} z^{-y} \frac{(1-z)}{x} \frac{d}{dz} \left(\ln^x \left(\frac{1}{1-z} \right) \right) dz$$

we obtain

$$\frac{(y-1)!}{(x-1)!} \left\{ \frac{z^{-y}(1-z)}{2\pi i x} \ln^x \left(\frac{1}{1-z} \right) \Big|_{r(|z|=r)}^{r} \right.$$

$$\left. + \frac{y}{2\pi i x} \oint z^{-y-1} \ln^x \left(\frac{1}{1-z} \right) dz - \frac{y-1}{2\pi i x} \oint z^{-y} \ln^x \left(\frac{1}{1-z} \right) dz \right\}$$

$$= \frac{y!}{x!} \oint z^{-y-1} \ln^x \left(\frac{1}{1-z} \right) dz - (y-1) \frac{(y-1)!}{x!} \oint z^{-y} \ln^x \left(\frac{1}{1-z} \right) dz$$

$$= \left[{y \atop x} \right] - (y-1) \left[{y-1 \atop x} \right],$$

and we have Proposition 2.1. □

Proposition 2.2. *If $x - y \in \mathbf{N}$, then*

$$\left\{ {y \atop x} \right\} = \left\{ {y-1 \atop x-1} \right\} + x \left\{ {y-1 \atop x} \right\}.$$

Proof. Since

$$\left\{ {y-1 \atop x-1} \right\} = \frac{(y-1)!}{(x-1)!} \frac{1}{2\pi i} \oint_{|z|=s} z^{-y} (e^z - 1)^{x-1} dz$$

and

$$x \left\{ {y-1 \atop x} \right\} = \frac{(y-1)!}{(x-1)!} \frac{1}{2\pi i} \oint_{|z|=s} z^{-y} (e^z - 1)^x dz$$

we have

$$\left\{ \begin{matrix} y-1 \\ x-1 \end{matrix} \right\} + x \left\{ \begin{matrix} y-1 \\ x \end{matrix} \right\} = \frac{(y-1)!}{(x-1)!} \frac{1}{2\pi i} \oint_{|z|=s} z^{-y}(e^z-1)^{x-1} e^z dz.$$

Integrating by parts gives

$$\frac{(y-1)!}{(x-1)!} \left\{ 0 - \frac{-y}{2\pi i x} \oint_{|z|=s} z^{-y-1}(e^z-1)^x dz \right\} = \left\{ \begin{matrix} y \\ x \end{matrix} \right\};$$

hence the proposition is proven. □

Remarks. The Γ function has singularities at the negative integers. The Stirling functions do not, however, because the integrals in their definitions are zero when y is a negative integer. The recursions in Propositions 2.1 and 2.2 can define the values for y as a negative integer (this can also be done using a limiting argument).

We now establish a result which suggests that with suitable restrictions many classical identities generalize to complex cases. See section 6.1 of Graham, Knuth, and Patashnik [3] and [4, 5] by Knuth for a fascinating survey of identities for Stirling numbers. Perhaps the most interesting one is the one below.

PROPOSITION 2.3. *If $x-y \in \mathbf{Z}$ then*

$$\left[\begin{matrix} -y \\ -x \end{matrix} \right] = \left\{ \begin{matrix} x \\ y \end{matrix} \right\}.$$

Proof. Set $u = 1-e^t, du = -e^t dt$ in the definition

$$\left[\begin{matrix} y \\ x \end{matrix} \right] = \frac{y!}{x!} \frac{1}{2\pi i} \oint_{C_1} u^{-y-1} \left(\ln \left(\frac{1}{1-u} \right) \right)^x du$$

to obtain

$$\frac{y!}{x!} \frac{1}{2\pi i} \oint_{C_2} \left((-1) \left(e^t - 1 \right) \right)^{-y-1} (-t)^x \left(-e^t \right) dt,$$

where C_2 is a closed path with the origin in its interior ($u = 0 \iff t = 0$). If we set $t = e^{\alpha + 2\pi i z}$, α a real number,

$$\frac{y!}{x!} \int_{-\frac{1}{2}}^{\frac{1}{2}} (-1)^{x-y} \left(e^{e^{\alpha+2\pi i z}} - 1 \right)^{-y-1} \left(e^{\alpha+2\pi i z} \right)^{x+1} e^{e^{\alpha+2\pi i z}} dz.$$

Here we observe that $(-1)^a$ is, unless a is an integer, a multivalued function. We have not derived an identity for general a so we will not do so for general x and y. With our hypothesis we have

$$\frac{y!}{x!}(-1)^{x-y} \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(e^{e^{\alpha+2\pi i z}} - 1 \right)^{-y-1} \left(e^{\alpha+2\pi i z} \right)^{x+1} \left(e^{e^{\alpha+2\pi i z}} - 1 + 1 \right) dz$$

$$= \frac{y!}{x!}(-1)^{x-y} \left\{ \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(e^{e^{\alpha+2\pi i z}} - 1 \right)^{-y} \left(e^{\alpha+2\pi i z} \right)^{x+1} dz \right.$$

$$\left. + \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(e^{e^{\alpha+2\pi i z}} - 1 \right)^{-y-1} \left(e^{\alpha+2\pi i z} \right)^{x+1} \right\}.$$

Then

$$\begin{bmatrix} -y \\ -x \end{bmatrix} = \frac{(-y)!}{(-x)!}(-1)^{y-x} \left\{ \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(e^{e^{\alpha+2\pi iz}} - 1 \right)^{y} \left(e^{\alpha+2\pi iz} \right)^{-x+1} dz \right.$$

$$\left. + \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(e^{e^{\alpha+2\pi iz}} - 1 \right)^{y-1} \left(e^{\alpha+2\pi iz} \right)^{-x+1} \right\},$$

and using the identity $(-x)! = 1/((x-1)!\sin(\pi x))$,

$$\frac{\sin(\pi x)}{\sin(\pi y)}(-1)^{y-x} \left(y \left\{ \begin{matrix} x-1 \\ y \end{matrix} \right\} + \left\{ \begin{matrix} x-1 \\ y-1 \end{matrix} \right\} \right).$$

If $x - y$ is an integer then $\sin(\pi x)/\sin(\pi y) = (-1)^{x-y}$, and by applying Proposition 2.2 we conclude our proof. □

If $x - y$ is not an integer, since the integrand is not single valued, the values of r and s become important. We choose to define r and s by saddlepoint conditions, that is, by $1/((1-r)\ln(1/(1-r))) = y/x$ and $s\exp(s)/(\exp(s)-1) = y/x$. We then choose the contour $z = s\exp(i\theta)$ or $z = r\exp(i\theta)$ and integrate from $\theta = -\pi$ to $\theta = \pi$. (When x and y are real this ensures that, as will be seen, the asymptotic behavior of the Stirling numbers can be obtained from the formulas for integer n and k by replacing n and k by y and x.) When x and y are complex we again propose defining r and s by the same equations. (They are defined as analytic functions of x and y by the implicit function theorem since the derivatives of the left-hand side with respect to r or s are not zero for r and s sufficiently close to the positive real axes (thus for x and y sufficiently close to the real axes).) We now choose the contours $z = r\exp(i\theta)$ or $z = s\exp(i\theta)$, where again θ goes from $-\pi$ to π. Note that for r and s small we can expand the integrand as a power series in z and integrate term by term. This gives a uniformly convergent series of analytic functions of x and y. Note that when x and y are positive integers we get the standard Stirling numbers. The analytic function can be analytically continued.

Remark. It seems to us that the ideas used to derive identities and recurrences when x and y differ by an integer 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 with x and y.

We found that the contour integrals in the above definitions can be evaluated readily using Maple (see [1]) if they are written as an integral over z from $-1/2$ to $1/2$, as was done in the proof of Proposition 2.3. All the propositions above were checked for several values of x and y. For example, when $y = 7.675$ and $x = 3.675$ we have $\left\{ {y \atop x} \right\} \approx 1011.174104$ and $\left[{-x \atop -y} \right] \approx 1011.174040$. Finally, note that Sprugnoli and Del Lungo [9] considered the problem of generalizing the identity

$$\sum_k \binom{n}{k} \left\{ \begin{matrix} k \\ m \end{matrix} \right\} = \left\{ \begin{matrix} n+1 \\ m+1 \end{matrix} \right\}$$

to real n and m. They showed that such a generalization would only be possible for $|m| < 1$ using asymptotic estimates. We had difficulty evaluating the relevant integrals and consequently could not test the identity. Nevertheless, our definition gives the same asymptotic behavior they obtained, so with our definition we also need $-1 < m < 1$ for the identity to be true. We do not continue to prove generalized recurrences and identities for other combinatorial counting functions here; rather, we give examples showing that they generalize results concerning unimodality and log concavity. They also simplify the analytic proofs of such results.

3. Log concavity and unimodality results. The definitions in section 2 are implicit in the asymptotic analysis of many combinatorial numbers since the Cauchy integral formula is often used. The analytical approach to prove that $a_{n,k}$ is unimodal, i.e., that

$$a_{n,1} \leq a_{n,2} \leq \cdots a_{n,k} \geq a_{n,k+1} \geq \cdots \geq a_{n,n},$$

or to prove that $a_{n,k}$ is log concave, i.e., that

$$a_{n,k+1}a_{n,k-1} \geq a_{n,k}^2,$$

involves studying the asymptotic behavior of the first difference of $a_{n,k}$ or the second difference of $\ln a_{n,k}$. When the saddlepoint method is used to do this the contour chosen depends upon n and k, so when k changes so does the contour. The change in contour usually is not important but it is necessary to prove this. If k is a real variable we can study the derivative of $a_{n,k}$, which gives us a different point of view and, as we shall see, the same contour may be used for all the derivatives.

We illustrate this with the entries in convolution matrices, where

$$a_{n,k} = \frac{n!}{k!}[t^n]h(t)^k.$$

We have seen that the Stirling numbers of both kinds are of this form. The asymptotic behavior of such $a_{n,k}$ has been studied by many authors. We shall rely heavily upon the paper by Gardy [2], which includes an excellent survey of the results so far. Gardy [2] defines

$$\Delta f(z) = z\frac{d}{dz}\ln f(z) = z\frac{f'(z)}{f(z)}, \qquad \delta f(z) = \frac{f''(z)}{f(z)} - \left(\frac{f'(z)}{f(z)}\right)^2 + \frac{f'(z)}{zf(z)}$$

and supposes that f satisfies the following properties ($f(z) = f_0 + f_1 z + f_2 z^2 + \cdots$).

Assumption 3.1. The function f has real positive coefficients with $f_0 \neq 0$ and $f_1 \neq 0$ and a strictly positive, possibly infinite, radius of convergence R.

Gardy also supposes that Ψ satisfies the following property (her assumption is more general than that below but we do not need the more general form).

Assumption 3.2. The function Ψ has positive coefficients such that $\Psi(0) = 0$ and has a strictly positive radius of convergence.

The following theorems of Gardy [2] will be very useful to us (Theorems 8, 5, and 6 of [2]).

THEOREM 3.3. *Let f satisfy Assumption* 3.1, *and let Ψ satisfy Assumption* 3.2. *Assume that the equation $\Delta f(z) = n/d$ has a real positive solution ρ smaller than the radius of convergence of f and of Ψ. Then for $n, d \to \infty$ and $\eta \leq n/d \leq M$, where η and M are positive constants,*

$$[z^n]\left\{f^d(z)\Psi(z)\right\} = \frac{f^d(\rho)\Psi(\rho)}{\rho^{n+1}\sqrt{2\pi d\delta f(\rho)}}(1 + o(1)). \qquad \square$$

Theorem 3.4. *Let f be a function satisfying Assumption 3.1 such that*

$$f(z) = e^{P(z)},$$

where $P(z) = \sum_{0 \le j \le q} p_j z^j$ is a polynomial of degree $q > 1$ with positive coefficients. Let $n, d \to \infty$ in such a way that $d = o(n)$ but $(\ln n)^{3q} n^{2q-3} = o\left(d^{2(q-1)}\right)$, and define ρ as the unique real positive solution of $\rho P'(\rho) = n/d$. Then

$$[z^n] e^{dP(z)} = \frac{e^{dP(\rho)}}{\rho^n \sqrt{2\pi n}}(1 + o(1)). \qquad \square$$

Theorem 3.5. *Let f be a meromorphic function with positive coefficients whose singularity of smallest modulus is a pole at 1 of order p: $f(z) = g(z)/(1-z)^p$, where g is a function analytic for $|z| \le 1$ and with positive coefficients. Assume that $f_1 \neq 0$ and define ρ by $\Delta f(\rho) = n/d$. Then if $d = o(n)$ and $\ln(n/\sqrt{d}) = o(d^{1/3})$ we have*

$$[z^n] f^d(z) = \sqrt{\frac{\rho d}{2\pi}} \frac{f^d(\rho)}{n\rho^n} (1 + o(1)). \qquad \square$$

We shall make some minor changes to these three theorems for our purposes. Note that these theorems are proven by the saddlepoint method and are true when n and d are real numbers with our definition of $[z^n]f^d(z)$.

Suppose $a_{y,x}$ is defined by

$$a_{y,x} = \frac{y!}{x!} \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-y(\alpha+i\theta)+x\ln h(e^{\alpha+i\theta})} d\theta = \frac{y!}{x!} \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x,y,\theta) d\theta,$$

where $\Delta h(e^\alpha) = y/x$. Then

$$\frac{da_{y,x}}{dx} = y! \frac{-(x!)'}{(x!)^2} \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x,y,\theta) d\theta$$

$$+ \frac{y!}{x!} \int_{-\pi}^{\pi} f(x,y,\theta) \left(-y \frac{d\alpha}{dx} + \ln h(e^{\alpha+i\theta}) + x \frac{h'(e^{\alpha+i\theta})}{h(e^{\alpha+i\theta})} e^{\alpha+i\theta} \frac{d\alpha}{dx} \right) d\theta.$$

If we consider the proof of Theorem 3.3 (8 of [2]), we see that one difference is the factor $\ln h(\rho e^{i\theta})$ in the integrand, where $\rho = e^\alpha$. while Gardy shows that $\delta h(\rho) \sim cy$ and

$$\ln h(\rho e^{i\theta}) = \ln h(\rho) + i\theta \Delta h(\rho) - \theta^2 \delta h(\rho) + \cdots .$$

She also shows that the integral over $\theta = -\alpha$ to $\theta = \alpha$, where $\alpha = \ln y/\sqrt{y}$, gives the asymptotic behavior of the whole integral. Note, however, that the coefficient of θ^2 in $\ln h(\rho e^{i\theta})$ is $1/x$ times that in $x \ln h(\rho e^{i\theta})$. Thus with this choice of α we have $\delta h(\rho)\alpha^2 = o(1)$ and the terms involving higher powers of θ are even smaller, while Gardy shows that all the coefficients of the powers of θ are the same size. Now

$$\int_{-\alpha}^{\alpha} e^{-x\rho \ln h(\rho)\theta^2} \theta d\theta = 0.$$

Thus

$$\frac{y!}{x!} \int_{-\pi}^{\pi} f(x,y,\theta) \ln h(e^{\alpha+i\theta}) d\theta = \ln h(\rho) \frac{y!}{x!} \int_{-\pi}^{\pi} f(x,y,\theta) d\theta \left(1 + O\left(\frac{\ln^2 y}{y} \right) \right)$$

$$= a_{y,x} \ln h(\rho) \left(1 + O\left(\frac{\ln^2 y}{y}\right)\right).$$

Since $|h(\rho e^{i\theta})| \leq |h(\rho)|$ by Assumption 3.2 and since $h(\rho e^{i\theta})^x \ln h(\rho e^{i\theta}) = 0$ for $x > 0$, if $h(\rho e^{i\theta}) = 0$ then the $\ln h(e^{\alpha + i\theta})$ term is unimportant for Gardy's analysis for $|\theta| > |\alpha|$; hence this range of θ is negligible. Similar considerations apply to

$$\frac{h'(\rho e^{i\theta})}{h(\rho e^{i\theta})} \frac{d\rho}{dx} e^{i\theta} = \frac{d\rho}{dx} \left(\frac{h'(\rho)}{h(\rho)} + i\theta\rho \ln(\rho) - \theta^2 \left. \frac{d^3 \ln h(\rho e^{i\theta})}{d^3\theta} \right|_{\theta=0} + \cdots \right) e^{i\theta}.$$

Furthermore,

$$\int_{-\pi}^{\pi} f(x,y,\theta) \frac{h'(e^{\alpha+i\theta})}{h(e^\alpha + i\theta)} e^{\alpha+i\theta} \frac{d\alpha}{dx} d\theta = \int_{-\pi}^{\pi} f(x,y,\theta) \frac{h'(e^\alpha)}{h(e^\alpha)} e^\alpha \frac{d\alpha}{dx} d\theta \left(1 + O\left(\frac{\ln^2 y}{y}\right)\right).$$

If $\rho = e^\alpha$ then $d\alpha/dx = \rho^{-1} d\rho/dx$. Thus

$$-y\frac{d\alpha}{dx} + x\frac{h'(e^\alpha)}{h(e^\alpha)} e^\alpha \frac{d\alpha}{dx} = \left(\frac{-y}{\rho} + x\frac{h'(\rho)}{h(\rho)}\right) \frac{d\rho}{dx} = 0$$

since $\Delta h(e^\alpha) = y/x$. Thus

$$\frac{da_{y,x}}{dx} = a_{y,x} \left(\frac{d}{dx}(-\ln\Gamma(x+1)) + \ln(\rho) \right) \left(1 + O\left(\frac{\ln^2 y}{y}\right)\right).$$

Moreover,

$$\frac{d^2 a_{y,x}}{d^2 x} = \frac{y!}{x!} \frac{d^2\left(-\ln\Gamma(x+1)\right)}{d^2 x} \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x,y,\theta) d\theta$$

$$+2\frac{y!}{x!} \frac{d\left(-\ln\Gamma(x+1)\right)}{dx} \frac{1}{2\pi} \frac{d}{dx} \int_{-\pi}^{\pi} f(x,y,\theta) d\theta$$

$$+\frac{y!}{x!} \frac{1}{2\pi} \frac{d^2}{d^2 x} \int_{-\pi}^{\pi} f(x,y,\theta) d\theta.$$

We again can evaluate all the derivatives with respect to x at $\theta = 0$. We also use the fact that the expression $-y/\rho + (xh'(\rho)/h(\rho))(d\rho/dx)$ and its derivative equal zero. Hence

$$\frac{d^2 a_{y,x}}{d^2 x} = a_{y,x} \left(-\frac{1}{x} - 2\ln x h(\rho) + \left(\ln^2 h(\rho) + \frac{h'(\rho)}{h(\rho)} \frac{d\rho}{dx} \right) \right) \left(1 + O\left(\frac{\ln^2 y}{y}\right)\right).$$

Note also that

$$\frac{d \ln a_{y,x}}{dx} = \frac{(a_{y,x})'}{a_{y,x}}, \qquad \frac{d^2 \ln a_{y,x}}{d^2 x} = \frac{(a_{y,x})''}{a_{y,x}} - \left(\frac{(a_{y,x})'}{a_{y,x}} \right)^2.$$

Using the fact that the asymptotic expansion for $\ln\Gamma(x)$ may be differentiated term by term, we obtain the following theorem.

Theorem 3.6. *Under the assumptions of Theorem* 3.3 *(see above)*

$$\frac{da_{y,x}}{dx} \sim a_{y,x} \left(\ln h(\rho) - \ln x\right),$$

$$\frac{d^2 \ln a_{y,x}}{d^2 x} \sim a_{y,x} \left(- \ln^2 x - \frac{1}{x} + \frac{h'(\rho)}{h(\rho)} \frac{d\rho}{dx} \right). \qquad \square$$

Let us now consider the proof of Theorem 3.4 (5 of [2]). The arguments concerning $\ln h(\rho e^{i\theta})$ and $h'(\rho e^{i\theta})/h(\rho e^{i\theta})$ in the proof of Theorem 3.6 are valid. With Gardy's choice of α we have $\alpha^2 x \delta h(\rho) \to \infty$ but $\alpha^2 \delta h(\rho) \to 0$. The analysis of Gardy is easily modified to handle the case $q = 1$; we found that $\alpha = \sqrt{\ln y}/y$ works if $q = 1$. We therefore conclude the following.

Theorem 3.7. *The conclusions of Theorem* 3.6 *hold when* $h(z) = \exp(P(z))$, *where* $P(z)$ *is a polynomial of degree* $q \geq 1$ *with positive coefficients provided* $x, y \to \infty$ *in such a way that* $x > y^{\epsilon}$, ϵ *a positive constant, if* $q = 1$ *and* $x > y^{a+\epsilon}$, $a = (2q-3)/(3(q-1))$, *if* $q \geq 2$. □

We now consider the proof of Theorem 3.5 (6 of [2]). First of all, when $p \geq 1$ the singularity may be of the form $g(z)\ln(1/(1-z))/(1-z)^p$ since $\ln(1/(1-z))$ is a slowly-varying function. Furthermore, since $d\ln(1/(1-z))/dz = (1-z)^{-1}$ the analysis of Gardy is easily modified to handle the case in which the singularity is of the form $g(z)\ln(1/(1-z))$. The derivatives of $\ln\ln(1/(1-z))$ are much like those for a singularity $1/(1-z)$; there are various powers of $\ln(1/(1-z))$ which do not matter. Also, $|\ln(1/(1-\rho e^{i\theta}))| \leq |\ln(1/(1-\rho e^{i\alpha}))|$ for $\theta \geq \alpha$ (this seems to be well known). The terms $\ln h(\rho e^{i\theta})$ and $h'(\rho e^{i\theta})/h(\rho e^{i\theta})$ may be handled as above, so we conclude the following.

Theorem 3.8. *The conclusions of Theorem* 3.6 *hold when* h *is a meromorphic function with positive coefficients whose singularity of smallest modulus is at* r *and is of the form* $g(z)\ln(1/(r-z))/(r-z)^p$, *where* $p \geq 0$, *or of the form* $g(z)/(z-r)^p$, *where* $p \geq 1$. *Here* $g(z)$ *is a function analytic for* $z \leq r$ *and with positive coefficients. Assume* $[z]h(z) \neq 0$ *and define* ρ *by* $\Delta f(\rho) = y/x$. *Then if* $x = o(y)$ *but* $x \geq y^{\epsilon}$, ϵ *a constant, the conclusion of Theorem* 3.6 *holds.* □

With Theorems 3.6, 3.7, and 3.8, the following corollary is useful.

Corollary 3.9. *If*

$$\frac{1}{\rho} + \frac{h''(\rho)}{h'(\rho)} - \frac{h'(\rho)}{h(\rho)} > 0,$$

then $d\rho/dx < 0$ *and since* $h'(\rho) > 0$ *it follows that* $a_{y,x}$ *is log concave (hence unimodal).*

Proof. From the saddlepoint condition

$$\rho \frac{h'(\rho)}{h(\rho)} = \frac{y}{x} \qquad \text{or} \qquad \ln \rho + \ln h'(\rho) - \ln h(\rho) = \ln y - \ln x.$$

Hence

$$\left(\frac{1}{\rho} + \frac{h''(\rho)}{h'(\rho)} - \frac{h'(\rho)}{h(\rho)} \right) \frac{d\rho}{dx} = \frac{-1}{x}. \qquad \square$$

4. Applications. In the applications we shall use the fact that $a_{y-x,x}$ defined with a certain $f(z)$ ($h(z)$ in our definition) is equal to $a_{y,x}$ defined with $h(z)=zf(z)$. Since the integrand is the same in both cases we shall apply Theorems 3.6, 3.7, or 3.8 with $h(z)/z$ so that Assumption 3.1 holds but use $h(z)$ in Corollary 3.9. We consider some examples of Merlini, Sprugnoli, and Verri [8]. Our log concavity results hold for every y sufficiently large, of course.

4.1. Stirling numbers of the second kind. Set $h(z)=\exp(z)-1$. Then

$$\frac{h''(\rho)}{h'(\rho)}=1, \qquad \frac{h'(\rho)}{h(\rho)}=\frac{e^\rho}{e^\rho-1}.$$

Thus

$$\frac{1}{\rho}+1-\frac{e^\rho}{e^\rho-1}=\frac{1}{\rho}-\frac{1}{e^\rho-1}=\frac{1}{\rho}-\frac{1}{\rho+\frac{\rho^2}{2}\cdots}>0.$$

Thus the Stirling numbers of the second kind are log concave for $y^\epsilon \le x \le y$ since the conditions of Theorems 3.6 and 3.7 are satisfied (Corollary 3.9 also holds). Note also that the maximum is achieved at x_0, where

$$\ln(e^\rho-1)=\ln x_0 \qquad \text{or} \qquad e^\rho=x_0+1.$$

Also, since $y/x_0=\Delta\ln(e^\rho-1)=\rho e^\rho/(e^\rho-1)$, we have

$$\frac{y}{x_0}=\rho\frac{x_0+1}{x_0}=\ln(x_0+1)\frac{x_0+1}{x_0}.$$

Thus $x_0\sim y/\ln y$, a well-known result, of course.

The same analysis obviously holds for $h(z)=\exp(P(z))$ and identifies the maximum. For the Stirling numbers it is easy to see that $\left\{ {y \atop x} \right\}=y^x/\Gamma(x+1)(1+o(y^{-\delta}))$ if $x=O(y^\epsilon), y\to\infty, x\ge 1$, so we can conclude that $\left\{ {y \atop x} \right\}$ is log concave for $1\le x\le y$.

4.2. Stirling numbers of the first kind. If $h(z)=\ln(1/(1-z))$ then

$$\frac{h''(\rho)}{h'(\rho)}-\frac{h'(\rho)}{h(\rho)}=\frac{1}{1-\rho}-\frac{1}{(1-\rho)\ln\left(\frac{1}{1-\rho}\right)}>0,$$

so $\left[{y \atop x} \right]$ is log concave for $y^\epsilon\le x\le y$ since Theorem 3.8 applies. We see, however, that the maximum would be at $\ln(1/(1-\rho))=x$, so $\rho=1-1/\rho+o(1/\rho^2)$ and hence $x\sim\ln y$. This is correct but all we have proven is that $\left[{y \atop x} \right]$ is log concave and monotone decreasing for $y^\epsilon\le x\le y$ using Theorem 3.8 and Corollary 3.9.

4.3. Tree polynomials. Let $h(z)=\ln(1/(1-T(z)))$, where $T(z)$ is the tree function defined by $T(z)=z\exp(T(z))$. We are now studying the tree polynomials of Knuth and Pittel [6]. It is not hard to see that Theorem 3.8 and Corollary 3.9 apply. Furthermore,

$$\frac{h'(\rho)}{h(\rho)}=\frac{T'(\rho)}{\ln\left(\frac{1}{1-T(\rho)}\right)(1-T(\rho))}, \qquad \frac{h''(\rho)}{h'(\rho)}=\frac{T''(\rho)}{T'(\rho)}+\frac{T'(\rho)}{1-T(\rho)},$$

so

$$\frac{h''(\rho)}{h'(\rho)}-\frac{h'(\rho)}{h'(\rho)}>0.$$

Thus the tree polynomials are log concave and monotone decreasing for $y^\epsilon < x < y$. (The results of Meir and Moon [7] apply to the case $x < y^\epsilon$ for x and y integers, so hopefully one can prove that the Knuth–Pittel tree polynomials are log concave by proving their results for real x and y.)

Remark. The range $x > y^\epsilon$ can be replaced by $x > (\ln y)^M$ in Theorems 3.7 and 3.8 if $q = 1$ (and Gardy specifies the M in the latter case). The saddlepoint method does not deal well with $x = O(\ln y)^M$. It would be useful to extend the range of Theorems 3.7 and 3.8 to this range of x so that the log concavity results would hold for all interesting real values of x and y. This would seem feasible since it amounts to extending the results for finite x (provable by standard methods) to $x = O(\ln y)^M$ and $y \to \infty$.

Acknowledgments. We are greatly indebted to Renzo Sprugnoli for suggesting this research, for providing valuable references, and for his advice and encouragement. We are also greatly indebted to Andrew Odlyzko for clearing up our difficulties resulting from the integrand being multivalued in section 2.

REFERENCES

[1] B. W. Char, K. O. Geddes, G. H. Gonnet, B. L. Leong, M. B. Monagan, and S. M. Watt, *Maple V Library Reference Manual*, Springer-Verlag, New York, Berlin, 1992.

[2] D. Gardy, *Some results on the asymptotic behaviour of coefficients of large powers of functions*, Discrete Math., 139 (1995), pp. 189–217.

[3] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics*, Addison–Wesley, Reading, MA, 1989.

[4] D. E. Knuth, *Convolution polynomials*, Mathematica, 4(2) (1992), pp. 67–78.

[5] D. E. Knuth, *Two notes on notation*, Amer. Math. Monthly, 99 (1992), pp. 403–422.

[6] D. E. Knuth and B. Pittel, *A recurrence related to trees*, Proc. Amer. Math. Soc., 105 (1989), pp. 335–349.

[7] A. Meir and J. W. Moon, *The asymptotic behaviour of coefficients of powers of certain generating functions*, European J. Combin., 11 (1990), pp. 581–587.

[8] D. Merlini, R. Sprugnoli, and M. C. Verri, *Asymptotics for Two-Dimensional Arrays: Convolution Matrices*, Technical report 27, Dipartimento di Sistemi e Informatica, Università di Firenze, Firenze, Italy, 1994.

[9] R. Sprugnoli and A. Del Lungo, *Semireal Stirling Numbers of the Second Kind*, Technical report, Dipartimento di Sistemi e Informatica, Università di Firenze, Firenze, Italy, 1994.

SIAM J. Discrete Math.
Vol. 10, No. 1, pp. 83–95, February 1997

007

SUPER ARROVIAN DOMAINS WITH STRICT PREFERENCES[*]

PETER C. FISHBURN[†] AND JERRY S. KELLY[‡]

Abstract. Given $m \geq 3$ alternatives and $n \geq 2$ voters, let $\sigma(m,n)$ be the least integer k for which there is a set of k strict preference profiles for the voters on the alternatives with the following property: Arrow's impossibility theorem holds for this profile set and for each of its strict preference profile supersets. We show that $\sigma(3,2) = 6$ and that for each m, $\sigma(m,n)/4^n$ approaches 0 monotonically as n gets large. In addition, for each n and $\epsilon > 0$, $\sigma(m,n)/(\log_2 m)^{2+\epsilon}$ approaches 0 as m gets large. Hence for many alternatives or many voters, a robust version of Arrow's theorem is induced by a very small fraction of the set of all $(m!)^n$ strict preference profiles.

Key words. Arrow's impossibility theorem, voter preference profiles, minimum profile sets

AMS subject classifications. 05A05, 90A08

PII. S0895480194263508

1. Introduction. Arrow's celebrated impossibility theorem [2], which occurs in a variety of forms [6, 7, 9], says that if a domain of voter preference profiles is sufficiently diverse and if each profile in the domain is mapped into a social order on the alternatives that satisfies a few appealing conditions, then a specific voter is a dictator in the sense that all of his or her strict preferences are preserved by the mapping. In this paper we consider the smallest domains of profiles of linear orders (strict rankings with no ties) for voters that induce an Arrovian dictator and are such that every superset domain also induces an Arrovian dictator. The latter restriction is motivated by an example in Bordes and Le Breton [5] which shows that domain enlargement can change a dictatorial conclusion into nondictatorship. Although their example uses weak orders (rankings that allow ties) rather than linear orders for voters, their point remains valid in our restricted context of linear orders for voter preferences. Further analysis of the domain-enlargement anomaly of [5] is provided in Kelly [11].

Our present concern is diametric to the search for social choice rules for *large* domains that satisfy Arrow's conditions, including no dictator [1, 3, 6, 8, 12]. Examples of such domains arise from restrictions on profiles which ensure that the pairwise simple majority relations based on the profiles will be transitive.

We consider a finite set X of $m \geq 3$ alternatives and a set of $n \geq 2$ voters, indexed by $i = 1, 2, \ldots, n$. Let $\mathbf{R}$ denote the set of all weak orders (transitive and complete binary relations) on X, and let $\mathbf{S}$ be the set of all strict rankings or linear orders (transitive, asymmetric, and weakly connected binary relations) on X. A *profile* of voter preference orders is an n-tuple $d = (S_1, S_2, \ldots, S_n)$ in $\mathbf{S}^n$. A *domain* D 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 $f(d) = \succsim_d$ on X to every $d \in D$. When $S \in \mathbf{S}$, xSy means that x is preferred to y, and $S = x_1 x_2 \cdots x_m$ means that x_j is preferred to x_k whenever $j < k$. The strict or asymmetric part of a social weak order $\succsim_d$ is denoted by $\succ_d$. That is,

$$x \succ_d y \quad \text{if} \quad x \succsim_d y \quad \text{and not} \quad (y \succsim_d x).$$

[*] Received by the editors February 23, 1994; accepted for publication (in revised form) February 5, 1996.
http://www.siam.org/journals/sidma/10-1/26350.html

[†] AT&T Bell Laboratories, 600 Mountain Ave., P.O. Box 636, Murray Hill, NJ 07974-0636 (fish@research.att.com).

[‡] Syracuse University, Syracuse, NY 13244.

When the profile d is clear from the context, we often write $\succ$ in place of $\succ_d$. The restriction to $Y \subseteq X$ of R on X is denoted by $R|_Y$.

Since we are concerned with the impossibility side of Arrow's result, we say that domain D is *Arrovian* if there does *not* exist a social choice rule f that satisfies the following three conditions proposed by Arrow:

(P) *Pareto condition.* For all x, $y \in X$ and all $d \in D$, (xS_iy in d for $i = 1, \ldots, n$) $\Rightarrow x \succ_d y$.

(IIA) *Independence of irrelevant alternatives.* For all d, $e \in D$ and all $Y \subseteq X$,

$$(d|_Y = e|_Y) \Rightarrow (\succsim_d|_Y = \succsim_e|_Y).$$

(ND) *Nondictatorship.* There is no $i \in \{1, \ldots, n\}$ such that $\succ_d = S_i$ for all $d \in D$. Given $(m, n) \geq (3, 2)$, let $\mathcal{A}$ denote the set of all Arrovian domains. It appears that $\mathcal{A}$ has a very complex structure. It is not closed under supersets since the single-profile unanimity domain $D = \{(S, S, \ldots, S)\}$ is Arrovian ((P) forces every i to be a "dictator") but many domains D' that include D are not Arrovian. And it is easily seen that $\mathcal{A}$ is not closed under unions.

Nontrivial examples of domains in $\mathcal{A}$ are $\mathbf{S}^n$ [6, p. 208] and every other domain in the set $\mathcal{T}$ of *free triple domains* [4], where $D \in \mathcal{T}$ if for every n-tuple d^* of linear orders on any 3-element set $Y \subseteq X$ there is a $d \in D$ for which $d|_Y = d^*$. Because the free triple property is inherited by supersets, $\mathcal{T}$ provides an example of Arrovian closure for supersets:

$$(D \in \mathcal{T}, D \subseteq D^+ \subseteq \mathbf{S}^n) \Rightarrow D^+ \in \mathcal{T} \cap \mathcal{A}.$$

Kelly [10] describes an Arrovian domain that is smaller than every free triple domain and for which every superset domain is also Arrovian.

The preceding observations suggest that interesting strengthenings of Arrow's theorem to *small* domains in $\mathcal{A}$ should satisfy a similar superset closure condition. Accordingly, we say that a domain D is *super Arrovian* (for strict preferences) if $\{D^+ : D \subseteq D^+ \subseteq \mathbf{S}^n\} \subseteq \mathcal{A}$. As just noted, every $D \in \mathcal{T}$ is super Arrovian. However, much smaller domains are often super Arrovian. For example, the only member of $\mathcal{T}$ for $(m, n) = (3, 2)$ is $\mathbf{S}^2$, which has 36 profiles, but there are 6-profile super Arrovian domains for $(m, n) = (3, 2)$, as will be proved shortly.

We focus on the smallest possible super Arrovian domains by defining $\sigma(m, n)$ as the least positive integer t such that some domain $D \subseteq \mathbf{S}^n$ with $|D| = t$ is super Arrovian. A super Arrovian domain is *minimum* when D is super Arrovian and $|D| = \sigma(m, n)$. We have not confirmed exact values of σ for $(m, n) \geq (3, 2)$ apart from $\sigma(3, 2) = 6$. However, bounds that reveal interesting aspects of σ's behavior are obtained.

Consider the case in which the number m of alternatives is fixed. For three alternatives we will show that $\sigma(3, n+1)/4^{n+1} \leq \frac{13}{14}[\sigma(3, n)/4^n]$, from which it follows that

$$\frac{\sigma(3, n)}{4^n} \underset{n}{\rightarrow} 0 \quad \text{monotonically.}$$

For larger fixed m we prove the generalization

$$\frac{\sigma(m, n)}{4^n} \underset{n}{\rightarrow} 0 \quad \text{monotonically.}$$

Thus, as n gets large for fixed m, the proportion of the $(m!)^n$ profiles in $\mathbf{S}^n$ needed for a super Arrovian domain becomes vanishingly small. Indeed, since a free triple domain for (m,n) has at least 6^n profiles, the number of profiles needed for a super Arrovian domain for fixed m becomes a tiny fraction of the number needed to generate a free triple domain.

A result used in the monotone convergence proofs is combined with other observations to yield a general bound on σ that summarizes much of what we know about σ.

Proposition 1. *For all $(m,n) \geq (3,2)$,*

$$\sigma(m,n) \leq 3^{n-2}a_m + (3^n - 2n - 5)/4,$$

where

$$a_m = \begin{cases} 6(2^{m-3}) & \textit{for} \quad m \leq 9, \\ (7\log_2 m)^2 & \textit{for} \quad m \geq 10. \end{cases}$$

This includes $\sigma(3,2) \leq 6$, $\sigma(3,3) \leq 22$, and $\sigma(4,2) \leq 12$. It also implies for each fixed n and $\epsilon > 0$ that

$$\frac{\sigma(m,n)}{(\log m)^{2+\epsilon}} \underset{m}{\rightarrow} 0.$$

Here, and later, logarithms are to base 2. Since $(\log m)^{2+\epsilon} \ll m!$ for large m, for any fixed n the number of profiles in a minimum super Arrovian domain is a tiny fraction of the number of possible linear orders for any one voter when m is large relative to n.

The next section proves that $\sigma(3,2) = 6$ and $\sigma(3,3) \leq 22$. Section 3 establishes the monotone convergence result for fixed $m = 3$ as $n \to \infty$, showing along the way that $2^n - 2 < \sigma(3,n) \leq 3\sigma(3,n-1) + n + 1$. Section 4 proves monotone convergence in n for larger fixed m. Section 5 gives results for variable m that lead to Proposition 1. A brief discussion in section 6 that features open problems concludes the paper.

We end this introduction with a lemma, a theorem, and a comment. The lemma and the theorem give insight into the structure of super Arrovian domains. The comment says something important about later proofs. The lemma shows that every two voters must disagree in some profile of a super Arrovian domain.

Lemma 1. *Suppose $D \subseteq \mathbf{S}^n$ is super Arrovian. Then for all distinct i, $j \in \{1,\ldots,n\}$ there is a $d = (S_1,\ldots,S_n)$ in D for which $S_i \neq S_j$.*

Proof. Suppose to the contrary that D is super Arrovian and two voters, say 1 and 2, have $S_1 = S_2$ for every profile in D. Let x, y, and z be three alternatives in X, and if $m \geq 4$ let r denote a fixed ranking of the other $m-3$. Define two new profiles:

$$e_1 = (xyzr, yxzr, xyzr, \ldots, xyzr),$$
$$e_2 = (yzxr, yxzr, yxzr, \ldots, yxzr).$$

Define a social choice rule f on $D \cup \{e_1, e_2\}$ by

$$f(d) = S_1 = S_2 \quad \text{for every} \quad d \in D,$$
$$f(e_1) = yxzr,$$
$$f(e_2) = yzxr.$$

It is easily seen that (P), (IIA), and (ND) hold: $f(e_1)$ shows that none of voters $1, 3, 4, \dots, n$ is a dictator, and $f(e_2)$ shows that voter 2 is not a dictator. But this contradicts D as super Arrovian. □

The following theorem characterizes Arrovian domains that are also super Arrovian. We say that an n-tuple d' of linear orders on $\{x, y\}$ is *nonunanimous* if it is neither $(xy, \dots, xy)$ nor $(yx, \dots, yx)$; i.e., both xy and yx appear in the n-tuple. Then $D \subseteq \mathbf{S}^n$ satisfies the *near-free doubles condition* if for every nonunanimous n-tuple d' of linear orders on any 2-element $Y \subseteq X$ there is a $d \in D$ for which $d|_Y = d'$.

Theorem 1. *$D \subseteq \mathbf{S}^n$ is super Arrovian if and only if it is Arrovian and satisfies the near-free doubles condition.*

Proof. Suppose D does not satisfy the near-free doubles condition. Without loss of generality, assume that there exists a nonunanimous n-tuple $d' = (xy$ for voters $i \leq j$, yx for voters $i > j)$ for a fixed $j \in \{1, \dots, n-1\}$ which is not the restriction to $Y = \{x, y\}$ for any $d \in D$. Let r be a fixed ranking on $X \backslash Y$ and define profile e by

$$e = (xyr \text{ for } i \leq j, \quad yxr \text{ for } i > j).$$

Note that $e|y = d'|y$. Define f on $D \cup \{e\}$ by

$$f(d) = S_1 \quad \text{for every} \quad d = (S_1, \dots, S_n) \quad \text{in} \quad D,$$
$$f(e) = yxr.$$

Suppose D is super Arrovian. Then, by Lemma 1, none of $2, \dots, n$ is a dictator; by the definitions of e and $f(e)$, individual 1 is not dictatorial in $D \cup \{e\}$. But it is easily seen that (P) and (IIA) hold for f, so we contradict the supposition that D is super Arrovian. This proves that if D is super Arrovian then it must satisfy the near-free doubles condition.

To prove the converse, suppose that D is Arrovian and satisfies the near-free doubles condition. Let f be a social choice rule for D that satisfies (P) and (IIA), and let i be a dictator for f. If e is any profile not in D, the only way to preserve (P) and (IIA) in extending f to e is $f(e) = S_i$ when S_i is i's order in e, for if e is unanimous on a pair then (P) implies that $f(e)$ agrees with the unanimous order on the pair, and if e is nonunanimous on a pair then the near-free doubles condition coupled with (IIA) and i's dictatorship implies that $f(e)$ agrees with S_i on the pair. It follows that every superset of D is also Arrovian, hence, that D is super Arrovian. □

In working with super Arrovian or potentially super Arrovian domains, it is often convenient to consider a social choice rule f that satisfies conditions (P) and (IIA), as in the preceding proof. We refer to such an f as a *P+IIA rule.* These rules always exist because dictatorial functions satisfy (P) and (IIA). Whether they are necessarily dictatorial depends on their domains. This is summarized in the following corollary, which is an easy consequence of our definitions and the proof of Theorem 1.

Corollary 1. *Let D be a domain in $\mathbf{S}^n$ and let F be the set of all* P+IIA *rules on D. Then*

(1) *D is Arrovian if and only if no $f \in F$ satisfies* (ND)*, i.e., if and only if there is a dictator for every $f \in F$;*

(2) *if D is super Arrovian, then for every $f \in F$ there is a unique dictator who is also the dictator for every* P+IIA *rule on every superset of D whose restriction to D is f.*

2. Three alternatives and few voters.

Lemma 2. $\sigma(3, 2) = 6$.

TABLE 1

	Profile	Pareto	Conclusion
$d1.$	(zyx, yxz)	$y \gg x$	$z1x$ or $y2z$
$d2.$	(yzx, xyz)	$y \gg z$	$y1x$ or $x2z$
$d3.$	(yxz, xzy)	$x \gg z$	$y1z$ or $x2y$
$d4.$	(xyz, zxy)	$x \gg y$	$x1z$ or $z2y$
$d5.$	(xzy, zyx)	$z \gg y$	$x1y$ or $z2x$
$d6.$	(zxy, yzx)	$z \gg x$	$z1y$ or $y2x$

Proof. Let

$$D_* = \{(zyx, yxz), (yzx, xyz), (yxz, xzy), (xyz, zxy), (xzy, zyx), (zxy, yzx)\}.$$

Note that for each (S_1, S_2) in D_*, S_1 and S_2 never have the same alternative ranked in the same position. Moreover, the six S_1 orders are the six linear orders on $\{x, y, z\}$, and similarly for the six S_2 orders. We show that D_* is super Arrovian (Part 1) and then prove that it is a minimum super Arrovian domain (Part 2). Both parts assume that $f : D \to \mathbf{R}$ is a P+IIA rule.

Part 1. Let $\gg$ denote unanimous preference, so $a \gg b$ if both voters prefer a to b. If $a \gg b$ in profile d, then $a \succ_d b$ by (P). Also let aib mean that $a \succ b$ whenever aS_ib. In other words, aib means that i is a dictator for the *ordered* pair (a, b). By definition, i is a dictator if aib for all six ordered pairs in $\{(x, y), (y, x), (x, z), (z, x), (y, z), (z, y)\}$.

Consider profile (zyx, yxz) in D_*. Since $y \gg x$ we have $y \succ x$ by (P), and therefore the assumption of weak order for f requires either $z \succ x$ or $y \succ z$. If $z \succ x$, then (IIA) and (P) imply $z1x$; if $y \succ z$, (IIA) and (P) imply $y2z$. A similar analysis for each profile in D_* yields Table 1.

Suppose $z1x$ from $d1$. We cannot also have $x2z$, for then $d1$ yields $z \succ x$ and $x \succ z$, a violation of asymmetry. It then follows from $d2$ that $y1x$. We cannot also have $x2y$, so $y1z$ by $d3$. Continuation gives $x1z$, $x1y$, and $z1y$, so individual 1 is a dictator. Similarly, if we begin with $y2z$ from $d1$, we conclude that individual 2 is a dictator.

Therefore, the other conditions show that one individual dictates all preferences for f, so D_* is Arrovian. It is super Arrovian by Theorem 1.

Part 2. To prove that D_* is a minimum super Arrovian domain, let D_0 be a domain of five or fewer profiles. We claim that D_0 is not super Arrovian; i.e., either it or a superset is not Arrovian. Let

$$A = \{(x, y), (y, x), (x, z), (z, x), (y, z), (z, y)\}.$$

We begin by examining three cases. These are then used to examine general situations for D_0.

Case 1. Hypothesis: D_0 does not satisfy the near-free doubles condition. Then D_0 is not super Arrovian by Theorem 1.

Case 2. Hypothesis: there is an $(a, b) \in A$ such that (ab, ba) is in exactly one profile in D_0 (every other profile has (ab, ab) or (ba, ba) or (ba, ab)), and for this profile a and b are adjacent (not separated by the third alternative) in at least one voter's order; moreover, the hypothesis of Case 1 is false. Let $(a, b) = (x, y)$ for definiteness and, without loss of generality (by symmetry considerations), assume that the special profile has (xyz, yx). To show that this situation admits a P+IIA rule f that is not dictatorial for D_0 or some superset, let $\succ_d = yxz$ for the special profile, and let f

assign voter 1's order to the other profiles in D_0. Then D_0 is non-Arrovian with the possible exception that voter 2 is a dictator. If so, $S_2 = S_1$ for all profiles other than the special profile, which must have yxz for voter 2. In this case, add to D_0 the new profile (xyz, yzx) with f assignment yxz to negate dictatorship and render $D_0 \cup \{(xyz, yzx)\}$ non-Arrovian.

Case 3. Hypothesis: there is an $(a, b) \in A$ such that (ab, ba) is in exactly two profiles in D_0, and for at least one voter, a and b are adjacent in both profiles; moreover, the hypotheses of Cases 1 and 2 are false. Let $(a, b) = (x, y)$ for definiteness. There are two subcases to consider. In the first, a voter with the x, y adjacencies in the two special profiles has the same order, e.g., zxy for the two. This subcase is explained by the analysis of Case 2. For the second subcase, we assume without loss of generality that the special profiles have (zxy, yx) and (xyz, yx). Let f assign zyx to the first of these profiles, yxz to the second, and voter 1's order to the other profiles in D_0. Then D_0 is non-Arrovian unless $S_2 = S_1$ on the other profiles and, in the special profiles, we have (zxy, zyx) and (xyz, yxz). If so, voter 2's dictatorial status is negated by adding (zxy, yzx) to D_0 with f assignment zyx.

We now use our cases to examine general situations for D_0. Given $|D_0| \leq 5$, consider voter 1's orders in the profiles of D_0. At least one of the six strict orders on $\{x, y, z\}$ is absent. Suppose without loss of generality that zyx is absent, so

$$S_1 \in \{xyz, xzy, yxz, yzx, zxy\}.$$

Suppose all five orders are present, so $|D_0| = 5$. Focus on zx in the last two. To avoid the conclusion that D_0 is not super Arrovian, Case 1 requires voter 2 to have xz in at least one of $(yzx, \cdot)$ and $(zxy, \cdot)$; Case 2 then requires voter 2 to have xz in both, but then Case 3 yields the conclusion that, in fact, D_0 is not super Arrovian.

If D_0 has more than two instances of yzx and zxy for voter 1 in its profiles, then the natural extension of Case 3 shows that it is not super Arrovian. The same thing is true if it has no (Case 1) or one (Case 2) occurrence of yzx and zxy in its profiles.

Since this covers all possible cases, we conclude that no domain with fewer than six profiles is super Arrovian. □

Remark. Up to permutations of alternatives and voters, D_* is the only minimum super Arrovian domain that has been verified for $(m, n) \geq (3, 2)$.

LEMMA 3. $\sigma(3, 3) \leq 22$.

In the following proof, we say that a pair of voters is *decisive* within a designated subset of profiles if for this subset the social preference on every ordered pair of alternatives is the same as their preference when they agree.

Proof. As before, we work with P+IIA rules. Consider a group of 6 profiles for three voters in which the first two voters have the profile orders $d1$ through $d6$ in Table 1, and $S_3 = S_2$ in each profile. By the Part 1 analysis, either voter 1 is a dictator or the combination of voters 2 and 3 is decisive. Similarly, by pairing 1 and 3, we have another 6-profile group in which either 2 is a dictator or $\{1, 3\}$ is decisive; by pairing 1 and 2 we get a third sextet of profiles in which 3 is a dictator or $\{1, 2\}$ is decisive.

Add profile (zxy, yzx, zyx) to the 18 used above which have a duplicated order. If voter 1 is dictatorial in the first group of 6 profiles, then at the new profile we have

$x \succ y$ since the restriction of the profile to $\{x, y\}$, namely (xy, yx, yx), fits the group-1 set and (IIA) applies. Similarly, $y \succ z$ if 2 is dictatorial in the second group of 6, which includes restricted profile (zy, yz, zy). Transitivity then forces $x \succ z$, contrary to $z \gg x$ and (P). So, with the new profile, we cannot have both 1 dictatorial versus $\{2, 3\}$ and 2 dictatorial versus $\{1, 3\}$.

In a similar manner, two more special profiles show that 1 dictatorial in group 1 and 3 dictatorial in group 3 are incompatible, as are 2 in group 2 and 3 in group 3.

We have 21 profiles thus far. The 22nd is the cyclic profile (zxy, yzx, xyz). This shows that we cannot simultaneously have $\{2, 3\}$ decisive for group 1, $\{1, 3\}$ decisive for group 2, and $\{1, 2\}$ decisive for group 3; otherwise $y \succ z$, $x \succ y$, and $z \succ x$, contrary to transitivity for $\succ$.

The only possible dictatorial and decisive matches left are

$$[1 \text{ from group } 1; \{1,3\} \text{ from group } 2; \{1,2\} \text{ from group } 3], \text{ or}$$
$$[\{2,3\} \text{ from group } 1; 2 \text{ from group } 2; \{1,2\} \text{ from group } 3], \text{ or}$$
$$[\{2,3\} \text{ from group } 1; \{1,3\} \text{ from group } 2; 3 \text{ from group } 3].$$

In each case, the voter who is listed for all three groups is a dictator overall, so the 22-profile domain is Arrovian. For example, if this is true for voter 1 then $a \succ b$ whenever 1 prefers a to b, regardless of the preferences of 2 and 3 on $\{a, b\}$ for all $(a, b) \in A$. It follows from Theorem 1 that the domain is super Arrovian since it satisfies the near-free doubles condition. □

3. Three alternatives and many voters. We continue toward our main result for three alternatives with two lemmas that apply to all $n \geq 2$. Let D be any nonempty set of profiles for voter set $N = \{1, 2, \ldots, n\}$.

LEMMA 4. $\sigma(3, n) > 2^n - 2$.

Proof. Assume that D is super Arrovian. Let f_i be the P+IIA rule that coincides with i's preferences throughout D. By Corollary 1, every P+IIA rule for D is one of the f_i. Whichever it might be, Theorem 1 says that every nonunanimous n-tuple of preferences for every pair from $\{x, y, z\}$ must appear in some profile in D.

There are exactly $2^n - 2$ n-tuples in $\{xyz, zyx\}^n$ that are not unanimous, and these $2^n - 2$ profiles satisfy the near-free doubles condition. This cannot be true for a smaller set of profiles. Moreover, if a domain D_0 has $2^n - 2$ members and satisfies the near-free doubles condition, then it has no profile with unanimity on some pair, and no two distinct members of D_0 have the same preference pattern on some pair.

It follows that $|D| \geq 2^n - 2$. Moreover, $|D| > 2^n - 2$, for if $|D| = 2^n - 2$, we can choose a profile p in D at which voters 1 and 2 differ. Then let f agree with voter 1 on all profiles except p, and let f agree with voter 2 at p. Then f is not dictatorial and, by the observations in the preceding paragraph, it is a P+IIA rule. Consequently, every super Arrovian D has $|D| > 2^n - 2$, so $\sigma(3, n) > 2^n - 2$. □

Our next lemma completes most of what will be needed to show that $\sigma(3, n)/4^n$ converges monotonically to 0.

LEMMA 5. $\sigma(3, n + 1) \leq 3\sigma(3, n) + n + 2$.

Proof. Lemmas 2 and 3 confirm this for $n = 2$. Assume henceforth that $n \geq 3$. Let D be a minimum super Arrovian domain for $N = \{1, 2, \ldots, n\}$, so $|D| = \sigma(3, n)$. We construct a super Arrovian domain for $N' = N \cup \{n + 1\}$ that uses no more than $3\sigma(3, n) + n + 2$ profiles.

For each profile $p = (v_1, v_2, \ldots, v_n)$ in D, form three profiles for N' as follows:

$$p_1 = (v_1, v_1, v_2, v_3, \ldots, v_n),$$
$$p_2 = (v_1, v_2, v_1, v_3, \ldots, v_n),$$
$$p_3 = (v_2, v_1, v_1, v_3, \ldots, v_n).$$

Let $D_i = \{p_i : p \in D\}$ so $|D_i| = \sigma(3, n)$ for $i = 1, 2, 3$. Because D is super Arrovian, D_1 is super Arrovian under the restriction that voters 1 and 2 always have the same preference order in an N' profile, i.e., when $\{1, 2\}$ behaves like a single voter. Similar remarks apply to $D_2(S_1 = S_3)$ and $D_3(S_2 = S_3)$.

Suppose D_1^+ is any superset of D_1 in which voters 1 and 2 always have the same order. Considering $\{1, 2\} = 12$ as a unit, it follows from Corollary 1 that every P+IIA rule on D_1^+ has a unique dictator $d_1 \in \{12, 3, 4, \ldots, n+1\}$ and that all nonunanimous preference patterns on each pair in $\{x, y, z\}$ (voter 1 agrees with voter 2) can be found in the profiles of D_1^+. Similar remarks apply to $D_2^+ \supseteq D_2$ with unique dictator $d_2 \in \{13, 2, 4, \ldots, n+1\}$ and to $D_3^+ \supseteq D_3$ with unique dictator $d_3 \in \{23, 1, 4, \ldots, n+1\}$ for P+IIA rules on D_2^+ and D_3^+, respectively.

Now for each $i \in \{4, 5, \ldots, n+1\}$ let pi be a profile with xyz for i and zyx for all other voters. There are $n - 2$ such special profiles. Let their set be D_0 and define $D_i' = D_i \cup D_0$ for $i = 1, 2, 3$. Then each D_i' is a D_i^+ as defined above. Also let $D^* = D_1' \cup D_2' \cup D_3'$, with

$$|D^*| \leq 3\sigma(3, n) + n - 2.$$

Let f^* be a P+IIA rule for D^*, with restriction f_i^* to D_i' for $i = 1, 2, 3$, and dictator d_i for D_i'. Because D_0 is common to each D_i', we must have either

(I) $d_1 = d_2 = d_3 = k$ for some $k \geq 4$, or

(II) each $d_i \subseteq \{1, 2, 3\}$.

Suppose (I) holds and, for definiteness, let $k = 4$. Suppose p is an $(n+1)$-tuple preference pattern for a *pair* of alternatives that is not unanimous. Then p will be found in some profile of D^* since at least two of voters 1, 2, and 3 have the same preference on the pair. It then follows from (P) and (IIA) for unanimous patterns that voter 4 is the unique dictator for any P+IIA rule that extends f^* to a superset of D^*. Consequently, D^* is super Arrovian when (I) holds.

Suppose (II) holds, so

$$d_1 \in \{12, 3\}, \qquad d_2 \in \{13, 2\}, \qquad d_3 \in \{23, 1\}.$$

We follow the lead of the proof of Lemma 3. Add profile

$$(zyx, yzx, zxy, zxy, \ldots, zxy)$$

to D^* if it is not already in D^* and extend f^* by P+IIA. Suppose $d_1 = 3$ and $d_2 = 2$. Then the order $\succ$ assigned by the rule to the new profile has $x \succ y$ (use $d_1 = 3$ and the fact that yx holds for voters 1 and 2, along with (IIA) and the availability of patterns for D_1') and $y \succ z$ (use $d_2 = 2$; zy holds for voters 1 and 3; (IIA)), and therefore $x \succ z$ by transitivity. However, this violates (P) since $z \gg x$. Hence the given profile prohibits the combination of $d_1 = 3$ and $d_2 = 2$.

In a similar manner, two other profiles can be added to D^* if they are not already present to prohibit $\{d_1 = 3, d_3 = 1\}$ and $\{d_2 = 2, d_3 = 1\}$. Finally, profile

$$(zxy, yzx, xyz, \ldots)$$

shows that we cannot simultaneously have $d_1 = 12$, $d_2 = 13$, and $d_3 = 23$, or else $z \succ x$, $x \succ y$, $y \succ z$ for a cycle.

Let D^{**} equal D^* in union with the four special profiles from the preceding two paragraphs, so

$$|D^{**}| \leq 3\sigma(3,n) + n - 2 + 4 = 3\sigma(3,n) + n + 2.$$

Let f^{**} be a P+IIA rule for D^{**} that extends f^* for D^* under (II). Then either

$$(d_1, d_2, d_3) = (12, 13, 1), \text{ or}$$
$$(d_1, d_2, d_3) = (12, 2, 23), \text{ or}$$
$$(d_1, d_2, d_3) = (3, 13, 23).$$

In each case, the voter who appears in all three d_i is a dictator overall and will continue to be the dictator for any P+IIA extension of f^{**} to a superset of D^{**}. For example, if $(d_1, d_2, d_3) = (12, 13, 1)$ and voter 1 prefers x to y, then $x \succ y$: all patterns that begin with xy, yx, yx are guaranteed by D_3, all that begin with xy, xy, yx are guaranteed by D_1, all that begin with xy, yx, xy are guaranteed by D_2, and all that begin with xy, xy, xy are guaranteed either by all three D_i (if nonunanimous) or by (P).

Hence D^{**} suffices to show that, regardless of whether (I) or (II) holds for f^*, every P+IIA rule f on a superset of D^{**} has a unique dictator. We conclude that D^{**} is super Arrovian for $N' = \{1, 2, \ldots, n+1\}$ and therefore that

$$\sigma(3, n+1) \leq 3\sigma(3,n) + n + 2. \qquad \square$$

We conclude this section with the following theorem.

THEOREM 2. *$\sigma(3,n)/4^n$ converges monotonically to 0 as n increases.*

Proof. Monotonic convergence to 0 follows from

$$\frac{\sigma(3,n+1)}{4^{n+1}} \leq \frac{13}{14}\left(\frac{\sigma(3,n)}{4^n}\right),$$

which is true by Lemmas 2 and 3 for $n = 2$ and by Lemmas 4 and 5 $[\sigma(3,n+1) \leq 3\sigma(3,n) + n + 2 \leq 3\sigma(3,n) + 5(2^n - 1)/7 \leq 3\sigma(3,n) + 5\sigma(3,n)/7 = 4(13/14)\sigma(3,n)]$ for $n \geq 3$. $\square$

4. More alternatives and many voters.

THEOREM 3. *For each $m \geq 4$, $\sigma(m,n)/4^n \underset{n}{\rightarrow} 0$ monotonically.*

The proof is similar to the proof of Theorem 2 with minor changes for the greater number of alternatives. We indicate these changes here.

LEMMA 6. *$\sigma(m,n) > 2^n - 2$.*

Proof. In the first paragraph of the proof of Lemma 4, replace $\{x,y,z\}$ by $X = \{x_1, x_2, \ldots, x_m\}$. In the second paragraph, replace $\{xyz, zyx\}^n$ by $\{x_1x_2\cdots x_m, x_m\cdots x_2x_1\}^n$. $\square$

LEMMA 7. *$\sigma(m,n+1) \leq 3\sigma(m,n) + n + 2$ for each $n \geq 2$.*

Proof. Let $\{x,y,z\}$ be a 3-alternative subset of X and let r be a fixed ranking of the other $m - 3$ alternatives.

Suppose $n = 2$. Modify the proof of Lemma 3 as follows. In the first paragraph, replace the set of six profiles which verify $\sigma(3,2) = 6$ by a set of $\sigma(m,2)$ profiles for a minimum super Arrovian domain for $(m,2)$. This change carries through the rest of the proof of Lemma 3.

In the second paragraph of that proof, replace the profile (zxy, yzx, zyx) by $(zxyr, yzxr, zyxr)$. Two more special profiles bring the total to $3\sigma(m,2)+3$. The final profile is $(zxyr, yzxr, xyzr)$, which replaces the cyclic profile (zxy, yzx, xyz). The rest of the Lemma 3 proof applies with 22 replaced by $3\sigma(m,2)+4$.

Suppose $n \geq 3$. Refer to the proof of Lemma 5. Take D as minimum for (m,n) and replace $\sigma(3,n)$ by $\sigma(m,n)$. In the fourth paragraph, replace "xyz for i and zyx for all..." by "$xyzr$ for i and (r inverse)zyx for all ...." In (II)'s analysis, replace the special profiles by similar profiles that conclude with r for every voter. □

Using Lemmas 6 and 7, the method in the final paragraph of the preceding section gives

$$\frac{\sigma(m,n+1)}{4^{n+1}} < \frac{15}{16}\left(\frac{\sigma(m,n)}{4^n}\right),$$

and monotone convergence to 0 for each fixed $m \geq 4$ follows.

5. Variable numbers of alternatives. We use two more lemmas in conjunction with Lemma 7 to yield Proposition 1. The lemmas focus on two voters and one voter, respectively.

LEMMA 8. $\sigma(m,2) \leq 6(2^{m-3})$ *for all* $m \geq 3$.

Proof. Equality holds at $m=3$ with D_* of Lemma 2. Let $X_m = \{x, y, z, v_4, \ldots, v_m\}$ and, with $D_3 = D_*$, define D_m for $m \geq 4$ as $D_m^{(1)} \cup D_m^{(2)}$, where $|D_m^{(1)}| = |D_m^{(2)}| = |D_{m-1}|$ with each $D_m^{(j)}$ formed from a copy of D_{m-1} as follows:

$D_m^{(1)}$ is obtained by inserting v_m as voter 1's second element and voter 2's penultimate element in each profile of D_{m-1}.

$D_m^{(2)}$ is obtained by inserting v_m as voter 2's second element and voter 1's penultimate element in each profile of D_{m-1}.

Thus $|D_m| = 2|D_{m-1}| = 6(2^{m-3})$. The 12 profiles in D_4 are as follows:

$D_4^{(1)}$	$D_4^{(2)}$
$(zv_4yx,\ \ yxv_4z)$	$(zyv_4x,\ \ yv_4xz)$
$(yv_4zx,\ \ xyv_4z)$	$(yzv_4x,\ \ xv_4yz)$
$(yv_4xz,\ \ xzv_4y)$	$(yxv_4z,\ \ xv_4zy)$
$(xv_4yz,\ \ zxv_4y)$	$(xyv_4z,\ \ zv_4xy)$
$(xv_4zy,\ \ zyv_4x)$	$(xzv_4y,\ \ zv_4yx)$
$(zv_4xy,\ \ yzv_4x)$	$(zxv_4y,\ \ yv_4zx)$

.

By inspection, D_4 satisfies the near-free doubles condition. Hence, by Theorem 1, it is super Arrovian if it is Arrovian.

To show that D_4 is Arrovian, let f be a P+IIA rule on D_4. The restriction of f to $D_4^{(1)}$ includes a P+IIA rule on D_3. By Corollary 1, one voter, whom we assume without loss of generality is voter 1, dictates social preferences on $\{x, y, z\}$ within $D_4^{(1)}$ and, by (IIA), within $D_4^{(2)}$. We show that 1 is also a dictator for every ordered pair of distinct alternatives in X_4 that involves v_4.

Consider profile (xzv_4y, zv_4yx). Then $x1z \Rightarrow x \succ z$, $z \gg v_4 \Rightarrow z \succ v_4$, and therefore $x \succ v_4$, so $x1v_4$. For profile (yv_4zx, xyv_4z) we have $v_4 \succ z$ by $v_4 \gg z$, and $z \succ x$ by $z1x$, so $v_4 \succ x$ and therefore v_41x. Four more profiles establish $y1v_4$, v_41y,

$z1v_4$ and v_41z in a similar manner. (If we had begun with 2 as the $\{x, y, z\}$ dictator, the other six profiles in D_4 would be used.) It follows that 1 dictates on all ordered pairs, hence, that D_4 is Arrovian. Therefore $\sigma(4, 2) \leq 12$.

For $m \geq 5$, it is easily seen from our recursive definition for D_m that it includes the following 12 profiles in which $r' = v_m v_{m-1} \cdots v_4$ and $r = v_4 v_5 \cdots v_m$:

$$\begin{array}{ll} (zr'yx, yxrz) & (zyrx, yr'xz) \\ (yr'zx, xyrz) & (yzrx, xr'yz) \\ (yr'xz, xzry) & (yxrz, xr'zy) \\ (xr'yz, zxry) & (xyrz, zr'xy) \\ (xr'zy, zyrx) & (xzry, zr'yx) \\ (zr'xy, yzrx) & (zxry, yr'zx) \end{array}.$$

A similar list holds for D_{m-1} when v_m is removed.

By inspection, D_m satisfies the near-free doubles condition, so it is super Arrovian if it is Arrovian. To show that it is Arrovian let f be a P+IIA rule on D_m. The restriction of f to $D_m^{(1)}$ induces a P+IIA rule on D_{m-1}. By Corollary 1, one voter, again assumed to be voter 1, dictates social preferences on X_{m-1} within $D_m^{(1)}$ and, by (IIA), within $D_m^{(2)}$. It remains to show that 1 is also a dictator for every ordered pair of distinct alternatives in X_m that includes v_m.

We treat separately the comparisons of v_m with $\{x, y, z\}$ and with $\{v_4, \ldots, v_{m-1}\}$. The first comparisons are similar to those for v_4 given earlier. For example, for $(zyrx, yr'xz)$, $(z1y, y \gg v_m) \Rightarrow (z \succ y, y \succ v_m) \Rightarrow z \succ v_m \Rightarrow z1v_m$ and for $(xr'yz, zxry)$, $(v_m \gg y, y1z) \Rightarrow (v_m \succ y, y \succ z) \Rightarrow v_m \succ z \Rightarrow v_m1z$.

For the other case, consider v_m versus v_j, $4 \leq j < m$. The presence of $(yv_{m-1} \cdots v_4zx,\ xyv_4 \cdots v_{m-1}z)$ in D_{m-1} ensures

$$(yv_{m-1} \cdots v_4zv_mx, xv_myv_4 \cdots v_{m-1}z) \in D_m.$$

This profile has $v_j \gg z$, which with $z1v_m$ yields $v_j \succ v_m$ and therefore v_j1v_m. In a similar manner, the presence of $(yxv_4 \cdots v_{m-1}z, xv_{m-1} \cdots v_4zy)$ in D_{m-1} ensures

$$(yv_mxv_4 \cdots v_{m-1}z, xv_{m-1} \cdots v_4zv_my) \in D_m.$$

Here we have v_m1x and $x \gg v_j$ to obtain $v_m \succ v_j$ and v_m1v_j. It follows that $\sigma(m, 2) \leq |D_m| = 6(2^{m-3})$. □

For our final lemma, which is Theorem 2.2.1 in Spencer [13], we define $\lambda(m)$ as the least integer k such that some k linear orders on m alternatives contain every ranking on every subset of three alternatives. It follows for n voters that there is a free triple domain in $\mathcal{T}$ that has $[\lambda(m)]^n$ profiles. As noted earlier, such a domain is super Arrovian, but when n is much larger than m there are super Arrovian domains with far fewer profiles. Hence we are mainly concerned here with m large in relation to n. Theorem 1 and remarks in Kelly [10] suggest that somewhat smaller super Arrovian domains than those with $[\lambda(m)]^n$ profiles exist for $m \gg n$, but we shall see that Spencer's theorem already gives powerful results for this case within $\mathcal{T}$.

LEMMA 9 (Spencer). *For all $m \geq 3$, $\log m < \lambda(m) < 7 \log m$.*

Exact values of $\lambda(m)$ are known only for small m. In particular, $\lambda(3) = \lambda(4) = 6$ and $\lambda(5) = 7$ as shown by

$m = 4$	$m = 5$
1234	51234
4231	54231
2143	21543
3142	31542
4132	41532
3241	32451
	14235

and two auxiliary results: (1) the displayed realization of $\lambda(4) = 6$ is unique up to permutations on the alternatives, and (2) it is impossible to insert a fifth alternative into the six orders shown for $m = 4$ so that every ranking of the fifth alternative and any two others appears in some augmented order. The proofs of (1) and (2) are elementary but involve detailed consideration of cases.

Lemma 9 and the super Arrovian property for $\mathcal{T}$ imply that $\sigma(m, 2) \leq (7 \log m)^2$. For comparison with Lemma 8 let

$$a_m = \min\{6(2^{m-3}), \quad (7 \log m)^2\}$$

so that for all $m \geq 3$, $\sigma(m, 2) \leq a_m$. Computation gives

$$a_m = \begin{cases} 6(2^{m-3}) & \text{for} \quad m \leq 9, \\ (7 \log m)^2 & \text{for} \quad m \geq 10. \end{cases}$$

By Lemma 7, $\sigma(m, 3) \leq 3a_m + 4, \sigma(m, 4) \leq 3(3a_m + 4) + 5$, and in general

$$\begin{aligned} \sigma(m, n) &\leq 3^{n-2} a_m + \sum_{i=3}^{n} 3^{n-i}(i + 1) \\ &= 3^{n-2} a_m + (3^n - 2n - 5)/4, \end{aligned}$$

which is the conclusion of Proposition 1.

Theorem 4. *For each* $n \geq 2$ *and* $\epsilon > 0$, $\sigma(m, n)/(\log m)^{2+\epsilon} \underset{m}{\to} 0$.

Proof. For $m \geq 10$ and $\epsilon > 0$, Proposition 1 gives

$$\frac{\sigma(m, n)}{(\log m)^{2+\epsilon}} \leq \frac{49(3^{n-2})}{(\log m)^{\epsilon}} + \frac{(3^n - 2n - 5)/4}{(\log m)^{2+\epsilon}}.$$

The right side vanishes for fixed n and ϵ as m gets large. □

For each $m \geq 3$ and $\epsilon > 0$, Proposition 1 also says that $\sigma(m, n)/(3 + \epsilon)^n \underset{n}{\to} 0$, but we cannot assert that the convergence is monotone when ϵ is small.

6. Discussion. Super Arrovian domains were introduced as a meaningful way of extending Arrow's classic impossibility theorem to a greater variety of domains. We have focused on the cardinality $\sigma(m, n)$ of the smallest super Arrovian domains for m alternatives and n voters. As either m or n gets large, the proportion of the $(m!)^n$ preference profiles in $\mathbf{S}^n$ needed to construct a super Arrovian domain goes quickly to 0. Some bounds on $\sigma(m, n)$ were derived for the general case.

The only specific σ value verified in the paper is $\sigma(3, 2) = 6$. We noted also that $\sigma(4, 2) \leq 12$ and $\sigma(3, 3) \leq 22$. Other best current bounds for $m = 3$ are

$$\sigma(3,4) \leq 66, \quad \sigma(3,5) \leq 133, \quad \text{and} \quad \sigma(3,6) \leq 362.$$

Proofs for these do not appear in the paper.

Feasible open problems include determination of the exact values of $\sigma(4,2)$ and $\sigma(3,3)$, along with a better lower bound on $\sigma(m,n)$ than that of Lemmas 4 and 6. We are also interested in tighter bounds that will yield a good asymptotic approximation to $\sigma(m,n)$.

Other open questions concern the structure of minimum and minimal super Arrovian domains, where a super Arrovian domain D is *minimal* if no proper subset of D is super Arrovian. One question is whether there exist minimal super Arrovian domains that are not also minimum, i.e., for which $|D| > \sigma(m,n)$. Another is whether minimum super Arrovian domains are unique up to permutations of alternatives and voters.

REFERENCES

[1] J. M. Abello and C. R. Johnson, *How large are transitive simple majority domains*?, SIAM J. Alg. Disc. Meth., 5 (1984), pp. 603–618.

[2] K. J. Arrow, *Social Choice and Individual Value*, 2nd ed., Wiley, New York, 1963.

[3] D. Black, *The Theory of Committees and Elections*, Cambridge University Press, London, England, 1958.

[4] J. H. Blau, *The existence of social welfare functions*, Econometrica, 25 (1957), pp. 302–313.

[5] G. A. Bordes and M. Le Breton, *Arrovian theorems for economic domains: The case where there are simultaneously private and public goods*, Soc. Choice Welf., 7 (1990), pp. 1–17.

[6] P. C. Fishburn, *The Theory of Social Choice*, Princeton University Press, Princeton, NJ, 1973.

[7] P. C. Fishburn, *Interprofile Conditions and Impossibility*, Harwood Academic, Chur, Switzerland, 1987.

[8] J. S. Kelly, *Voting anomalies, the number of voters and the number of alternatives*, Econometrica, 42 (1974), pp. 239–251.

[9] J. S. Kelly, *Arrow Impossibility Theorems*, Academic Press, New York, 1978.

[10] J. S. Kelly, *The free triple assumption*, Soc. Choice Welf., 11 (1994), pp. 97–101.

[11] J. S. Kelly, *The Bordes–LeBreton exceptional case*, Soc. Choice Welf., 11 (1994), pp. 273–281.

[12] A. K. Sen and P. K. Pattanaik, *Necessary and sufficient conditions for rational choice under majority decision*, J. Econom. Theory, 1 (1969), pp. 178–202.

[13] J. Spencer, *Probabilistic Methods in Combinatorics*, Ph.D. thesis, Harvard University, Cambridge, MA, 1970.

SIAM J. Discrete Math.
Vol. 10, No. 1, pp. 96–108, February 1997

A VARIANT OF THE BUCHBERGER ALGORITHM FOR INTEGER PROGRAMMING*

REGINA URBANIAK†, ROBERT WEISMANTEL†, AND GÜNTER M. ZIEGLER‡

Abstract. In this paper we modify Buchberger's S-pair reduction algorithm for computing a Gröbner basis of a toric ideal so as to apply it to an integer program (IP) in inequality form with fixed right-hand sides and fixed upper bounds on the variables. We formulate the algorithm in the original space and interpret the reduction steps geometrically. In fact, three variants of this algorithm are presented, and we give elementary proofs for their correctness. A relationship among these (exact) algorithms, iterative improvement heuristics, and the Kernighan–Lin procedure is established. Computational results are also presented.

Key words. integer programming, upper bounds, test sets, Buchberger algorithm, Gröbner bases, iterative improvement heuristics

AMS subject classification. 90C

PII. S0895480195281209

1. Introduction.

In this paper we consider an integer programming (IP) problem of the type

$$\max \{c^T x : \quad Ax \le b, \quad 0 \le x \le u, \quad x \text{ integral}\},$$

where $A \in \mathbb{N}^{m \times n}$ is a given matrix, $b \in \mathbb{N}^m$ is the right-hand-side vector, $u \in \mathbb{N}^n$ denotes a vector of upper bounds for the variables, and $c \in \mathbb{Z}^n$ is the objective function. (Here and in the following $\mathbb{N}$ denotes the natural numbers including 0.) We assume that $c_i > 0$ for all i: otherwise we can set $x_i = 0$ for the corresponding variable and eliminate a column from our program. If upper bounds are not explicitly given, they may be generated by setting $u_i := \min\{b_j/a_{ij} : a_{ij} > 0\}$. The algorithms and proofs that we present can easily be adapted to the solution of families of IPs with varying right-hand sides b, as long as finite upper bounds for the variables are given.

Integer programs can, in principle, be solved by applying the Buchberger algorithm [4] for computing the Gröbner basis of a toric ideal. The connection between test sets for IP and Gröbner bases of certain ideals was first established by Conti and Traverso [5]. For details on this approach we refer to Thomas [15], Thomas and Weismantel [16], Pottier [13], and Hoşten and Sturmfels [10]. Whereas the algorithms of [5] and [15] deal with families of IP problems of the form $Ax = b$, $x \ge 0$ for varying right-hand-side vectors b, here we show how to handle the case of a fixed right-hand side and fixed upper bounds on the variables. This is essential, since most IPs arising "in practice" have upper bounds, often $u_i = 1$.

Moreover, the procedures formulated in [5] and [15] are applied to an "extended" IP with additional variables of the form

$$(EIP(b)) \quad \min \{c^T x + M\mathbf{1}^T y : \quad Ax + Ey = b, \; x \in \mathbb{N}^n, \; y \in \mathbb{N}^m, \; x, y \text{ integral}\},$$

* Received by the editors January 3, 1995; accepted for publication (in revised form) February 5, 1996. This work was partially supported by Science Program SC1-CT91-620 of the EEC.
http://www.siam.org/journals/sidma/10-1/28120.html

† Konrad–Zuse–Zentrum für Informationstechnik Berlin (ZIB), Heilbronner Strasse 10, 10711 Berlin, Germany (urbaniak@zib-berlin.de, weismantel@zib-berlin.de).

‡ Department of Mathematics 6–1, Technical University of Berlin, 10623 Berlin, Germany (ziegler@math.tu-berlin.de). The work of this author was supported by a Gerhard-Hess-Forschungsförderpreis of the German Science Foundation (DFG).

where $M \in \mathbb{N}$ is a "large" integer, E is the $m \times m$ identity matrix, and $\mathbf{1}$ denotes the vector of all ones. In practice the additional variables may lead to a considerable increase in the space and time requirements of the algorithms considered. The original proofs for correctness and finiteness of those algorithms needed an algebraic machinery (which only applies in the case $c \geq 0$); however, the geometric version by Thomas [15] only needed the Gordan–Dickson lemma.

We formulate our algorithm in the original space and interpret all steps geometrically. In fact, three variants of a (simple) algorithm are presented, and we give elementary geometric proofs for their correctness. A relationship among these (exact) algorithms, iterative improvement heuristics, and the Kernighan–Lin procedure is established as well. Finally, preliminary computational results show that this type of algorithm has potential for becoming a useful tool in the solution of practical IP problems.

As mentioned above, our variant of the Buchberger algorithm deals with an IP problem of the type

$$\max\,\{c^T x: \quad Ax \leq b, \quad 0 \leq x \leq u, \quad x \text{ integral}\}. \tag{1}$$

In order to handle more general programs

$$\max\,\{c^T x: \quad Ax \leq b, \quad Cx = d, \quad 0 \leq x \leq u, \quad x \text{ integral}\}, \tag{2}$$

we apply the following simple transformation. We define $c' := c + M\mathbf{1}^T C$ (where $\mathbf{1}$ is the vector with all components equal to 1 and M is a sufficiently large integer) and solve the IP problem

$$\max\,\{c'^T x: \quad Ax \leq b, \quad Cx \leq d, \quad 0 \leq x \leq u, \quad x \text{ integral}\}. \tag{3}$$

Then every optimal solution x^0 of (3) will satisfy $Cx^0 = d$, provided that the program (2) is feasible. If (2) is infeasible, then the objective function value of an optimal solution to (3) is less than $M\mathbf{1}^T d$. Therefore, in terms of optimal solutions both formulations (2) and (3) are in a one-to-one correspondence, and we can always assume that the IP problem is given in the form (1).

Throughout the paper we use the following notation. N denotes the set $\{1, \ldots, n\}$. We say that $x \leq y$ holds for vectors $x, y \in \mathbb{Z}^n$ if $x_i \leq y_i$ for all $i \in N$. Thus "$\leq$" is a partial order on $\mathbb{Z}^n$.

From the objective function c we obtain a linear order on $\mathbb{Z}^n$ as follows: we choose an arbitrary term order $\prec_0$ (for example, lexicographic) and use it as a "tie breaker" on the points that have the same objective function value under c; that is, we define

$$x \prec_c y \qquad :\Longleftrightarrow \qquad \begin{cases} c^T x < c^T y & \text{or} \\ c^T x = c^T y & \text{and } x \prec_0 y. \end{cases}$$

In the following "$\prec$" always denotes a linear order $\prec_c$ that refines the (fixed) objective function c in this way. One might note that $\prec$ is a term order in the sense of Gröbner basis theory if and only if $c \geq 0$. (In case of doubt we write "$\prec_c$" for "$\prec$".)

For a vector $d \in \mathbb{Z}^n$ we define $d^{\succ} := d$ if $d \succ 0$; in the case where $d \prec 0$, we set $d^{\succ} := -d$. For $v \in \mathbb{Z}^d$ we denote by v^+ the vector with $v_i^+ = v_i$ if $v_i \geq 0$ and $v_i^+ = 0$ otherwise. Accordingly, v^- is the vector with $v_i^- = -v_i$ if $v_i \leq 0$ and $v_i^- = 0$ otherwise. Clearly $v = v^+ - v^-$.

Definition 1.1. *Given a matrix $A \in \mathbb{N}^{m \times n}$, objective function vector $c \in \mathbb{Z}^n$, and a right-hand-side vector $b \in \mathbb{N}^m$, we denote by $IP_{A,b,c,u}$ the optimization problem*

$$\max\{c^T x: \quad Ax \leq b, \quad 0 \leq x \leq u, \quad x \textit{ integral}\}. \tag{$IP_{A,b,c,u}$}$$

We say that x is feasible *for $IP_{A,b,c,u}$ if $Ax \leq b$, $0 \leq x \leq u$, and x is integral.*

A subset $B \subseteq \mathbb{Z}^n$ is a test set *for $IP_{A,b,c,u}$ if and only if $v \succ 0$, the vectors v^+, v^- are feasible for all $v \in B$, and for every nonoptimal point $x \in \mathbb{N}^n$ there is some $v \in B$ such that $x + v$ is feasible.*

The paper is organized as follows. In section 2 we present the new variants of the Buchberger algorithm to compute test sets for IPs. A link of these variants to iterative improvement heuristics and to the Kernighan–Lin heuristic is established in section 3. We also show computational results when our algorithms are applied to small- and medium-sized real world problems in section 4.

2. Three variants of the Buchberger algorithm. In this section we present three variants of an algorithm that compute a test set for the IP problem

$$(IP_{A,b,c,u}) \qquad \max\{c^T x : Ax \leq b,\ x_i \in \{0, 1, \ldots, u_i\},\ i \in N\},$$

where $A \in \mathbb{N}^{m \times n}$, $c \in \mathbb{N}^n$, $b \in \mathbb{N}^m$ is a *fixed* right-hand-side vector, and u is the vector of upper bounds on the variables. In the following $\prec$ always denotes a term order refining c.

We start with an outline of the basic form of the algorithm. Having proved that this version of the algorithm terminates after finitely many steps with a test set for $IP_{A,b,c,u}$, we show how to speed up the computations by excluding certain vectors in the computation of the test set.

Roughly speaking, a test set B can be computed as follows. Start with the n unit vectors $B := \{e_i : i \in N\}$. Iteratively, compute the difference vectors between all pairs of vectors in B and direct each such difference vector such that it is greater than 0 with respect to the order. All such difference vectors that are not in B and that are differences of feasible vectors for $IP_{A,b,c,u}$ are added to B. The algorithm terminates if no more vectors are added to B.

Algorithm 2.1.

```
(1) Set $B_{old} := \emptyset$, $B := \{e_i : i \in N\}$.
(2) While $B_{old} \neq B$ perform the following steps:
    (2.1) Set $B_{old} := B$.
    (2.2) For all pairs of vectors $v, v' \in B$ with $v \prec v'$, $(v'-v)^+, (v'-v)^-$ feasible,
          set $B := B \cup \{v' - v\}$.
```

Whenever step 2 of this algorithm is executed (except for the last time), a new vector is added to the set B. Since the number of different vectors $w = v' - v$ satisfying $-u \leq w \leq u$ is bounded by $\prod_{i \in N}(2u_i + 1)$, the above algorithm terminates after finitely many steps.

We now show that the set B generated by Algorithm 2.1 is a test set for $IP_{A,b,c,u}$. Suppose that x is a feasible point ($Ax \leq b$, $0 \leq x \leq u$) that is not optimal and that cannot be improved by adding an element in B. Let x' be some feasible vector with $x' \succ x$. Then $x' - x$ can be written as an integral combination of unit vectors: we can decrease from x to reach 0 (staying feasible), then increase to reach x', using only vectors in B. Hence, there is a sequence $P = (x^0, \ldots, x^p)$ of vectors x^i with the following properties:

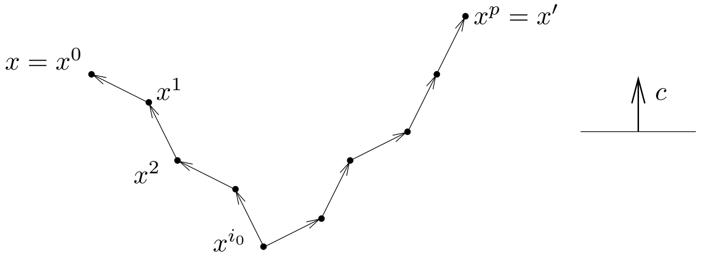
(i) $x^0 = x$, $x^p = x'$,

(ii) for all $i = 1, \ldots, p$, $(x^i - x^{i-1})^{\succ} \in B$, and

(iii) all the points $x^i \in P$ are feasible.

In fact, in the specific current situation, we know more for (ii): there is some i_0 such that $-(x^i - x^{i-1}) \in B$ for $i \leq i_0$, and $x^i - x^{i-1} \in B$ for $i > i_0$.

In the following sketches, the vertical direction “upward” represents increasing objective function. Thus the small vectors, depicting elements of B, are directed upward.



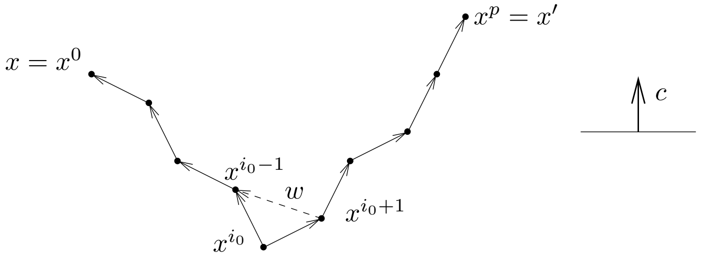
Among all sequences that satisfy (i), (ii), and (iii), let $P = (x^0, \ldots, x^p)$ be a sequence such that the minimum point in P, x^{i_0}, is maximal with respect to the order $\prec$. (Such a sequence exists since the number of feasible points is finite. The minimum point x_{i_0} is unique since $\succ$ is a total order.)

We have $i_0 \neq 0$ (otherwise $x = x^0$ could be improved by $x^1 - x^0 \in B$) and $i_0 \neq p$ (otherwise we would have $x' = x^p \prec x^0 = x$).

Both vectors x^{i_0+1} and x^{i_0-1} are feasible. It follows that

$$w \quad := \quad \Big((x^{i_0+1} - x^{i_0}) \; - \; (x^{i_0-1} - x^{i_0}) \Big)^{\succ} \quad = \quad (x^{i_0+1} - x^{i_0-1})^{\succ}$$

is a difference of two feasible vectors. Moreover, since $x^{i_0-1} - x^{i_0} \in B$ and $x^{i_0+1} - x^{i_0} \in B$, the difference vector w has been computed in step 2.2 of Algorithm 2.1 and was added to B.



Thus

$$P' \quad := \quad (x^0, \ldots, x^{i_0-1}, x^{i_0+1}, \ldots, x^p)$$

again satisfies properties (i)–(iii); yet the minimum element in P' is larger than x^{i_0}, which is a contradiction.

With this we have proved the following theorem.

Theorem 2.2. *Algorithm* 2.1 *terminates after a finite number of steps. The output is a test set for the IP problem* $IP_{A,b,c,u}$.

Example 2.3. Consider the 0/1 knapsack problem

$$\max\{x_1 + 3x_2 + 2x_3 : \; x_1 + 2x_2 + 3x_3 \leq 3, \; x_i \in \{0, 1\}, \; i = 1, 2, 3\}.$$

The algorithm starts with the vectors e_1, e_2, e_3. Then the vectors $e_2 - e_1$, $e_3 - e_1$, and $e_3 - e_2$ are added to the set B. In the next iteration the vector $e_1 + e_2 - e_3$ is added. The algorithm terminates with the set

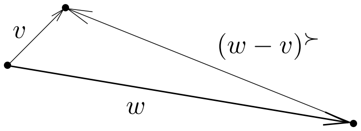
$$B \quad = \quad \{e_1, e_2, e_3, e_2 - e_1, e_3 - e_1, e_2 - e_3, e_1 + e_2 - e_3\},$$

since in step 2 no new vectors are found.

This set is, indeed, a test set for the above 0/1 knapsack problem. Yet not all of these vectors are really needed to guarantee that one can go from any feasible point of this program to the optimal solution without decreasing the objective function value in each step. Namely, the vector $e_1 + e_2 - e_3$ can always be replaced by the two vectors e_1 and $e_2 - e_3$, both being elements in B. In this situation we call $e_1 + e_2 - e_3$ *reducible.* Elimination of reducible vectors from the computation of a test set is a very important issue in order to keep the size of the test set small. We now formalize this situation.

DEFINITION 2.4. *For an IP problem $IP_{A,b,c,u}$, let B be a family of improvement vectors* (*that is, $v \succ 0, v^+, v^-$ are feasible for all $v \in B$*).

A vector $w \neq 0$ can be reduced *by $v \in B$ if $v^+ \leq w^+$, $v^- \leq w^-$, and $(Av)^+ \leq (Aw)^+$. In this situation, we say that we obtain $(w - v)^{\succ}$ by* reducing *w.*



In this situation, a trivial computation shows that if at any feasible point $x \in \mathbb{Z}^n$ the vector w can be applied (that is, if $x + w$ is feasible as well), then one could also apply v instead of w and obtain a feasible point $x + v \succ x$ (and after that one can apply $w - v$ to reach $x + w$). Thus we note

- if x and $x + w$ are feasible, then so is $x + v$;
- $|v|_1 \leq |w|_1$, with equality only if $v = w$, and $|w - v|_1 < |w|_1$; and
- $x + v \succ x$.

We have *not* assumed that $w \succ 0$, because in the following proofs of this section we need to reduce difference vectors of the form $w - w'$.

ALGORITHM 2.5 (reduction). This algorithm computes the *reduction* $\overline{w}^B$ of a vector $w \in \mathbb{Z}^n$ by a set B of improvement vectors.

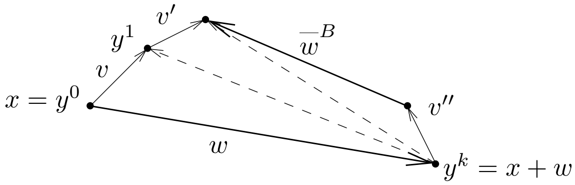
(1) Input $B \subseteq (\mathbb{Z}^n)^{\succ 0}$, $w \in \mathbb{Z}^n$.
(2) As long as possible, find $v \in B$ such that $r \in \{w, -w\}$ can be reduced by v, and replace r by $r - v$.
(3) Output $\overline{w}^B := r^{\succ}$.

The vector $\overline{w}^B$ is called the *reduced vector* of w with respect to B.

PROPOSITION 2.6. *Assume that x and $x + w$ are feasible and compute $\overline{w}^B$. Then there is a sequence of distinct integral points $x = y^0, y^1, \ldots, y^{k-1}, y^k = x + w$, with the following properties:*

- *each y^j is feasible*;
- *$y^0 \prec y^1 \prec \cdots \prec y^{j_0} \succ \cdots \succ y^k = x + w$ for some $0 \leq j_0 \leq k$*;
- *in particular, for $0 < j < k$ we get $y^j \succ x$ or $y^j \succ x + w$ or both*;
- *$(y^j - y^{j-1})^{\succ} \in B$ for each $1 \leq j \leq k$, except that if $\overline{w}^B \neq 0$, then we either have $y^{j_0} - y^{j_0 - 1} = \overline{w}^B$ or $y^{j_0} - y^{j_0+1} = \overline{w}^B$*; *and*
- *$|y^j - y^{j-1}|_1 \leq |y^k - y^0|_1 = |w|_1$, with equality only if $k = 1$.*

The following sketch shows how this sequence of points y^i may be generated by reducing by v, v', and v'' (in that order).



Using reduction, we can modify our initial algorithm as follows.

Algorithm 2.7.

```
(1) Set $B_{old} := \emptyset$, $B := \{e_i : i \in N\}$.
(2) While $B_{old} \neq B$ repeat the following:
    (2.1) Set $B_{old} := B$.
    (2.2) For all pairs of vectors $v, v' \in B_{old}$ such that $v \prec v'$ perform the following steps:
        (2.2.1) If $(v' - v)^+$ and $(v' - v)^-$ are feasible, set $w := v' - v$.
        (2.2.2) Compute $r := \overline{w}^B$ by Algorithm 2.5.
        (2.2.3) Set $B := B \cup \{r\}$.
```

Again, Algorithm 2.7 terminates after finitely many steps, since there exists an upper bound on the number of different vectors that can be added to the set B.

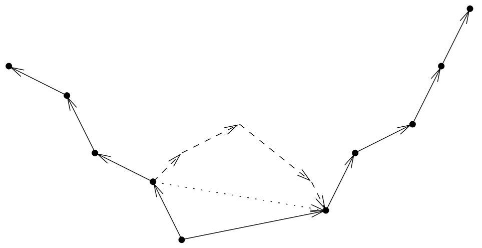
To show that it computes a test set, we can nearly proceed as before. For any nonoptimal feasible point x that cannot be improved by a vector in B, and every feasible $x' \succ x$, there is a sequence $P = (x^0, \ldots, x^p)$ that satisfies properties (i)–(iii) above for which the minimum point x^{i_0} occurring in P is maximal (with respect to $\prec$). We again have $0 < i_0 < p$, which means that the difference vector

$$w \quad = \quad (x^{i_0+1} - x^{i_0-1})^{\succ}$$

was considered in Algorithm 2.7. Thus, by Proposition 2.6 we get a new sequence of feasible points

$$P' \quad := \quad (x^0, \ldots, x^{i_0-1} = y^0, y^1, \ldots, y^k = x^{i_0+1}, \ldots, x^p),$$

where the minimum point of P' is larger than that of P.



Thus we have proved the following theorem.

Theorem 2.8. *Algorithm 2.7 terminates after a finite number of steps. The output is a test set for the IP problem $IP_{A,b,c,u}$.*

As a postprocessing step, the size of the final output B of Algorithm 2.7 can be further reduced by the following theorem.

Theorem 2.9 (postprocessing). *Let B be any test set for $IP_{A,b,c,u}$. Then successively for each $w \in B$, one can perform the following operations:*

- *if w is reducible by some $v \in B\backslash w$, replace B by $B\backslash w$ and*
- *if $-w$ is reducible by some $v \in B\backslash w$, replace B by $(B\backslash w) \cup \{\overline{w}^{B\backslash w}\}$ otherwise.*

After these operations, B is still a test set.

Next we deal with the following question: what are sufficient conditions for a vector b to be successively reducible to 0? This question is of computational relevance, because vectors that can be reduced to 0 (by a sequence of reduction steps) need not be added to the set B during the run of Algorithm 2.7 but can be excluded in advance.

The first criterion in this respect can be adapted from Buchberger [4] (see also [6]). Suppose that $v, w \in B$ are two vectors in the current set B of Algorithm 2.7 with $v \succ w$. If the following three conditions are satisfied

- the vectors v^+ and w^+ have disjoint support,
- the vectors v^- and w^- have disjoint support,
- the vectors $(Av)^+$ and $(Aw)^+$ have disjoint support,

then the vector $d = v - w$ (to be computed in step 2.2 of Algorithm 2.7) is reducible by v and can be reduced to 0 (see step 2.2.2). This follows because under the above assumptions we obtain

$$d^+ = v^+ + w^-, \quad d^- = v^- + w^+, \quad \text{and} \quad (Ad)^+ = (Av)^+ + (Aw)^-.$$

Hence, every component in v^- is less than or equal to the corresponding component of d^-, every component in v^+ is less than or equal to the corresponding component of d^+, and every component in $(Av)^+$ is less than or equal to the corresponding component of $(Ad)^+$. Therefore, d is reducible by v and since $r := v - d = w$ can be reduced to 0 by w, the statement follows.

Though such criteria help in reducing the running time of the overall procedure, the main bottleneck is that iteratively for every pair of vectors in the current set B the associated difference vector needs to be computed. Excluding parts of these computations in advance is one of the main issues for applying this algorithm to the solution of IP instances of nontrivial size.

We have found one such criterion. Namely, we show that for every element in the current set B it is sufficient to compute just n difference vectors instead of $|B| - 1$. Then, however, a difference vector v of two elements in the current set B can be reduced only by a vector $v' \in B$ that satisfies $v'^+ \leq v^+$, $v'^- \leq v^-$, $(Av')^+ \leq (Av)^+$, and $v' \leq v$.

Algorithm 2.10.

```
(1) Set $B_{old} := \emptyset$, $E := \{e_i : i \in N\}$, $B := E$.
(2) While $B_{old} \neq B$ repeat the following:
    (2.1) Set $B_{old} := B$.
    (2.2) For every $v := (w - e_i)^{\succ}$ with
        (2.2.1) $w \in B$, $e_i \in E$ with $w \neq e_i$,
        (2.2.2) $v^+, v^-$ feasible, and
        (2.2.3) $v$ is not reducible by any $v' \in B$ with $v - v' \geq 0$
            set $B := B \cup \{v\}$.
```

Theorem 2.11. *Algorithm 2.10 terminates after a finite number of steps. The output is a test set for $IP_{A,b,c,u}$.*

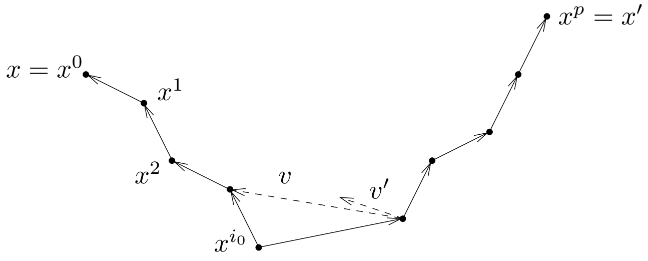
Proof. Finiteness of the algorithm is clear.

Suppose that there exists a nonoptimal feasible point x that cannot be improved by any element of the set B that is computed by Algorithm 2.10. Let x' be the feasible vector with $x' \succ x$. As for Algorithm 2.1, there exists a sequence $P = (x^0, x^1, \ldots, x^{i_0}, \ldots, x^p)$ with $0 < i_0 < p$ and

(i) $x^0 = x$, $x^p = x'$;

(ii) $x^{i-1} - x^i \in E$ for $i \leq i_0$, and $x^{i+1} - x^i \in E$ for $i \geq i_0$ (except that one of $x^{i_0-1} - x^{i_0}$ and $x^{i_0+1} - x^{i_0}$ is permitted to be in $B \backslash E$);

(iii) every point x^i is feasible.



Now choose a sequence of points P with these properties such that its minimum point x^{i_0} is maximal.

In the course of Algorithm 2.10, one has computed the difference vector $v := (x^{i_0+1} - x^{i_0-1})^{\succ}$. This vector clearly satisfies steps 2.2.1 and 2.2.2. If it fails 2.2.3 then it can be written in the form $v = v' + \sum_{k=1}^{s} e_{i_k}$ for $v' \in B$ and $e_{i_1}, \ldots, e_{i_s} \in E$. Furthermore, we have that $x^{i_0 \pm 1} + v = x^{i_0 \mp 1}$ (where the sign "±" is "+" if $x^{i_0+1} \prec x^{i_0-1}$ and "−" otherwise), and thus all the points in the sequence

$$x^{i_0 \pm 1}, x^{i_0 \pm 1} + v', x^{i_0 \pm 1} + v' + e_{i_1}, \ldots, x^{i_0 \pm 1} + v' + \sum_{k=1}^{s} e_{i_k} = x^{i_0 \mp 1}$$

are feasible. From this we obtain a new sequence P' that again satisfies the above properties, yet the minimal point in this sequence is larger than x^{i_0}: a contradiction. □

It is shown in [16] that the minimal reduced test set B for $IP_{A,b,c,u}$ is unique. This test set is computed by Algorithm 2.7. Algorithms 2.1 and 2.10 produce test sets for $IP_{A,b,c,u}$ that are in general supersets of B. After applying the postprocessing Theorem 2.9, these test sets coincide with B. However, these algorithms proceed in different orders. While Algorithms 2.1 and 2.7 might produce exchange vectors of ℓ_1-norm 2^k in their kth iteration of step 2, Algorithm 2.10 will generate improvement vectors according to increasing ℓ_1-norm: it produces *all* improvement vectors of ℓ_1-norm k in the kth iteration of step 2.

3. A relation to iterative improvement heuristics. To simplify the discussions we will now only consider 0/1 problems, which have $u_i = 1$ for all $i \in N$. (Generalizations of what follows to arbitrary upper bounds are straightforward.) We furthermore assume that $Ae_i \leq b$ for all $i \in N$ and that for $i, j \in N$ the vectors Ae_i and Ae_j do not have disjoint support. These assumptions are usually satisfied by instances coming from traveling salesman problems, graph partitioning problems, or knapsack problems, etc.

One approach for obtaining good solutions for the problem

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} c_i x_i, \\ & Ax \leq b, \\ & x_i \in \{0,1\} \quad \text{for } i = 1, \ldots, n \end{aligned}$$

is to start with some feasible solution, i.e., a set $S \subseteq N$ such that $A\chi^S \leq b$. (For any subset $S \subseteq N$, χ^S denotes the incidence vector, with $\chi_i^S = 1$ if $i \in S$ and $\chi_i^S = 0$ otherwise.) Iteratively we replace items which belong to S by items which are not in the current solution via a certain rule such that the incidence vector of the resulting set, S' say, is feasible. Exchange the role of S with S' and repeat these steps until a certain stopping criterion is satisfied.

This procedure is certainly too general to be analyzed and needs specification of (a) the rule according to which items are replaced by others and (b) the stopping criterion.

In most implementations, exchange operations are allowed only if the number of items involved is less than or equal to a certain threshold value, λ say. More precisely, the cardinality of the symmetric difference between the sets S and S' must not exceed λ. The reason for that is simply to keep the running time of the procedure in acceptable limits. Indeed, usually a value of $\lambda = 2$ or $\lambda = 3$ is chosen. (The resulting algorithms are the "2-OPT" and "3-OPT" heuristics.) In addition, *iterative improvement heuristics* only allow exchanging items of S with items not in S if the objective function value $c\chi^S$ increases by this. Those algorithms terminate if the current solution x cannot be improved by replacing items with $x_i = 1$ against items with $x_i = 0$ such that the number of items involved is less than or equal to λ.

In the case where $\lambda = 2$ or $\lambda = 3$ there is a nice relationship between iterative improvement heuristics and our Algorithm 2.7. Similar statements can be made for Algorithms 2.1 and 2.10.

Proposition 3.1. *Let $v \in \{0, -1, 1\}^n$ be a vector such that $\sum_{i=1}^n |v_i| \leq 3$, $-b \leq Av \leq b$, and $v \succ 0$. After performing step 2 in Algorithm 2.7 twice, the set B either contains v or v is the sum of vectors in B.*

Proof. We start initially with the n unit vectors. When step 2 is performed the first time all the vectors $e_i - e_j \succ 0$, $i, j \in \{1, \ldots, n\}$ are computed and added to B. Note that under the assumptions introduced at the beginning of this section those vectors cannot be reduced. Thus, after a first processing of step 2 all vectors $y \succ 0$ with entries $0, +1, -1$, and $\sum_{i=1}^n |y_i| \leq 2$, $-b \leq Ay \leq b$ have been generated.

Now let $v \in \{0, 1\}^n$ be a 0/1 vector such that $\sum_{i=1}^n |v_i| = 3$, $-b \leq Av \leq b$, and $v \succ 0$. v can be written in one of the following forms: (i) $v = e_i - (e_u - e_w)$ with $(e_u - e_w) \succ 0$, (ii) $x = (e_u - e_w) - e_i$ with $(e_u - e_w) \succ 0$, (iii) $x = (e_u - e_w) + e_i$ with $(e_u - e_w) \succ 0$, or (iv) $v = e_u + e_w + e_i$ where $i, u, w \in \{1, \ldots, n\}$, $i \neq u \neq w \neq i$. Suppose that (iii) or (iv) holds. Then v is a sum of elements in B. Otherwise, (i) or (ii) is true. Then v is the difference vector of elements in B. This difference vector was computed by processing step 2 of Algorithm 2.7 a second time. Since (iii) and (iv) are not true, this difference vector is not reducible via the elements in the current set B. □

As a corollary we obtain that via Algorithm 2.7 certain iterative improvement heuristics can be simulated. In fact, this algorithm is a strong generalization of the idea of iterative improvement heuristics and it is obvious that by restricting the number of times step 2 is to be processed, the output can be used to (iteratively) improve feasible solutions.

Instead of admitting exchanges that always improve the current objective function value, Kernighan and Lin [11, 12] used a slightly different strategy. Again suppose that a set $S \subseteq N$ is given such that $A\chi^S \leq b$. Iteratively we either exchange one item which belongs to S by one item which does not so that the new solution is feasible again or we add to the current set S a new item if this yields a feasible solution. In other words, in order to move from a feasible solution x to a feasible solution x' we either have that $x + (e_i - e_j) = x'$ or $x + e_i = x'$ for some $i \in N$ (and $j \in N \setminus \{i\}$). Let $B^2 := \{e_i : \ i \in N\} \cup \bigcup_{e_i - e_j \succ 0} \{e_i - e_j\}$ and $B^2_- = \bigcup_{b \in B^2} \{-b\}$. Then, at a current feasible point x, Kernighan and Lin choose a vector v in $B^2 \cup B^2_-$ with $c^T v = \max\{c^T b : b \in B^2 \cup B^2_-, x + b \text{ is feasible}\}$. The procedure terminates if a given number of iterations have been performed.

Following this approach it is clear that at some point x an exchange operation may be performed that (locally) yields a decrease in the objective function. However, by a sequence of exchanges, some of which might have a negative objective function value and some of which have a positive objective function value, we might reach some feasible point $x' \succ x$. Suppose this is the case and in order to make our analysis easy let us also assume that x' can be reached from x by first applying an exchange step $v \in B^2_-$ and then an exchange step $w \in B^2$. We have already seen that the set B^2 is generated by performing step 2 in Algorithm 2.7 once. Therefore, $-v \in B^2$ and $w \in B^2$ and as $x' \succ x$, so is $v + w \succ 0$. Since $v + w = w - (-v)$, with $(-v), w \in B^2$, the vector $x' - x$ is computed by performing step 2 in Algorithm 2.7 a second time. Either $x' - x$ is not reducible or it is. In the first case, it is added to our set B generated by Algorithm 2.7. In the latter case we can reach a point $\tilde{x}$ by using elements in B such that in each step the objective function is not decreased.

Computing a difference vector w between pairs of elements in a current improvement set B and directing it such that $w \succ 0$ can be viewed as a two step procedure, first locally getting worse, but afterward globally improving the objective function value.

4. Computational results. In this section we present preliminary computational results with Algorithms 2.7 and 2.10. We have applied both algorithms to small- and medium-sized instances coming from set covering problems (Steiner triple systems; see [14]), knapsack and multidimensional knapsack problems (see [8]), set partitioning problems [9], and experimental design problems (see [1]). For all examples, except for those arising in experimental design problems, the number of columns is in the range of 6 to 105 and the number of rows is between 1 and 331. The set partitioning instances reported in [9] involve up to several thousand columns and rows. From these original data we took subsets of the rows and columns and solved the set partitioning problem associated with this subset. For the instances of experimental design problems the number of columns varies from 147 to 2,205 and the number of rows is between 28 and 121.

We always start the computations with the vector 0, which is feasible for all instances (via the transformation of section 1 we replace conditions $Ax = b$ by $Ax \leq b$). Iteratively we improve the current (feasible) solution by elements in the set B (computed according to Algorithms 2.7 and 2.10) until either we prove optimality or we exceed a time limit. We performed all tests on a SUN Sparc10 workstation with a limit of 30 minutes CPU time.

Tables 1 and 2 summarize our results. In order to distinguish the instances we use the following convention: "knap" means that the instance is a (multidimensional) knapsack problem. The prefix "cov" and "part" stands for instances of set covering and set partitioning problems, respectively. The "des" is used for experimental design problem instances. The first number following the prefix corresponds to the number of columns. The second number is the number of rows. For example, knap.20.1 is an instance of a knapsack problem consisting of 20 items and 1 row, etc.

Column 2 of the tables gives the optimal value of the corresponding problem. The

Table 1

EXAMPLE	OPT	SOL (2.7)	TIME (2.7)	SOL (2.10)	TIME (2.10)
cov.9.13	5	5	0:00	5	0:00
cov.15.36	9	9	0:00	9	0:00
cov.27.118	18	18	0:00	18	0:00
cov.45.331	30	30	23:16	29	0:00
knap.6.10	3800	3800	0:00	3800	0:00
knap.10.10	87061	87061	0:00	87061	0:00
knap.15.10	4015	4015	0:00	4015	0:00
knap.20.10	6120	6120	0:00	6120	0:00
knap.28.10	12400	12400	0:00	12400	0:00
knap.39.5	10618	10618	15:34	10605	5:59
knap.49.5	15223	15205	0:20	15223	26:22
knap.20.1	7708	7708	0:00	7708	0:00
knap.50.1	19928	19928	0:02	19928	0:02
knap.100.1	41773	41773	2:30	41773	2:38
knap.30.5	4561	4561	0:00	4561	0:01
knap.40.5	5557	5557	0:06	5557	0:05
knap.50.5	6159	6159	0:03	6159	0:03
knap.60.5	6954	6954	0:19	6954	0:16
knap.60.5	7486	7486	1:37	7486	1:28
knap.60.5	7289	7289	0:27	7289	0:28
knap.60.5	8633	8633	1:09	8633	1:09
knap.70.5	7698	7698	1:00	7698	0:54
knap.80.5	8947	8947	0:10	8947	0:10
knap.80.5	8344	8344	31:28	8341	1:08
knap.90.5	9492	9492	0:40	9492	0:45
knap.28.4	3418	3418	1:17	3418	2:49
knap.35.4	3186	3186	0:12	3186	0:12
knap.27.4	3090	3090	9:40	3090	2:53
knap.34.4	3186	3186	0:03	3186	0:03

optimal values for the set partitioning problems were obtained by the cutting plane code of [3]. For the knapsack problems this value was obtained with the cutting plane code reported in [7]. For the experimental design problem we refer to [1] for the optimal values. The optimal values for the set covering instances are taken from MIPLIB [2]. Columns 3 and 4 report on the objective function value of the best solution found via Algorithm 2.7 and the corresponding time that was needed. Accordingly, columns 5 and 6 show the appropriate values if Algorithm 2.10 is applied.

The results show that in all examples except for four instances the solution computed by Algorithm 2.10 is the same as the one given by Algorithm 2.7. In fact, both procedures behave quite similarly concerning both running time and quality of the solution. It seems that neither of the two variants is significantly superior over the other. Algorithm 2.10 can prove optimality for the nine instances cov.9.13, knap.6.10, knap.10.10, knap.15.10, knap.20.1, part.10.4, part.24.11, part.30.9, and part.39.3 within five minutes of CPU time, whereas Algorithm 2.7 terminates with a provably optimal solution in only seven cases. For the remaining examples both procedures did not succeed in proving optimality. Nevertheless for 56 out of 59 examples Algorithm 2.7 or 2.10 found an optimal solution.

A special behavior can be observed when the two algorithms are applied to the experimental design data: either the optimal solution is found immediately or we do not find any feasible solution.

Summarizing our experiments, we think that on very hard combinatorial prob-

Table 2

EXAMPLE	OPT	SOL (2.7)	TIME (2.7)	SOL (2.10)	TIME (2.10)
knap.29.2	95168	95168	0:00	95168	0:00
knap.20.10	2139	2139	0:08	2139	6:19
knap.40.30	776	776	0:03	776	0:03
knap.37.30	1035	1035	0:21	1035	0:21
knap.28.2	130883	130883	0:06	130883	0:06
knap.105.2	624319	624319	0:15	624319	0:15
knap.60.30	7772	7772	0:08	7772	0:08
part.10.4	–4248	–4248	0:00	–4248	0:00
part.24.11	–7983	–7983	0:00	–7983	0:00
part.30.9	–1816	–1816	0:00	–1816	0:00
part.39.3	–2874	–2874	0:00	–2874	0:00
part.40.16	–8061	–8061	0:00	–8061	0:00
part.42.23	–35818	–35818	0:05	–35818	0:05
part.43.18	–11493	–11493	0:00	–11493	0:00
part.48.14	–7634	–7634	0:01	–7634	0:01
part.47.20	–6792	–6792	0:01	–6792	0:01
part.49.15	–22959	–22959	0:01	–22959	0:01
part.49.15	–5782	–5782	0:00	–5782	0:00
part.59.8	–2698	–2698	0:01	–2698	0:01
part.67.12	–4942	–4942	0:00	–4942	0:00
part.74.16	–11268	–13820	0:37	–13820	0:38
part.77.22	–16812	–16812	0:03	–16812	0:03
part.86.22	–9933	–9933	4:54	–9933	4:49
part.92.10	–2800	–2800	0:00	–2800	0:00
des.147.28	21	21	0:00	21	0:00
des.294.35	42	42	0:00	42	0:01
des.432.48	36	36	0:00	36	0:01
des.675.60	90	—	—	—	—
des.1014.91	78	78	0:03	78	0:02
des.2205.126	210	—	—	—	—

lems such as experimental design problems we are still far from having an "effective" optimization algorithm. It is not good enough to run Algorithm 2.10 as a "black box" that will hopefully find a good solution. Here, a *combinatorial* understanding of the test set B seems to be indispensable. For the knapsack, multidimensional knapsack, set partitioning, and set covering problems that we tested, the situation is different. Algorithms 2.7 and 2.10 work quite well on those instances and usually produce very good solutions. Moreover, the performance is stable. Certainly the running times are still too high and our implementation cannot in reasonable time handle instances with a couple of hundred columns. Storing and computing all the difference vectors would exceed the memory requirements.

To conclude, the algorithms that we presented here are very general; they are not adapted to special purpose problems and we implemented the routines straightforwardly. The quality of the solutions that were produced on many of the test samples is quite high and very stable. We think that there is still a lot of research to be done in order to understand test sets combinatorially, but the results certainly indicate that the construction, analysis, and adaptation of such methods are worth further efforts.

Conclusions. Whereas dual methods like cutting plane algorithms have proven to be extremely successful in the solution of (large-scale) IP problems, there is a lack of primal algorithms that have the potential to prove optimality of an IP. In particular, it would be desirable to have both primal and dual algorithms that make it possible to systematically and simultaneously improve current primal and dual solutions. The three Buchberger algorithms that we presented in this paper might have the potential to satisfy those needs and requirements.

Yet we are still far from applying our algorithms to large-scale problems. From our point of view the computational results in section 5 show that the "Buchberger-type" algorithms generate very good solutions starting from scratch. Certainly the running time and the memory requirements form a bottleneck. The number of exchange vectors that are generated might even be squared when proceeding from one single iteration to the next. Hence, further research must concentrate on the combinatorial understanding of the exchange vectors that need to be contained in the test set. Then one could work with "classes of exchange vectors" implicitly rather than have to generate all such vectors explicitly. This would be analogous to the treatment of "classes of facets" in a cutting plane approach.

If one can make progress in this direction, then primal algorithms based on ideas as presented in this paper might become a powerful tool in the solution of IP problems.

REFERENCES

[1] Th. Beth, D. Jungnickel, and H. Lenz, *Design Theory,* B.I.-Wissenschaftsverlag, Bibliographisches Institut, Zürich, 1985; Cambridge University Press, London, 1993.

[2] R. E. Bixby, E. A. Boyd, and R. R. Indovina, *MIPLIB: A test set of mixed integer programming problems,* SIAM News, 25 (1992), p. 16; available from Rice University SOFTLIB at `softlib.cs.rice.edu` and from ZIB eLib electronic library at `elib.zib-berlin.de`.

[3] R. Borndörfer and R. Weismantel, *Solving Set Partitioning Problems via Cutting Planes,* ZIB-Berlin, 1995, preprint.

[4] B. Buchberger, *Gröbner bases: An algorithmic method in polynomial ideal theory,* in Multidimensional Systems Theory, N. K. Bose, ed., D. Reidel, 1985, pp. 184–232.

[5] P. Conti and C. Traverso, *Buchberger algorithm and integer programming,* Lecture Notes in Comput. Sci., 539 (1991), pp. 130–139.

[6] D. A. Cox, J. B. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra,* Undergraduate Texts in Mathematics, Springer-Verlag, New York, 1992.

[7] C. E. Ferreira, A. Martin, and R. Weismantel, *Solving multiple knapsack problems by cutting planes,* SIAM J. Optim., 6 (1996), pp. 858–877.

[8] H. Heitkotter, *Personal communication,* Universität Dortmund, 1993.

[9] K. L. Hoffman and M. Padberg, *Solving airline crew-scheduling problems by branch-and-cut,* Management Science, 39 (1993), pp. 657–682.

[10] S. Hoşten and B. Sturmfels, *GRIN: An implementation of Gröbner bases for integer programming,* Lecture Notes in Comput. Sci., 920 (1995), pp. 267–276.

[11] B. W. Kernighan and S. Lin, *An efficient heuristic procedure for partitioning graphs,* Bell Systems Technical Journal, 49 (1970), pp. 291–307.

[12] B. W. Kernighan and S. Lin, *An effective heuristic algorithm for the traveling salesman problem,* Oper. Res., 21 (1973), pp. 498–516.

[13] L. Pottier, *Gröbner Bases of Toric Ideals: Properties, Algorithms, and Applications,* Rapport de Recherche de l'INRIA RR 2224, Sophia Antipolis, 1994.

[14] A. Sassano, *On the facial structure of the set covering polytope,* Math. Programming, 44 (1989), pp. 181–202.

[15] R. R. Thomas, *A geometric Buchberger algorithm for integer programming,* Math. Oper. Res., 20 (1995), pp. 864–884.

[16] R. R. Thomas and R. Weismantel, *Truncated Gröbner bases for integer programming,* AAECC, to appear.

SIAM J. DISCRETE MATH.
Vol. 10, No. 1, pp. 109–127, February 1997

CLIQUE r-DOMINATION AND CLIQUE r-PACKING PROBLEMS ON DUALLY CHORDAL GRAPHS[*]

ANDREAS BRANDSTÄDT[†], VICTOR D. CHEPOI[‡], AND FEODOR F. DRAGAN[‡]

Abstract. Let $\mathcal{C}$ be a family of cliques of a graph $G = (V, E)$. Suppose that each clique C of $\mathcal{C}$ is associated with an integer $r(C)$, where $r(C) \geq 0$. A vertex v *r-dominates* a clique C of G if $d(v, x) \leq r(C)$ for all $x \in C$, where $d(v, x)$ is the standard graph distance. A subset $D \subseteq V$ is a *clique r-dominating set* of G if for every clique $C \in \mathcal{C}$ there is a vertex $u \in D$ which r-dominates C. A *clique r-packing set* is a subset $P \subseteq \mathcal{C}$ such that there are no two distinct cliques $C', C'' \in P$ r-dominated by a common vertex of G. The *clique r-domination problem* is to find a clique r-dominating set with minimum size and the *clique r-packing problem* is to find a clique r-packing set with maximum size. The formulated problems include many domination and clique-transversal-related problems as special cases. In this paper an efficient algorithm is proposed for solving these problems on dually chordal graphs which are a natural generalization of strongly chordal graphs. The efficient algorithm is mainly based on the tree structure and special vertex elimination orderings of dually chordal graphs. In some important particular cases where the algorithm works in linear time the obtained results generalize and improve known results on strongly chordal graphs.

Key words. graphs, hypergraphs, tree structure, hypertrees, covering and packing problems, transversal and matching problems, dually chordal graphs, clique hypergraphs, generalization of strongly chordal graphs

AMS subject classifications. 05C65, 05C70, 68Q25, 68R10

PII. S0895480194267853

1. Introduction.

Strongly chordal graphs introduced in [13, 9] and [19] are a well-known subclass of chordal graphs [10] for which several problems remaining NP complete for chordal graphs are efficiently solvable. Among them are the problems of r-domination [7], clique transversal [8], and k-neighborhood covering [17] which are solved in linear time if a *strong elimination ordering* of a strongly chordal graph is given together with the input graph. The best algorithms for finding strong elimination orderings are not linear time.

A very natural generalization of strong elimination orderings is given by *maximum neighborhood orderings* [2]. These orderings lead to *dually chordal graphs* [5, 12, 24]—a generalization of strongly chordal graphs. For a dually chordal graph a maximum neighborhood ordering can be computed in linear time.

For dually chordal graphs in [4] a linear time solution of the r-domination problem is given using only a maximum neighborhood ordering.

In this paper we present a unified method to solve different types of clique r-domination and clique r-packing problems on dually chordal graphs. In some particular cases the obtained results generalize and improve results of [7, 8] and [17] on strongly chordal graphs.

2. Problem formulations.

Let $G = (V, E)$ be a finite connected simple (i.e., without loops and multiple edges) and undirected graph. A *clique* is a subset of pairwise adjacent vertices of V. A *maximal clique* is a clique that is not a proper

[*] Received by the editors May 4, 1994; accepted for publication (in revised form) February 20, 1996.
http://www.siam.org/journals/sidma/10-1/26785.html

[†] Universität Rostock FB Informatik, D 18051 Rostock, Germany (ab@informatik.uni-rostock.de).

[‡] Department of Mathematics and Cybernetics, Moldova State University, A. Mateevici Str. 60, Chişinău 277009 Moldova (chepoi@cinf.usm.md, dragan@cinf.usm.md). The research of these authors was supported by the VW–Stiftung Project No. I/69041.

subset of any other clique. The *distance* $d(u, v)$ between vertices $u, v \in V$ is the length (i.e., number of edges) of a shortest path connecting u and v. The *disk* centered at vertex v with radius k is the set of all vertices having distance at most k to v:

$$N^k[v] = \{u \in V : d(v, u) \leq k\}.$$

For a clique C and vertex $v \in V$ we denote by

$$\delta(v, C) = \max\{d(v, u) : u \in C\}$$

the *deviation* of v from C. Evidently, $\delta(v, C)$ is the smallest integer $k \geq 0$ such that $C \subseteq N^k[v]$.

Let $\mathcal{C}$ be a family of cliques of a graph G. Suppose that each clique C of $\mathcal{C}$ is associated with an integer $r(C)$, where $r(C) \geq 0$ and $r(C) > 0$ for cliques with size $|C| > 1$. A vertex v *r-dominates* a clique C of G if $\delta(v, C) \leq r(C)$, i.e., $C \subseteq N^{r(C)}[v]$. (For cliques of size $|C| > 1$ and $r(C) = 0$ there is no vertex v which r-dominates C.) A subset $D \subseteq V$ is a *clique r-dominating set* of G if for every clique $C \in \mathcal{C}$ there is a vertex $u \in D$ which r-dominates C. A *clique r-packing set* is a subset $P \subseteq \mathcal{C}$ such that there are no two distinct cliques $C', C'' \in P$ r-dominated by a common vertex of G. The *clique r-domination problem* is to find a clique r-dominating set with minimum size $\gamma_{\mathcal{C},r}(G)$, and the *clique r-packing problem* is to find a clique r-packing set with maximum size $\pi_{\mathcal{C},r}(G)$. Then $\gamma_{\mathcal{C},r}(G)$ and $\pi_{\mathcal{C},r}(G)$ are called the *clique r-domination* and *clique r-packing numbers* of G.

The formulated problems include many domination and clique-transversal-related problems as special cases. First, if $\mathcal{C}$ is the family of maximal cliques of G and $r(C) = 1$ for all $C \in \mathcal{C}$ then we obtain the *clique-transversal* and the *clique-independence problems* [8]. If $\mathcal{C}$ is a family of edges of G and $r(C) = k$ (where k is an integer) for all $C \in \mathcal{C}$ then we obtain the *k-neighborhood covering* and *k-neighborhood independence problems* considered in [17]. Finally, if $\mathcal{C}$ consists only of single vertices of G then we obtain the *r-domination* and *r-packing problems* [22, 19, 7, 12, 11, 4] whose particular instances are the *domination, k-domination, packing*, and *k-packing problems* [14, 9].

3. Dually chordal graphs. In this section we recall the definitions and some results and algorithms on dually chordal graphs, which we use in what follows.

For a vertex $v \in V$ of a graph $G = (V, E)$ we denote by $N(v)$ the *open neighborhood* $N(v) = \{v : uv \in E\}$ and by $N[v] = N(v) \cup \{v\}$ the *closed neighborhood.* For $Y \subseteq V$ let $G(Y)$ be the *subgraph induced by* Y. For a graph G with the vertex set $V = \{v_1, \ldots, v_n\}$ let $G_i = G(\{v_i, v_{i+1}, \ldots, v_n\})$ and $N_i[v]$ ($N_i(v)$) be the *closed (open) neighborhood* of v in G_i.

A vertex v is *simplicial* if and only if (iff) $N[v]$ is a clique. The ordering $(v_1, \ldots, v_n)$ of V is a *perfect elimination ordering* iff for all $i \in \{1, \ldots, n\}$ the vertex v_i is simplicial in G_i. The graph G is *chordal* iff G has a perfect elimination ordering [15].

The ordering $(v_1, \ldots, v_n)$ is a *strong elimination ordering* iff for all $i \in \{1, \ldots, n\}$ $N_i[v_j] \subseteq N_i[v_k]$ when $v_j, v_k \in N_i[v_i]$ and $j < k$. The graph G is *strongly chordal* iff G has a strong elimination ordering [13].

A vertex $u \in N[v]$ is a *maximum neighbor* of v iff for all $w \in N[v]$ the inclusion $N[w] \subseteq N[u]$ holds (note that $u = v$ is not excluded). The ordering $(v_1, \ldots, v_n)$ is a *maximum neighborhood ordering* if for all $i \in \{1, \ldots, n\}$ there is a maximum neighbor $u_i \in N_i[v_i]$:

$$\text{for all } w \in N_i[v_i], N_i[w] \subseteq N_i[u_i] \text{ holds.}$$

The graph G is *dually chordal* [5] iff G has a maximum neighborhood ordering. The graph G is *doubly chordal* [20] iff G is chordal and dually chordal.

There is a close connection between chordal and dually chordal graphs which can be expressed in terms of hypergraphs (for hypergraph notions we follow [3]). Let $\mathcal{N}(G) = \{N[v] : v \in V\}$ be the (*closed*) *neighborhood hypergraph* of G, and let $\mathcal{C}(G) = \{C : C \text{ is a maximal clique of } G\}$ be the *clique hypergraph* of G. By $\mathcal{D}(G) = \{N^k[v] : v \in V,\ k \text{ a nonnegative integer}\}$ we denote the *disk hypergraph* of G.

Now let $\mathcal{E}$ be a hypergraph with underlying vertex set V, i.e., $\mathcal{E}$ is a set of subsets of V. The *dual hypergraph* $\mathcal{E}^*$ has $\mathcal{E}$ as its vertex set and $\{e \in \mathcal{E} : v \in e\}$ $(v \in V)$ as its edges. The *underlying graph* (or *two-section graph*) $\Gamma(\mathcal{E})$ of the hypergraph $\mathcal{E}$ has vertex set V and two distinct vertices are adjacent iff they are contained in a common edge of $\mathcal{E}$. The *line graph* $L(\mathcal{E}) = (\mathcal{E}, E)$ of $\mathcal{E}$ is the intersection graph of $\mathcal{E}$, i.e., $ee' \in E$ iff $e \cap e' \neq \emptyset$. A *partial hypergraph* of hypergraph $\mathcal{E}$ has V as the underlying vertex set and some edges of $\mathcal{E}$.

A hypergraph $\mathcal{E}$ is a *hypertree* (called *arboreal hypergraph* in [3]) iff there is a tree T with vertex set V of $\mathcal{E}$ such that every edge $e \in \mathcal{E}$ induces a subtree in T. Equivalently, $\mathcal{E}$ is a hypertree iff the line graph $L(\mathcal{E})$ is chordal and $\mathcal{E}$ has the *Helly property*, i.e., any pairwise intersecting subfamily of edges of $\mathcal{E}$ has a common vertex; see [3]. A hypergraph $\mathcal{E}$ is a *dual hypertree* (*α-acyclic hypergraph*) iff there is a tree T with vertex set $\mathcal{E}$ such that for all vertices $v \in V$ $T_v = \{e \in \mathcal{E} : v \in e\}$ induces a subtree of T, i.e., $\mathcal{E}^*$ is a hypertree.

THEOREM 3.1 (see [12], [5]). *Let $G = (V, E)$ be a graph. Then the following conditions are equivalent:*

(i) *G is a dually chordal graph,*
(ii) *$\mathcal{N}(G)$ is a hypertree,*
(iii) *$\mathcal{D}(G)$ is a hypertree,*
(iv) *$\mathcal{C}(G)$ is a hypertree,*
(v) *G is the underlying graph of a hypertree.*

It is well known [6] that G is chordal iff $\mathcal{C}(G)$ is a dual hypertree, i.e., G is the underlying graph of some dual hypertree. Therefore the equivalence of parts (i) and (iv) of Theorem 3.1 justifies the name "dually chordal graphs" for graphs with maximum neighborhood ordering.

Since hypertrees are the dual hypergraphs of α-acyclic hypergraphs by Theorem 3.1 we immediately get that dually chordal graphs can be recognized in time proportional to the size of the corresponding hypergraph. This is a consequence of the linear time algorithm for testing α-acyclicity of hypergraphs [25]. It is easy to see that the hypergraph $\mathcal{N}(G)$ used in Theorem 3.1 has size proportional to the number of edges of the corresponding graphs. Therefore it can be tested in linear time $O(|E|)$ whether a graph $G = (V, E)$ is dually chordal.

Below we present a special algorithm for determining a maximum neighborhood ordering of dually chordal graphs. Its complexity is $O(|E|)$; for more details and a correctness proof we refer to [4].

ALGORITHM 3.2. **MNO** *(Find a maximum neighborhood ordering of G)*

```
Input:  A dually chordal graph $G = (V, E)$ with $|V| = n > 1$.
Output: A maximum neighborhood ordering of $G$.
    (0) initially all $v \in V$ are unnumbered and unmarked;
    (1) choose an arbitrary vertex $v \in V$, number $v$ with $n$, i.e., $v_n = v$, and let
        $mn(v_n) := v$;
repeat
```

```
    (2) among all unmarked vertices select a numbered vertex $u$ such that $N[u]$ contains a maximum number of numbered vertices;
    (3) number all unnumbered vertices $x$ from $N[u]$ consecutively with maximal possible numbers between $1$ and $n-1$ which are still free;
        for all of them let $mn(x) := u$;
    (4) mark $u$;
until all vertices are numbered
```

The meaning of $mn(x)$ is a maximum neighbor of x. Note that the algorithm also yields a maximum neighbor for each vertex, and all vertices of $N[v_n]$ occur consecutively in the ordering on the left of v_n and have v_n as their maximum neighbor. Furthermore, since G is assumed to be connected, for all v_i with $i \leq n-1$, $mn(v_i) \neq v_i$ holds.

Maximum neighborhood orderings of graphs produced by the MNO algorithm immediately lead to optimal algorithms for computing the distance matrix for all graphs having such orderings. Let $(v_1, \ldots, v_n)$ be a maximum neighborhood ordering of a graph G produced by the MNO algorithm. Subsequently we only use such maximum neighborhood orderings. This assumption is of crucial importance for the correctness of the main algorithm. The maximum neighbor $mn(v_i)$ of the vertex v_i in $G(\{v_i, v_{i+1}, \ldots, v_n\})$ has an important metric property: for every vertex $v_j, j > i$, which is nonadjacent to v_i there exists a shortest path of $G(\{v_i, v_{i+1}, \ldots, v_n\})$ between v_i and v_j which passes through $mn(v_i)$.

To see this assume by way of contradiction that all shortest paths between v_i and v_j contain vertices not in G_i. Among these paths choose a path P whose leftmost vertex u with respect to the maximum neighborhood ordering $(v_1, \ldots, v_n)$ has rightmost position. Let v, w be the neighbors of u in P. Since P is a shortest path the distance of v, w is 2. Now the maximum neighbor $mn(u)$ which according to the MNO algorithm is distinct from u is also adjacent to v and w and on the right of u. Thus replacing u by $mn(u)$ in P we obtain a shortest path P' between v_i and v_j whose leftmost vertex is on the right of u—a contradiction.

In particular, we obtain that every graph $G(\{v_{i+1}, v_{i+2}, \ldots, v_n\})$ is a distance-preserving subgraph of $G(\{v_i, v_{i+1}, \ldots, v_n\})$. Thus it follows that $G(\{v_i, v_{i+1}, \ldots, v_n\})$ is a distance-preserving subgraph of G for all $i \in \{1, \ldots, n\}$.

Let $G = (V, E)$ be a dually chordal graph, and let $D(G) = (d(v_i, v_j))_{i,j \in \{1,\ldots,n\}}$ denote the distance matrix of G. By $D_{i+1}(G)$ we denote the submatrix of $D(G)$ which contains the distances between the vertices $v_{i+1}, \ldots, v_n$. The next submatrix $D_i(G)$ is obtained from $D_{i+1}(G)$ by adding the ith row and ith column according to the following rule:

for all $k > i$ define

$$d(v_i, v_k) = d(v_k, v_i) = \begin{cases} 1 & \text{if } v_i \text{ and } v_k \text{ are adjacent,} \\ d(mn(v_i), v_k) + 1 & \text{otherwise.} \end{cases}$$

Evidently, this procedure correctly finds the whole matrix $D(G)$ in optimal time $O(n^2)$. Moreover, the maximum neighborhood ordering of G for every two query vertices u and v allows us to find in time $O(c \cdot d(u, v))$ a shortest path between u and v (c is the necessary time to verify the adjacency of two vertices). Let $num(v) = i$ if $v = v_i$ in the maximum neighborhood ordering of G.

Procedure 3.3 (**sh–path**(u, v)).

```
if $u$ and $v$ are adjacent then return $(u, v)$
else
```

```
if $num(u) < num(v)$ then
  return $(u, sh\text{–}path(mn(u), v))$
else
  return $(sh\text{–}path(u, mn(v)), v))$
```

According to [4] the shortest path between two vertices u and v of a dually chordal graph G computed by procedure *sh–path*(u, v) is called a *maximum neighbor path.* Such a path $P(u, v)$ has an important property: it splits into two subpaths $P' = (u_0, u_1, \ldots, u_p)$ and $P'' = (v_0, v_1, \ldots, v_q)$, where $u_0 = u$ and $v_0 = v$ such that each u_i is the maximum neighbor of u_{i-1} and each v_i is the maximum neighbor of v_{i-1} and vertices u_p, v_q are adjacent.

We conclude this section with the following property of cliques of dually chordal graphs.

LEMMA 3.4. *Let G be a dually chordal graph and C be a clique of G. If the vertex v r-dominates C then there exists a vertex v^* with $d(v, v^*) = \delta(v, C) - 1$ which 1-dominates C.*

Proof. If $d(v, u) < \delta(v, C)$ for some vertex $u \in C$, then u is the required vertex. Thus assume that all vertices of C are equidistant from v. Applying the Helly property to the family of pairwise intersecting disks consisting of $N^{\delta(v,C)-1}[v]$ and closed neighborhoods of vertices of C, we obtain a vertex v^* adjacent to all vertices of C and satisfying the required equality $d(v, v^*) = \delta(v, C) - 1$. □

4. Hypergraph approach to clique r-domination and clique r-packing.

For a family of cliques $\mathcal{C}$ of a graph G and a function $r : \mathcal{C} \to N \cup \{0\}$ with $r(C) > 0$ for cliques of size $|C| > 1$ define the hypergraph $\mathcal{D}_{\mathcal{C},r}(G)$ as follows:

$$\mathcal{D}_{\mathcal{C},r}(G) = \left\{ \bigcap_{v \in C} N^{r(C)}[v] : C \in \mathcal{C} \right\}$$

where $\bigcap_{v \in C} N^{r(C)}[v] = \{x : x\ r\text{-dominates } C\}$. (Note that this notation can be used in the same way for families of arbitrary vertex sets instead of cliques.)

Using this notation, the clique r-domination and clique r-packing problems on G may be formulated as the transversal and matching problems on the hypergraph $\mathcal{D}_{\mathcal{C},r}(G)$. Recall that a *transversal* of a hypergraph $\mathcal{E}$ is a subset of vertices which meets all edges of $\mathcal{E}$. A *matching* of $\mathcal{E}$ is a subset of pairwise disjoint edges of $\mathcal{E}$. For a hypergraph $\mathcal{E}$, the *transversal problem* is to find a transversal with minimum size $\tau(\mathcal{E})$, and the *matching problem* is to find a matching with maximum size $\nu(\mathcal{E})$. From the definitions we obtain the following lemma.

LEMMA 4.1. *D is a clique r-dominating set of a graph G iff D is a transversal of $\mathcal{D}_{\mathcal{C},r}(G)$. P is a clique r-packing set of a graph G iff P is a matching of $\mathcal{D}_{\mathcal{C},r}(G)$. Thus $\tau(\mathcal{D}_{\mathcal{C},r}(G)) = \gamma_{\mathcal{C},r}(G)$ and $\nu(\mathcal{D}_{\mathcal{C},r}(G)) = \pi_{\mathcal{C},r}(G)$ hold for every graph G and every function $r : \mathcal{C} \to N \cup \{0\}$ defined on every family $\mathcal{C}$ of cliques.*

The parameters $\gamma_{\mathcal{C},r}(G)$ and $\pi_{\mathcal{C},r}(G)$ are always related by a min–max duality inequality $\gamma_{\mathcal{C},r}(G) \geq \pi_{\mathcal{C},r}(G)$. The next result shows that for dually chordal graphs the converse inequality holds.

LEMMA 4.2. *For every family of cliques $\mathcal{C}$ of a dually chordal graph G and every function $r : \mathcal{C} \to N \cup \{0\}$ the hypergraph $\mathcal{D}_{\mathcal{C},r}(G)$ is a hypertree. In particular, the equality $\gamma_{\mathcal{C},r}(G) = \pi_{\mathcal{C},r}(G)$ holds.*

Proof. By definition each edge of the hypergraph $\mathcal{D}_{\mathcal{C},r}(G)$ is the intersection of some disks of G. By Theorem 3.1 the disk hypergraph $\mathcal{D}(G)$ of G is a hypertree. This means that there exists a tree T such that each edge of $\mathcal{D}(G)$ is a subtree of T.

Since the intersection of subtrees is a subtree too, all edges of $\mathcal{D}_{\mathcal{C},r}(G)$ are subtrees of T. Therefore the hypergraph $\mathcal{D}_{\mathcal{C},r}(G)$ is a hypertree. It is well known [3] that the equality $\tau(\mathcal{E}) = \nu(\mathcal{E})$ holds for every hypertree. By Lemma 4.1 we obtain the required equality. ☐

Note that Lemma 4.2 is true not only for families of cliques but also for arbitrary families of vertex sets of a dually chordal graph assuming only that the given r-values do not lead to empty hyperedges. (For hypergraphs with empty edges there is no transversal.)

It is known that in a hypertree $\mathcal{E}$ the transversal and matching problems can be solved in time proportional to the size of $\mathcal{E}$ [25]. The hypergraph $\mathcal{D}_{\mathcal{C},r}(G)$ can be computed in $O(|V|^2 + |V|\sum_{i=1}^{p}|C_i|)$ time for $\mathcal{C} = \{C_1, C_2, \ldots, C_p\}$. This can be done by applying the distance matrix of G. As we already mentioned this matrix can be computed in optimal time $O(|V|^2)$. In the worst case the hypergraph $\mathcal{D}_{\mathcal{C},r}(G)$ has size $O(|V||\mathcal{C}|)$. Thus the whole time to solve the clique r-domination and clique r-packing problems is $O(|V|^2 + |V|\sum_{i=1}^{p}|C_i|)$. Below we present an algorithm of the same complexity for solving these problems, which avoids the construction of the hypergraph $\mathcal{D}_{\mathcal{C},r}(G)$. For two particular cases when $\mathcal{C}$ consists only of maximal cliques or only of vertices of G its complexity becomes linear. The algorithm simultaneously finds a clique r-dominating set D and a clique r-packing set P such that $|D| = |P|$. This provides an algorithmic proof of duality results between these two problems on dually chordal graphs.

We refer also to [1], where similar duality results were used to provide efficient algorithms for three covering and packing problems on families of subtrees of a tree.

5. The algorithm. Let $G = (V, E)$ be a dually chordal graph, $\mathcal{C}$ be an arbitrary family of cliques of G, and $(v_1, \ldots, v_n)$ be the ordering of V generated by the MNO algorithm. By $r(C_1), \ldots, r(C_p)$ we denote the dominating radii of the corresponding cliques of $\mathcal{C}$. The algorithm processes the vertices in the order from v_1 to v_n. In iteration i the algorithm decides whether the vertex v_i has to be put into the clique r-dominating set D. If v_i is included in D then a certain clique C which is r-dominated by v_i is included in the clique r-packing set P. Initially, both sets D and P are empty. After processing, vertex v_i is deleted from the graph and information concerning whether or not v_i was included in D is given to its maximum neighbor $mn(v_i)$ and/or to its other neighbors and cliques from $N_i(v_i)$.

For technical reasons we extend the initial family of cliques $\mathcal{C}$ by including in $\mathcal{C}$ as one-vertex cliques all vertices $v \in V$ such that $\{v\} \notin \mathcal{C}$. For each of them initially $r(\{v\}) = \infty$. Let $\mathcal{C}(v)$ be a set of all cliques of $\mathcal{C}$ that contain the vertex v. At the next step we redefine the r-dominating radii of one-vertex cliques by putting $r(\{v\}) = \min\{r(C) : C \in \mathcal{C}(v)\}$. As in [7, 4] we associate to each clique $C \in \mathcal{C}$ the dominating radius $r(C)$ and the nonnegative integer $a(C)$. Initially $a(C) = \infty$ for all $C \in \mathcal{C}$. $a(C)$ keeps decreasing during the execution of the algorithm, while the dominating radii decrease only for one-vertex cliques. At each step $r(\{v\})$ becomes the current radius within which the clique $\{v\}$ and all other yet undominated cliques from $\mathcal{C}(v)$ must be r-dominated in the remaining graph. Unlike the dominating radii of cliques the value $a(C)$ indicates an upper bound for the distances of vertices $v \in D$ from the current clique r-dominating set D to the clique C; i.e., there is a $v \in D$ such that $d(v, C) \leq a(C)$. The value of $r(\{v_i\})$ decreases in the case where v_i is the maximum neighbor of a vertex v_j, $j < i$, such that there is a clique $C \in \mathcal{C}(v_j)$ that is not properly r-dominated by a vertex of D within distance $r(C)$ in iteration j. Then necessarily $r(C) = r(\{v_j\})$; for a proof see Lemma 5.5. In this case, $r(\{v_i\})$ is set to be $r(\{v_j\})-1$. Similarly, in a previous iteration, $r(\{v_j\})$ is set to be $r(\{v_k\})-1, k < j$. Continuing this argument, we find that there is a smallest (i.e., leftmost) vertex v_{i^*} and a clique $C^* \in \mathcal{C}(v_{i^*})$ that forces $\ldots, k, j, i$ to decrease their $r(\cdot)$ values, although $r(\{v_{i^*}\})$ never changes. We use $fn(\{v_i\})$ (*clique furthest neighbor* of $\{v_i\}$) to denote this initial clique C^* from which $r(\{v_i\})$ decreases.

In step i the algorithm processes vertex v_i according to the following rules. When $r(\{v_i\}) = 0$ then v_i must be in D because no other vertex r-dominates $\{v_i\}$. Moreover, we include $fn(\{v_i\})$ in P and set $a(\{v_i\}) = 0$. Otherwise, if $r(\{v_i\}) > 0$ then we distinguish between two cases. Either we find a clique $C \in \mathcal{C}(v_i)$ which is not yet r-dominated (lines (8) and (9)) or we establish that all cliques of $\mathcal{C}(v_i)$ are properly r-dominated by vertices of D in iteration i. In the second case we do nothing. So suppose that the first case holds. Then either C coincides with $\{v_i\}$ or C contains v_i as the smallest vertex in the MNO ordering and $r(C) = r(\{v_i\})$. In both cases we update $r(\{mn(v_i)\})$ by

$$r(\{mn(v_i)\}) = \min\{r(\{mn(v_i)\}), r(\{v_i\}) - 1\}.$$

At each step we have to update $a(C)$ for all $C \subseteq N_i(v_i)$:

$$a(C) = \min\{a(C), a(\{v_i\}) + 1\}.$$

The algorithm solves the transversal and matching problems on the hypergraph $\mathcal{D}_{\mathcal{C},r}(G)$ without constructing this hypergraph and works in linear time when $\mathcal{C}$ consists only of maximal cliques or only of vertices of G.

ALGORITHM 5.1. **(CRDP)** *(Find a minimum clique r-dominating set and a maximum clique r-packing set of a dually chordal graph G)*

Input: *A dually chordal graph $G = (V, E)$ with a maximum neighborhood ordering $(v_1, \ldots, v_n)$ obtained by the MNO algorithm and a family $\mathcal{C} = \{C_1, \ldots, C_p\}$ of cliques in G with radii $r(C_1), \ldots, r(C_p) \geq 0$ and $r(C_i) > 0$ for $|C_i| > 1$*

Output: *A minimum clique r-dominating set D and a maximum clique r-packing set P of G*

```
(1)  $D := \emptyset; P := \emptyset;$
(2)  for all $v \in V$ such that $\{v\} \notin \mathcal{C}$ do begin $\mathcal{C} := \mathcal{C} \cup \{\{v\}\}$; $r(\{v\}) := \infty$ end;
(3)  for all $C \in \mathcal{C}$ do begin $a(C) := \infty$; $fn(C) := C$ end;
(4)  for all $v \in V$ do
       begin
         choose a clique $C$ from $\mathcal{C}(v)$ with minimal radius $r(\cdot)$;
         $r(\{v\}) := r(C)$; $fn(\{v\}) := C$
       end;
(5)  for $i := 1$ to $n - 1$ do
       begin
         $par := 1$;
(6)      if $r(\{v_i\}) = 0$ then
           begin
(7)          $D := D \cup \{v_i\}; P := P \cup \{fn(\{v_i\})\}$; $a(\{v_i\}) := 0$
           end
         else
           begin
(8)          if $a(\{v_i\}) > r(\{v_i\})$ and $\forall v \in N_i(v_i)$ $a(\{v\}) + 1 > r(\{v_i\})$ and $r(\{v\}) > 0$
               then $C := \{v_i\}$
             else
```

```
(9)                 if $\exists C' \in \mathcal{C}(v_i)$ such that $[i = \min\{num(v) : v \in C'\}$ and $a(C') > r(C')$
                      and $(\forall v \in N_i[v_i]$
                        $(a(\{v\}) + 1 > r(C')$ or $(a(\{v\}) + 1 = r(C')$ and $C' \not\subseteq N[v]))$
                        and $(r(\{v\}) > 0$ or $r(\{v\}) = 0$ and $r(C') = 1$ and $C' \not\subseteq N[v]))]$
                      then $C := C'$
                    else
(10)                  $par := 0$;
(11)              if $par = 1$ and $r(\{mn(v_i)\}) \geq r(\{v_i\})$ then
                    begin
(12)                  $r(\{mn(v_i)\}) := r(\{v_i\}) - 1$;
(13)                  $fn(\{mn(v_i)\}) := fn(C)$
                    end;
                end;
(14)        for all $C \in \mathcal{C}$ such that $C \subseteq N_i(v_i)$ do $a(C) := \min\{a(C), a(\{v_i\}) + 1\}$;
          end;
(15)  if $r(\{v_n\}) < a(\{v_n\})$ then $D := D \cup \{v_n\}$ and $P := P \cup \{fn(\{v_n\})\}$
```

Subsequently for one-vertex cliques the brackets { and } are omitted.

THEOREM 5.2. *Algorithm CRDP is correct.*

Proof. In order to prove that the set D constructed by the algorithm is clique r-dominating we use the following reformulation of the clique r-domination problem in terms of $r(C)$ and $a(C)$:

find a minimum size set $D \subseteq V$ such that for every clique $C \in \mathcal{C}$

(a) $a(C) \leq r(C)$, or

(b) $\delta(v, C) \leq r(C)$ for some $v \in D$, or

(c) $\delta(u, C) + a(u) \leq r(C)$ for some $u \in V$ (D dominates C via vertex u).

(Note that the cases (a),(b),(c) do not exclude each other—this case distinction is useful subsequently for technical purposes.)

The proof of the theorem is based on some auxiliary results. In all of them let $\mathcal{C}_1 = \mathcal{C}$ and $\mathcal{C}_{i+1} = \mathcal{C}_i \setminus \mathcal{C}(v_i)$, where $\mathcal{C}(v_i)$ are all cliques of $\mathcal{C}$ which contain the vertex v_i.

LEMMA 5.3. *If $r(v_i) = 0$ then D is a clique r-dominating set of $\mathcal{C}_i$ in G_i iff $D = D' \cup \{v_i\}$, where D' is a clique r-dominating set of $\mathcal{C}_{i+1}$ in G_{i+1} with $a(C) := 1$ for all cliques $C \subseteq N_i(v_i)$ and $a(C) := a(C)$ otherwise, and $r(C) := r(C)$ for all cliques $C \in \mathcal{C}_{i+1}$.*

Proof. If $r(v_i) = 0$ then the one-vertex clique $\{v_i\}$ is not r-dominated by any other vertex of G_i. So, necessarily v_i belongs to every clique r-dominating set of G_i, in particular $v_i \in D$. Then v_i r-dominates every clique $C \subseteq N_i[v_i]$ with $r(C) > 0$. Let $D' = D \setminus \{v_i\}$. Assume that D' is not a clique r-dominating set for $\mathcal{C}_{i+1}$, i.e., some clique $C \in \mathcal{C}_{i+1}$ is not dominated by D'. Evidently, $C \not\subseteq N_i[v_i]$. Then in both graphs G_i and G_{i+1} the clique C has the same values for $r(C)$ and $a(C)$. Since C is r-dominated by D this is possible only if $\delta(u, C) + a(u) \leq r(C)$ for some u of G_i or $\delta(v_i, C) \leq r(C)$. Consider the second case. By Lemma 3.4 there exists a vertex v^* with $d(v_i, v^*) = \delta(v_i, C) - 1$ which dominates C. Let w be a neighbor of v_i which belongs to a shortest path between v_i and v_i^*. Since $C \not\subseteq N_i[v_i]$ such a vertex w always exists. Then in G_{i+1} $a(w) = 1$ and $\delta(w, C) + a(w) \leq r(C)$. This means that C is r-dominated by D'. Next suppose that $\delta(u, C) + a(u) \leq r(C)$. Necessarily $u \in N_i(v_i)$ or $u = v_i$. In the first case we obtain a similar inequality $\delta(u, C) + a(u) \leq r(C)$ in G_{i+1} too because $a(u)$ does not increase in G_{i+1}. Otherwise, if $\delta(v_i, C) + a(v_i) \leq r(C)$ then for the vertex $w \in N_i(v_i)$ introduced above we have $\delta(w, C) = \delta(v_i, C) - 1$ and

$a(w) \leq a(v_i) + 1$. This means that $\delta(w, C) + a(w) \leq r(C)$. Thus D' is a clique r-dominating set of $\mathcal{C}_{i+1}$ in G_{i+1}. Conversely if $\delta(u, C) + a(u) \leq r(C)$ for some clique $C \in \mathcal{C}_{i+1}$ and some vertex $u \in N_i(v_i)$ then $\delta(v_i, C) \leq r(C)$ and C is r-dominated in G_i from v_i. Therefore if D' is a clique r-dominating set for $\mathcal{C}_{i+1}$ then $D' \cup \{v_i\}$ is a clique r-dominating set for $\mathcal{C}_i$ in G_i. □

LEMMA 5.4. *Suppose that $r(v_i) \geq 1$ and that the conditions of lines* (8) *and* (9) *are not fulfilled. A subset $D \subseteq \{v_{i+1}, \ldots, v_n\}$ is a clique r-dominating set of $\mathcal{C}_i$ in G_i iff D is a clique r-dominating set of $\mathcal{C}_{i+1}$ in G_{i+1} with $r(C) := r(C)$ for all $C \in \mathcal{C}_{i+1}$, and $a(C) := \min\{a(C), a(v_i) + 1\}$ when $C \subseteq N_i(v_i)$ and $a(C) := a(C)$ otherwise.*

Proof. Since the values of $a(\cdot)$ do not increase from left to right along the ordering $(v_1, \ldots, v_n)$ each clique r-dominating set $D \subseteq \{v_{i+1}, \ldots, v_n\}$ of $\mathcal{C}_i$ in G_i is clique r-dominating in G_{i+1} too.

Conversely, suppose that D is a clique r-dominating set of $\mathcal{C}_{i+1}$ in G_{i+1}. Pick an arbitrary clique C of $\mathcal{C}_i$. By the conditions of the lemma every clique of $\mathcal{C}(v_i)$ is already r-dominated: let $C \in \mathcal{C}(v_i)$. Recall that the conditions of (8) and (9) are not fulfilled. Condition (8) is not fulfilled iff $a(\{v_i\}) \leq r(\{v_i\})$ or there exists $v \in N_i(v_i)(a(\{v\}) + 1 \leq r(\{v_i\})$ or $r(\{v\}) = 0)$. Consequently $\{v_i\}$ is dominated by D. Condition (9) is not fulfilled iff

$$\forall C' \in \mathcal{C}(v_i)(a(C') \leq r(C'))$$

(i.e., D dominates C') or

$$\exists v \in N_i(v_i)(a(\{v\}) + 1 \leq r(C') \text{ and } (a(\{v\}) + 1 \neq r(C') \text{ or } C' \subseteq N[v]))$$

(if even $a(\{v\}) + 2 \leq r(C')$ holds then D dominates C' via v; if $a(\{v\}) + 1 = r(C')$ then $C' \subseteq N[v]$ and thus also D dominates C' via v) or

$$r(\{v\}) = 0 \text{ and } (r \neq 0 \text{ or } r(C') \neq 1 \text{ or } C' \subseteq N[v])$$

($r(\{v\}) = 0$ means that $v \in D$, $r(v) \neq 0$ does not hold; thus $r(C') > 1$ since $r(v_i) \leq 1$ by the suppositions of the lemma and thus because of $r(\{v\}) = 0$ also C' is dominated or ($r(C') = 1$ and $C' \subseteq N[v]$) in which case C' is also dominated).

Thus it is enough to consider only the case when $C \in \mathcal{C}_{i+1}$. If $a(C) \leq r(C)$ in G_{i+1} then the same inequality holds in G_i too, except the case when $C \subseteq N_i(v_i)$. Then $a(v_i) + \delta(v_i, C) \leq a(v_i) + 1 \leq a(C) \leq r(C)$. If the clique C is r-dominated by a vertex $v \in D$ in G_{i+1} then v dominates C in G_i too. This is so because G_{i+1} is a distance preserving subgraph of G_i. Therefore it is sufficient to consider only the case when $\delta(u, C) + a(u) \leq r(C)$ holds in G_{i+1} for some vertex $u \in N_i(v_i)$ (i.e., for some one-vertex clique $\{u\} \subseteq N_i(v_i)$). If $\min\{a(u), a(v_i) + 1\} = a(u)$ then we are done. Otherwise, since $\delta(v_i, C) \leq \delta(u, C) + 1$ we obtain that

$$\delta(v_i, C) + a(v_i) \leq \delta(u, C) + a(v_i) + 1 \leq r(C).$$

Hence, D is a clique r-dominating set of $\mathcal{C}_i$ in G_i. □

LEMMA 5.5. *Assume that C^+ is a clique of $\mathcal{C}(v_i)$ obtained in lines* (8), (9) *of the algorithm. A subset $D \subseteq (\{v_{i+1}, \ldots, v_n\} \setminus \{v \in N_i(v_i) : r(v) \neq 0\}) \cup \{mn(v_i)\}$ is a clique r-dominating set of $\mathcal{C}_i$ in G_i iff D is a clique r-dominating set of $\mathcal{C}_{i+1}$ in G_{i+1} with $a(C) := \min\{a(C), a(v_i) + 1\}$ for $C \subseteq N_i(v_i)$ and $a(C) := a(C)$ for all other cliques of $\mathcal{C}_{i+1}$, and $r(C) := r(C)$ for all $C \in \mathcal{C}_{i+1} \setminus \{\{mn(v_i)\}\}$ and $r(mn(v_i)) := \min\{r(v_i) - 1, r(mn(v_i))\}$.*

Proof. 1. "$\Longrightarrow$": From our conditions we immediately get that if D' is a clique r-dominating set of G_{i+1} then replacing all vertices of $D' \cap \{v \in N_i(v_i) : r(v) \neq 0\}$ by $mn(v_i)$ we obtain a clique r-dominating set D with $|D| \leq |D'|$.

First we prove that if the clique C^+ is different from $\{v_i\}$ then $r(C^+) = r(v_i)$. Assume the contrary. Then $r(C^+) > r(v_i)$ because for every vertex v_i and every clique $C \in \mathcal{C}(v_i)$ during the first $i-1$ steps of the algorithm the inequality $r(C) \geq r(v_i)$ holds; see lines (4) and (12) of the algorithm. If $a(v_i) \leq r(v_i)$ then

$$a(v_i) + \delta(v_i, C) = a(v_i) + 1 \leq r(C^+).$$

Next suppose that there exists a vertex $v \in N(v_i)$ such that either $a(v) + 1 \leq r(v_i)$ or $r(v) = 0$. In the first case we immediately get

$$\delta(v, C^+) + a(v) \leq 2 + a(v) \leq r(C^+).$$

Otherwise, if $r(v) = 0$ then by the choice of C^+ we obtain that $r(C^+) = 1$. But then $r(v_i) = 0$, which is impossible. In all cases we get a contradiction with the choice of the clique C^+. Thus $r(C^+) = r(v_i)$.

Let $D \subseteq (\{v_{i+1}, \ldots, v_n\} \setminus \{v \in N_i(v_i) : r(v) \neq 0\}) \cup \{mn(v_i)\}$ be a clique r-dominating set of $\mathcal{C}_i$ in G_i and let C be an arbitrary clique of $\mathcal{C}_{i+1}$. First suppose that $C \neq \{mn(v_i)\}$. Since C is r-dominated by D in G_i in order to establish the same property in G_{i+1} it is enough to assume that $\delta(v_i, C) + a(v_i) \leq r(C)$ in G_i; otherwise we immediately get this. If $C \not\subseteq N_i(v_i)$ then as in Lemma 5.4 we choose a vertex $w \in N_i(v_i)$ such that $\delta(w, C) = \delta(v_i, C) - 1$. In this case we have

$$\delta(w, C) + a(w) \leq \delta(v_i, C) + a(v_i) \leq r(C).$$

If $C \subseteq N_i(v_i)$ then as $a(C) \leq a(v_i) + 1$ and $\delta(v_i, C) = 1$ in G_{i+1} we obtain that $a(C) \leq r(C)$.

Next consider the one-vertex clique $\{mn(v_i)\}$ of G_{i+1}. We can suppose that

$$r(mn(v_i)) = r(v_i) - 1 = r(C^+) - 1;$$

otherwise, we can apply the preceding arguments. In G_i the clique C^+ is r-dominated by D. By the choice of C^+ this means that in G_i either $\delta(u, C^+) + a(u) \leq r(C^+)$ for some vertex $u \notin N_i[v_i]$ or $\delta(u, C^+) \leq r(C^+)$ for some $u \in D \backslash N_i[v_i]$. In both cases the vertex $mn(v_i)$ is one step closer to u than the vertices of C^+. This allows us to conclude that in G_{i+1} either

$$d(mn(v_i), u) + a(u) \leq r(C^+) - 1 = r(mn(v_i))$$

or

$$d(mn(v_i), u) \leq r(C^+) - 1 = r(mn(v_i)).$$

Therefore D is a clique r-dominating set of $\mathcal{C}_{i+1}$ in G_{i+1}.

2. "$\Longleftarrow$": Conversely, let $D \subseteq \{v_{i+1}, \ldots, v_n\}$ be a clique r-dominating set of $\mathcal{C}_{i+1}$ in G_{i+1}. The arguments for proving that D is a clique r-dominating set of $\mathcal{C}_i$ in G_i are the same as in Lemma 5.4 except when C is a clique of $\mathcal{C}(v_i)$. For all such cliques we have $r(C) \geq r(v_i)$. Then every such clique is r-dominated by the same vertex as the clique $\{mn(v_i)\}$ is r-dominated in G_{i+1}, except the case when $a(mn(v_i)) \leq r(mn(v_i))$

in G_{i+1}. Then $mn(v_i)$ r-dominates all cliques of $\mathcal{C}(v_i)$ because for all $C \in \mathcal{C}(v_i)$ we have

$$\delta(mn(v_i), C) + a(mn(v_i)) = 1 + a(mn(v_i)) \leq 1 + r(mn(v_i)) \leq r(v_i) \leq r(C).$$

Thus, D is a clique r-dominating set of $\mathcal{C}_i$ in G_i. □

LEMMA 5.6. *The set $D \subseteq V$ obtained by the algorithm is a clique r-dominating set of $\mathcal{C}$ in G.*

Proof. The proof follows from the preceding three lemmas by induction on the number of vertices and by the fact that the clique r-dominating set of a larger family of cliques remains clique r-dominating for every subfamily too. □

LEMMA 5.7. $|P| = |D|$.

Proof. By the algorithm we know that $|D| \geq |P|$. If $|P| < |D|$ then there are two vertices $u, v \in D$ and a clique $C \in P$ such that $fn(u) = C = fn(v)$. Let w be the vertex of C with the minimal index $num(w)$. By the algorithm, $fn(u)$ and $fn(v)$ reach vertices u and v along maximum neighbor paths connecting w, u and w, v, respectively. Let w^+ be the vertex belonging to both paths that is furthest from w. Denote by u' and v' the next neighbors of w^+ in these paths. Necessarily $u' = v'$ since both are the maximum neighbors $mn(w^+)$. Thus the unique maximum neighbor path from w^+ reaches the unique vertex $u = v$, which is a contradiction to the assumption that there are two vertices $u, v \in D$ with the property $fn(u) = C = fn(v)$. □

LEMMA 5.8. *Let $R = (u_0, u_1, \ldots, u_k)$ be a path between vertices u_0 and u_k such that $u_i = mn(u_{i-1}), i = 1, \ldots, k$. If u is a vertex with $num(u) > num(u_{k-1})$ then either the maximum neighbor path between u_0 and u contains R or u is adjacent to all vertices $u_j, \ldots, u_k$ for some $j < k$.*

Proof. By the conditions of the lemma we get that the procedure *sh–path* will include in the maximum neighbor path between u_0 and u all vertices of R until a neighbor u_j of u is achieved. If $j = k$ then the whole path R belongs to the constructed path. Otherwise, if $j < k$ then because $num(u_j) < \cdots < num(u_k) < num(u)$ and $u_{i+1} = mn(u_i)$ for each $i \leq k-1$, we obtain that u must be adjacent to all vertices $u_j, u_{j+1}, \ldots, u_k$. □

LEMMA 5.9. *P is a clique r-packing set of $\mathcal{C}$ in G.*

Proof. First of all, note that P is a subset of the initial family of cliques $\mathcal{C}$; see line (4) of the algorithm.

Assume that P is not a clique r-packing set; i.e., there are two cliques $C', C'' \in P$ which are r-dominated by a common vertex of G. In particular we obtain that $d(w', w'') \leq r(C') + r(C'')$ for arbitrary vertices $w' \in C'$ and $w'' \in C''$. According to the algorithm there are vertices $x, y \in D$ such that $C' = fn(x)$ and $C'' = fn(y)$. Let $u \in C'$ and $v \in C''$ be vertices with minimal indices $num(u)$ and $num(v)$ in the cliques C' and C'', respectively. By the algorithm $fn(u) = C'$ and $fn(v) = C''$. Let R' and R'' be maximum neighbor paths between u, x and v, y. Both of these paths are increasing. By the algorithm we obtain that R' and R'' are disjoint; otherwise a common vertex of R' and R'' is a “bottleneck” for the transmission of cliques $C' = fn(x)$ and $C'' = fn(y)$ (see also Lemma 5.7). Moreover, the lengths of these paths are $r(C')$ and $r(C'')$, respectively.

Let R be the maximum neighbor path between vertices u and v. As we know already R splits into two increasing maximum neighbor paths $R_u = (u, \ldots, u^+)$ and $R_v = (v, \ldots, v^+)$ and an edge u^+v^+. Comparing the paths R', R_u and R'', R_v we conclude that they must be comparable with respect to $\subseteq$. Because of $d(u, v) \leq r(C') + r(C'') = r(u) + r(v)$ at least one of the inequalities $|R'| > |R_u|$ or $|R''| > |R_v|$ holds. In particular, at least one of the incidences $u^+ \in R'$ or $v^+ \in R''$ is satisfied.

Case 1. $u^+ \in R'$ and $v^+ \in R''$.

Let R'_0 and R''_0 be subpaths of R' and R'' which connect vertices u^+, x and v^+, y, respectively. Since $d(u, v) \leq r(C') + r(C'')$ at least one of these paths has an edge, i.e., $u^+ \neq x$ or $v^+ \neq y$. Among adjacent vertices $u' \in R'_0$ and $v' \in R''_0$ with $u' \neq x$ or $v' \neq y$, we choose adjacent vertices $u^* \in R'_0$ and $v^* \in R''_0$ whose sum $d(u^*, x) + d(v^*, y)$ is minimal. Assume without loss of generality that $num(u^*) < num(v^*)$.

We claim that $u^* \neq u$ or $v^* \neq v$ holds. Assume to the contrary that $u^* = u$ and $v^* = v$. Then necessarily $r(C') + r(C'') > 0$. If $r(C') > 0$ then the vertex $mn(u^*)$ of R' must be adjacent to v^*. We get a contradiction with the choice of u^* and v^* except when $mn(u^*) = x$ and $v^* = y$. Then $r(C') = 1$ while $r(C'') = 0$ and thus $C'' = \{v\}$. If v is not adjacent to some vertex $u' \in C'$ then $d(v, u') = 2 > r(C') + r(C'')$ in contradiction to the choice of the cliques C' and C''. Otherwise if v is adjacent to all vertices of C' then $C' \subseteq N[v]$ and according to the algorithm we cannot transmit the clique $C' = fn(u)$ to the vertex $mn(u)$. Next assume that $r(C') = 0$, i.e., $C' = \{u\}$ and $x = u^* = u$. Again, as in the preceding case, if $r(C'') = 1$ then u must be adjacent to all vertices of C''. Hence in step $num(v)$ we have $r(C'') = a(C'') = 1$, which is a contradiction. So let $r(C'') \geq 2$. But then in step $num(v)$ we have $\delta(v, C'') + a(v) = 1 + 1 \leq r(C'')$ and according to lines (8) and (9) of the algorithm we cannot insert the clique C'' in P. Thus $u^* \neq u$ or $v^* \neq v$ holds.

Since $num(u^*) < num(v^*)$ we get that either u^* coincides with u or x or v^* is adjacent to the maximum neighbor $mn(u^*)$ of u^*. In the last case we obtain a contradiction with our choice of an edge u^*v^* except the case when $mn(u^*) = x$ and $v^* = y$. In this case we conclude that y is adjacent to x. If $r(C'') = 0$ and $C'' = \{y\}$ then in step $num(u^*)$ we have $r(y) = 0$ for $y \in N(u^*)$. By the algorithm we cannot transmit $C' = fn(u^*)$ to x. So $r(C'') > 0$. Let y^+ be the neighbor of y in P''. If $num(y^+) < num(u^*)$ then in step $num(y^+)$ we obtain $r(y) = 0$. Therefore in step $num(u^*)$ we already have a neighbor of u^* which violates the condition in line (8) of the algorithm. Hence $num(y^+) > num(u^*)$; i.e., the vertex $x = mn(u^*)$ is adjacent to y^+. Then $r(x) = 0$ in step $num(y^+)$ and again we can apply the condition in line (8) in order to obtain a contradiction with $C'' = fn(v) = \cdots = fn(y)$.

Therefore we obtain that if $num(u^*) < num(v^*)$ then u^* coincides with x or u. Consider the first case, i.e., $u^* = x$. Then $v^* \neq y$. Before step $num(x)$ we have $r(x) = 0$; after this step we obtain $a(v^*) = 1$. Then in step $num(v^*)$ the condition in line (8) of the algorithm is violated except the case when $v^* = v$ and $r(C'') \leq 1$. By the choice of cliques C' and C''

$$d(u, v') \leq r(C') + r(C'') \leq r(C') + 1$$

holds for every vertex $v' \in C''$. Because of $num(x) = num(u^*) < num(v^*) = num(v) < num(v')$ we can apply Lemma 5.8 to path R' and every vertex $v' \in C''$. If $r(C'') = 0$ then necessarily $C'' = \{v\}$ and by this lemma we get that v is adjacent to some vertex $z \neq x$ of R'. Then in step $num(z)$ we have a neighbor v of z with $r(v) = 0$ and we can apply the condition in line (8) in order to obtain a contradiction with $C' = fn(u) = \cdots = fn(z) = \cdots = fn(x)$. So suppose that $r(C'') = 1$. If x is adjacent to all vertices of C'' then in step $num(x)$ we obtain $a(C'') = 1$. Then in step $num(v)$ we have $a(C'') = r(C'')$ and we cannot include C'' in P. So there is a vertex $v' \in C''$ nonadjacent with x. By Lemma 5.8 the whole path R' belongs to the maximum neighbor path between u and v'. Then $d(u, v') \geq d(u, x) + 2 > r(C') + r(C'')$ in contradiction with the choice of the cliques C' and C''.

Finally consider the case when $u^* = u$. Then $v^* \neq v$ and $u^* \neq x$; otherwise, the conditions of the preceding cases are fulfilled. In particular we obtain that $r(C') > 0$.

Since $num(u^*) < num(v^*)$ the vertex v^* must be adjacent to the vertex $mn(u^*)$. By the choice of vertices u^* and v^* and our conditions we get $v^* = y$ and the vertices $u^* = u$ and x are adjacent. Thus $r(C') = 1$. Let z be the neighbor of v^* in the subpath of R'' connecting the vertices v and v^*. If $num(u^*) < num(z)$, then from $v^* = mn(z)$ it follows that $v^* = mn(u^*)$. Then we get a contradiction with the disjointness of the paths R' and R''. So suppose that $num(u^*) > num(z)$. If y is adjacent to all vertices of the clique C' then in step $num(u)$ we have $r(y) = 0$ and $C' \subseteq N[y]$. This leads to a contradiction with the fact that $C' = fn(x)$ is included in P. So suppose that y is nonadjacent to some vertex $u' \in C'$. Because of

$$num(u') > num(u) > num(z) > \cdots > num(v)$$

we can apply Lemma 5.8 to the vertex u' and the path R''. Then we obtain that

$$d(v, u') \geq d(v, y) + d(y, u') \geq r(C'') + 2 > r(C'') + r(C')$$

in contradiction with the choice of the cliques C' and C''.

Case 2. $u^+ \in R'$ and $v^+ \notin R''$ (the case when $u^+ \notin R'$ and $v^+ \in R''$ is similar).

Since the paths R'' and R_v are comparable, the inclusion $R'' \subseteq R_v$ holds; i.e., the vertex y belongs to the path R_v. Let z be the neighbor of v^+ in R_v. Then $v^+ = mn(z)$ and z belongs to the subpath of R_v between v^+ and y. Let $d(u^+, x) = l'$ and $d(v^+, y) = l''$. Since $d(u, v) \leq r(C') + r(C'')$ and $d(u, v) = r(C') - l' + 1 + l'' + r(C'')$ holds, the inequality $l'' < l'$ necessarily is fulfilled. Moreover since $v^+ \neq y$ and $2 + r(C'') \leq d(u, v) \leq r(C') + r(C'')$ holds we have $r(C') \geq 2$.

If $num(u^+) < num(z)$ then the vertex $mn(u^+) \in R'$ must be adjacent to both v^+ and z. Then since u^+ and z are adjacent to both vertices $mn(u^+)$ and $v^+ = mn(z)$ by the MNO algorithm we conclude that $v^+ = mn(u^+)$. By the CRDP algorithm in step $num(v^+)$ we have $a(v^+) = l''$ and $r(v^+) = l' - 1$. Since $l'' < l'$ the inequality $a(v^+) \leq r(v^+)$ holds. Comparing with the conditions in lines (8) and (9) we get that $x = v^+$. But then

$$d(u, v) \geq r(C') + d(x, y) + r(C'') > r(C') + r(C''),$$

which leads to a contradiction.

Now assume that $num(u^+) > num(z)$. If $num(u^+) < num(v^+)$ then in step $num(u^+)$ for vertex v^+ we have $a(v^+) = l'' < l' = r(u^+)$. Therefore in step $num(u^+)$ $a(v^+) + 1 \leq r(u^+)$ and if $u \neq u^+$ we cannot transmit the value $fn(u^+) = C'$ to the maximum neighbor of u^+. Otherwise if $num(u^+) > num(v^+)$ and $u^+ \neq u$ then before step $num(u^+)$ we already have $a(u^+) \leq l'' + 1$. Then $a(u^+) \leq r(u^+)$ in step $num(u^+)$ and again we can apply the condition in line (8) in order to obtain a contradiction with $C' = fn(u) = \cdots = fn(u^+) = \cdots = fn(x)$.

Finally suppose that $u^+ = u$ and $num(u) = num(u^+) > num(z)$. First assume that $l'' + 1 = l'$. We claim that $C' \subseteq N[v^+]$. Assume the contrary and let u' be a vertex of C' nonadjacent with v^+. Since

$$num(u') > num(u) > num(z) > \cdots > num(v),$$

by applying Lemma 5.8 to the path R_v and the vertex u' we get

$$d(u', v) = d(v, v^+) + d(v^+, u') \geq r(C'') + l'' + 2 > r(C'') + r(C'),$$

which is a contradiction. So $C' \subseteq N[v^+]$. Then in step $num(u)$ we have $a(C') \leq l'' + 1 = l' = r(C')$ if $num(u) > num(v^+)$ and $a(v^+) + 1 \leq l'' + 1 = l' = r(C')$

and $C' \subseteq N[v^+]$ if $num(u) < num(v^+)$. In both cases we get a contradiction with $C' = fn(u) = \cdots = fn(x)$. So let $l''+1 < l'$. Then in step $num(u)$ we have $1+a(u) \leq l''+2 \leq l' = r(C'')$ if $num(u) > num(v^+)$ and $1 + a(v^+) = l'' + 1 < l' = r(C')$ if $num(u) < num(v^+)$. We obtain the same contradiction as in the preceding case. This concludes the proof of the lemma. ∎

From Lemmas 5.3–5.9 and the duality between the clique r-domination and clique r-packing problems we immediately obtain that the sets $D \subseteq V$ and $P \subseteq \mathcal{C}$ computed by the algorithm CRDP are minimum clique r-dominating and maximum clique r-packing sets. ∎

6. Time bounds for special cases. In this section we consider the time bounds for some important special input cases of the CRDP algorithm. The obtained results are collected in the following.

THEOREM 6.1. *Let $\mathcal{C} = \{C_1, \ldots, C_p\}$ be a family of cliques of a dually chordal graph $G = (V, E)$ and $r : \mathcal{C} \to N \cup \{0\}$ be the radius function on $\mathcal{C}$. Then the clique r-domination and clique r-packing problem for $\mathcal{C}$ can be solved in time*

(1) $O(|E| + |V| \sum_{i=1}^{p} |C_i|)$ *if $\mathcal{C}$ is an arbitrary family of cliques,*
(2) $O(|E| + \sum_{i=1}^{p} |C_i|)$ *if $\mathcal{C}$ is a family of maximal cliques,*
(3) $O(|E|)$ *if $\mathcal{C}$ is a family of one-vertex cliques,*
(4) $O(|V||E|)$ *if $\mathcal{C}$ is a family of edges,*
(5) $O(|E|)$ *if G is doubly chordal and $\mathcal{C}$ is a family of maximal cliques,*
(6) $O(|E|)$ *if G is doubly chordal without induced sun S_3 and $\mathcal{C} = E$ and $r(e) \equiv k$ for all $e \in E$.*

The running time of the algorithm for an arbitrary family $\mathcal{C} = \{C_1, \ldots, C_p\}$ of cliques of a dually chordal graph $G = (V, E)$ can be estimated as follows. For a vertex $v_i \in V$ let s_i be the degree of v_i and k_i be the number of cliques containing v_i (i.e., $k_i = |\mathcal{C}(v_i)|$) and l_i be the number of cliques of $\mathcal{C}$ which are dominated by v_i, i.e.,

$$l_i = |\{C \in \mathcal{C} : C \subseteq N[v_i]\}|.$$

Evidently lines (2) and (3) of the algorithm use $O(|\mathcal{C}| + |V|)$ operations, while line (4) takes $O(|V| + \sum_{i=1}^{p} |C_i|)$ time. The overall time bound of lines (5)–(8), (10)–(13), and (15) is $O(|E|)$. In order to implement lines (9) and (14) we compute in advance in $O(\sum_{i=1}^{|V|} k_i s_i + \sum_{i=1}^{|V|} l_i)$ time the 0-1-matrix $M = (m_{ij})_{j=1,\ldots,|\mathcal{C}|}^{i=1,\ldots,|V|}$, where $m_{ij} = 1$ if and only if $C_j \subseteq N[v_i]$. Using this matrix, lines (9) and (14) can be executed in time $O(\sum_{i=1}^{|V|} k_i s_i)$ and $O(\sum_{i=1}^{|V|} l_i)$, respectively. So the total time of the algorithm is

$$O\left(|E| + \sum_{i=1}^{|V|} k_i s_i + \sum_{i=1}^{|V|} l_i\right),$$

which in the worst case can be estimated as $O(|E| + |V| \sum_{i=1}^{|\mathcal{C}|} |C_i|)$ because $\sum_{i=1}^{|V|} k_i = \sum_{i=1}^{|\mathcal{C}|} |C_i|$ and $\sum_{i=1}^{|V|} l_i \leq |V||\mathcal{C}|$.

Now we consider the complexity of the algorithm CRDP in some important particular cases. First suppose that $\mathcal{C}$ consists only of all one-vertex cliques. Then we obtain the r-domination and r-packing problems. Since

$$\sum_{i=1}^{|\mathcal{C}|} |C_i| = |\mathcal{C}| \leq |V|, \quad \sum_{i=1}^{|V|} l_i = \sum_{i=1}^{|V|} s_i = 2|E|$$

and line (9) is omitted the total time complexity for these problems is $O(|E|)$.

Next let $\mathcal{C}$ be a family of maximal cliques of G. Then the time complexity of all lines is the same as in the general case except lines (9) and (14). Line (14) takes only $O(|E|)$ time because the condition $C \subseteq N_i(v_i)$ is fulfilled only for the additional one-vertex cliques and not for any maximal clique of G. Line (9) can be implemented directly without computing the matrix M. For this purpose recall that it is enough to verify the condition in this line only for cliques $C \subseteq N_i[v_i]$ such that v_i has minimal index in C and $r(C) = r(v_i)$; see the proof of Lemma 5.5. Using this fact first we can select all vertices $v \in N_i[v_i]$ such that either $a(v) + 1 = r(v_i)$ or $r(v) = 0$. This can be done in time $O(s_i)$. After that for each clique $C \in \mathcal{C}(v_i)$ such that $r(C) = r(v_i)$ and v_i has minimal index in C we decide whether $C \not\subseteq N[v]$ for some selected vertex v. This operation can be implemented in time $O(k_i + \sum_{C_i \in \mathcal{C}(v_i)} |C_i|)$. So for each $i, i \in \{1, \dots, |V|\}$ the time complexity of line (9) is $O(k_i + l_i + \sum_{C_i \in \mathcal{C}(v_i)} |C_i|)$. Since each clique is considered only once in step $\min\{num(v) : v \in C\}$ we conclude that the overall time amount of line (9) and of the whole algorithm is $O(|E| + \sum_{i=1}^{|\mathcal{C}|} |C_i|)$. Assume that a dually chordal graph $G = (V, E)$ is also chordal; i.e., G is a doubly chordal graph. It is known that for chordal graphs $\sum_{i=1}^{|\mathcal{C}|} |C_i|$ does not exceed $O(|E|)$. So for doubly chordal graphs the algorithm requires only $O(|E|)$ time. Since strongly chordal graphs are doubly chordal [5] this result improves and generalizes the algorithm for the clique transversal and clique independence problem presented in [8].

Finally consider the case when $\mathcal{C}$ is a subset of edges of a dually chordal graph $G = (V, E)$. Since $\sum_{i=1}^{|\mathcal{C}|} |C_i| \leq 2|E|$ the time complexity of the algorithm is $O(|V||E|)$.

Next assume that $\mathcal{C} = E$ and $r(e) \equiv k$ for every edge $e \in E$, where k is a positive integer. As we already mentioned these problems are known as the k-neighborhood covering and k-neighborhood independence problems [17]. In [17] a linear time algorithm for solving these problems in strongly chordal graphs is presented under the assumption that a strong elimination ordering of such graphs is given. Unfortunately the fastest known algorithms for deriving such orderings have time complexity $O(|V|^2)$ [23] or $O(|E| \log |V|)$ [21]. Our algorithm can be modified in order to solve the k-neighborhood covering and k-neighborhood independence problems on strongly chordal graphs. In fact the approach presented below solves these problems on a more general subclass of dually chordal graphs, namely on doubly chordal graphs containing no induced sun S_3 (see Figure 1). This is mainly due to the next result.

Denote by $\gamma_{E,k}$ and $\pi_{E,k}$ the k-neighborhood covering and the k-neighborhood independence numbers of G.

LEMMA 6.2. *Let $G = (V, E)$ be a doubly chordal graph containing no induced sun S_3. Then*

$$\gamma_{E,k}(G) = \pi_{E,k}(G) = \pi_{\mathcal{C},k}(G) = \gamma_{\mathcal{C},k}(G),$$

where $\mathcal{C}$ is the family of all maximal cliques of G and $r(C) \equiv k$ for all $C \in \mathcal{C}$.

Proof. By Lemma 4.2 we have

$$\gamma_{E,k}(G) = \pi_{E,k}(G),\ \pi_{\mathcal{C},k}(G) = \gamma_{\mathcal{C},k}(G).$$

Moreover for arbitrary graphs $\pi_{E,k}(G) \leq \pi_{\mathcal{C},k}(G)$ holds. In order to show this we extend each edge of the k-neighborhood independent set to a maximal clique of G. The obtained family of cliques represents a clique k-packing set of $\mathcal{C}$. So it is enough to show the converse inequality. Let P be a maximal clique k-packing set of G. By

[18, Proposition 3] every maximal clique of a chordal graph G without induced sun S_3 has an edge which is not contained in any other maximal clique of G. Include in the set I all such representative edges of cliques from P. We claim that I is a k-neighborhood independent set of G. Let $e' = u'v'$ and $e'' = u''v''$ be edges which represent cliques $C', C'' \in P$. Assume that there exists a vertex w such that $\delta(w, e') \leq k$ and $\delta(w, e'') \leq k$. We will show that $\delta(w, C') \leq k$ and $\delta(w, C'') \leq k$. Assume the contrary and let $\delta(w, C') > k$. This means that $d(w, v) > k$ for some vertex $v \in C'$. Then necessarily $d(w, v') = d(w, u') = k$. By Lemma 3.4 there is a common neighbor w^* of vertices v' and u' which is at distance $k - 1$ from w. Since the edge e' is contained in the unique clique C' we have $w^* \in C'$ and thus the vertices w^* and v must be adjacent. This contradicts the assumption that $d(w, v) > k$. □

Thus in order to solve the k-neighborhood covering and k-neighborhood independence problems on a doubly chordal graph G without induced sun S_3 we can apply the algorithm CRDP for the family of all maximal cliques $\mathcal{C}$ of G when $r(C) \equiv k$ for all $C \in \mathcal{C}$. As we already mentioned for doubly chordal graphs this problem can be solved in time $O(|E|)$. Let D and P be the output of the algorithm. Evidently D is a minimum k-neighborhood covering set too while the set I defined in the proof of Lemma 6.2 is a maximum k-neighborhood independent set. Hence it remains only to efficiently compute such a set I in $O(|E|)$ time under the assumption that the set P is given.

In order to do this we use the perfect elimination ordering of a chordal graph G and obtain in $O(|E|)$ time all representative edges for the family of maximal cliques; see the procedure presented below. In each clique of the set P we select such an edge and obtain the required set I.

Let $G = (V, E)$ be a chordal graph and $v_1, \dots, v_n$ be a perfect elimination ordering of G. As before $num(v)$ is the index of a vertex v in this ordering.

Procedure 6.3 (**representative edges).**

```
$E^* := \emptyset$;
for all $i \in \{1, \dots, n\}$ do $A_i := N_i(v_i)$;
for $i := 1$ to $n - 1$ do
  if $A_i$ is not empty then
    begin
      $v :=$ arbitrary vertex from $A_i$;
      $E^* := E^* \cup v_i v$;
      for all $v_j \in N_i(v_i)$ do $A_j := A_j \setminus N_i(v_i)$;
    end
```

Lemma 6.4. *Let $G = (V, E)$ be a chordal graph without induced sun S_3. Then the presented procedure correctly finds within $O(|E|)$ steps a set of representative edges for the family of all maximal cliques of G.*

Proof. The correctness proof is based on two claims.

(a) Each maximal clique of G contains an edge selected by the procedure.

(b) The family of selected edges E^* consists only of representative edges.

In order to prove (a) let C be an arbitrary maximal clique. By [18, Proposition 3], C has a representative edge $v_i v_j, i < j$. If this or any other edge of C is not selected by the procedure then in step i we must have $A_i = \emptyset$. This means in particular that in some step $t < i$ the vertex v_j must be deleted from A_i. By the procedure in this step some edge $v_t v_l$ must be included into E^* where $v_l \in N_t(v_t)$. Since the edge $v_i v_j$ is representative for the clique C and v_t is adjacent to both v_i and v_j, certainly v_t is a vertex of C. Moreover, since v_t is simplicial in $G(\{v_t, v_{t+1}, \dots, v_n\})$ the vertex v_l is

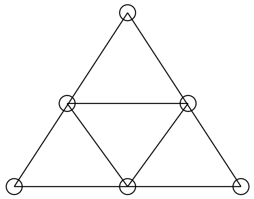


FIG. 1.

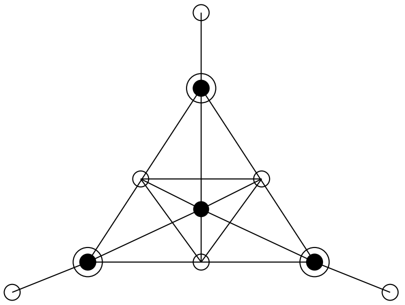


FIG. 2.

adjacent to v_i and v_j. Using the same argument for the edge v_iv_j, we conclude that $v_l \in C$ and thus the edge $v_tv_l \in E^*$ is contained in C.

Next suppose that in E^* there is an edge v_iv_j, $i < j$ which belongs to two maximal cliques C' and C''. Let v_t be the vertex with minimal index in $C' \cup C''$. Let, for example, $v_t \in C'$. Since both cliques C' and C'' are completely contained in $G(\{v_t, \ldots, v_n\})$ and v_t is a simplicial vertex of this graph, we obtain that $v_t \notin C''$. Then v_t is not adjacent to some vertex $v_l \in C''$. By the procedure the edge v_iv_j is included in E^* in step i. Therefore, before this step we have $v_j \in A_i$. In particular, v_j remains in A_i after step $t < i$. This is possible only if $A_t = \emptyset$. Since initially both vertices v_i and v_j belong to A_t they must be deleted from this set in some steps k' and k''. Necessarily $k' \neq k''$; otherwise we must delete the vertex v_j from A_i in step k', which is impossible. But in this case the vertices $v'_k, v_t, v_i, v_j, v''_k, v_l$ induce a sun S_3. This is the case because G is chordal and there are no edges between the vertex pairs (v_l, v_t), (v'_k, v_j) and (v''_k, v_i).

Time bound. Let $\sigma = (v_1, \ldots, v_n)$ again be the perfect elimination ordering of G. For every $i \in \{1, \ldots, n\}$ determine the position $c(v_i)$ of the leftmost neighbor of v_i in σ. Obviously this can be done in $2|E|$ steps.

Rearrange the vertices of V in increasing order with respect to the parameter $c(v_i)$, $i \in \{1, \ldots, n\}$. This can be done using bucket sort in $O(|E|)$ time, obtaining the ordering $\tau = (v_{j_1}, \ldots, v_{j_n})$. Note that the nonempty elements of the family $L_k = \{v_i : c(v_i) = k\}$, $k \in \{1, \ldots, n\}$, represent a clique partition of the chordal graph G along σ.

Using the ordering τ and the standard technique of how to get an ordered adjacency list from a nonordered adjacency list of G (see, e.g., [16]) we get an ordered representation of A_i as linearly linked list $L^i_{j_1}, \ldots, L^i_{j_i}$, ordered with respect to τ, of linearly linked lists $L^i_{j_k} \subseteq L_{j_k}$, $k \in \{1, \ldots, i\}$, called *segments* subsequently.

Then, having this structure for every A_i, the operation $A_j := A_j \setminus N_i(v_i)$ can be performed in the following way: due to claim (a) each maximal clique contains a representative edge, and hence we have to delete only the leftmost segment (corresponding to $c(v_i)$) from the current list A_j.

Since for fixed j this can be done in constant time the whole procedure takes linear time. □

Unfortunately the equality in Lemma 6.2 does not hold for all dually chordal graphs. Figure 2 provides an example of a doubly chordal graph G with $\gamma_{\mathcal{C},1}(G) = 4$ and $\gamma_{E,1}(G) = 3$.

7. Conclusions. In this paper we presented a unified approach to solve different types of clique r-domination and clique r-packing problems on dually chordal graphs. For three particular cases of these problems, namely for r-domination and r-packing, clique transversal and clique independence, and k-neighborhood covering and k-neighborhood independence we obtain linear time algorithms. The corresponding results generalize and improve results of [7, 8, 17] for the same problems on strongly chordal graphs.

Concerning the clique transversal and clique independence problems, the complexity of our algorithm is proportional to the sum of the sizes of all maximal cliques of a dually chordal graph. Unlike chordal graphs where the number of maximal cliques does not exceed the number of vertices, dually chordal graphs may have an exponential number of maximal cliques. The reason for this is that an arbitrary graph G can be transformed into a dually chordal graph by adding a new vertex adjacent to all vertices of G. Moreover, it is not known whether the clique transversal problem is in NP.

REFERENCES

[1] I. Bárány, J. Edmonds, and L. A. Wolsey, *Packing and covering a tree by subtrees*, Combinatorica, 6 (1986), pp. 221–233.

[2] H. Behrendt and A. Brandstädt, *Domination and the Use of Maximum Neighbourhoods*, Technical Report SM-DU-204, University of Duisburg, 1992.

[3] C. Berge, *Hypergraphs*, North–Holland, Amsterdam, The Netherlands, 1989.

[4] A. Brandstädt, V. D. Chepoi, and F. F. Dragan, *The Algorithmic Use of Hypertree Structure and Maximum Neighbourhood Orderings*, Technical Report SM-DU-244, University of Duisburg, 1994; in Graph-Theoretic Concepts in Computer Science, Lecture Notes in Computer Science 903, E. W. Mayr, G. Schmidt, and G. Tinhofer, eds., Springer-Verlag, Berlin, New York, 1995, pp. 65–80.

[5] A. Brandstädt, F. F. Dragan, V. D. Chepoi, and V. I. Voloshin, *Dually Chordal Graphs*, Technical Report SM-DU-225, University of Duisburg 1993; Graph-Theoretic Concepts in Computer Science, Lecture Notes in Computer Science 790, J. van Leeuwen, ed., Springer-Verlag, Berlin, New York, 1994, pp. 237–251.

[6] P. Buneman, *A characterization of rigid circuit graphs*, Discrete Math., 9 (1974), pp. 205–212.

[7] G. J. Chang, *Labeling algorithms for domination problems in sun–free chordal graphs*, Discrete Appl. Math., 22 (1988/89), pp. 21–34.

[8] G. J. Chang, M. Farber, and Z. Tuza, *Algorithmic aspects of neighbourhood numbers*, SIAM J. Discrete Math., 6 (1993), pp. 24–29.

[9] G. J. Chang and G. L. Nemhauser, *The k-domination and k-stability problems on sun-free chordal graphs*, SIAM J. Alg. Disc. Meth., 5 (1984), pp. 332–345.

[10] G. A. Dirac, *On rigid circuit graphs*, Abh. Math. Sem. Univ. Hamburg, 25 (1961), pp. 71–76.

[11] F. F. Dragan, *Domination and packings in triangulated graphs*, Metody Diskret. Analiz., 51 (1991), pp. 17–36. (In Russian.)

[12] F. F. Dragan, C. F. Prisacaru, and V. D. Chepoi, *The location problem on graphs and the Helly problem*, Disk. Math., 4 (1992), pp. 67–73. (In Russian.) (The full version appeared as preprint: F.F. Dragan, C.F. Prisacaru, and V.D. Chepoi, r-Domination and p-Center Problems on Graphs: Special Solution Methods and Graphs for Which This Method is Usable, Kishinev State University, preprint MoldNIINTI, N. 948–M88, 1987 (in Russian).)

[13] M. Farber, *Characterizations of strongly chordal graphs*, Discrete Math., 43 (1983), pp. 173–189.

[14] M. Farber, *Domination, independent domination and duality in strongly chordal graphs*, Discrete Appl. Math., 7 (1984), pp. 115–130.

[15] D. R. Fulkerson and O. R. Gross, *Incidence matrices and interval graphs*, Pacific J. Math., 15 (1965), pp. 835–855.

[16] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[17] S. H. Hwang and G. J. Chang, *k-Neighbourhood Covering and Independence Problems*, DIMACS Report 93-03, DIMACS Center, Rutgers University, New Brunswick, NJ, 1993.

[18] J. Lehel and Z. Tuza, *Neighbourhood perfect graphs*, Discrete Math., 61 (1986), pp. 93–101.

[19] A. Lubiw, *Doubly lexical orderings of matrices*, SIAM J. Comput., 16 (1987), pp. 854–879.

[20] M. Moscarini, *Doubly chordal graphs, Steiner trees and connected domination*, Networks, 23 (1993), pp. 59–69.

[21] R. Paige and R.E. Tarjan, *Three partition refinement algorithms*, SIAM J. Comput. 16 (1987), pp. 973–989.

[22] P. J. Slater, *R-domination in graphs*, J. Assoc. Comput. Mach., 23 (1976), pp. 446–450.

[23] J. P. Spinrad, *Doubly lexical ordering of dense* 0–1*–matrices*, Inform. Process. Lett., 45 (1993), pp. 229–235.

[24] J. L. Szwarcfiter and C. F. Bornstein, *Clique graphs of chordal and path graphs*, SIAM J. Discrete Math., 7 (1994), pp. 331–336.

[25] 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., 13 (1984), pp. 566–579.

SIAM J. DISCRETE MATH.
Vol. 10, No. 1, pp. 128–145, February 1997

ON COSET WEIGHT DISTRIBUTIONS OF THE 3-ERROR-CORRECTING BCH-CODES[*]

PASCALE CHARPIN[†] AND VICTOR ZINOVIEV[‡]

Abstract. We study the coset weight distributions of the 3-error-correcting binary narrow-sense BCH-codes and of their extensions, whose lengths are, respectively, $2^m - 1$ and 2^m, m odd. We prove that all weight distributions are known as soon as those of the cosets of minimum weight 4 of the extended code are known. We point out that properties of the cosets which are orphans yield interesting properties on the other cosets. We describe the classes of cosets which are equivalent under the affine permutations. At the end we produce significant numerical results, proving that the number of distinct weight distributions of cosets increases with the length of the codes.

Key words. BCH-codes, uniformly packed codes, coset, coset weight distribution, orphan, affine permutation

AMS subject classifications. 94B15, 51E30

PII. S089548019529443X

1. Introduction.

This paper was initiated by the papers of Camion, Courteau, Fournier, and Kanetkar [6]; Camion, Courteau, and Montpetit [7]; and Charpin [9], [10]. Charpin showed in [10] that there are eight distinct weight distributions of cosets of 2-error-correcting binary primitive BCH-codes of length $2^m - 1$, m even, and of length 2^m for the extended such codes. For the length $2^m - 1$, m odd, it is well known [3], [20] that there are four such distinct weight distributions. We examine here the coset weight distributions of the 3-error-correcting binary narrow-sense BCH-codes of length $2^m - 1$ with m odd, also extended or not. The results of this paper were announced in [11].

We denote by B the 3-error-correcting BCH-code and by $\widehat{B}$ its extension. For length 32 the coset weight distribution of $\widehat{B}$ was given by Camion, Courteau, and Montpetit [7]; this code is in fact the self-dual Reed–Muller code $[32, 16, 8]$ and there are eight distinct weight distributions for its cosets. Our main result is that *the number of weight distributions of cosets of $\widehat{B}$ (respectively, of B) increases with the value of m.* Of course, we suppose that this property holds also when m is even, although we do not study this case here. At any rate, we prove that the code $\widehat{B}$ gives us an example of an infinite class of codes whose dual distance is constant while the number of distinct lines in the distance matrix increases with the length.

In section 2, we present the fundamental equations which give as solutions the coefficients of the distance matrices of B and $\widehat{B}$. Throughout the equations $(A.i)$ and $(E.i)$, what is easy and what is hard appear clearly, and the next sections are in fact a precise explanation of both aspects.

We begin in section 3 with the easy cases. They are globally the cosets of weight 1, 2, 3, and 5. We don't know all about the cosets of B of weight 3 and 5, but we prove that any unsolved problem about these cosets is an unsolved problem about the cosets of $\widehat{B}$ of weight 4 and 6. We consider these last cases as the hard cases. In

[*] Received by the editors November 8, 1995; accepted for publication (in revised form) March 15, 1996.

http://www.siam.org/journals/sidma/10-1/29443.html

[†] Institut National de Recherche en Informatique et en Automatique, Codes, Domaine de Voluceau-Rocquencourt, BP 105-78153, Le Chesnay, France (pascale.charpin@inria.fr).

[‡] Institute for Problems of Information Transmission of the Russian Academy of Sciences, Bol'shoi Karetnyi 19, Moscow 101447, Russia.

section 4 we study the action of affine permutations on cosets of $\widehat{B}$. It is natural to do that because it is well known that the code $\widehat{B}$ is invariant under these permutations. We characterize the classes of equivalent cosets by their syndromes, and we give some properties about the cosets of weight 4. The cosets of weight 4 and 6 are studied in section 5. We point out the significant role of the cosets of $\widehat{B}$ which are orphans, taking here the terminology of [5]. In section 6 we summarize our results, showing clearly that our problem is reduced to the study of the weight distributions of cosets of $\widehat{B}$ of weight 4. By using the classification of section 4, we were able to compute the full weight distribution for length 128. That is given by Table 5 in section 7. We found 12 distinct weight distributions for the cosets of $\widehat{B}$. Moreover, we found at least 18 distinct weight distributions for the length 512. At the end we give several conjectures.

The *distance* and the *weight* are the Hamming distance and the Hamming weight. The weight of any code word x is denoted by $wt(x)$, and the distance between any two code words x and y is denoted by $d(x,y)$. Denote by K the Galois field of order 2. Let C be any binary code of length n. Recall that the *covering radius* of C, generally denoted by ρ, is the following distance:

$$\rho = \max_{x \in K^n} \ \min_{c \in C} \{ \ d(x,c) \ \}.$$

Let $D = x + C$ be a coset of C. *The weight of the coset* D is the minimum weight of the code words of D. *A leader* of D is a code word of D of minimum weight.

2. The fundamental equations. Let C be any code of length n over K and let ρ be its covering radius. We will say that such a code is *uniformly packed*, in the sense of [3], if there exist rational numbers $\alpha_0, \ \ldots, \ \alpha_\rho$ such that for any $v \ \in \ K^n$

$$\sum_{k=0}^{\rho} \alpha_k \ f_k(v) \ = \ 1, \tag{1}$$

where $f_k(v)$ is the number of code words at distance k from v. Let B denote here the 3-error-correcting primitive binary BCH-code of length $n = 2^m - 1$, where m is odd, and let $B^\perp$ denote as usual the dual code of B. The minimal distance of B is $d = 7$. It was shown by Kasami [17] that the *external distance* of B, i.e., the number of nonzero weights in $B^\perp$, is $s = 5$ (see also [19], p. 669). According to the well-known result due to Delsarte [12], we have the following inequality for the covering radius of B:$\rho \ \leq \ 5$. But on the other hand, we know from the result of Gorenstein, Peterson, and Zierler [14] that for these codes $\rho \ \geq \ 5$. Hence we have $\rho \ = \ 5$ for the code B. Note that this result was obtained by Helleseth [15], who proved even more: all binary 3-error-correcting BCH-codes have covering radius 5 (essential steps in this result also belong to Assmus and Mattson [1] and van der Horst and Berger [16]). Now we use the following result from the paper of Bassalygo and Zinoviev [4, Theorem 1]: *the code C is a uniformly packed code* (*in the sense of* [3]) *if and only if the covering radius ρ of C is equal to the external distance s: $\rho \ = \ s$.* Therefore B is a uniformly packed code in the sense of [3]. Note that Goethals and Van Tilborg [13] have previously showed that the code B is a *uniformly packed code of order* $j = 2$ (see [13, 21]). From this last paper we have the following parameters α_i for the code B:

$$\begin{aligned} &\alpha_0 = \alpha_1 = 1, \\ &\alpha_2 = \alpha_3 = -120/(n-1)(n-7), \\ &\alpha_4 = \alpha_5 = 120/(n-1)(n-7). \end{aligned} \tag{2}$$

Now let $\widehat{B}$ be the 3-error-correcting primitive binary extended BCH-code of length $N = 2^m$, where m is odd; $\widehat{B}$ is obtained from B by overall parity check. Assume that the position we add to the code words of B is always the first position of $\widehat{B}$. The minimal distance of $\widehat{B}$ is $d = 8$, of course. Now we can use the following result [4, Theorem 2]: *an extension of a binary uniformly packed code with parameters α_i, $i \in [0,\rho]$, is a uniformly packed code if and only if the parameters α_i satisfy*

$$\alpha_{\rho-2i} \;=\; \alpha_{\rho-2i-1},\;\; i = 0,1,\dots,[(\rho-1)/2],$$

where $[a]$ denotes the integer part of a. Applying this to the code B, the condition above becomes $\alpha_5 = \alpha_4$, $\alpha_3 = \alpha_2$, and $\alpha_1 = \alpha_0$. So we deduce from (2) that the code $\widehat{B}$ is uniformly packed with covering radius 6. Note that the external distance of the code $\widehat{B}$ (respectively, of B) is equal to its covering radius. Then, by applying the general result of Assmus and Pless, *the weight distribution of cosets of weight 5 in B is uniquely determined, as are the weight distributions of cosets of weight 5 and 6 in $\widehat{B}$* [2, Corollary 1–2].

From now on, the notation for the parameters of codes B and $\widehat{B}$ will be as follows: we will use the same symbols for both codes, but for $\widehat{B}$ all the corresponding symbols will have a hat. The parameters $\widehat{\alpha}_i$ of the code $\widehat{B}$ are connected with the parameters α_i. This connection is given by [4, Theorem 2]. That is,

$$\widehat{\alpha}_{\rho-2i} \;=\; \alpha_{\rho-2i},\quad i = 0,1,\dots,[\rho/2]$$

and for $i = 0,1,\dots,[(\rho+1)/2]$,

$$\widehat{\alpha}_{\rho-2i+1} \;=\; ((\rho+1-2i)\alpha_{\rho-2i} \;+\; (n-\rho+2i)\alpha_{\rho-2i+2}\;)/(n+1),$$

where by convention $\alpha_{-1} = \alpha_{\rho+1} = \alpha_{\rho+2} = 0$. We have

$$\begin{array}{lll} & \widehat{\alpha}_0 = \widehat{\alpha}_1 = 1, & \widehat{\alpha}_2 = 2(N-68)/N(N-8), \\ (3) & \widehat{\alpha}_3 = -120/(N-2)(N-8), & \widehat{\alpha}_4 = 120/N(N-2), \\ & \widehat{\alpha}_5 = -\widehat{\alpha}_3, & \widehat{\alpha}_6 = 720/N(N-2)(N-8). \end{array}$$

Recall that $N = 2^m$ denotes here the length of the code $\widehat{B}$.

Let D be any coset of B. Recall that the *weight of* D is the minimum weight of the code words of D. Since the covering radius of B is 5, the weight i of B is in the range $[0,5]$. We will denote by $\mu_{i,j}$ the number of code words of weight j in such a coset of weight i:

$$\mu_{i,j} = \texttt{card}\;\{\; x \in D \mid wt(x) = j \;\}.$$

Similarly, we will denote by $\widehat{\mu}_{i,j}$ the number of code words of weight j in a coset of $\widehat{B}$ of weight i, $i \in [0,6]$.

For a coset D with weight distribution

$$\mu_{i,i},\;\mu_{i,i+1},\dots,\;\mu_{i,n}$$

we denote by $A_i(x)$ the weight polynomial of D:

$$A_i(x) \;=\; \sum_{k=i}^{n} \mu_{i,k}\; x^k. \tag{4}$$

To write out a general expression for the polynomial $A_i(x)$ we need some results from [3], which we give, for simplicity, only for the binary case. First denote by $P_u(n,\xi)$ the Krawtchouk polynomial of degree u:

$$P_u(n,\xi) = \sum_{j=0}^{u} (-1)^{u-j} \binom{n-\xi}{j} \binom{\xi}{u-j},$$

where

$$\binom{a}{b} \;=\; \frac{a(a-1)\dots(a-b+1)}{b!}$$

for any real a. Lloyd's type theorem for the uniformly packed codes asserts (Theorem 1 in [3]) that the existence of a uniformly packed code C of length n with the parameters $\alpha_i, i=0,1,\dots,\rho$, implies that the Lloyd polynomial $L_\rho(n,\xi)$,

$$L_\rho(n,\xi) \;=\; \sum_{i=0}^{\rho} \alpha_i \; P_i(n,\xi),$$

has ρ distinct integer roots between 0 and n. Denote by ξ_i the ith root of $L_\rho(n,\xi)$, where $i=0,1,\dots,\rho$. Now suppose that D is an arbitrary coset of C of weight i with the weight polynomial $A(x)$ of type (4). We want to know the weight distribution of D (or, in other words, to know the coefficients of $A_i(x)$).

Theorem 2 in [3] gives us the following result: *the weight polynomial $A_i(x)$ of a coset (of weight i) of a uniformly packed code C, with the roots ξ_j of the Lloyd polynomial $L_\rho(n,\xi)$, might be written in the following general form:*

$$A_i(x) = \frac{|C|(1+x)^n}{2^n} + \sum_{j=1}^{\rho} c_{i,j}(1+x)^{n-\xi_j}(1-x)^{\xi_j},$$

where $|C|$ is the cardinality of the code C and $c_{i,j}$ are constants depending on the initial known coefficients of $A_i(x)$ and therefore determined by solving the corresponding system of linear equations. So to know the weight polynomial $A_i(x)$ of C we must know any ρ numbers $\mu_{i,j}$ for $j \in [0,n]$ enough to find the unknown values $c_{i,j}$ from the corresponding equations.

Now we return to our BCH-codes B and $\widehat{B}$. The determination of the coset weight distribution of B is reduced to the resolution of the following equations, considered separately. In other words, if we consider the weight distribution of the coset of weight i, then we use the equation $(A.i)$:

$$\begin{array}{ll}
(A.1) & \alpha_1\;\mu_{1,1} \;=\; 1,\\
(A.2) & \alpha_2\;\mu_{2,2} \;+\; \alpha_5\;\mu_{2,5} \;=\; 1,\\
(A.3) & \alpha_3\;\mu_{3,3} \;+\; \alpha_4\;\mu_{3,4} \;+\; \alpha_5\;\mu_{3,5} \;=\; 1,\\
(A.4) & \alpha_4\;\mu_{4,4} \;+\; \alpha_5\;\mu_{4,5} \;=\; 1,\\
(A.5) & \alpha_5\;\mu_{5,5} \;=\; 1,
\end{array}$$

where the numbers α_i are given above by (2). These equations are obtained from (1) for each weight $i \in [1,5]$ for the case when the vector v is a zero vector. Each equation

$(A.i)$ corresponds to the weight distributions of cosets of minimum weight i, implying $\mu_{i,j} = 0$ for $j < i$. Moreover, since the minimum weight of B is 7, the sum of two weights in a given coset cannot be less than 7.

Now consider the corresponding equations for the code $\widehat{B}$. By the definition of the extension, a coset of $\widehat{B}$ has either only even weights or only odd weights. Therefore, in the same manner as we obtained the equations $(A.i)$, we obtain from (1) the equations $(E.i)$ corresponding to the weights $i \in [1,6]$ of the cosets of $\widehat{B}$:

$$\begin{array}{ll}
(E.1) & \widehat{\alpha}_1\ \widehat{\mu}_{1,1} \ = \ 1, \\
(E.2) & \widehat{\alpha}_2\ \widehat{\mu}_{2,2} \ + \ \widehat{\alpha}_6\ \widehat{\mu}_{2,6} \ = \ 1, \\
(E.3) & \widehat{\alpha}_3\ \widehat{\mu}_{3,3} \ + \ \widehat{\alpha}_5\ \widehat{\mu}_{3,5} \ = \ 1, \\
(E.4) & \widehat{\alpha}_4\ \widehat{\mu}_{4,4} \ + \ \widehat{\alpha}_6\ \widehat{\mu}_{4,6} \ = \ 1, \\
(E.5) & \widehat{\alpha}_5\ \widehat{\mu}_{5,5} \ = \ 1, \\
(E.6) & \widehat{\alpha}_6\ \widehat{\mu}_{6,6} \ = \ 1.
\end{array}$$

From the results of Kasami [17] and Bassalygo and Zinoviev [4] we have all the roots $\widehat{\xi}_i$ of the Lloyd polynomial $\widehat{L}_6(N,\xi)$ for the code $\widehat{B}$ (these roots are exactly the values of nonzero weights in the dual code $\widehat{B}^{\perp}$):

$$\begin{array}{llll}
\widehat{\xi}_1 \ = \ N/2 \ - \ \sqrt{2N}, & \widehat{\xi}_2 \ = \ N/2 \ - \ \sqrt{N/2}, \\
\widehat{\xi}_3 \ = \ N/2, & \widehat{\xi}_4 \ = \ N/2 \ + \ \sqrt{N/2}, \\
\widehat{\xi}_5 \ = \ N/2 \ + \ \sqrt{2N}, & \widehat{\xi}_6 \ = \ N.
\end{array}$$

Note that the five roots of the Lloyd polynomial $L_5(n,\xi)$ for the code B are the first five roots $\widehat{\xi}_i, i \in [1,5]$, of $\widehat{L}_6(N,\xi)$. This is so because the all-one vector, which corresponds to the root $\widehat{\xi}_6$, cannot belong to the code $B^{\perp}$.

Now we give some definitions and notation which we will use in the next sections.

Let $v \in K^n$, $v = (v_1, \dots,\ v_n)$. The *support* of v is

$$\mathrm{supp}(v) = \{\ \ell \mid v_\ell \neq 0\ \}.$$

Note that the Hamming weight $wt(v)$ of v is equal to the cardinality of the support of v.

We will use here the terminology of [5], where special cosets, so-called *orphans*, are introduced.

DEFINITION 2.1. *Let C be an arbitrary linear code C of length n and let D be a coset of C of weight i. Let D' be the coset*

$$D' = D \ + \ v^{(j)},$$

where $v^{(j)}$ denotes a binary vector with exactly one nonzero position at the jth coordinate.

If the weight of D' is $i-1$, then D' is said to be a child of D.

If the weight of D' is $i+1$, then D' is said to be a parent of D.

The coset D is said to be an orphan if and only if it has no parent. In other words, an orphan of C is a coset D with the following property:

$$\bigcup_{v\ is\ a\ leader\ of\ D} \mathrm{supp}(v) = \{\ 1, \dots, n\ \}.$$

Notation. From now on let us denote by $\mathcal{D}$ (respectively, by $\widehat{\mathcal{D}}$) the full set of the cosets of B (respectively, of $\widehat{B}$). We will denote by $\mathcal{D}_i$ (respectively, by $\widehat{\mathcal{D}}_i$) the subset of $\mathcal{D}$ (respectively, of $\widehat{\mathcal{D}}$) which consists of all the cosets of weight i.

The number of cosets of B will be denoted by Γ and the number of such cosets of minimum weight i will be denoted by $\Gamma(i)$. Similarly, for the extended code $\widehat{B}$, a notation is as follows:

$$\widehat{\Gamma} \;=\; |\widehat{\mathcal{D}}| \quad \text{and} \quad \widehat{\Gamma}(i) \;=\; |\widehat{\mathcal{D}}_i|.$$

3. Cosets weight distribution: The easy cases. Since the dimension of both codes B and $\widehat{B}$ is $2^m - 3m - 1$, $m \geq 5$, we obviously obtain

$$\Gamma = 2^{3m} \quad \text{and} \quad \widehat{\Gamma} = 2^{3m+1}.$$

The weight distribution of B is known, due to Kasami, who in [17] gave the weight distribution of the dual of B. In fact, we use here the table given in [19, p. 669]; it is the weight distribution of $B^{\perp}$. Since we also need the weight distribution of $\widehat{B}$, we give the weight distribution of the dual code in Table 1.

TABLE 1
The weight distribution of the dual of the binary 3*-error-correcting extended BCH-code of length* 2^m*, m odd.*

Weights	Number of code words
0	1
$2^{m-1} \pm 2^{(m+1)/2}$	$2^{m-3}(2^m-1)(2^{m-1}-1)/3$
$2^{m-1} \pm 2^{(m-1)/2}$	$2^{m-1}(2^m-1)(5.2^{m-1}+4)/3$
2^{m-1}	$(2^m-1)(5.2^{2m-1}+7.2^{m-2}(2^{m-1}-1)+2^{m+2}+6)/3$
2^m	1

Remark. Recall that a *tactical configuration* $T(n,w,\ell,\beta)$ is a set of binary vectors of length n and weight w such that any ℓ, $1 \leq \ell \leq w$, positions are simultaneously occupied by ones in precisely β vectors of $T(n,w,\ell,\beta)$. If $\beta = 1$, a configuration $T(n,w,\ell,1)$ is called a *Steiner system* and is denoted by $S(n,w,\ell)$.

Let B_7 be the set of code words of weight 7 in B and $\widehat{B}_8$ be the set of code words of weight 8 in $\widehat{B}$. Using equation (1) for arbitrary vectors v of weights 2 and 3 we have immediately the following: *the set* $\widehat{B}_8$ *is a tactical configuration* $T(N,8,3,\beta)$ *and the set* B_7 *is a tactical configuration* $T(n,7,2,\beta)$*, where*

$$\beta \;=\; \frac{1-\widehat{\alpha}_3}{\widehat{\alpha}_5} \;=\; \frac{(N-2)(N-8)}{120} \;+\; 1. \tag{5}$$

This result can be also deduced from Theorem 3 in [4].

3.1. Cosets of minimum weights 1, 2, and 3. Since the minimum distance of codes B and $\widehat{B}$ are, respectively, 7 and 8, any coset of weight i, $1 \leq i \leq 3$, has only one code word of weight i. So the number of such cosets of weight i is exactly the number of code words of weight i in the ambient space. That is, for cosets of B and $\widehat{B}$

$$\Gamma(1) = n, \quad \Gamma(2) = n(n-1)/2, \quad \text{and} \quad \Gamma(3) = n(n-1)(n-2)/6, \tag{6}$$

$$\widehat{\Gamma}(1) = N, \quad \widehat{\Gamma}(2) = N(N-1)/2, \quad \text{and} \quad \widehat{\Gamma}(3) = N(N-1)(N-2)/6. \tag{7}$$

The condition $\widehat{\mu}_{i,i} = 1$ for $i \in [1, 3]$ immediately gives us the solution of the corresponding equations $(E.i)$. We then obtain the values of $\widehat{\mu}_{2,6}$ and $\widehat{\mu}_{3,5}$. Similarly, the condition $\mu_{i,i} = 1$ for $i \in [1, 2]$ immediately gives us the solution of the corresponding equations $(A.i)$. We can then obtain the value of $\mu_{2,5}$. Note that $\mu_{2,5}$ and $\widehat{\mu}_{3,5}$ are also given by the remark above. These results can be summarized as follows.

Proposition 3.1. *There is only one coset weight distribution for the cosets of B of weight* 1 *and* 2. *The number of code words of weight* 5 *in the coset of weight* 2 *is* $\mu_{2,5} = \beta$ (*see* (5)).

There is only one coset weight distribution for the cosets of $\widehat{B}$ of weight 1, 2, *and* 3. *The number of code words of weight* 6 *in the coset of weight* 2 *is*

$$\widehat{\mu}_{2,6} = \frac{1 - \widehat{\alpha}_2}{\widehat{\alpha}_6} = \frac{(N-2)(N^2 - 10N + 136)}{720}.$$

The number of code words of weight 5 *in the coset of weight* 3 *is* $\widehat{\mu}_{3,5} = \beta$ (*see* (5)).

Finally, we cannot describe the set $\mathcal{D}_3$ of cosets of B of weight 3; we only know its cardinality. Moreover, according to (2), by using $(A.2)$ and $(A.3)$ we can state the following relation:

$$\mu_{3,4} \; + \; \mu_{3,5} \; = \; \mu_{2,5}, \tag{8}$$

where $\mu_{2,5}$ is known to be equal to β. Note also that $\mu_{2,5} = \widehat{\mu}_{3,5}$. Hence we can conclude that *to describe $\mathcal{D}_3$ is equivalent to describing $\widehat{\mathcal{D}}_4$*. Indeed, a coset of $\mathcal{D}_3$ can be seen as a shortened coset of $\widehat{\mathcal{D}}_4$, with

$$\mu_{3,4} = \widehat{\mu}_{4,4} - 1.$$

Such a coset of $\widehat{\mathcal{D}}_4$ must have a leader which has zero in its first position (this position is the parity check position of $\widehat{B}$). We will explain in section 4 that any coset of $\widehat{\mathcal{D}}_4$ is equivalent to such a coset.

3.2. Cosets of minimum weight 5. All cosets of $\mathcal{D}_5$ have the same weight distribution—it is immediate from $(A.5)$(see also [1]). However, we are not able to give the cardinality of $\mathcal{D}_5$; we only can say that it is equal to the cardinality of $\widehat{\mathcal{D}}_6$.

Proposition 3.2. *There is only one weight distribution for the cosets of $\mathcal{D}_5$. Any coset of $\mathcal{D}_5$ is an orphan, and it contains*

$$\mu_{5,5} \; = \; \frac{1}{\alpha_5} \; = \; \frac{(n-1)(n-7)}{120}$$

code words of weight 5. *Moreover, the cardinality of $\mathcal{D}_5$ is equal to the number of cosets of $\widehat{B}$ of weight* 6*:*

$$\Gamma(5) \; = \; \widehat{\Gamma}(6).$$

Proof. The value $\mu_{5,5}$ follows from $(A.5)$. From Definition 2.1, we know that an orphan is a coset without parent. Since the covering radius of B is 5, it is clear that any coset $G \in \mathcal{D}_5$ is an orphan. Now for any coset $H \in \widehat{\mathcal{D}}_6$, we obtain a coset $G \in \mathcal{D}_5$ by deleting one position of H. We always delete the first position, which corresponds to the overall parity checking position of $\widehat{B}$. Two such cosets G and G' are distinct, as soon as we got two distinct cosets H and H'. Actually, this correspondence is one-to-one: by the definition of the extension, two distinct cosets of $\mathcal{D}_5$ cannot give the same extension. So $\Gamma(5) = \widehat{\Gamma}(6)$. □

Now for $\widehat{\mathcal{D}}_5$, equations $(E.i)$ involve a full description. Moreover, we will end this section by explaining some links between $\widehat{\mathcal{D}}_5$ and $\widehat{\mathcal{D}}_4$.

PROPOSITION 3.3. *There are*

$$\widehat{\Gamma}(5) = N(N-1)(5N+8)/6$$

distinct cosets of $\widehat{B}$ *of weight* 5*. All of these cosets have the same weight distribution and each of them contains*

$$\widehat{\mu}_{5,5} \; = \; (N-2)(N-8)/120 \tag{9}$$

vectors of weight 5*. Note that* $\widehat{\mu}_{5,5} = \mu_{5,5}$*.*

Proof. All cosets of minimum weight 3 have the same weight polynomial. We know from $(E.3)$ that the number of the code words of weight 5 in the coset of minimum weight 3 is

$$\widehat{\mu}_{3,5} = \beta,$$

where β is defined in (5). From the equation $(E.5)$ we have $\widehat{\mu}_{5,5} \; = \; 1/\widehat{\alpha}_5$. Taking into account the value of $\widehat{\alpha}_5$ in (3) we obtain (9). Now the total number of binary vectors of length N and weight 5 is

$$T \; = \; \binom{N}{5},$$

and we have

$$T = \widehat{\Gamma}(5)\; \widehat{\mu}_{5,5} \; + \; \widehat{\Gamma}(3)\; \widehat{\mu}_{3,5}.$$

Then we can compute $\widehat{\Gamma}(5)$ using the value of $\widehat{\Gamma}(3)$ given by the equation (7). □

PROPOSITION 3.4. *Let* $G \; \in \; \widehat{\mathcal{D}}_5$*, let* F *be a child of* G*, that is,*

$$F \; = \; G \; + \; v^{(j)}, \quad F \in \widehat{\mathcal{D}}_4$$

for some $j \; \in \; \{\; 1, \ldots, \; N \;\}$*, and let* $k_j(G)$ *denote the weight of the* j*th column of the binary matrix formed by the leaders of* G*. Then the weight distribution of* F *is defined by* $\widehat{\mu}_{4,4} \; = \; k_j(G)$*, where* $k_j(G) \; < \; N/4$*.*

Proof. Consider the jth column of the matrix formed by all the leaders of G. So we have $k_j(G)$ vectors $u_s, \; s = 1, \ldots, k_j(G)$, which have "1" at jth position. Then the coset F has weight 4 and the $k_j(G)$ vectors

$$u_s \; + \; v^{(j)}, \; s \; = \; 1, \ldots, k_j(G)$$

are the only vectors in F that have weight 4. Hence, such a coset F is not an orphan since it has some parent. That gives the inequality at the statement, completing the proof. □

Note that any $F \in \widehat{\mathcal{D}}_4$, which is not an orphan, is a child of some coset of $\widehat{\mathcal{D}}_5$. In this section we have proved that each unsolved problem on cosets of B can be seen as an unsolved problem on cosets of $\widehat{B}$. We will see in section 5 that the general problem we treat here is reduced to the determination of the weight distribution of cosets of $\widehat{\mathcal{D}}_4$, more precisely to the determination of the possible values of $\widehat{\mu}_{4,4}$. The proposition above suggests an equivalent point of view: we know all about the weight distribution of cosets of $\widehat{\mathcal{D}}_5$, but we do not know, for such a coset, how much leaders have for one given position in its support.

4. Equivalent cosets. At the end of this paper we will give numerical results on the coset weight distributions of the code $\widehat{B}$ for $m = 7$ and $m = 9$. We obtain these results with the aid of a computer; however, the computation was possible because of some properties on the equivalent cosets. In this section we want to present these properties and their corollaries.

Let K and $\mathbf{G}$ be, respectively, the fields of order 2 and of order N. Since we treat primitive binary codes, we can consider extended codes as K-subspaces in the group algebra of the additive group of $\mathbf{G}$. This representation is more convenient when we want to describe the permutations on cosets which conserve the code $\widehat{B}$. So, in this section, the ambient space is the group algebra $\mathcal{A} = K[\{\mathbf{G}, +\}]$ and a code word is a formal sum:

$$x = \sum_{g \in \mathbf{G}} x_g X^g, \; x_g \in K.$$

Recall that the code $\widehat{B}$ is invariant under the affine permutations on $\mathbf{G}$. That means that any permutation

$$\sigma_{u,v} \;:\; \sum_{g \in \mathbf{G}} x_g X^g \quad \longmapsto \quad \sum_{g \in \mathbf{G}} x_g X^{ug+v} \;, \;\; u \neq 0, \; u \in \mathbf{G}, \; v \in \mathbf{G}$$

is an automorphism of the code $\widehat{B}$ [18]. Therefore, for any coset $D = x + \widehat{B}$, we have obviously $\sigma_{u,v}(D) = \sigma_{u,v}(x) + \widehat{B}$. Let us define, for any integer $s \in [0, N-1]$, the mapping $\phi_s(x)$,

$$\phi_s : A \;\rightarrow\; \mathbf{G}, \; \phi_s(x) \;=\; \sum_{g \in \mathbf{G}} x_g g^s, \tag{10}$$

where by convention $\phi_0(x) = \sum_{g \in \mathbf{G}} x_g$.

DEFINITION 4.1. *The extended* 3*-error-correcting BCH-code* $\widehat{B}$ *is the following subspace of* $\mathcal{A}$*:*

$$\widehat{B} = \{ \; x \mid \phi_s \; (x) = 0, \; s \in \{0\} \cup cl(1) \cup cl(3) \cup cl(5) \; \},$$

where $cl(t)$ *is the cyclotomic coset of* 2 *(mod* n*) containing* t *and* $m \geq 5$*. So the dimension of* $\widehat{B}$ *equals* $N - 3m - 1$*, where* $N = 2^m$ *and* $n = N - 1$*.*

DEFINITION 4.2. *There are* 2^{3m+1} *cosets of* $\widehat{B}$*. Each coset* $x + \widehat{B}$ *is uniquely defined by its so-called syndrome:*

$$S(x) = (\; \phi_0(x), \; \phi_1(x), \; \phi_3(x), \; \phi_5(x) \;).$$

When $\phi_0(x) = 0$*, all weights of the coset are even and we will say that the coset is even*; *otherwise, all weights of the coset are odd and we will say that the coset is odd.*

We will see that our problem is in fact the determination of the weight distributions of the cosets of $\widehat{B}$ of weight 4. Moreover, the odd cosets can be studied simply from the even cosets. For this reason we now study even equivalent cosets. Recall that we denote by $\widehat{\mathcal{D}}$ the set of all cosets of $\widehat{B}$.

LEMMA 4.3. *Let us define the following subsets of* $\widehat{\mathcal{D}}$*:*

$$\mathcal{B}_1 = \{ \; x + \widehat{B} \mid \phi_0(x) = 0 \;\;\textit{and}\;\; \phi_1(x) \neq 0 \; \}, \tag{11}$$

$$\mathcal{B}_2 = \{ \; x + \widehat{B} \mid \phi_0(x) = 0 \;\;\textit{and}\;\; \phi_1(x) = 0 \; \}, \tag{12}$$

$$\mathcal{B}_3 = \{\ x + \widehat{B} \mid \phi_0(x) = \phi_1(x) = \phi_3(x) = 0\ \}. \tag{13}$$

Then $\mathcal{B}_1$ is contained in the Reed–Muller code $R(m-1,m)$ of order $m-1$ and not contained in $R(m-2,m)$; $\mathcal{B}_2$ is contained in $R(m-2,m)$; $\mathcal{B}_3$ is contained in the extended 2-error correcting BCH-code.

Proof. Recall the definition of the Reed–Muller code of length N and order r, denoted by $R(r,m)$. For any $t \in [0,n]$ let us define the 2-weight of t to be $\omega_2(t) = \sum_{i=0}^{m-1} t_i$, where

$$t \ = \ \sum_{i=0}^{m-1} t_i 2^i$$

is the binary expansion of t. Let I_r be the set of integers from $[0,n]$ such that $\omega_2(t) < m-r$. The code $R(r,m)$ is the set of code words x satisfying $\phi_t(x) = 0$ for all $t \in I_r$. We have $I_{m-1} = \{0\}$ and $I_{m-2} = \{0\} \cup cl(1)$. The extended 2-error correcting BCH-code is the set of code words satisfying $\phi_t(x) = 0$ for t in $\{0\} \cup cl(1) \cup cl(3)$. □

LEMMA 4.4. *Let u and v be in $\mathbf{G}$, where $u \neq 0$. Consider a coset $x + \widehat{B}$ whose syndrome is $S(x) = (0, \delta, \gamma, \lambda)$. Then the syndrome of the coset $\sigma_{u,v}(x) + \widehat{B}$ is as follows:*

$$S(\sigma_{u,o}(x)) = (0,\ u\delta,\ u^3\gamma,\ u^5\lambda) \tag{14}$$

and

$$S(\sigma_{1,v}(x)) = (0,\ \delta,\ \gamma + \delta v^2 + \delta^2 v,\ \lambda + \delta v^4 + \delta^4 v). \tag{15}$$

Proof. For any code word $x = \sum_{g\in\mathbf{G}} x_g X^g$, we have

$$\phi_t(\sigma_{u,o}(x)) = \sum_{g\in\mathbf{G}} x_g (ug)^t = u^t \phi_t(x).$$

Thereby (14) follows immediately. Now $\phi_t(\sigma_{1,v}(x)) = \phi_t(X^v x)$. So, for $t = 1, 3$ and 5 we obtain

$$\begin{aligned}
\phi_1(X^v x) &= \sum_{g\in\mathbf{G}} x_g (g+v) = \phi_1(x) + v\ wt(x) = \phi_1(x) = \delta,\\
\phi_3(X^v x) &= \sum_{g\in\mathbf{G}} x_g (g+v)^3 = \phi_3(x) + v^2\phi_1(x) + v(\phi_1(x))^2 = \gamma + \delta v^2 + \delta^2 v,\\
\phi_5(X^v x) &= \sum_{g\in\mathbf{G}} x_g (g+v)^5 = \phi_5(x) + v^4\phi_1(x) + v(\phi_1(x))^4 = \lambda + \delta v^4 + \delta^4 v,
\end{aligned}$$

where the sums are computed modulo 2. Then we obtain (15), therefore completing the proof. □

Let us define an equivalence relation Δ on the set $\widehat{\mathcal{D}}$ of the cosets of $\widehat{B}$. Let u and v be any elements in $\mathbf{G}$, where $u \neq 0$; for any $D_1 \in \widehat{\mathcal{D}}$ and any $D_2 \in \widehat{\mathcal{D}}$,

$$D_1 \Delta D_2 \quad \Leftrightarrow \quad \exists\ \ u, v, u \neq 0 \ \text{ such that } \ D_1 = \sigma_{u,v}(D_2). \tag{16}$$

From now on, D_1 *is equivalent to* D_2 means that $D_1 \Delta D_2$. For a given D, we are interested in the number of cosets D_1 such that $D \Delta D_1$. Moreover, we want to characterize explicitly the cosets D_1 by its syndromes. We here study even cosets; hence the syndrome of D will always be of the form $(0, \delta, \gamma, \lambda)$, and the weight of such a coset should be 2, 4, or 6.

Since m is odd then 3 (respectively, 5) and $2^m - 1$ are relatively prime. Hence it follows from (14) that there are always $N - 1$ distinct cosets $\sigma_{u,0}(D)$, $u \in \mathbf{G}^*$. Suppose that $\delta = 0$, meaning $D \in \mathcal{B}_2$. It follows from (15) that $\sigma_{1,v}(D) = D$ for any v. In this case the coset D is an *orphan*, because each coordinate position is covered by at least one leader of D (see Definition 2.1). The weight of D could be 4 or 6. When it is 4 the supports of two leaders cannot intersect, proving that the number of leaders is $N/4$. Since $\mathcal{B}_2$ is contained in $R(m-2, m)$, the support of any code word of weight 4 is an affine subspace of dimension 2. As there are $(N-1)(N-2)/6$ linear subspaces of dimension 2, there are the same number of cosets of weight 4 in $\mathcal{B}_2$. On the other hand, there are N^2 cosets in $\mathcal{B}_2$, implying that the number of cosets of weight 6 in $\mathcal{B}_2$ is

$$N^2 - \ (N-1)(N-2)/6 - 1 = (N-1)(5N+8)/6.$$

Moreover, by definition, $\mathcal{B}_3$ is composed of $N-1$ cosets of weight 6, if we except $\widehat{B}$ itself.

So we have proved the following.

Proposition 4.5. *Let $D \in \mathcal{B}_2$. Then D is an orphan and*

$$\texttt{card}\ \{\ D_1 \mid D \Delta D_1\ \} = \texttt{card}\ \{\ \sigma_{u,0}(D) \mid u \in \mathbf{G}^*\ \} = N - 1.$$

When the weight of D is 4*, D has $N/4$ leaders.*

There are $(N-2)/6$ *nonequivalent cosets of weight* 4 *and* $(5N+8)/6$ *nonequivalent cosets of weight* 6 *in* $\mathcal{B}_2$*.*

There is only one coset D of weight 6 *in* $\mathcal{B}_3$ *up to equivalence. The cosets of $\mathcal{B}_3$ are $\sigma_{u,0}(D)$, $u = \alpha^k$, whose syndromes are $(0,0,0,\alpha^k)$ (α denotes here a primitive element of $\mathbf{G} = GF(2^m)$).*

Suppose now that $\delta \neq 0$; i.e., we consider cosets D in $\mathcal{B}_1$. It comes from (15) that D is invariant under a permutation $\sigma_{1,v}$ if and only if

$$\delta v^2 + \delta^2 v = 0 \quad \text{and} \quad \delta v^4 + \delta^4 v = 0.$$

The mapping $v \ \rightarrow \ \ \delta v^2 + \delta^2 v$ is linear; its kernel has dimension 1. Hence it takes exactly 2^{m-1} distinct values. Since m is odd, we obtain the same result for the mapping $v \ \rightarrow \ \delta v^4 + \delta^4 v$. In both cases the kernel is $\{0, \delta\}$; so, by applying $\sigma_{1,v}$, we obtain exactly 2^{m-1} different syndromes. Suppose that the weight of D is 4. Whenever D contains the code words a whose support is $\{\ a_1,\ a_2,\ a_3,\ a_4\ \}$, it contains also the word $X^\delta a$ whose support is $\{\ a_1 + \delta,\ a_2 + \delta,\ a_3 + \delta,\ a_4 + \delta\ \}$. These code words do not intersect. Indeed, the equalities $a_1 = a_2 + \delta$ and $a_3 = a_4 + \delta$ would imply $\sum_{i=1}^4 a_i = 0$, meaning that D is contained in $R(m-2, m)$ (i.e., $\delta = 0$). So we have proved the following.

Proposition 4.6. *The set $\mathcal{B}_1$ contains $N^2(N-1)$ elements. For any $D \in \mathcal{B}_1$ we have*

$$\texttt{card}\ \{\ D_1 \mid D \Delta D_1\ \} = N(N-1)/2.$$

So there are $2N$ classes of nonequivalent cosets in $\mathcal{B}_1$.

The permutation $\sigma_{1,v}$ leaves a coset D with the syndrome $(0,\delta,\gamma,\lambda)$ invariant if and only if $v=\delta$. Therefore, when the weight of D is 4, the number of leaders in D is even: whenever D contains a word a, it contains also the word $X^{\delta}a$, which cannot be equal to a.

There are $N(N-1)/2$ distinct code words of weight 2 and each coset of weight 2 contains only one code word of weight 2. All cosets of weight 2 are in $\mathcal{B}_1$, because the minimum weight of $R(m-2,m)$ is 4. Since the group of the $\sigma_{u,v}$ is doubly transitive, they are equivalent. The syndromes can be calculated from the formulas of Lemma 4.4.

PROPOSITION 4.7. *The cosets of weight 2 are in $\mathcal{B}_1$. The corresponding syndromes are of the form*

$$(\ 0,\ u,\ u^3+uv^2+u^2v,\ u^5+uv^4+u^4v\ ),\quad u\in\mathbf{G}\backslash\{0\},\quad v\in\mathbf{G}.$$

These cosets are the $\sigma_{u,v}(D)$, where D is the coset whose leader is $1+X$ and whose syndrome is $(0,1,1,1)$.

Note that the coset $\sigma_{u,v}(D)$ is equal to the coset $\sigma_{u,v'}(D)$ if and only if $v'=v$ or $v'=v+u$. This gives us $N(N-1)/2$ different cosets of weight 2.

5. Cosets weight distribution: The hard cases.

5.1. Cosets of minimum weight 4. We begin by giving the results we have on cosets of weight 4 of B, the elements of $\mathcal{D}_4$. Moreover we claim that the weight distributions of cosets of $\mathcal{D}_4$ can be precisely obtained from those of the cosets of $\widehat{\mathcal{D}}_4$.

PROPOSITION 5.1. *Let F be any coset of $\mathcal{D}_4$. The weight distribution of F is uniquely defined by the value $\mu_{4,4}$, where $\mu_{4,4}$ is an even number in the interval*

$$2\ \le\ \mu_{4,4}\ \le\ (n+1)/4\ -\ 2.$$

Moreover,

$$\mu_{4,4}\ +\ \mu_{4,5}\ =\ \mu_{5,5}\ =\ \frac{(n-1)(n-7)}{120}.$$

The coset F can be seen as a shortened coset of $\widehat{\mathcal{D}}_4$ with parameter $\widehat{\mu}_{4,4}=\mu_{4,4}$.

Proof. From equation $(A.4)$ and the equality $\alpha_4=\alpha_5$ (see (2)) we have for an arbitrary coset F of weight 4

$$\mu_{4,4}\ +\ \mu_{4,5}=\ \frac{1}{\alpha_5}\ =\ \frac{(n-1)(n-7)}{120}.$$

Extending F, we clearly obtain a coset of weight 4 of $\widehat{B}$, which has as its set of leaders the set of leaders of F. So $\mu_{4,4}$ is even according to Proposition 4.6. Of course, F cannot be an orphan, since n is an odd number, implying $\mu_{4,4}\ <\ n/4$ and therefore $\mu_{4,4}\ <\ (n+1)/4\ -\ 1$ (because $(n+1)/4\ -\ 1$ is also odd). □

PROPOSITION 5.2. *Let F be any coset of weight 4 of $\widehat{B}$, i.e., $F\in\ \widehat{\mathcal{D}}_4$. The weight distribution of F is uniquely defined by the value $\widehat{\mu}_{4,4}$, where $\widehat{\mu}_{4,4}$ is an even number in the interval*

$$2\ \le\ \widehat{\mu}_{4,4}\ \le\ N/4.$$

Proof. Suppose that F is an arbitrary coset of $\widehat{B}$ of weight 4:$F\ \in\ \widehat{\mathcal{D}}_4$. Since every weight of F is even we obtain from formula $(E.4)$ the value $\widehat{\mu}_{4,6}$:

$$\widehat{\mu}_{4,6}\ =\ \frac{1-\widehat{\alpha}_4\widehat{\mu}_{4,4}}{\widehat{\alpha}_6}. \tag{17}$$

Therefore, the weight distribution of F is uniquely determined from the value $\widehat{\mu}_{4,4}$. Now note that two leaders of F have disjoint supports, since the minimum weight of B is 8. Hence $\mu_{4,4} \leq N/4$. From Proposition 4.6 we have that the number $\widehat{\mu}_{4,4}$ is always even. ∎

It is clear that any coset $F \in \widehat{\mathcal{D}}_4$ with $\widehat{\mu}_{4,4}$ leaders has $N - 4\widehat{\mu}_{4,4}$ different parents from $\widehat{\mathcal{D}}_5$. As we already know from Proposition 4.5, there are at least $(N-1)(N-2)/6$ cosets in $\widehat{\mathcal{D}}_4$ with weight distribution

$$\widehat{\mu}_{4,4} = N/4 \text{ and } \widehat{\mu}_{4,6} = N(N-8)(N-32)/720. \tag{18}$$

These cosets have no parent; they are orphans. There are N different cosets in $\widehat{\mathcal{D}}_3$ which are generated by any such orphan. They are the N children of the orphan. Can two different orphans R and R' give the same children? If yes, that implies that the distance between these two cosets is 2, i.e., that the set of code words

$$R + R' = \{\ x + x' \mid x \in R,\ x \in R'\ \}$$

has minimum weight 2. So, if the set above has minimum weight 4 there is a contradiction. Particularly, if the orphans R and R' are in the RM-code of order $m-2$, the set of the children of R and the set of the children of R' do not intersect. In this way, we obtain at least $N(N-1)(N-2)/6$ cosets of weight 3. In accordance with (7), we have the following.

Proposition 5.3. *Any coset in $\widehat{\mathcal{D}}_3$ is a child of some orphan of $\widehat{B}$ of weight* 4 *which is contained in the RM-code of order $m-2$.*

5.2. Cosets of minimum weight 6. At the end, we have to study the cosets of $\widehat{\mathcal{D}}_6$. It is the same situation we had for cosets of $\mathcal{D}_5$. Although we know the weight distribution of such cosets, we cannot give the cardinality of $\widehat{\mathcal{D}}_6$. However, we can give a property analogous to those stated in Proposition 5.3.

Proposition 5.4. *All cosets of $\widehat{B}$ of weight* 6 *have the same weight distribution. Such a coset is an orphan and it contains*

$$\widehat{\mu}_{6,6} \ = \ N(N-2)(N-8)/720 \tag{19}$$

code words of weight 6*.*

Proof. It is clear that the equation $(E.6)$ has only one solution (it can be deduced also from [1]). That is $\widehat{\mu}_{6,6} = 1/\widehat{\alpha}_6$. We deduce (19) from the formula (3), which gives the value of $\widehat{\alpha}_6$. Then all cosets in $\widehat{\mathcal{D}}_6$ have the same weight distribution. Such cosets are orphans since the covering radius of $\widehat{B}$ is 6. ∎

Now take $F \ \in \ \widehat{\mathcal{D}}_6$ and consider its children. They are cosets $G \ \in \ \widehat{\mathcal{D}}_5$ such that

$$G \ = \ F \ + \ v^{(i)}$$

for some $i \in [1, N]$. So if we denote

$$\operatorname{supp}(G) \ = \ \bigcup_{v \text{ is a leader of } G} \operatorname{supp}(v),$$

then we have for such a child of F

$$\operatorname{supp}(G) \ \subseteq \ \{1, \ldots, N\} \ \setminus \ \{i\}.$$

PROPOSITION 5.5. *Let G be any coset from $\widehat{\mathcal{D}}_5$. Then G is not an orphan, and there is $i \in [1, N]$ and a coset $F \in \mathcal{B}_2$ (i.e., a coset of weight 6, which belongs to Reed–Muller code $R(m-2, m)$) such that G is a child of F with $G = F + v^{(i)}$. Moreover, we have*

$$\mathrm{supp}(G) \;=\; \{\; 1, \ldots, N \;\} \setminus \; \{i\}.$$

Proof. Let F and F' be two arbitrary cosets from $\widehat{\mathcal{D}}_6$. Using the same idea we used for the proof of Proposition 5.3, we can say *if $F + F'$ has minimum weight 4, then the set of the children of F and the set of the children of F' do not intersect.* That is particularly true when we consider cosets in $\mathcal{B}_2$.

From Proposition 4.5 we know that there are $(N-1)(5N+8)/6$ distinct cosets of weight 6 in $\mathcal{B}_2$. Each such coset has exactly N children because any coset of weight 6 is an orphan. Since all children of such cosets are distinct, we obtain $N(N-1)(5N+8)/6$ distinct cosets of weight 5. But from Proposition 3.3 we know that this is exactly the number $\widehat{\Gamma}(5)$ of different cosets of weight 5. Therefore, any coset G from $\widehat{\mathcal{D}}_5$ is a child of some coset F from $\widehat{\mathcal{D}}_6$. We have $G = F + v^{(i)}$ for some i. Clearly, a leader of the coset G cannot have the position i in its support. So G is not an orphan and we have $\mathrm{supp}(G) \subseteq \{\; 1, \ldots, N \;\} \setminus \; \{i\}$. Suppose now that there is another position j which is not covered by $\mathrm{supp}(G)$. Then there is a contradiction with the fact that any coset of $\mathcal{D}_5$ is an orphan. Indeed, we can suppose that $j = 0$ because of the invariance of cosets of $\mathcal{B}$ under affine permutations. With this hypothesis, shortening G we obtain a coset of B of weight 5 which is not an orphan because ith position is not covered by the nonzero position of its leaders. According to Proposition 3.2 we have a contradiction. □

6. Summary of results. In this section we summarize the results we have about the weight distribution of the cosets of the code B and of its extension. These results are explained in sections 3, 4, and 5. In Table 2, the values we know for the number of cosets of a given weight are presented. We give the distance matrices of B and $\widehat{B}$ in Tables 3 and 4. Let C be a code with the dual distance t. Recall that the distance matrix of C is the $u \times (t+1)$ matrix containing the $t+1$ first coefficients of the u distinct weight distributions of cosets of C. The weight distributions of the cosets of C can be fully calculated from these elements [12].

TABLE 2
The number $\Gamma(i)$ of cosets of B of weight i and the number $\widehat{\Gamma}(i)$ of cosets of $\widehat{B}$ of weight i. We denote by γ the number of cosets of $\widehat{B}$ of weight 4 which are not in $R(m-2, m)$.

i	$\Gamma(i)$	$\widehat{\Gamma}(i)$
1	n	N
2	$n(n-1)/2$	$N(N-1)/2$
3	$n(n-1)(n-2)/6$	$N(N-1)(N-2)/6$
4	?	$(N-1)(N-2)/6 \;+\; \gamma$
5	$= \widehat{\Gamma}(6)$	$N(N-1)(5N+8)/6$
6	0	?

In Table 2, it clearly appears that the knowledge of γ involves the knowledge of any $\widehat{\Gamma}(i)$, implying the knowledge of any $\Gamma(i)$ since we know the total number of cosets. The coefficients of the distance matrix of B (see Table 3) depend only on those of the distance matrix of $\widehat{B}$ (see Table 4). Moreover, we have proved that all

Table 3
The distance matrix of the code B of length n, $n = 2^m - 1$, m odd.

0 1 2	3	4	5
1 0 0	0	0	0
0 1 0	0	0	0
0 0 1	0	0	$(n-1)(n-7)/120 + 1$
0 0 0	1	$\widehat{\mu}_{4,4} - 1$	$\widehat{\mu}_{3,5} - \widehat{\mu}_{4,4} + 1$
0 0 0	1	$\cdots$	$\cdots$
0 0 0	0	$\widehat{\mu}_{4,4} \leq (n-7)/4$	$\widehat{\mu}_{5,5} - \widehat{\mu}_{4,4}$
0 0 0	$\cdots$	$\cdots$	$\cdots$
0 0 0	0	0	$(n-1)(n-7)/120$

Table 4
The distance matrix of the code $\widehat{B}$ of length N, $N = 2^m$, m odd.

0 1 2 3	4	5	6
1 0 0 0	0	0	0
0 1 0 0	0	0	0
0 0 1 0	0	0	$(N-2)(N^2 - 10N + 136)/720$
0 0 0 1	0	$(N-2)(N-8)/120 + 1$	0
0 0 0 0	$\widehat{\mu}_{4,4} \leq (N-8)/4$	0	$\widehat{\mu}_{4,6}$
0 0 0 0	$\cdots$	0	$\cdots$
0 0 0 0	$N/4$	0	$N(N-8)(N-32)/720$
0 0 0 0	0	$(N-2)(N-8)/120$	0
0 0 0 0	0	0	$N(N-2)(N-8)/720$

coefficients of the distance matrix of $\widehat{B}$ are known as soon as the possible values of $\widehat{\mu}_{4,4}$ are known (see Proposition 5.2).

Therefore, we conclude that *the problem of the weight distribution of the cosets of the* 3*-error-correcting BCH-codes, extended or not (i.e., B or $\widehat{B}$), is reduced to the problem of the weight distribution of the cosets of weight* 4 *of $\widehat{B}$, which are not in the Reed–Muller code of order $m-2$.*

7. Numerical results and conjectures. For length 128 we have computed the cosets weight distribution of $\widehat{B}$. We give in Table 5 the distance matrix and the number of cosets for each weight. Note that in this case, we obtain 12 distinct weight distributions, whereas we had 8 weight distributions for length 32. So we conjecture that the number of weight distributions increases with the length. We will make our conjecture precise later. Now we want to explain how Table 5 was completed.

- The number of cosets and the corresponding lines of the distance matrix are known for cosets of weight 1, 2, 3, or 5 for any length (see sections 3 and 6).
- So it remains to determine the number of cosets of weight 4 or 6 and the weight distributions of the cosets of weight 4. For the computation of weight distributions we only need to determine the number of leaders. We use the definition of cosets by syndrome (see Definition 4.2).
- We know the number of cosets of weight 4 or 6 contained in $\mathcal{B}_2$, i.e., in $R(m-2, m)$ (see Proposition 4.5). There are 127×21 cosets of weight 4 and 127×108 cosets of weight 6. Such a coset of weight 4 has 32 leaders; it is an orphan. Our numerical results prove that all orphans of weight 4 are in $\mathcal{B}_2$.
- From now on we study the cosets of weight 4 or 6 contained in $\mathcal{B}_1$, i.e., in $R(m-1, m)\backslash R(m-2, m)$. There are 127×2^{14} cosets in $\mathcal{B}_1$, whose 127×64 have

weight 2. So there remain 127 × 16320 cosets of weight 4 or 6. Actually, we have computed the syndrome of any code word of weight 4 which is not in $R(m-2,m)$. Taking into account the results of section 4 it is sufficient to consider the syndromes

$$(\ 0,\ 1,\ 0,\ \lambda\ ) \quad \text{and} \quad (\ 0,\ 1,\ 1,\ \lambda\ ), \quad \lambda \in GF(128).$$

Indeed, they define $128+127$ cosets of weight 4 or 6; the syndrome $(0,1,1,1)$ corresponds to a coset of weight 2. From Proposition 4.6 each of these cosets has 127×64 equivalent cosets. Then we obtain

$$127 \times 64 \times (128+127) = 127 \times 16320$$

distinct cosets, and it is exactly the number of cosets of weight 4 or 6 in $\mathcal{B}_1$. So we need to examine a few code words of weight 4; the number of such code words of the same syndrome is the number of leaders.

- We found that 127×192 syndromes correspond to cosets of weight 6. By adding the number of such cosets in $\mathcal{B}_2$, we obtain the total number of cosets of weight 6. There remain 127×16128 cosets of weight 4 in $\mathcal{B}_1$. The number of leaders is even, in accordance with Proposition 4.6. This number takes all even value in the range $[2,10]$.

Table 5

The distance matrix of the 3*-error-correcting extended BCH-code of length* 128; $W_{\min}$ *is the minimum weight of the coset.*

$W_{\min}$	Number of cosets	Number of words of weight:						
		0	1	2	3	4	5	6
0	1	1	0	0	0	0	0	0
1	128	0	1	0	0	0	0	0
2	$127 \times 64 = 8128$	0	0	1	0	0	0	2667
3	$127 \times 2688 = 341376$	0	0	0	1	0	127	0
4	$127 \times 1792 = 227584$	0	0	0	0	2	0	2648
4	$127 \times 6272 = 796544$	0	0	0	0	4	0	2608
4	$127 \times 5376 = 682752$	0	0	0	0	6	0	2568
4	$127 \times 2240 = 284480$	0	0	0	0	8	0	2528
4	$127 \times 448 = 56896$	0	0	0	0	10	0	2488
4	$127 \times 21 = 2667$	0	0	0	0	32	0	2048
5	$127 \times 13824 = 1755648$	0	0	0	0	0	126	0
6	$127 \times 300 = 38100$	0	0	0	0	0	0	2688

By using Tables 3 and 5, it is very easy to compute the distance matrix of the code B (of length 127). We also easily obtain the number of cosets of B of weight i, $i \in [0,5]$, by using Table 2. It is more complicated if we want to compute to number of cosets of weight 3 or 4 for each weight distribution. We proceed as follows.

- Let $x(i)$ be the number of cosets of $\widehat{B}$ of weight 4 such that $\widehat{\mu}_{4,4} = i$, $i < N/4$.
- Then $x(i) = 127 \times 64 \times y(i)$, where $y(i)$ is the number of nonequivalent cosets in the sense of (16); we can suppose that the $y(i)$ cosets have position zero in their support.
- Let F be such a coset. The cardinality of its support is $4i$. Consider the 64 cosets $\sigma_{1,v}(F)$. Among these cosets $2i$ have position zero in their support and $64-2i$ have not.
- So we obtain from F, $127 \times 2i$ cosets of weight 3 of B and $127 \times (64-2i)$ cosets of weight 4 of B. Multiplying these numbers by $y(i)$, we obtain the number of cosets of weight 3 and 4 whose weight distributions are defined by $\widehat{\mu}_{4,4} = i$.

- From the 127×21 orphans of weight 4, we obtain the same number of cosets of B of weight 3. They correspond to one and only one weight distribution.

Recall that, for length 32, all cosets of weight 4 have the same weight distribution with $\widehat{\mu}_{4,4} = 2$. It is because in this case the code $\widehat{B}$ is exactly the Reed–Muller code of order 2. Any coset of weight 4 is a coset of the RM-code of minimum weight 8. Since the supports of these code words of weight 8 are the affine subspaces of K^5 of dimension 3, it is clear that such a coset cannot contain more than two words of weight 4.

For length 128, we have found six different weight distributions for the cosets of weight 4. For length 512, we made a random exploration of cosets of weight 4. Our numerical results allow us to state the following conjecture.

Conjecture 1. Let $\widehat{B}$ be the extended 3-error-correcting BCH-code of length 512. There are 12 different weight distributions for the cosets of $\widehat{B}$ of weight 4. These distributions are determined by the number $\widehat{\mu}_{4,4}$ of code words of weight 4. This number is

1. $\widehat{\mu}_{4,4} = 128$ for the orphans contained in the RM-code of order 7. (We did not find other cosets corresponding to this value.)
2. $\widehat{\mu}_{4,4} = i$ for all even integers i in the range $[12, 32]$.

So we have shown that the situation here is completely different from those we had for the 2-error-correcting BCH-codes. In both cases the external distance is a constant not depending on the length. The number of weight distributions of cosets is constant for any length for the 2-error-correcting BCH-codes. And that is true not only when m is odd (and codes are completely regular) but also when m is even [10, 20]. For the 3-error-correcting BCH-codes, we strongly conjecture that this number increases with the length. When m is odd these codes are uniformly packed, and we point out this property for $m = 5, 7$, and 9. Moreover, we are able to propose general conjectures.

Conjecture 2. Let $\widehat{B}$ be the extended 3-error-correcting BCH-code of length N, m odd. *Then any coset of $\widehat{B}$ of weight 4, which is an orphan, is contained in the RM-code of order $m-2$.*

Conjecture 3. Denote by G the Galois field of order 2^m, m odd. For any (A, B), where A and B are any elements in G, let us denote by $\mathcal{E}(A, B)$ the following system of three equations, with four variables, on G:

$$\begin{aligned} W + X + Y + Z &= 1, \\ W^3 + X^3 + Y^3 + Z^3 &= A, \\ W^5 + X^5 + Y^5 + Z^5 &= B. \end{aligned}$$

Let $\mathcal{N}(A, B)$ be the number of solutions of $\mathcal{E}(A, B)$ satisfying $X \neq Y \neq Z \neq W$. Consider the (A, B) such that $\mathcal{N}(A, B)$ is not zero and recall that $\mathcal{N}(A, B)$ is always even (see Proposition 4.6). *Then there exist two even integers depending on m, say ℓ_m and u_m, $\ell_m < u_m < 2^{m-2}$, such that*

$$\ell_m \leq N(A, B) \leq u_m.$$

Moreover, for any even value i in the range $[\ell_m, u_m]$, there is an (A, B) such that $\mathcal{N}(A, B) = i$.

Acknowledgments. The authors are indebted to Nicolas Sendrier for computing some numerical results and checking others with his own programs. They are grateful to the anonymous referee who indicated several improvements of the manuscript.

The second author thanks Paul Camion for the possibility to work in INRIA (project CODES) as an invited professor in 1994 and 1995.

REFERENCES

[1] E. F. Assmus, Jr. and H. F. Mattson, Jr., *Some* 3*-error-correcting BCH codes have covering radius* 5, IEEE Trans. Inform. Theory, IT-22 (1976), pp. 348–349.

[2] E. F. Assmus, Jr. and V. Pless, *On the covering radius of extremal self-dual codes*, IEEE Trans. Inform. Theory, IT-29 (1983), pp. 359–363.

[3] L. A. Bassalygo, G. V. Zaitsev, and V. A. Zinoviev, *Uniformly packed codes*, Problems Inform. Transmission, 10 (1974), pp. 9–14.

[4] L. A. Bassalygo and V. A. Zinoviev, *Remark on uniformly packed codes*, Problems Inform. Transmission, 13 (1977), pp. 22–25.

[5] R. A. Brualdi and V. S. Pless, *Orphans of the first order Reed-Muller codes*, IEEE Trans. Inform. Theory, IT-36 (1990), pp. 399–401.

[6] P. Camion, B. Courteau, G. Fournier, and S. V. Kanetkar, *Weight distribution of translates of linear codes and generalized Pless Identities*, J. Inform. Optim. Sci., 8 (1987), pp. 1–23.

[7] P. Camion, B. Courteau, and A. Montpetit, *Coset weight enumerators of the extremal self-dual binary codes of length* 32, Eurocode '92: International Symposium on Coding Theory and Applications, CISM International Centre for Mechanical Sciences Ser. Vol. 339, P. Camion and P. Charpin, eds., Springer-Verlag, Amsterdam, The Netherlands, 1993, pp. 17–30.

[8] P. Camion, B. Courteau, and P. Delsarte, *On r-partition designs in Hamming spaces*, Applicable Algebra in Eng. Comm. and Comput., 2 (1992), pp. 147–162.

[9] P. Charpin, *Tools for cosets weight enumerators of some codes*, in Proceedings of Finite Fields: Theory, Applications and Algorithmes, G. L. Mullen and P. Jau-Shyong, eds., Contemporary Mathematics, Vol. 168, AMS, Providence, RI, 1994.

[10] P. Charpin, *Weight distributions of cosets of* 2*-error-correcting binary BCH codes, extended or not*, IEEE Trans. Inform. Theory, IT-40 (1994), pp. 1425–1442.

[11] P. Charpin and V. A. Zinoviev, *On weight distributions of cosets of* 3*-error-correcting extended BCH codes of length* 2^m*, m odd*, in Proceedings ACCT4 '94 Fourth International Workshop. Algebraic and Combinatorial Coding Theory, Novgorod, Russia, 1994, pp. 66–69.

[12] P. Delsarte, *Four fundamental parameters of a code and their combinatorial significance*, Inform. and Control, 23 (1973), pp. 407–438.

[13] J. M. Goethals and H. C. A. Van Tilborg, *Uniformly packed codes*, Philips Res. Rep., 30 (1975), pp. 9–36.

[14] D. Gorenstein, W. W. Peterson, and N. Zierler, *Two-error-correcting Bose-Chaudhuri codes are quasi-perfect*, Inform. and Control, 3 (1960), pp. 291–294.

[15] T. Helleseth, *All binary* 3*-error-correcting BCH codes of length* $2^m - 1$ *have covering radius* 5, IEEE Trans. Inform. Theory, IT-24 (1978), pp. 257–258.

[16] J. A. van der Horst and T. Berger, *Complete decoding of triple-error-correcting binary BCH codes*, IEEE Trans. Inform. Theory, IT-22 (1976), pp. 138–147.

[17] T. Kasami, *Weight distributions of Bose-Chaudhuri-Hocquenghen codes*, in Combinatorial Mathematics and its Applications, R. C. Bose and T. A. Dowling, eds., Univ. of North Carolina Press, Chapel Hill, NC, 1969.

[18] T. Kasami, S. Lin, and W. W. Peterson, *Some results on cyclic codes which are invariant under the affine group and their applications*, Inform. and Control, 11 (1967), pp. 475–496.

[19] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes*, North–Holland, Amsterdam, The Netherlands, 1986.

[20] N. V. Semakov, V. A. Zinoviev, and G. V. Zaitsev, *Uniformly packed codes*, Problems Inform. Transmission, 7 (1971), pp. 38–50.

[21] H. C. A. Van Tilborg, *Uniformly Packed Codes*, Ph.D. thesis, Tech. Univ. Eindhoven, Eindhoven, The Netherlands, 1976.

[22] X.-D. Hou, *Classification of cosets of the Reed-Muller code* $R(m-3, m)$, Discrete Math., 128 (1994), pp. 203–224.

SIAM J. Discrete Math.
Vol. 10, No. 1, pp. 146–157, February 1997

011

THE STRUCTURE AND NUMBER OF OBSTRUCTIONS TO TREEWIDTH[*]

SIDDHARTHAN RAMACHANDRAMURTHI[†]

Abstract. For each pair of nonadjacent vertices in a graph, consider the greater of the degrees of the two vertices. The minimum of these maxima is a lower bound on the treewidth of a graph, unless it is a complete graph. This bound has three consequences. First, the obstructions of order $w + 3$ for treewidth w have a simple structural characterization. Second, these graphs are exactly the pathwidth obstructions of order $w + 3$. Finally, although there is only one obstruction of order $w + 2$ for width w, the number of obstructions of order $w + 3$ is bounded below by an exponential function of $\sqrt{w}$.

Key words. treewidth, k-trees, graph minors, pathwidth, lower bounds, obstructions

AMS subject classifications. 05C85, 68R10, 05C35, 05C30

PII. S0895480195280010

1. Introduction.

Ever since its introduction, interest in the notion of *treewidth* of a graph has been growing (see [8]). This is mainly because, for many problems that are intractable on general graphs, polynomial-time and even linear-time algorithms can be found for graphs of bounded treewidth (see [3, 5], for example).

Arnborg, Corneil, and Proskurowski [2] showed that it is NP-hard to determine the treewidth of an arbitrary graph. Using dynamic programming, they also showed that for every fixed constant w, there exists an $O(n^{w+2})$-time algorithm to decide whether the treewidth is at most w [2]. Robertson and Seymour developed an $O(n^2)$-time algorithm [24] for this problem by showing (nonconstructively) that graphs of treewidth at most w can be characterized by a finite number of minimal forbidden minors (called *obstructions*) [23]. They started by computing an approximate tree decomposition (of width bounded by a constant, say, $c.w$) and used it to check whether the graph contains any of the obstructions for treewidth w.

Algorithms that do not rely on obstructions have also been designed for treewidth. One approach has been the development of an explicit linear-time algorithm to determine whether a graph has treewidth at most w when given an approximate tree decomposition for it [9]. By combining this with a linear-time algorithm to compute an approximate tree decomposition, Bodlaender [7] invented a linear-time algorithm to decide whether the treewidth of a graph is at most a constant w. Unfortunately, this algorithm is exponential in a polynomial in w and hence appears to be impractical even for $w = 4$.

Another approach for designing algorithms without using obstructions is to identify a set of reductions such that a graph has treewidth at most a fixed constant w if and only if it can be reduced to the null graph by a finite sequence of these reductions. This method was used in [4] for $w = 3$ and later in [26] to develop a linear-time algo-

[*] Received by the editors January 13, 1995; accepted for publication (in revised form) March 21, 1996. This research was funded in part by National Science Foundation grant NSF-BIR-9318160. An extended abstract of a previous version of this paper was presented at the 20th International Workshop on Graph-Theoretic Concepts in Computer Science, WG94, Herrsching, Germany, 1994 and was published in Lecture Notes in Comput. Sci., 903 (1995), pp. 14–25.

http://www.siam.org/journals/sidma/10-1/28001.html

[†] Department of Mathematics, University of Tennessee, Knoxville, TN 37996-1300 (siddhart@cs.utk.edu).

rithm for $w = 4$. Although both these algorithms are practical, no general techniques are known for finding a complete set of reductions for $w > 4$.

The *pathwidth* of a graph is a concept akin to treewidth (see [9, 17], for example). There has been considerable interest in the obstructions for treewidth and for pathwidth [6, 14, 17, 19, 27]. Obstruction-based algorithms have been used for integrated circuit design and other applications [12, 15, 18]. The reasons for studying the obstructions are two-fold. First, a better comprehension of their structure and number can help one design better algorithms for the fixed-parameter problem. Second, being minimal graphs, analyzing their structure can give us insights into developing better lower bounds for the treewidth of a graph.

Only the obstruction sets for treewidth 1, 2, and 3 have been found so far [6, 27], and none of the methods used to find them seem to generalize to larger values of w. Lagergren [19] found a weak upper bound on the number of edges in an obstruction for treewidth w. This bound is triple exponential in w^4 and is purely of theoretical significance. The obstruction sets for pathwidth 1 and 2 have also been computed [17], but the general constructions produce only relatively large and sparse obstructions (i.e., of order at least $3w + 3$ for pathwidth w).

Some general methods for computing the obstructions to a given family of graphs have also been proposed [14, 20]. These methods are nontrivial since their use requires a tree decomposition of bounded width and additional problem-specific information [11]. Hence, their application has been limited [10, 11].

In this paper we define a new metric γ of a graph and show that it is a lower bound for the treewidth of the graph. In practice, this bound can be computed in time linear in the size of the graph. Besides being of independent interest, our bound gives a new perspective on the obstructions for treewidth and for pathwidth. As a result, we obtain the following three important consequences.

A. For every $w \geq 3$,
   1. every obstruction of order $w + 3$ for treewidth w has a simple structural characterization and
   2. there exists at least one obstruction of order $w + 3$ for treewidth w.

B. The graphs in A are exactly the obstructions of order $w + 3$ for pathwidth w.

C. For treewidth w, the number of obstructions of order $w + 3$ is bounded below by an exponential function of $\sqrt{w}$.

Consequences A and B are significant in their generality and simplicity. Our characterization shows that these obstructions can be constructed and recognized easily. This is the first direct and general method for constructing nontrivial obstructions for treewidth and dense obstructions for pathwidth. In light of the fact that the complete graph is the only obstruction of order $w + 2$ for treewidth w, consequence C is very surprising. Ours is the first proof that there is an exponential number of obstructions for treewidth.

The rest of this paper is organized as follows. Section 2 consists of preliminaries. In section 3 we present a new lower bound for the treewidth of a graph and sketch a linear-time algorithm to compute this bound. The proof of consequence A is spread over sections 4 and 5. Consequences B and C are proven in sections 4 and 5, respectively. In section 6, we discuss the implications of our results and some open problems.

2. Preliminaries. The graphs that we consider are finite, simple, and undirected. The *order* of a graph is the number of vertices in the graph; the *size* of a graph is the number of edges. If G is a graph of order n, we usually denote the set of vertices of G by $V = \{v_1, v_2, \ldots, v_n\}$; the set of edges of G is denoted by E. The degree of vertex v_i is $\delta_i = |\{v_j : (v_i, v_j) \in E\}|$ and the minimum degree of G is $\delta = \min_i\{\delta_i\}$. K_n denotes the complete graph of order n.

Definition (see [22]). Given a graph G, a pair (T, Y) is a *tree decomposition* of G if T is a tree and $Y = \{X_i\}$ is a family of subsets of $V(G)$ indexed by $V(T)$ such that

(a) $\cup X_i = V(G)$,

(b) for every edge $(v_a, v_b) \in E(G)$, $\exists i \in V(T) \ni \{v_a, v_b\} \subseteq X_i$, and

(c) for $i, j, k \in V(T)$, if j is on the path between i and k in T, then $X_i \cap X_k \subseteq X_j$.

The *width* of a tree decomposition (T, Y) is $\max_{i \in V(T)}\{|X_i| - 1\}$. The *treewidth*$(G)$ is the minimum width over all possible tree decompositions of G.

A *path decomposition* of G is just a tree decomposition (T, Y), where T is a simple path. Note that $treewidth(K_n) = pathwidth(K_n) = n - 1$. Therefore, in general, $0 \leq treewidth(G) \leq pathwidth(G) \leq n - 1$.

We use a special kind of tree decomposition called a *smooth* decomposition [7]. A similar notion was independently developed by Yan [28] for path decompositions.

Definition (see [7]). A tree decomposition (T, Y) of width w is *smooth* if

(a) for every $i \in V(T)$, $|X_i| = w + 1$ and

(b) if (i, j) is an edge in T then $|X_i - X_j| = |X_j - X_i| = 1$.

As shown in [7, 28], any tree decomposition can be easily transformed into a smooth tree decomposition without changing the treewidth. The following lemma gives a useful relation between the number of vertices in a graph and the number of vertices in a smooth tree decomposition of the graph.

Lemma 2.1 (see [7, 28]). *If (T, Y) is a smooth tree decomposition of width w for a graph G, then $|V(T)| = |V(G)| - w$.*

If H and G are graphs, then H is a *minor* of G, denoted by $H \leq_m G$, if and only if a graph isomorphic to H can be obtained from a subgraph of G by contracting edges. If $H \leq_m G$ and H is not isomorphic to G, we write $H <_m G$. A family $\mathcal{F}$ of graphs is *closed* in the minor order if $\forall G \in \mathcal{F},\ H \leq_m G \Rightarrow H \in \mathcal{F}$. The *obstruction set* for a minor-closed family $\mathcal{F}$, written *obs*$(\mathcal{F})$, is the set of all graphs in the complement of $\mathcal{F}$ that are minimal in the minor order. In other words, $G \in obs(\mathcal{F})$ if and only if $G \in \overline{\mathcal{F}}$ and $H <_m G \Rightarrow H \in \mathcal{F}$. Therefore, if $\mathcal{F}$ is a minor-closed family, then $G \in \mathcal{F}$ if and only if $H \not\leq_m G\ \forall H \in obs(\mathcal{F})$. If we know all the graphs in *obs*$(\mathcal{F})$, then we can decide whether $G \in \mathcal{F}$ in polynomial time using the fact that, for every fixed graph H, there exists a polynomial-time algorithm that when given an input graph G decides whether $H \leq_m G$ [24].

Let TW(k) denote the family of graphs with treewidth at most k. One can easily verify that for any fixed k, TW(k) is minor-closed. It is known that *obs*(TW(k)) is a finite set [23]. Unfortunately, the proof of this is nonconstructive [13]. Therefore, we must invent other methods to learn about the structure of these obstructions as well as their number.

3. A new lower bound for treewidth. Using smooth decompositions, Yan [28] obtained the following lower bound for pathwidth. We observe that it holds for treewidth as well.

Lemma 3.1 (see [28]). *For any graph G of order n and size e, treewidth$(G) \geq \frac{2n-1-\sqrt{(2n-1)^2-8e}}{2}$.*

Proof. It is known [7] that if $w = treewidth(G)$, then $e \leq nw - \frac{w(w+1)}{2}$. This inequality can be rewritten as $w^2 - (2n-1)w + 2e \leq 0$. The general solution of this quadratic inequality is $\frac{2n-1\pm\sqrt{(2n-1)^2-8e}}{2}$. Since $w \leq n-1$ always, we get $w \geq \frac{2n-1-\sqrt{(2n-1)^2-8e}}{2}$. $\square$

Next we introduce a new metric γ of a graph and show that it is a lower bound for treewidth. We start with a structural result showing the existence of vertices of bounded degree in graphs of bounded treewidth.

LEMMA 3.2. *If G is not a complete graph and treewidth$(G) = w$, then there exists a pair of nonadjacent vertices, each of degree at most w in G.*

Proof. $G \subset K_n \Rightarrow w \leq n-2$. Let (T, Y) be a smooth tree decomposition of width w for G. In T, let 1 and $n-w$ be two leaves, let 2 be the neighbor of 1, and let $n-w-1$ be the neighbor of $n-w$. Also, let $\{v_1\} = X_1 - X_2$ and $\{v_2\} = X_{n-w} - X_{n-w-1}$. Since 2 is on every path from 1 to another vertex in T and $v_1 \notin X_2$, $\forall i$, $|X_i \cap \{v_1, v_2\}| \leq 1$. Hence, $(v_1, v_2) \notin E(G)$. Moreover, $|X_i| = w+1 \Rightarrow \delta(v_1) \leq w$ and $\delta(v_2) \leq w$. $\square$

This lemma immediately motivates the definition of a new metric γ of a graph.

DEFINITION. For a graph $G = (V, E)$ of order n,

$$\gamma(G) = \begin{cases} n-1 & \text{if } G \text{ is a complete graph,} \\ \min_{(v_i,v_j)\notin E}\{\max\{\delta_i, \delta_j\}\} & \text{otherwise.} \end{cases}$$

LEMMA 3.3. *For every graph G, treewidth$(G) \geq \gamma(G)$.*

Proof. If G is a complete graph, then $\gamma(G) = n-1 = \mathit{treewidth}(G)$. Otherwise, let $w = \mathit{treewidth}(G)$ and let v_1, v_2 be a pair of nonadjacent vertices of degree $\leq w$ in G. Then $w \geq \max\{\delta_1, \delta_2\} \geq \gamma(G)$. $\square$

This shows that γ is a lower bound for treewidth. There are families of graphs for which γ is greater than the lower bound given by Lemma 3.1. Complete bipartite graphs of the form $K_{m,m}$ with $m \geq 3$ are an example. Moreover, Lemma 3.1 is based on the total number of edges in the graph. In contrast, our lower bound γ is based on the neighborhood of individual vertices and hence reveals useful structural information, as will be evident in what follows.

By checking each pair of vertices in a graph, it is easy to compute γ in $O(n^2)$ time. Instead, if we first sort the vertices in nondecreasing order of their degree and observe that γ equals the degree of some vertex within the first $\delta+2$ vertices in the sorted list, then γ can be computed in $O(n+e)$ time.

In the next section we use the metric γ to examine the structure of obstructions for treewidth and prove consequences A and B.

4. Obstructions of order $w+3$ for treewidth w.

The complete graph is the smallest obstruction for treewidth (and pathwidth). It is also the only general obstruction known for treewidth. For each $w \geq 0$, $K_{w+2} \in \mathit{obs}(\mathrm{TW}(w))$. What is the next smallest obstruction for treewidth? In this section, we explore obstructions of order $w+3$ for $\mathrm{TW}(w)$.

Let $n = w+3$ be the order of an obstruction for $\mathrm{TW}(w)$. The treewidth of such a graph would be $n-2$. The following lemma shows that the γ of this graph equals its treewidth.

LEMMA 4.1. *For a graph G of order n, treewidth$(G) = n-2$ if and only if $\gamma(G) = n-2$.*

Proof. ($\Leftarrow$): $\gamma(G) = n-2 \Rightarrow G \subset K_n$. Therefore, $n-2 \leq \mathit{treewidth}(G) < n-1$.

($\Rightarrow$): $\mathit{treewidth}(G) = n-2 \Rightarrow \gamma(G) \leq n-2$. Suppose $\gamma(G) < n-2$. Then there exist two vertices v_1 and v_2 in $V(G)$ such that the edge $(v_1, v_2) \notin E(G), \delta_1 < n-2$, and $\delta_2 < n-2$. $\delta_1 < n-2$ implies that there exists $v_x \in V(G)$ such that $(v_1, v_x) \notin E(G)$. Similarly, there exists v_y (x may be equal to y) such that $(v_2, v_y) \notin$

$E(G)$. Then the following is a smooth tree decomposition of width $n-3$ for G: $X_1 = V - \{v_2, v_x\}$, $X_2 = V - \{v_1, v_2\}$, and $X_3 = V - \{v_1, v_y\}$. This contradicts the fact that $treewidth(G) = n-2$. □

A similar result can be obtained for pathwidth.

LEMMA 4.2. *For a graph G of order n, $pathwidth(G) = n-2$ if and only if $\gamma(G) = n-2$.*

Proof. Notice that the tree decomposition of width $n-3$ used in the previous proof is also a path decomposition. The rest is trivial. □

The next theorem gives the structural characterization promised by consequence A.

THEOREM 4.3. *For every $n \geq 6$, a graph G of order n is an obstruction for $TW(n-3)$ if and only if G satisfies the following four conditions:*

(i) $\gamma(G) = n-2$,

(ii) *the maximum degree of G is $n-2$,*

(iii) $\delta(G) \geq \max\{4, 3q-3\}$, *where $q \leq \lfloor \frac{n}{3} \rfloor$ is the number of vertices of degree $< n-2$, and*

(iv) *G is missing at least three disjoint edges (i.e., $\overline{G}$ has a matching of size ≥ 3).*

Proof. ($\Rightarrow$): First we show that if $G \in obs(\mathrm{TW}(n-3))$ then it satisfies (i)–(iv). Since $treewidth(G) = n-2$, (i) follows directly from the previous lemma. For (ii), suppose there exists v_1 with $\delta_1 = n-1$. Let v_2 be a vertex of minimum degree in G. Then $\delta_2 = \delta(G) \leq \gamma(G) = n-2$. Let $H = (V(G), E(G) - \{(v_1, v_2)\})$. $\gamma(H) = \min\{\gamma(G), \max\{\delta_1, \delta_2\}\} = \min\{\gamma(G), \delta_1\} = n-2$. This implies $treewidth(H) = n-2$, contradicting the fact that G is minor-minimal. Therefore, the maximum degree of $G = n-2$.

The proof of (iii) is by contradiction. Suppose $1 \leq \delta_1 = \delta \leq 3$. Let $v_2, \ldots, v_{\delta+1}$ be the neighbors of v_1. Each vertex in $V_2 = \{v_i : \delta+2 \leq i \leq n\}$ is nonadjacent to v_1 and has degree $n-2$. Therefore, each neighbor of v_1 has degree $\geq n-\delta$. If $\delta = 1$ then $\delta_2 = n-1$, which contradicts (ii). If $\delta = 2$ then contracting the edge (v_1, v_2) to v_2 gives us $K_{n-1} <_m G$ and $treewidth(K_{n-1}) = n-2 = treewidth(G)$, contradicting the minor minimality of G. If $\delta = 3$ then we claim that at least one of the edges $(v_2, v_3), (v_2, v_4), (v_3, v_4)$ is in G, because otherwise $\delta_2 = \delta_3 = \delta_4 = n-3$ and $\gamma(G) < n-2$. Assume that $(v_2, v_3) \in E$. Contract edge (v_1, v_4) to v_4 to obtain $K_{n-1} \leq_m G$. Therefore, $\delta \geq 4$.

Let $V_1 = \{v_1, \ldots, v_q\}$ be the set of vertices of degree $< n-2$ in G. Each vertex in V_1 has at least two nonneighbors in G. $\gamma = n-2 \Rightarrow$ the vertices in V_1 are mutually adjacent and any vertex in G not adjacent to a vertex in V_1 has degree $= n-2$. Clearly, there are at least $2q$ such vertices of degree $n-2$. Therefore, $q \leq \lfloor \frac{n}{3} \rfloor$ and $\delta \geq q-1+2q-2 = 3q-3$. Hence, $\delta \geq \max\{4, 3q-3\}$.

It is clear from the proof of (iii) that each vertex of degree $< n-2$ in G contributes an edge to a maximum matching in the complement of G. Thus, when $q \geq 3$, (iv) follows from (iii). Otherwise, we have three different cases.

When $q = 0$, $\delta_i = n-2 \, \forall i$ and there is a matching of size $\lfloor \frac{n}{2} \rfloor$ in the complement of G. For $n \geq 6$, $\lfloor \frac{n}{2} \rfloor \geq 3$.

If $q = 1$, let $\delta_1 < n-2$ and $(v_1, v_2) \notin E$. Then $\delta \geq 4 \Rightarrow \exists\{v_3, v_4, v_5, v_6\} \ni \{(v_1, v_i), (v_2, v_i) : 3 \leq i \leq 6\} \subset E$ and $\{(v_3, v_4), (v_5, v_6)\} \cap E = \phi$.

If $q = 2$, let $\delta_1 \leq \delta_2 < n-2$ and $\{(v_1, v_3), (v_2, v_4)\} \cap E = \phi$. Then $\delta \geq 4 \Rightarrow \exists\{v_5, v_6\} \ni \{(v_1, v_i), (v_2, v_i) : 5 \leq i \leq 6\} \subset E$ and $(v_5, v_6) \notin E$. Thus, when $q = 1$ or 2, $(v_1, v_2), (v_3, v_4), (v_5, v_6)$ is a set of three disjoint edges missing from G. This concludes the proof of (iv).

($\Leftarrow$): Suppose G satisfies conditions (i)–(iv). We want to prove that $G \in obs(\mathrm{TW}(n-3))$. It follows from condition (i) and Lemma 4.1 that $treewidth(G) = n - 2$. It remains to show that $H <_m G \Rightarrow treewidth(H) < treewidth(G)$. Recall the definition that $H \leq_m G$ if and only if a graph isomorphic to H can be obtained from a subgraph of G by contracting edges. Condition (iii) implies that there are no isolated vertices in G. Therefore, in order to obtain a graph H such that $H <_m G$, at least one edge of G must either be deleted or contracted.

Suppose we delete an edge (v_x, v_y) from G to obtain graph H. Since condition (ii) stipulates that $\delta_x \leq n-2$ and $\delta_y \leq n-2$ in G, we know that in H, v_x and v_y are a pair of nonadjacent vertices each of degree less than $n-2$. Therefore, $\gamma(H) < n-2$, which implies that $treewidth(H) < n - 2$.

Suppose we obtain graph H by contracting an edge (v_x, v_y) in G. Since the order of H is $n - 1$, we know that $treewidth(H) \leq n - 2$. In order for $treewidth(H)$ to be $n-2$, H must be a complete graph. Condition (iv) implies that there exist six distinct vertices, say, $v_i, 1 \leq i \leq 6$ in G such that $\{(v_1, v_2), (v_3, v_4), (v_5, v_6)\} \cap E(G) = \phi$. Since the vertices v_x and v_y can cover at most two of the matchings in $\overline{G}$, we know that at least one of the three edges $(v_1, v_2), (v_3, v_4)$, and (v_5, v_6) does not exist in H. Therefore, H is not a complete graph and $treewidth(H) < n - 2$. □

Given a graph G, it is simple to verify in polynomial time whether G satisfies conditions (i)–(iv) of Theorem 4.3. Hence, the obstructions of order $w+3$ for $\mathrm{TW}(w)$ are easily recognizable.

In the following lemma we prove consequence B.

LEMMA 4.4. *A graph G of order n is an obstruction for* $\mathrm{PW}(n-3)$ *if and only if G is also an obstruction for* $\mathrm{TW}(n-3)$.

Proof. ($\Leftarrow$): Let $G \in obs(\mathrm{TW}(n-3))$. Then $treewidth(G) = n - 2$ and $\gamma(G) = n - 2$, which implies that $pathwidth(G) = n - 2$. If H is a minor of G, then $treewidth(H) < treewidth(G) = n - 2$ and $\gamma(G) < n - 2$. Therefore, $pathwidth(H) < n - 2$ and $G \in obs(\mathrm{PW}(n-3))$.

($\Rightarrow$): Let $G \in obs(\mathrm{PW}(n-3))$. Then $pathwidth(G) = n-2$ and $\gamma(G) = n-2$, which implies that $treewidth(G) = n - 2$. Suppose $H <_m G$. Then $pathwidth(H) < pathwidth(G)$. But $treewidth(H) \leq pathwidth(H)$. Therefore, for every $H <_m G$, $treewidth(H) < treewidth(G)$ and $G \in obs(\mathrm{TW}(n-3))$. □

The proof of consequence A is not complete because Theorem 4.3 does not tell us whether there are any graphs that satisfy all four conditions. In the next section, we show that such obstructions do exist for every $n \geq 6$.

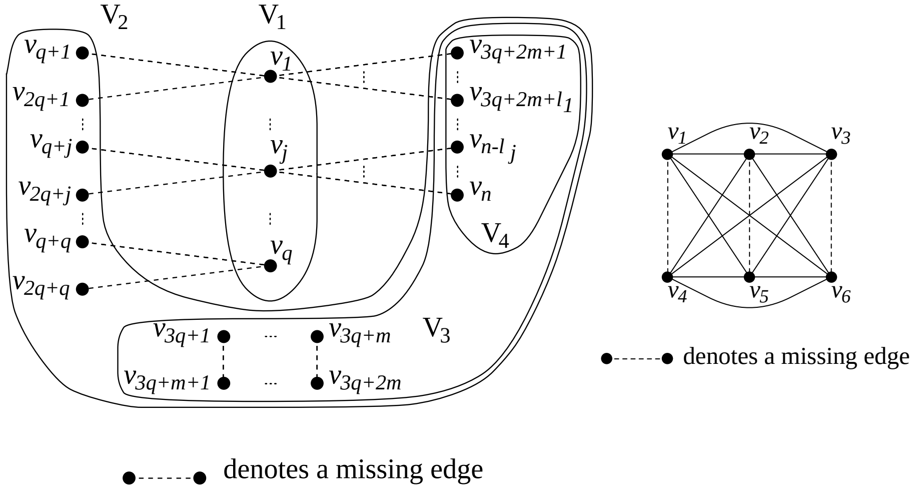
5. Enumerating the obstructions. Given w, we would like to find all the obstructions of order $w + 3$ for $\mathrm{TW}(w)$. For this, we use the structural description given by Theorem 4.3 and the theory of partitions of integers.

DEFINITION (see [1]). A *partition* of a positive integer n is a finite nonincreasing sequence of positive integers $l_1, l_2, \ldots, l_r$ such that $\sum_{i=1}^{r} l_i = n$. The l_i are called *parts* of the partition. $p_r(n)$ is the number of partitions of n into r parts. The total number of partitions of n is $p(n) = \sum_{r=1}^{n} p_r(n)$.

We adopt the convention that $p_0(0) = 1, p_0(n) = 0$ if $n > 0$, and $p_r(n) = 0$ if $r > n$ or if $n < 0$. Also, $p(0) = 1$ and $p(n) = 0$ for $n < 0$.

5.1. A canonical representation. Let $S(n)$ be the set of obstructions of order n for $\mathrm{TW}(n-3)$ and let $S(n, q)$ be the set of graphs in $S(n)$ with exactly q vertices of degree less than $n - 2$. Recall from condition (iii) of Theorem 4.3 that $0 \leq q \leq \lfloor \frac{n}{3} \rfloor$. Hence, $S(n) = \dot\bigcup_{q=0}^{\lfloor \frac{n}{3} \rfloor} S(n, q)$ and $|S(n)| = \sum_{q=0}^{\lfloor \frac{n}{3} \rfloor} |S(n, q)|$. We describe a canonical representation for each obstruction in $S(n,q)$ for $\mathrm{TW}(n-3)$.

Consider an obstruction G of order n for $\mathrm{TW}(n-3)$ with q vertices of degree less than $n-2$. Let us label the q vertices of degree less than $n-2$ as $v_1,\ldots,v_q$, and let $V_1=\{v_i : 1\le i\le q\}$. Let $V_2=V-V_1=\{v_i : q+1\le i\le n\}$ be the set of all vertices of degree $n-2$ in G. Since each vertex in V_1 has degree less than $n-2$, $\forall v_i\in V_1$, let $\{(v_i,v_{q+i}),(v_i,v_{2q+i})\}\cap E=\phi$. Then v_{q+i} and v_{2q+i} are adjacent to all other vertices in G. The canonical form is shown in Figure 1(a). For clarity, only the missing edges are shown. All other edges are present.



(a)

(b)

FIG. 1. (a) *The canonical form*; (b) *obstruction for* $\mathrm{TW}(3)$.

Let $V_3=\{v_i : 3q+1\le i\le n\}\subseteq V_2$. If m is the size of the maximum matching in the complement of the subgraph induced by V_3 then $\forall i,\ 1\le i\le m$, let $(v_{3q+i},v_{3q+m+i})\notin E$. Therefore, v_{3q+i} and v_{3q+m+i} are adjacent to all other vertices in G. Clearly, $m\le\lfloor\frac{n-3q}{2}\rfloor$.

Let $V_4=\{v_i : 3q+2m+1\le i\le n\}\subseteq V_3\subseteq V_2$. Since every vertex in V_2-V_4 already has degree $n-2$, for each vertex in V_4 there is exactly one nonneighbor in V_1. There exist $p_j(n-3q-2m)$ distinct mappings of V_4 into j elements of V_1. Suppose $l_1,\ldots,l_j$ is a j-partition of $n-3q-2m$. Then, $\forall i,\ 1\le i\le j\le q$ and $\forall k,\ 1\le k\le l_i$, we let $(v_i,v_{3q+2m+l+k})\notin E$, where $l=l_1+\cdots+l_{i-1}$. Therefore, $\forall v_i\in V_1$, if $1\le i\le j$, then $\delta_i=n-3-l_i$; otherwise $\delta_i=n-3$.

Observe that each graph that satisfies conditions (i)–(iv) of Theorem 4.3, and hence each obstruction in $S(n)$, has a unique representation in the form of a 5-tuple $(n,q,m,j,(l_1,\ldots,l_j))$. This is stated more formally in the following lemma.

LEMMA 5.1. *A graph G of order n is an obstruction for* $\mathrm{TW}(n-3)$ *if and only if G can be uniquely represented by a 5-tuple $(n,q,m,j,(l_1,\ldots,l_j))$, where*

(i) *$n\ge 6$ is the order of the G,*

(ii) *$0\le q\le\lfloor\frac{n}{3}\rfloor$ is the number of vertices of degree $<n-2$ (i.e., $q=|V_1|$),*

(iii) *$\max\{0,3-q\}\le m\le\lfloor\frac{n-3q}{2}\rfloor$ is the size of the maximum matching in the complement of the subgraph induced by those vertices of degree exactly $n-2$ (i.e., $m=|E(\overline{G(V_3)})|$),*

(iv) *$\min\{1,n-3q-2m\}\le j\le\min\{q,n-3q-2m\}$ is the number of vertices of degree less than $n-3$ in G, and*

(v) $l_1 \geq \cdots \geq l_j$ *is a* j*-partition of* $n-3q-2m$ *such that* $n-3-l_1 \leq \cdots \leq n-3-l_j$ *is the degree sequence of the set of vertices of degree less than* $n-3$ *in* G.

Proof. ($\Rightarrow$): The proof of this follows from the preceding description of the canonical form. Given a graph $G \in obs(\mathrm{TW}(n-3))$, the unique 5-tuple representing G is obtained as follows: $n = |V(G)|$; $q =$ the number of vertices of degree $< n-2$ (i.e., $q = |V_1|$); $m =$ the size of the maximum matching in the complement of the subgraph induced by those vertices of degree exactly $n-2$; $j =$ the number of vertices of degree less than $n-3$ in G; and $n-3-l_1 \leq \cdots \leq n-3-l_j$ is the degree sequence of the set of vertices of degree less than $n-3$ in G.

($\Leftarrow$): Now we show that each 5-tuple that satisfies (i)–(v) yields a unique graph $G \in obs(\mathrm{TW}(n-3))$. We start with a complete graph of order n and delete a set of edges from it so that the resulting graph G satisfies conditions (i)–(iv) of Theorem 4.3. Let $V_1 = \{v_1, \ldots, v_q\}$ and let $V_2 = V(G) - V_1$. For each $v_i \in V_1$, delete edges (v_i, v_{q+i}) and (v_i, v_{2q+i}). Let $V_3 = V_2 - \{v_i : q+1 \leq i \leq 3q\}$. Delete the m disjoint edges $(v_{3q+i}, v_{3q+m+i}), 1 \leq i \leq m$ between vertices in V_3. Let $V_4 = V_3 - \{v_{3q+i} : 1 \leq i \leq 2m\}$. For each vertex $v_i, 1 \leq i \leq j$ in V_1, delete the set of edges $\{(v_i, v_{3q+2m+k}) : l_1 + \cdots + l_{i-1} + 1 \leq k \leq l_i\}$. Call the modified graph G.

We claim that G satisfies conditions (i)–(iv) of Theorem 4.3. Observe that only the vertices in V_1 have degree less than $n-2$ and they are all mutually adjacent. Therefore, $\gamma(G) = n-2$. Since each vertex of G has at least one nonneighbor and $q \leq \lfloor \frac{n}{3} \rfloor$, we have maximum degree$(G) = n-2$. Notice that each vertex in V_1 is adjacent to the other $q-1$ vertices in V_1, at least $2(q-1)$ vertices in $V_2 - V_3$ and $2m$ vertices in V_3. Therefore, $\delta(G) \geq 3q-3+2m$. Since $m \geq 0$, we have $\delta(G) \geq 3q-3$. We also need to show that $\delta(G) \geq 4$. When $q = 0$, every vertex in G has degree $n-2 \geq 4$. Condition (iii) of this theorem implies that $q+m \geq 3$. Therefore, $\delta(G) \geq 3q-3+2m = 2(q+m)-3+q \geq 3+q$. Consequently, if $q \geq 1$, then $\delta \geq 4$. Notice that the size of the maximum matching in $\overline{G}$ is $q+m \geq 3$. This completes the proof. □

5.2. An exact formula and an exponential lower bound. Given w, we would like to know the number of obstructions of order $w+3$ for $\mathrm{TW}(w)$. Using Lemma 5.1, we now derive an exact formula for this.

THEOREM 5.2. $|S(n)| = \sum_{q=0}^{\lfloor \frac{n}{3} \rfloor} |S(n,q)|$, *where* $|S(n,q)|$ *is as follows.*
When $q = 0$,

$$|S(n,0)| = \begin{cases} 1 & \textit{for even } n \geq 6, \\ 0 & \textit{otherwise.} \end{cases}$$

When $q = 1$,

$$|S(n,1)| = \begin{cases} 0 & \textit{for } n \leq 6, \\ \sum_{m=2}^{\lfloor \frac{n-3}{2} \rfloor} 1 & \textit{otherwise.} \end{cases}$$

When $q = 2$,

$$|S(n,2)| = \begin{cases} 0 & \textit{for } n \leq 7, \\ \sum_{m=1}^{\lfloor \frac{n-6}{2} \rfloor} \sum_{j=0}^{2} p_j(n-6-2m) & \textit{otherwise.} \end{cases}$$

When $3 \leq q \leq \lfloor \frac{n}{3} \rfloor$,

$$|S(n,q)| = \begin{cases} 0 & \textit{for } n \leq 8, \\ \sum_{m=0}^{\lfloor \frac{n-3q}{2} \rfloor} \sum_{j=0}^{q} p_j(n-3q-2m) & \textit{otherwise.} \end{cases}$$

Table 1
The number of obstructions of order $w+3$ for treewidth w.

Treewidth	1	2	3	4	5	6	7	8	9	10	15	20	25
No. of obstructions	0	0	1	1	3	4	7	9	15	18	79	242	694

Proof. Each 5-tuple that satisfies the conditions stated in Lemma 5.1 represents a unique graph in $S(n)$. Therefore, for every n, the number of distinct 5-tuples equals the number of obstructions in $S(n)$.

When $q = 0$, every vertex in G has degree $n-2$, which implies that n should be even, and G is a complete graph of order n with $\frac{n}{2}$ disjoint edges missing. For even $n \geq 6$, G satisfies all four conditions of Theorem 4.3 and is in $S(n,0)$.

We can see from our canonical representation and the proof of Theorem 4.3 that G is in $S(n)$ whenever $q \geq 1$, $q+m \geq 3$, and $n \geq 3q+2m$. Therefore, for $q \geq 1$ we have the general expression $|S(n,q)| = \sum_{m=\max\{0,3-q\}}^{\lfloor\frac{n-3q}{2}\rfloor} \sum_{j=0}^{q} p_j(n-3q-2m)$. Recall that $p_0(n-3q-2m) \neq 0 \Leftrightarrow n-3q-2m = 0$.

If $q = 1$ then $m \geq 2$ and $n \geq 7$. Therefore, $|S(n,1)| = 0$ for $n \leq 6$. For $n \geq 7$, $|S(n,1)| = \sum_{m=2}^{\lfloor\frac{n-3}{2}\rfloor} \sum_{j=0}^{1} p_j(n-3-2m) = p_0(n-3-2\lfloor\frac{n-3}{2}\rfloor) + p_1(n-3-2\lfloor\frac{n-3}{2}\rfloor) + \sum_{m=2}^{\lfloor\frac{n-3}{2}\rfloor-1} p_1(n-3-2m) = \sum_{m=2}^{\lfloor\frac{n-3}{2}\rfloor} 1$.

If $q = 2$ then $m \geq 1$ and $n \geq 8$. Therefore, $|S(n,2)| = 0$ for $n \leq 7$. For $n \geq 8$, $|S(n,2)| = \sum_{m=1}^{\lfloor\frac{n-6}{2}\rfloor} \sum_{j=0}^{2} p_j(n-6-2m)$.

If $q \geq 3$, then $m \geq 0$ and $n \geq 9$. Therefore, $|S(n,q)| = 0$ when $q \geq 3$ and $n \leq 8$. For $q \geq 3$ and $n \geq 9$, $|S(n,q)| = \sum_{m=0}^{\lfloor\frac{n-3q}{2}\rfloor} \sum_{j=0}^{q} p_j(n-3q-2m)$. □

This completes the proof of consequence A.

We can compute the value of $|S(n)|$ using Theorem 5.2 and the recurrence relation $p_k(n) = p_k(n-k) + p_{k-1}(n-1)$. Table 1 lists some representative values.

For treewidth 3, the graph shown in Figure 1(b) is the only obstruction of order 6. This graph was called M_6 in [6] and $K_{2,2,2}$ in [27]. Curiously, while the entire obstruction set for TW(3) was previously known, the fact that $S(w+3) \subset obs(\mathrm{TW}(w))$ was not suspected.

Similarly, even though the entire obstruction set for PW(2) was known (see [17]), the existence of obstructions of order $w+3$ for PW(w) was previously unknown. This is because $|S(w+3)| = 0$ for $w = 2$, and the general methods in [17] can only produce obstructions of order at least $3w+3$.

It is evident from Table 1 that as the treewidth increases, the number of obstructions increases rapidly. In what follows, we show that $|S(n)|$ grows exponentially in $\lfloor\sqrt{n}\rfloor$.

Corollary 5.3. *For $n \geq 12$, $|S(n)| \;\geq\; p(\lfloor\frac{n}{4}\rfloor - 2)$.*

Proof. For $n \geq 12$, we know that $|S(n)| \geq \sum_{q=3}^{\lfloor\frac{n}{3}\rfloor} |S(n,q)|$. Taking only the $m = 0$ terms of $S(n,q)$ for each $q \geq 3$ and using the fact that $n \geq 12 \Rightarrow \lceil\frac{n}{4}\rceil \geq 3$, we get

$$|S(n)| \geq \sum_{q=3}^{\lfloor\frac{n}{3}\rfloor} \sum_{j=1}^{q} p_j(n-3q) \geq \sum_{q=\lceil\frac{n}{4}\rceil}^{\lfloor\frac{n}{3}\rfloor} \sum_{j=1}^{q} p_j(n-3q).$$

If $q \geq \frac{n}{4}$, then $q \geq n - 3q$ and we have

$$\begin{aligned} |S(n)| &\geq \sum_{q=\lceil \frac{n}{4} \rceil}^{\lfloor \frac{n}{3} \rfloor} \sum_{j=1}^{n-3q} p_j(n-3q) = \sum_{q=\lceil \frac{n}{4} \rceil}^{\lfloor \frac{n}{3} \rfloor} p(n-3q) \\ &\geq p(n - 3\lceil \tfrac{n}{4} \rceil) \\ &\geq p(\lfloor \tfrac{n}{4} \rfloor - 2). \qquad \square \end{aligned}$$

There is no closed-form expression known to compute either $p_k(n)$ or $p(n)$. It is known that as $n \to \infty$, $p(n) \sim \frac{e^{c\sqrt{n}}}{4n\sqrt{3}}$, where $c = \pi\sqrt{\frac{2}{3}}$ (see [1, page 70]). For our purposes, it is sufficient to note that $p(n) \geq 2^{\lfloor \sqrt{n} \rfloor}$ for $n > 1$ (see [25, page 222]). Hence it follows that $|S(n)|$ is bounded below by an exponential function of $\sqrt{n}$. This completes the proof of consequence C.

6. Conclusions. The lower bound for treewidth γ that we presented is a tight bound in the sense that for many families of graphs, γ equals the treewidth. This metric γ enabled us to characterize some of the densest obstructions for treewidth. None of the graphs in $S(w+3)$ was previously known to be an obstruction for either TW(w) or PW(w). We have proven that they are obstructions for both.

Because of Lemma 4.4, we naturally wonder about the extent of intersection between *obs*(TW(w)) and *obs*(PW(w)). It is known (see [16]) that there is a large number of tree obstructions to PW(w) for each $w > 0$. However, since the treewidth of a tree is 1, there cannot be any trees in *obs*(TW(w)) for any $w > 0$. Therefore, *obs*(PW(w)) $\not\subseteq$ *obs*(TW(w)). We can also show that trees are not the only obstructions that distinguish *obs*(PW(w)) from *obs*(TW(w)).

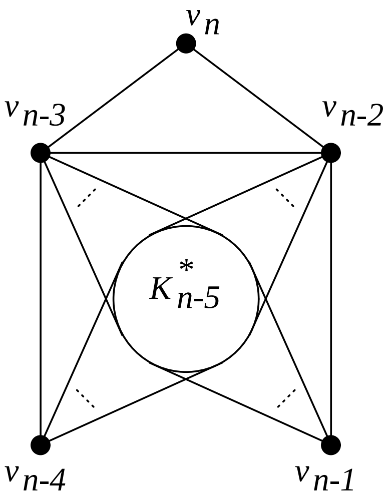


FIG. 2. *An obstruction of order n for* PW($n-4$).

Let K_n^* denote a complete graph of order n from which $\lfloor \frac{n}{2} \rfloor$ disjoint edges have been deleted. If n is odd then an edge incident on the odd vertex is also deleted. For every $n \geq 6$, the graph in Figure 2 is an obstruction of order n for PW($n-4$), but it is not an obstruction for TW($n-4$). This shows that Lemma 4.4 cannot be extended even to $n-4$. Is it true then that *obs*(TW(w)) $\subset$ *obs*(PW(w))? This is certainly true when $w = 1$ or $w = 2$ because in these two cases *obs*(TW(w)) $= \{K_{w+1}\}$. But what about the general case? We conjecture that for every $w \geq 3$, $obs(\mathrm{TW}(w)) \nsubseteq obs(\mathrm{PW}(w))$.

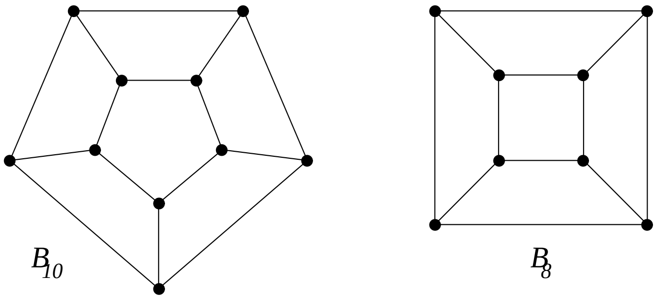


FIG. 3. *A* TW(3) *obstruction containing a* PW(3) *obstruction.*

Consider the prism graphs B_{10} and B_8 in Figure 3. It is known that $B_{10} \in obs(\mathrm{TW}(3))$ (see [6, 27]). We can easily verify that $B_8 \in obs(\mathrm{PW}(3))$. Since $B_8 \leq_m B_{10}$ and $B_8 \in obs(\mathrm{PW}(3))$, it is clear that $B_{10} \notin obs(\mathrm{PW}(3))$. Therefore, $obs(\mathrm{TW}(3)) \nsubseteq obs(\mathrm{PW}(3))$. We believe that such obstructions exist for every $w \geq 3$, and consequently $obs(\mathrm{TW}(w)) \nsubseteq obs(\mathrm{PW}(w))$. However, B_{10} is the only such obstruction we know and we have been unable to generalize B_{10} to values of $w > 3$. Extrapolating from the number of obstructions for PW(3), it seems likely that an obstruction of maximum order for TW(w) would contain as a proper minor an obstruction for PW(w). If this were true, then it would follow that for every $w \geq 3$, $obs(\mathrm{TW}(w)) \nsubseteq obs(\mathrm{PW}(w))$. Unfortunately, we have been unable to verify this conjecture.

Although an explicit knowledge of the obstructions may be helpful in the design of practical algorithms to decide membership in TW(w) or PW(w) (see [15, 21], for instance), the rapid growth of the number of obstructions of order $w + 3$ with increasing w poses a formidable new challenge. One way to surmount this potential difficulty would be to develop general tests for entire families of structurally-related obstructions rather than test for each obstruction individually.

Acknowledgments. We thank Rajeev Govindan, Nancy Kinnersley, and Mike Langston for their helpful comments and encouragement during this research. Thanks also to the anonymous referees for helping to improve the presentation of these results.

REFERENCES

[1] G. E. ANDREWS, *The theory of partitions*, in Encyclopedia of Mathematics and its Applications, Vol. 2, G.-C. Rota, ed., Addison–Wesley, Reading, MA, 1976.

[2] S. ARNBORG, D. G. CORNEIL, AND A. PROSKUROWSKI, *Complexity of finding embeddings in a k-tree*, SIAM J. Algebraic Discrete Meth., 8 (1987), pp. 277–284.

[3] S. ARNBORG, J. LAGERGREN, AND D. SEESE, *Problems easy for tree-decomposable graphs*, J. Algorithms, 12 (1991), pp. 308–340.

[4] S. ARNBORG AND A. PROSKUROWSKI, *Characterization and recognition of partial* 3*-trees*, SIAM J. Algebraic Discrete Meth., 7 (1986), pp. 305–314.

[5] S. ARNBORG AND A. PROSKUROWSKI, *Linear time algorithms for $\mathcal{NP}$-hard problems restricted to partial k-trees*, Discrete Appl. Math., 23 (1989), pp. 11–24.

[6] S. ARNBORG, A. PROSKUROWSKI, AND D. CORNEIL, *Forbidden minors characterization of partial* 3*-trees*, Discrete Math., 80 (1990), pp. 1–19.

[7] H. L. Bodlaender, *A linear time algorithm for finding tree-decompositions of small treewidth*, in Proc. 25th ACM Symposium on Theory of Computing, San Diego, CA, 1993, pp. 226–234.

[8] H. L. Bodlaender, *A tourist guide through treewidth*, Acta Cybernetica, 11 (1993), pp. 1–23.

[9] H. L. Bodlaender and T. Kloks, *Better algorithms for the pathwidth and treewidth of graphs*, in Proc. 18th ICALP, Madrid, Spain, 1991, Lecture Notes in Comput. Sci., 510 (1991), pp. 544–555.

[10] K. Cattell and M. J. Dinneen, *A characterization of graphs with vertex cover up to five*, in Proc. Workshop on Orders, Algorithms and Applications, Lyon, France, 1994, Lecture Notes in Comput. Sci., 831 (1994), pp. 86–99.

[11] K. Cattell, M. J. Dinneen, and M. R. Fellows, *Obstructions to within a few vertices or edges of acyclic*, in Proc. Workshop on Algorithms and Data Structures, Kingston, Ontario, Canada, 1995, Lecture Notes in Comput. Sci., 955 (1995), pp. 415–427.

[12] W. W.-M. Dai and M. Sato, *Minimal forbidden minor characterization of planar* 3*-trees and application to circuit layout*, in Proc. IEEE International Symposium on Circuits and Systems, New Orleans, LA, 1990, pp. 2677–2681.

[13] M. R. Fellows and M. A. Langston, *Nonconstructive tools for proving polynomial time decidability*, J. Assoc. Comput. Mach., 35 (1988), pp. 727–739.

[14] M. R. Fellows and M. A. Langston, *An analogue of the Myhill-Nerode theorem and its use in computing finite-basis characterizations*, in Proc. 30th Symposium on Foundations of Computer Science, Research Triangle Park, NC, 1989, pp. 520–525.

[15] R. Govindan, M. A. Langston, and S. Ramachandramurthi, *A practical approach to layout optimization*, in Proc. 6th International Conference on VLSI Design, Bombay, India, 1993, pp. 222–225.

[16] N. G. Kinnersley, *The vertex separation number of a graph equals its path-width*, Inform. Process. Lett., 42 (1992), pp. 345–350.

[17] N. G. Kinnersley and M. A. Langston, *Obstruction set isolation for the gate matrix layout problem*, Discrete Appl. Math., 54 (1994) pp. 169–213.

[18] A. Kornai and Z. Tuza, *Narrowness, pathwidth, and their application in natural language processing*, Discrete Appl. Math., 36 (1992), pp. 87–92.

[19] J. Lagergren, *An upper bound on the size of an obstruction*, in Graph Structure Theory, N. Robertson and P. Seymour, eds., AMS, Providence, RI, Contemp. Math., 147 (1993), pp. 601–621.

[20] J. Lagergren and S. Arnborg, *Finding minimal forbidden minors using a finite congruence*, in Proc. 18th ICALP, Madrid, Spain, 1991, Lecture Notes in Comput. Sci., 510 (1991), pp. 533–543.

[21] S. Ramachandramurthi, *Algorithms for VLSI Layout Based on Graph Width Metrics*, Doctoral dissertation, Department of Computer Science, University of Tennessee, Knoxville, TN, 1994.

[22] N. Robertson and P. D. Seymour, *Graph minors* II. *Algorithmic aspects of treewidth*, J. Algorithms, 7 (1986), pp. 309–322.

[23] N. Robertson and P. D. Seymour, *Graph minors* IV. *Tree-width and well-quasi-ordering*, J. Combin. Theory Ser. B, 48 (1990), pp. 227–254.

[24] N. Robertson and P. D. Seymour, *Graph minors* XIII. *The disjoint paths problem*, 1986, manuscript.

[25] H. E. Rose, *A Course in Number Theory*, 2nd ed., Clarendon Press, Oxford, 1994.

[26] D. P. Sanders, *On linear recognition of treewidth at most four*, SIAM J. Discrete Math., 9 (1996), pp. 101–117.

[27] A. Satyanarayana and L. Tung, *A characterization of partial* 3*-trees*, Networks, 20 (1990), pp. 299–322.

[28] X. Yan, *A Relative Approximation Algorithm for Computing the Pathwidth of Outerplanar Graphs*, Master's thesis, Department of Computer Science, Washington State University, Pullman, WA, 1989.

SIAM J. Discrete Math. © 1997 Society for Industrial and Applied Mathematics
Vol. 10, No. 1, pp. 158–170, February 1997 012

ON INTEGER MULTIFLOW MAXIMIZATION*

ANDRÁS FRANK[†], ALEXANDER V. KARZANOV[‡], AND ANDRÁS SEBŐ[§]

Abstract. Generalizing the two-commodity flow theorem of Rothschild and Whinston [*Oper. Res.*, 14 (1966), pp. 377–387] and the multiflow theorem of Lovász [*Acta Mat. Akad. Sci. Hungaricae*, 28 (1976), pp. 129–138] and Cherkasky [*Ekonom.-Mat. Metody*, 13 (1977), pp. 143–151], Karzanov and Lomonosov [*Mathematical Programming*, O. I. Larichev, ed., Institute for System Studies, 1978, pp. 59–66] in 1978 proved a min-max theorem on maximum multiflows. Their original proof is quite long and technical and relies on earlier investigations into metrics. The main purpose of the present paper is to provide a relatively simple proof of this theorem. Our proof relies on the locking theorem, which is another result of Karzanov and Lomonosov, and the polymatroid intersection theorem of Edmonds [*Combinatorial Structures and Their Applications*, R. Guy, H. Hanani, N. Sauer, and J. Schönheim, eds., Gordon and Breach, 1970, pp. 69–87]. For completeness, we also provide a simplified proof of the locking theorem. Finally, we introduce the notion of a node demand problem and, as another application of the locking theorem, we derive a feasibility theorem concerning it.

The presented approach gives rise to (combinatorial) polynomial-time algorithms.

Key words. multiflow, polymatroid, network flows, locking, node demands

AMS subject classifications. 05C38, 90B10, 90C27, 05C85

PII. S0895480195287723

1. Introduction.

Let $G = (V, E)$ and $H = (T, F)$ be two undirected graphs so that $T \subseteq V$. We call a path of G *H-admissible* if it connects two nodes x, y of T so that $xy \in F$. G will be called a *supply graph,* H a *demand graph*, and the elements of T *terminals* while the other elements of V are called *inner nodes.* The *maximization problem* consists of finding a maximum number of edge-disjoint H-admissible paths. If H consists of one edge, then Menger's theorem gives an answer.

In general, the problem is NP-complete even in the special case when G is Eulerian. (A graph is called *Eulerian* if the degree of every node is even.) There are, however, important special cases when the problem is tractable. Rothschild and Whinston [13] proved a max-flow-min-cut-type theorem when (G, T) is inner Eulerian and H consists of two edges. (We say that the pair (G, T) is *inner Eulerian* if the degree $d(v)$ is even for every inner node v.) Another result is due, independently, to Lovász [12] and Cherkasky [1]. They solved the maximization problem when H is a complete graph and G is inner Eulerian. In [8] Karzanov and Lomonosov found a common generalization of these two theorems. Their original proof is rather lengthy and technical and it is certainly much more difficult than those of the two special cases mentioned above. Details of these proofs were described in [4] and [10, 11]. Later Karzanov [6, 7] gave another proof which was based on the splitting-off technique and

* Received by the editors March 21, 1995; accepted for publication (in revised form) March 21, 1996.

http://www.siam.org/journals/sidma/10-1/28772.html

[†] Department of Operations Research, Eötvös University, Múzeum krt. 6-8, Budapest, Hungary H-1088. The research of this author was partially supported by Hungarian National Foundation for Scientific Research grant OTKA T17580 and was partially carried out while the author visited Leibniz-imag and was supported by the Région Rhône-Alpes.

[‡] Institute for System Analysis, Prospect 60 Let Okyabrya 9, 117312 Moscow, Russia (karzanov@cs.vniisi.msk.su).

[§] CNRS, Leibniz-imag, 46 avenue Felix Viallet, 38031 Grenoble cedex 1, France (andras.sebo@imag.fr). The research of this and the first author was partially supported by the European project DONET.

gave rise to a strongly polynomial solution algorithm. However, this latter proof was also rather complicated.

The main contribution of this paper is a relatively simple proof of the theorem of Karzanov and Lomonosov. The proof relies on two ingredients: the so-called locking theorem, which is another result of Karzanov and Lomonosov [8], and the polymatroid intersection theorem of Edmonds [2]. For completeness, we will also provide a simplified proof of the locking theorem. Since both of these ingredients can be solved by a (combinatorial) polynomial-time algorithm, the approach gives rise to an alternate strongly polynomial-time algorithm for the (capacitated) maximization problem in question which is faster than that in [6].

In what follows we do not distinguish between a one-element set $\{x\}$ and its only element x. For a set X and an element t let $X+t$ denote the union of X and t. For a vector $m : S \to \mathbf{R}$ we use the notation $m(X) := \sum(m(s) : s \in X)$. A family of pairwise disjoint nonempty subsets of a set S is called a *subpartition* of S. For two elements s, t a set X is called a *$t\bar{s}$-set* if $t \in X, s \notin X$. An integer-valued vector or function is called *even* if each of its values is an even integer. For a polyhedron P we use the notation $P/2 := \{x/2 : x \in P\}$.

For a graph $G = (V, E)$ the cut $[X, V - X]$ denotes the set of edges with precisely one end node in X. Its cardinality is denoted by $d(X)(= d(V - X))$. $d(X)$ is called the *degree function* of G. Let $d(X, Y)$ denote the number of edges with one in $X - Y$ and the other in $Y - X$. Let $\bar{d}(X, Y) := d(X \cap Y, V - (X \cup Y))$. It is easy to prove that d satisfies the following identities for every pair X, Y of subsets of V:

$$d(X) + d(Y) = d(X \cap Y) + d(X \cup Y) + 2d(X, Y), \tag{1.1}$$

$$d(X) + d(Y) = d(X - Y) + d(Y - X) + 2\bar{d}(X, Y). \tag{1.2}$$

Let A and B be two disjoint subsets of V. A path connecting an element of A and an element of B is called an *(A, B)-path.* A path connecting two distinct elements of A is called an *A-path.* $\lambda(A, B; G)$ or simply $\lambda(A, B)$ stands for the maximum number of edge-disjoint (A, B)-paths. By Menger's theorem $\lambda(A, B) = \min(d(X) : A \subseteq X \subseteq V - B)$.

One may consider a fractional version of the edge-disjoint paths problem. Let G and H be as before. By an *H-multiflow* or briefly *multiflow* x we mean a family $\{P_1, P_2, \ldots, P_k\}$ of paths of G along with nonnegative coefficients $\alpha_1, \alpha_2, \ldots, \alpha_k$ so that each P_i connects the end nodes of a demand edge. x is called *integer-valued* if each α_i is an integer.

If each P_i connects an element of A and an element of B (that is, when H is a complete bipartite graph with bipartition (A, B)), we speak of an *(A, B)-flow.* For an H-multiflow x let $x(e) := \sum(\alpha_i : P_i \text{ uses } e)$ $(e \in E)$ and $x(t) := \sum(\alpha_i : P_i \text{ ends at } t)$ $(t \in T)$. For a given capacity function $c : E \to \mathbf{R}_+$, x is called *c-admissible* if $x(e) \le c(e)$ for every $e \in E$.

2. The locking problem.

Let $G = (V, E)$ be a graph and $T \subseteq V$ a subset of terminal nodes. For a subset $A \subseteq T$ the notation $\lambda(A, T - A; G)$ will be abbreviated by $\lambda(A; G)$ or by $\lambda(A)$ when no confusion can arise. Throughout the paper we assume that the current (G, T) is inner Eulerian.

Lovász [12] and Cherkasky [1] proved the following theorem.

THEOREM 2.1. *For an inner Eulerian pair (G, T) the maximum number of edge-disjoint T-paths is equal to $(\sum \lambda(t) : t \in T)/2$.*

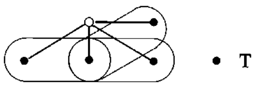


FIG. 1.

An equivalent formulation now follows.

THEOREM 2.1′. *Given an inner Eulerian pair* (G, T), *there is a family* $\mathcal{F}$ *of edge-disjoint* T*-paths in* G *so that* $\mathcal{F}$ *contains* $\lambda(t)$ *paths ending at* t *for each* $t \in T$.

In other words, there is a single family of edge-disjoint T-paths that includes maximum families of edge-disjoint $(t, T - t)$-paths simultaneously for all $t \in T$.

Karzanov and Lomonosov [8] extended this theorem. To formulate their result let us say that a family $\mathcal{F}$ of edge-disjoint T-paths *locks a subset* $A \subseteq T$ if $\mathcal{F}$ contains $\lambda(A)$ $(A, T - A)$-paths. Furthermore, we say that $\mathcal{F}$ *locks a family* $\mathcal{L}$ of subsets of T if $\mathcal{F}$ locks all members of $\mathcal{L}$.

Theorem 2.1′ asserts that there is a family $\mathcal{F}$ of paths that locks all singletons of T. Is it always possible to find a family of edge-disjoint T-paths that locks a specified family $\mathcal{L}$? The answer, in general, is no, as is shown by Figure 1. Here $\mathcal{L}$ consists of three pairwise crossing sets. (Two subsets X, Y of T are called *crossing* if none of $X - Y, Y - X, X \cap Y, T - (X \cup Y)$ is empty.)

Figure 1 indicates why it is natural to require $\mathcal{L}$ to be 3-cross-free. A family $\mathcal{L}$ of subsets of T is called 3*-cross-free* if it has no three pairwise crossing members.

LOCKING THEOREM 2.2 (see [8, 5, 10, 11]). *Let* (G, T) *be inner Eulerian and* $\mathcal{L}$ *a* 3*-cross-free family of subsets of* T. *Then there is a family of edge-disjoint* T*-paths that locks* $\mathcal{L}$.

A proof of a slightly weaker version was sketched in [8]. The present proof relies on an idea of splitting used previously in [5], but is technically simpler. *Splitting off* a pair of adjacent edges $e = st, f = sx$ of a graph G refers to an operation that replaces e and f by a new edge connecting x and t (this way we may introduce parallel edges between x and t). The resulting graph is denoted by G^{ef}.

Proof. We may assume that $T - A \in \mathcal{L}$ for each $A \in \mathcal{L}$ because for $A \in \mathcal{L}$ adding $T - A$ to $\mathcal{L}$ affects neither 3-cross-freeness nor lockability. Also assume that G is connected.

We proceed by induction on the number of edges incident to the elements of $V - T$. If this number is zero, then the statement is trivial. Therefore, there is an edge $e = st$ with $t \in T, s \notin T$. We are going to show that there is an edge $f = sx$ for which

$$(2.1) \qquad \lambda(A; G) = \lambda(A; G^{ef}) \quad \text{for every } A \in \mathcal{L}.$$

From this the theorem follows since, by induction, there is a family $\mathcal{F}$ of T-paths of G^{ef} locking $\mathcal{L}$. If a path $P \in \mathcal{F}$ uses the new edge h of G^{ef} having arisen from the splitting of e, f, then revise $\mathcal{F}$ by replacing h in P by e and f. By (2.1) the revised $\mathcal{F}$ locks $\mathcal{L}$ in G.

CLAIM 1. *Suppose for* $X, Y \subseteq V$ *that* $X \cap T \subseteq Y \cap T$ *and that* $d(X) = \lambda(X \cap T), d(Y) = \lambda(Y \cap T)$. *Then* $d(X \cap Y) = \lambda(X \cap T), d(X \cup Y) = \lambda(Y \cap T)$ *and* $d(X, Y) = 0$.

Proof. Since $X \cap T \subseteq Y \cap T$ we have $(X \cap Y) \cap T = X \cap T$ and hence $d(X \cap Y) \geq \lambda(X \cap T)$. Analogously, $(X \cup Y) \cap T = Y \cap T$ and $d(X \cup Y) \geq \lambda(Y \cap T)$. Therefore, by (1.1), $\lambda(X \cap T) + \lambda(Y \cap T) = d(X) + d(Y) = d(X \cap Y) + d(X \cup Y) + 2d(X, Y) \geq \lambda(X \cap T) + \lambda(Y \cap T) + 2d(X, Y)$, from which the claim follows. □

Call a set $X \subseteq V$ *tight* if $X \cap T \in \mathcal{L}$ and $d(X) = \lambda(X \cap T)$. Since $\mathcal{L}$ is closed under complementation, $V - X$ is tight if X is tight. Because (G, T) is inner Eulerian, a pair of edges $e = st, f = sx$ will satisfy (2.1) precisely if

(2.2) there is no tight set X with $t, x \in X \subseteq V - s$.

CLAIM 2. *There are no three maximal tight $t\bar{s}$-sets.*

Proof. Let X, Y, Z be maximal tight $t\bar{s}$-sets. Since $\mathcal{L}$ is 3-cross-free, two of the three sets $X \cap T, Y \cap T, Z \cap T$, say $X \cap T$ and $Y \cap T$, are noncrossing.

Then either $X \cap T \subseteq Y \cap T$ or $Y \cap T \subseteq X \cap T$ or $T \subseteq X \cup Y$. In the first two cases Claim 1 implies that $X \cup Y$ is tight, contradicting the maximality of X and Y. In the last case, by applying Claim 1 to $X' = V - X$ and Y, we obtain that $d(X', Y) = 0$, contradicting the existence of edge st. □

Let S denote the set of neighbors of s.

CLAIM 3. *It is not possible to cover S by two tight $t\bar{s}$-sets.*

Proof. Suppose that $S \subseteq X \cup Y$, where X and Y are tight $t\bar{s}$-sets. Let $\alpha := d(s, X - Y), \beta := d(s, Y - X), \gamma := d(s, X \cap Y)$. By symmetry we may assume that $\alpha \geq \beta$. $(X + s) \cap T = X \cap T$ implies that $d(X + s) \geq \lambda(X \cap T)$. On the other hand, since γ is positive, we have $d(X + s) = d(X) - \alpha - \gamma + \beta < d(X) = \lambda(X \cap T)$, which is a contradiction. □

By Claims 2 and 3 there is an edge $f = sx$ satisfying (2.2), and then (2.1) holds; the proof of Locking Theorem 2.2 is complete.

Remark. One may be interested in other possible locking theorems when, rather than 3-cross-freeness, some other property is assumed for the family $\mathcal{L} \subseteq 2^T$ to be locked. On the negative side, Karzanov and Pevzner [9] showed that for every $\mathcal{L}$, including three pairwise crossing sets, there is a graph G and a subset T of its nodes so that (G, T) is inner Eulerian and there is no family of T-paths locking all members of $\mathcal{L}$. On the other hand, there are other locking theorems in which some restrictions are imposed on the relationship of G and the family $\mathcal{L}$. For example, let G be a planar Eulerian graph and let $T := \{t_1, \dots, t_k\}$ denote the nodes of its outer face in the cyclic order. If we define $\mathcal{L}$ to consist of all subsets of T of form $\{t_i, \dots, t_j\}$ $(1 \leq i \leq j \leq k)$ then, although $\mathcal{L}$ is not 3-cross-free when $k \geq 4$, the locking theorem holds. (This is a theorem equivalent, by planar dualization, to a result of Hurkens, Schrijver, and Tardos [3].

We will need a slight extension of Theorem 2.2. Let $m : T \to \mathbf{Z}$ be a nonnegative integer-valued function on T. A family $\mathcal{F}$ of edge-disjoint T-paths is called *m-independent* if every terminal $t \in T$ is the end of at most $m(t)$ members of $\mathcal{F}$. Let $\lambda_m(A)$ denote the maximum number of edge-disjoint m-independent $(A, T-A)$-paths. We say that a family $\mathcal{F}$ of edge-disjoint T-paths *m-locks a subset* $A \subseteq T$ if $\mathcal{F}$ is m-independent and contains $\lambda_m(A)$ $(A, T - A)$-paths. Furthermore, we say that $\mathcal{F}$ *m-locks a family* $\mathcal{L}$ of subsets of T if $\mathcal{F}$ m-locks all members of $\mathcal{L}$.

The following theorem is a straightforward consequence of Theorem 2.2 and will be used in the proof of Theorem 4.3.

THEOREM 2.3. *Let G be inner Eulerian and $\mathcal{L}$ a 3-cross-free family of subsets of T. Let $m : T \to \mathbf{Z}_+$ be a vector so that $m(t) + d(t)$ is even for $t \in T$. Then there is a family $\mathcal{F}$ of edge-disjoint T-paths that m-locks $\mathcal{L}$.*

Proof. Let G' be a graph arising from G by splitting every node $t \in T$ in the following way: add a new node t' along with $m(t)$ parallel edges between t and t' and replace each edge xt of G by xt'. The result immediately follows when Theorem 2.2 is applied to (G', T). □

3. Flows and polymatroids. A nonnegative set function $b : 2^T \to \mathbf{R}_+$ is called a *polymatroid function* if

1. $b(\emptyset) = 0$,
2. b is monotone increasing, i.e., $b(X) \geq b(Y)$ when $Y \subseteq X \subseteq T$,
3. b is submodular, i.e., $b(X) + b(Y) \geq b(X \cup Y) + b(X \cap Y)$ for $X, Y \subseteq T$.

The degree function d of a graph G satisfies properties 1 and 3 but typically not 2.

A polyhedron $P(b) := \{x \in \mathbf{R}^T, x \geq 0, x(A) \leq b(A)$ for every $A \subseteq T\}$ is called a *polymatroid*. It is called *integral* if every vertex of P is integer-valued.

The concept of a polymatroid was introduced by Edmonds [2]. He proved that a polymatroid uniquely determines its defining polymatroid function. Furthermore, a polymatroid is integral if and only if b is integer-valued.

For a polymatroid $P(b)$ the face $B(b) := \{x : x \in P, x(T) = b(T)\}$ of $P(b)$ is called the *basis polyhedron* and its elements are the *bases*. Edmonds also proved the following result.

THEOREM 3.1 (see [2]). *For an (integral) polymatroid $P(b)$ and an (integer-valued) vector $x \in P(b)$ there is an (integer-valued) basis y with $y \geq x$.*

The polymatroid intersection theorem of Edmonds states that the linear system of two polymatroids is totally dual integral (TDI). Here we need only the following consequence.

THEOREM 3.2 (see [2]). *For two polymatroid functions a and b defined on the power set of T*

$$\max(x(T) : x \in P(a) \cap P(b)) = \min(a(X) + b(T - X) : X \subseteq T).$$

Furthermore, if a and b are integer-valued, the maximum is attained by an integer vector.

It follows that there is a vector x in $P(a) \cap P(b)$ and a bipartition $\{A, B\}$ of T so that $x(A) = a(A)$ and $x(B) = b(B)$, and if a and b are integral-valued, then so is x.

Let $G = (V, E)$ be a graph endowed with a capacity function $c : E \to \mathbf{R}_+$. Let T be a subset of nodes and $A \subset T, B := T - A$. Define $P_A := \{m \in \mathbf{R}_+^{\mathbf{A}}$: there is a c-admissible (A, B)-flow x for which $x(v) = m(v)$ for every $v \in A\}$.

For $X \subseteq A$ let $f_A(X) := \min(\delta_c(Y) : Y \subseteq V, X \subseteq Y \cap T \subseteq A)$. Here $\delta_c(Y) := \sum(c(e) : e \in [Y, V - Y])$. Clearly, f_A is submodular and monotone increasing. By a multiterminal version of the max-flow min-cut (MFMC) theorem a vector $m \in R_+^A$ belongs to P_A if and only if $m(X) \leq f_A(X)$. Therefore, P_A is a polymatroid. Furthermore, if c and m are integer-valued, then there is a c-admissible integer-valued (A, B)-flow x for which $x(v) = m(v)$ for every $v \in A$.

Let $G = (V, E)$ be an Eulerian graph and T a subset of nodes. Define c by $c(e) = 1$ for every $e \in E$. Let $\mathcal{T} := \{T_1, T_k, \ldots, T_k\}$ be a partition of T and $\lambda_i := \lambda(T_i, T - T_i)$. Let P denote the direct sum of polymatroids $P_{T_1}, P_{T_2}, \ldots, P_{T_k}$.

LEMMA 3.3. *Let q be an integer basis of P. Then there is a family $\mathcal{F}$ of edge-disjoint T-paths connecting distinct members of $\mathcal{T}$ so that each $t \in T$ is the end point of exactly $q(t)$ paths of $\mathcal{F}$.*

Proof. For each $T_i \in \mathcal{T}$ let X_i be a minimal subset of V for which $X_i \cap T = T_i$ and $d(X_i) = \lambda_i$. We claim that these sets are disjoint. If, indirectly, $X_i \cap X_j \neq \emptyset$ for some $1 \leq i < j \leq k$, then (1.2) implies $\lambda_i + \lambda_j \leq d(X_i - X_j) + d(X_j - X_i) \leq d(X_i) + d(X_j) = \lambda_i + \lambda_j$. Hence $\lambda_i = d(X_i - X_j)$, contradicting the minimality of X_i.

We claim that there is a family $\mathcal{F}_0$ of edge-disjoint paths in G connecting distinct X_i's and not using edges induced by any X_i so that $\mathcal{F}_0$ contains $\lambda_i = d(X_i)$ paths ending in X_i for each i $(1 \leq i \leq k)$. Indeed, apply Theorem 2.1$'$ to the pair (G', T'), where the graph G' arises from G by contracting each X_i into a node denoted by t_i and $T' := \{t_1, \dots, t_k\}$.

Since q is a basis of P, for each T_i there is a family $\mathcal{F}'_i$ of $q(T_i)$ $(= \lambda_i = d(X_i))$ edge-disjoint paths in G connecting T_i and $T - T_i$, so that each $t \in T_i$ is the end node of $q(t)$ members of $\mathcal{F}'_i$. For each member of $\mathcal{F}'_i$ erase the edges outside X_i and denote by $\mathcal{F}_i$ the family of the resulting paths. By glueing together the paths in $\mathcal{F}_0$ and the paths in $\mathcal{F}_i$ $(i = 1, \dots, k)$ we obtain a family $\mathcal{F}$ of paths satisfying the requirements. □

LEMMA 3.4. *Let q be an integer basis of P and m an integer vector for which $m \geq q$. Then an m-independent family $\mathcal{F}$ of T-paths that m-locks $\mathcal{T}$ contains at least $q(T)/2$ paths connecting distinct members of $\mathcal{T}$.*

Proof. By Lemma 3.3, $\lambda_q(T_i) = q(T_i)$. The assumption $m \geq q$ implies that $\lambda_m \geq \lambda_q$. Since $\mathcal{F}$ m-locks $\mathcal{T}$, there are $\lambda_m(T_i) \geq \lambda_q(T_i) = q(T_i)$ paths in $\mathcal{F}$ connecting T_i and $T - T_i$ for each $i = 1, \dots, k$, from which the lemma follows. □

Remark. Since q is a basis, $\mathcal{F}$ contains at most $q(T_i)$ $(T_i, T - T_i)$-paths, and therefore $\mathcal{F}$ contains at most $q(T)/2$ paths connecting distinct members of $\mathcal{T}$. That is, the number of such paths in $\mathcal{F}$ is precisely $q(T)/2$, but we will not need this fact.

Let $\mathcal{A}$ and $\mathcal{B}$ be two partitions of T and let $\mathcal{L} := \mathcal{A} \cup \mathcal{B}$. Let H be a demand graph on T so that uv is an edge of H if and only if no $X \in \mathcal{L}$ includes both u and v.

For $A_i \in \mathcal{A}$ let $a_i(X)$ $(X \subseteq A_i)$ be a set function defined by $a_i(X) := \lambda(X, T - A_i)$. We saw above that a_i is a polymatroid function. Define b_j analogously for $\mathcal{B}$. For $X \subseteq T$ let

$$a(X) := \sum a_i(X \cap A_i) \quad \text{and} \quad b(X) := \sum b_j(X \cap B_j). \tag{3.1}$$

Then a and b are polymatroid functions. Let $P(a)$ and $P(b)$ be the polymatroids defined by a and b, respectively.

LEMMA 3.5. *Let m' be an arbitrary even vector in $P(a) \cap P(b)$ and $h := m'(T)/2$. Then there are h edge-disjoint H-admissible paths.*

Proof. Since G is Eulerian, $P(a/2)(= P(a)/2)$ is an integral polymatroid. By applying Theorem 3.1 to $P(a/2)$ and to $x := m'/2$ we find that there is an even basis m_a of $P(a)$ so that $m_a \geq m'$. Analogously, there is an even basis m_b of $P(b)$ so that $m_b \geq m'$. Define a vector m by $m(t) := \max(m_a(t), m_b(t))$ for $t \in T$. Clearly, m is even and $m_a(t) + m_b(t) \geq m(t) + m'(t)$ for each $t \in T$. Hence

$$m_a(T) + m_b(T) - m(T) \geq m'(T).$$

Since $\mathcal{L}$ is 3-cross-free, we can apply Theorem 2.3. Let $\mathcal{F}$ denote the family of m-independent T-paths provided by the theorem. Then $|\mathcal{F}| \leq m(T)/2$.

We are going to prove that the number h' of H-admissible paths in $\mathcal{F}$ is at least h. (Note that a path is not H-admissible precisely if it connects two nodes belonging to the same member of $\mathcal{L}$.)

By applying Lemma 3.4 with the choice $\mathcal{T} := \mathcal{A}$, $P := P(a), q := m_a$, we find that there are at most $|\mathcal{F}|$ $-m_a(T)/2$ paths in $\mathcal{F}$ having both end nodes in the same member of $\mathcal{A}$. Analogously, there are at most $|\mathcal{F}|$ $-m_b(T)/2$ paths in $\mathcal{F}$ having both end nodes in the same member of $\mathcal{B}$.

Hence $h' \geq |\mathcal{F}| - (|\mathcal{F}| - m_a(T)/2) - (|\mathcal{F}| - m_b(T)/2) = (m_a(T) + m_b(T))/2 - |\mathcal{F}| \geq (m_a(T) + m_b(T) - m(T))/2 \geq m'(T)/2 = h$, as required. □

(Note that an element $t \in T$ need not be the end node of exactly $m'(t)$ members of the family assured by Lemma 3.5. For further comments, see the beginning of section 6.)

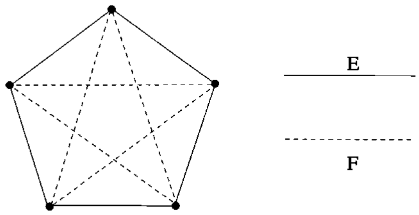


FIG. 2.

4. Maximization. Let $G = (V, E)$ be a supply graph and $H = (T, F)$ a demand graph so that $T \subseteq V$ and $E \cap F = \emptyset$. Throughout this section we assume that the pair (G, T) is inner Eulerian; that is, $d(v)$ is even for every $v \in V - T$, where d stands for the degree function of G.

The *maximization form* of the edge-disjoint paths problem consists of finding a maximum number $\mu = \mu(G, H)$ of edge-disjoint H-admissible paths. We can easily get an upper bound on μ. Let us call a subpartition $\{X_1, X_2, \ldots, X_k\}$ of V *admissible* if $T \subseteq \cup X_i$ and each $X_i \cap T$ is stable in $H (i = 1, \ldots, k)$. Clearly,

$$\mu(G, H) \leq \sum d(X_i)/2. \tag{4.1}$$

The value $\sum d(X_i)/2$ will be called the *value of the subpartition.* Let $\tau = \tau(G, H)$ denote the minimum value of an admissible subpartition. We have $\mu \leq \tau$.

Figure 2 shows that we do not have equality, in general.

There are two known special cases when equality holds. Theorem 2.1 shows that this is the case if H is a complete graph on T. Reformulating Theorem 2.1, we have the following theorem.

THEOREM 4.1. *Suppose that (G, T) is inner Eulerian and the demand graph H is complete. Then $\mu(G, H) = \tau(G, H)$.*

Another special case for which $\mu = \tau$ is when H consists of two edges; that is, $H = 2K_2$.

THEOREM 4.2. *Suppose that (G, T) is inner Eulerian and H consists of two edges $s_i t_i$ $(i = 1, 2)$. Then $\mu(G, H) =\tau(G, H)$.*

This is a theorem of Rothschild and Whinston. Actually, they proved it in the following simpler form.

THEOREM 4.2′ (see [13]). *Suppose that (G, T) is inner Eulerian and H consists of two edges $s_i t_i (i = 1, 2)$. Then $\mu(G, H)$ is the minimum cardinality τ' of a cut $[X, V - X]$ of G for which $\{s_i, t_i\} \cap X = 1$ $(i = 1, 2)$.*

(The equivalence of the two forms, that is, $\tau = \tau'$, may be proven as follows. Since a cut $[X, V - X]$ for which $\{s_i, t_i\} \cap X = 1$ $(i = 1, 2)$ provides an admissible partition of special form, clearly $\tau' \geq \tau$. To see the other direction let $\mathcal{P} := \{X_1, \ldots, X_k\}$ be a minimal admissible subpartition of G for which k is minimum. Then $\tau = \sum d(X_i)/2$, and since $|T| \leq 4$, we have $2 \leq k \leq 4$.

If $k = 2$, then both X_1 and X_2 contain exactly two terminal nodes which are not connected in H. Furthermore, if say $d(X_1) \leq d(X_2)$, then $\{X_1, V - X_1\}$ would also be an admissible partition whose value is not bigger than that of $\{X_1, X_2\}$. Therefore, $\{X_1, V - X_1\}$ is another optimal admissible partition and hence $\tau' = \tau$.

If $k \geq 3$, then there are two members of $\mathcal{P}$, say X_1 and X_2, such that each contains one terminal node and these two terminal nodes, say s_1 and s_2, are not connected in H. But now by replacing X_1 and X_2 by $X_1 \cup X_2$ we obtain another minimal admissible subpartition, contradicting the minimum choice of k. $\Box$)

Let us call a graph $H = (T, F)$ *bistable* if there are two partitions $\mathcal{A}$ and $\mathcal{B}$ of T such that for $x, y \in T$ xy is an edge of H precisely if x and y belong to different parts of $\mathcal{A}$ and different parts of $\mathcal{B}$. It is easily seen that a graph is bistable if and only if its complement is the line graph of a bipartite graph. (It can also be shown that bistable graphs are those for which the family of maximal stable sets of H can be partitioned into two parts, each consisting of disjoint sets.)

Clearly, a clique or, more generally, a complete k-partite graph, is bistable and $2K_2$ is also bistable. Therefore, Theorems 2.1 and 2.2 are special cases of the following.

THEOREM 4.3 (see [6, 10, 11]). *Suppose that (G, T) is inner Eulerian and $H = (T, F)$ is bistable. Then $\mu(G, H) = \tau(G, H)$.*

A proof of a slightly weaker, half-integral version was previously sketched in [8]. The reader may feel that bistable demand graphs form a rather peculiar class of graphs and there may be larger, more natural classes of graphs for which $\mu = \tau$ holds. Karzanov and Pevzner [9], however, showed that if $H = (T, F)$ is not bistable and contains no isolated nodes, then there is a supply graph $G = (V, E)$, with $T \subseteq V$ and (G, T) inner Eulerian, so that $\mu(G, H) < \tau(G, H)$.

In section 6 we will outline our original plan of proof, which was intended to use only Theorem 3.2, and we will point out why that attempt failed. This perhaps will help the reader understand how we were led to invoke the locking theorem in the proof below.

Proof. By (4.1) we have $\mu(G, H) \leq \tau(G, H)$. To see the other direction, first we prove that the theorem follows from its special case when the graph is completely Eulerian. So suppose the theorem is true for (G', H') whenever G' is Eulerian and we want to prove it for (G, H) when G is inner Eulerian. Let K denote the set of nodes of G with odd degree. Since (G, T) is inner Eulerian, $K \subseteq T$. If K is empty, we are done. If not, for a new node t, let $T' := T + t$ and $V' := V + t$. Let $E' := E \cup \{xt : x \in K\}$ and $F' := F \cup \{xt : x \in T\}$. Then $G' := (V', E')$ is Eulerian and $H' := (T', F')$ is bistable. Let μ' and τ' denote, respectively, the maximum and minimum in question concerning (G', H'). By the assumption $\mu' = \tau'$.

Obviously, there is an optimal solution to the maximization problem concerning (G', H') in which every edge $xt,\ x \in K$, is itself a path in the solution. Thus we have $\mu' = \mu + |K|$. Furthermore, let $\mathcal{M}'$ be an optimal admissible subpartition for (G', H') so that $t \in X \in \mathcal{M}$. Since every edge $xt,\ x \in T$, belongs to $H',\ X \cap T = \{t\}$. Hence $\mathcal{M} - \{X\}$ is an admissible subpartition for (G, H), and therefore $\tau \leq \tau' - |K|$. We can conclude that $\mu = \mu' - |K| = \tau' - |K| \geq \tau$, as required.

Let $\mathcal{A}$ and $\mathcal{B}$ be the two partitions of T defining the bistable graph H. Note that each stable set of H is a subset of some $S \in \mathcal{A} \cup \mathcal{B}$. Let a and b be defined by (3.1). Since $P(a)/2$ and $P(b)/2$ are integral polymatroids, by Theorem 3.2 there exist an even vector m' in $P(a) \cap P(b)$ and a bipartition $\{A, B\}$ of T so that

$$m'(A) = a(A) \quad \text{and} \quad m'(B) = b(B). \tag{4.2}$$

Hence we have $m'(T) = a(A) + b(B)$. By Lemma 3.5 there are $m'(T)/2$ edge-disjoint H-admissible paths in G. Thus the proof will be complete if we find an admissible subpartition of value $(a(A) + b(B))/2$. To this end let us assume that A is a maximal subset of T for which A and $B := T - A$ satisfy (4.2). We claim that

$$a(A + t) > a(A) \quad \text{for every element } t \in B. \tag{4.3}$$

Indeed, if we have $a(A+t) = a(A)$ for some element $t \in B$, then $a(A) = m'(A) \leq m'(A+t) \leq a(A+t) = a(A)$, from which $m'(A+t) = a(A+t)$ and $m'(t) = 0$. Furthermore, $b(B) = m'(B) = m'(B-t) \leq b(B-t) \leq b(B)$, and hence $m'(B-t) = b(B-t)$; that is, the bipartition $\{A+t, B-t\}$ of T would also satisfy (4.2), contradicting the maximal choice of A. (Note that, because of this choice of A and B, the role of $\mathcal{A}$ and $\mathcal{B}$ will not be fully symmetric.)

For each $A_i \in \mathcal{A}$ for which $A \cap A_i$ is nonempty there exists a set $X_i \subseteq V$ for which $A_i \cap A \subseteq X_i \cap T \subseteq A_i$ and $d(X_i) = a(A_i \cap A) = m'(A_i \cap A)$. Here the last equality follows from (4.2) and the definition of a. Analogously, for each $B_j \in \mathcal{B}$ for which $B \cap B_j$ is nonempty there exists a set Y_j for which $B_j \cap B \subseteq Y_j \cap T \subseteq B_j$ and $d(Y_j) = b(B_j \cap B) = m'(B_j \cap B)$. Assume that both X_i and Y_j are chosen minimal and let $\mathcal{K} := \{X_i : A_i \in \mathcal{A},\ A_i \cap A \text{ nonempty}\} \cup \{Y_j : B_j \in \mathcal{B}, B_j \cap B \text{ nonempty}\}$.

LEMMA 4.4. *$\mathcal{K}$ is an admissible subpartition of value $(a(A) + b(B))/2$.*

Proof. Clearly, each element of T belongs to at least one member of $\mathcal{K}$, and we show that no more than one. That is, we claim that

$$X_i \cap T \subseteq A \tag{4.4a}$$

and

$$Y_j \cap T \subseteq B. \tag{4.4b}$$

We have $m'(A_i \cap A) = a(A_i \cap A) = d(X_i) \geq a(A_i \cap X_i) \geq m'(A_i \cap X_i) \geq m'(A_i \cap A)$, and hence $a(A_i \cap A) = a(A_i \cap X_i)$. Hence (4.4a) must hold, for otherwise there is an element $t \in (X_i \cap T) - A$ and t would violate (4.3).

Also, $m'(B_j \cap B) = b(B_j \cap B) = d(Y_j) \geq b(B_j \cap Y_j) \geq m'(B_j \cap Y_j) \geq m'(B_j \cap B)$, and hence $m'(t) = 0$ for every $t \in Y_j \cap A$. We have $m'(X_i \cap A) + m'(Y_j \cap B) = m'(A_i \cap A) + m'(B_j \cap B) = d(X_i) + d(Y_j) \geq d(X_i - Y_j) + d(Y_j - X_i) \geq a((X_i - Y_j) \cap A) + b((Y_j - X_i) \cap B) \geq m'((X_i - Y_j) \cap A) + m'((Y_j - X_i) \cap B) = m'(X_i \cap A) + m'(Y_j \cap B)$. Hence $d(Y_j') = b(Y_j' \cap B) = m'(Y_j' \cap B)$ holds for $Y_j' := Y_j - X_i$. Therefore, if (4.4b) is not true and there is an element $t \in (Y_j \cap T) - B$ which belongs to, say X_i, then Y_j' is a proper subset of Y_j, contradicting the minimal choice of Y_j. Hence the proof of (4.4) is complete.

We claim that $\mathcal{K}$ is a subpartition. Assume to the contrary that $L \cap K \neq \emptyset$ for some $K, L \in \mathcal{K}$. By the definition of $\mathcal{K}$ and by (4.4) we have $L \cap K \cap T = \emptyset$. The minimal choice of the members of $\mathcal{K}$ implies that $d(K) < d(K - L)$. But then $d(K) + d(L) \geq d(K-L) + d(L-K) > d(K) + d(L)$, which is a contradiction.

By its definition, $\mathcal{K}$ is admissible and its value is $(\sum_i d(X_i) + \sum_j d(Y_j))/2 = (\sum_i a(A \cap A_i) + \sum_j b(B \cap B_j))/2 = m'(T)/2$, as required. □

By Lemmata 3.5 and 4.4 and by (4.2) we have $\mu \geq m'(T)/2 = (a(A) + b(B))/2 \geq \tau$, and the proof of Theorem 4.3 is complete.

5. Algorithmic aspects. In this section we briefly outline how the proof above gives rise to a strongly polynomial (combinatorial) algorithm in the capacitated case. (Informally, a polynomial-time algorithm is *strongly polynomial* if the number of steps does not depend on the magnitude of the occurring capacities and costs.)

The input of the algorithm consists of two graphs $G = (V, E)$ and $H = (T, F)$, where $T \subseteq V$. G is endowed with a nonnegative rational capacity function $c : E \Rightarrow \mathbf{Q}_+$. We assume that $H = (T, F)$ is given by two partitions $\mathcal{A}= \{A_1, A_2, \ldots, A_h\}$ and $\mathcal{B}= \{B_1, B_2, \ldots, B_k\}$ of T so that $xy \in F$ if and only if each A_i and each B_j contains at most one of x and y. (Note that if a graph H is given by its incidence matrix, one can test H efficiently for bistability. Namely, decide first by enumeration whether there are more than $2|T|$ maximal stable sets of H. If the answer is yes, then H is not bistable. If the answer is no, then H is bistable if and only if the intersection graph of the maximal stable sets is bipartite.)

The output of the algorithm consists of a c-admissible H-multiflow x, so that $\sum(x(t) : t \in T) = \sum(\delta_c(Z) : Z \in \mathcal{K})$, and an admissible subpartition $\mathcal{K} = \{Z_1, Z_2, \ldots, Z_t\}$ of V. Moreover, if c is integer-valued and *Eulerian* in the sense that $\delta_c(v)$ is even for every node $v \in V$, then the output x is integer-valued as well.

Actually, we will assume that c is integer-valued and Eulerian. If this is not the case, one can multiply through the capacities by $2N$, where N denotes the least common denominator of the capacities. If c is inner Eulerian, we can apply the reduction described in section 4 to obtain a completely Eulerian case.

First, we remark that the proof of Theorem 2.2 immediately provides a polynomial-time algorithm for the set system $\mathcal{L}= \mathcal{A}\cup\mathcal{B}$ when c is identically 1. It is not difficult to show that, for general integer-valued Eulerian c, if in every step one splits off as much capacity as possible, then the algorithm is strongly polynomial (cf. [5]). In what follows we comment on the use of the polymatroid intersection algorithm to construct an even vector m' and an admissible subpartition occurring in the proof of Theorem 4.3.

For disjoint sets $X, Y \subseteq V$ let $\lambda_c(X, Y)$ denote the value of a flow between X and Y. With the help of a MFMC computation $\lambda_c(X, Y)$ can be computed in (strongly) polynomial time.

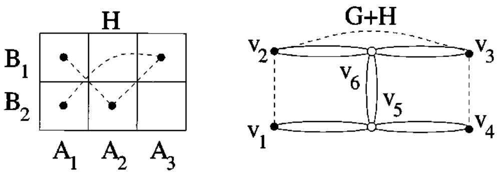
For $A_i \in \mathcal{A}$ let $a_i(X)$ $(X \subseteq A_i)$ be a set function defined by $a_i(X) := \lambda_c(X, T - A_i)$. Define b_j analogously. For $X \subseteq T$ let $a(X) := \sum a_i(X \cap A_i)$ and $b(X) := \sum b_j(X \cap B_j)$. Let $P(a)$ and $P(b)$ be the polymatroids defined by a and b. It is known from polymatroid theory that $P(a/2) = P(a)/2$ (and $P(b/2) = P(b)/2$). Since c is Eulerian, both $a/2$ and $b/2$ are integer-valued, and hence $P(a)/2$ and $P(b)/2$ are integral polymatroids. Therefore, if z is an integer-valued vector in $P(a/2) \cap P(b/2)$ for which $z(V)$ is maximum, then $m' := 2z$ is an even vector in $P(a) \cap P(b)$ for which $m'(V)$ is maximum. By Theorem 3.2 there is a bipartition $\{A, B\}$ of T so that $z(A) = a(A)/2$ and $z(B) = b(B)/2$ holds. Hence $m'(A) = a(A)$ and $m'(B) = b(B)$.

There is a (combinatorial) strongly polynomial algorithm, due to Schönsleben [14], for computing z (and hence m') and $\{A, B\}$. This algorithm works if an oracle is available to minimize $a(A) - z(A)$ and $b(A) - z(A)$ over $A \subseteq T$, where $z : T \to \mathbf{Q}$ is a vector. In our case this oracle can indeed be constructed by invoking the MFMC algorithm, and this way one obtains a purely combinatorial strongly polynomial algorithm for computing m' and A, B satisfying (4.2). Using the proofs of Claims 1 and 2 in the proof of Theorem 4.3, one may compute in strongly polynomial time an integer-valued maximum multiflow and an admissible subpartition of minimum value.

Karzanov [6] described a more direct way to compute m' and a minimal admissible subpartition. His method consists of one MFMC computation on an appropriately defined auxiliary digraph on $|V||T|$ nodes, and its complexity is $O(\varphi(|T||V|))$, where $\varphi(n)$ denotes the complexity of an MFMC computation on a network with n nodes.

Next, the even basis m_a of $P(a)$ (respectively, $m_b \in P(b)$) defined in the proof of Theorem 4.3 can be constructed by $|\mathcal{A}|$ (respectively, $|\mathcal{B}|$) MFMC computations. Thus, vector m defined in (4.4) can be computed from m' in $O(|T|\varphi(|V|))$ steps.

The m-locking problem can be solved by applying at most $O(|V|)$ splitting-off operations at every node $v \in V - T$; each operation consists of finding $|\mathcal{A}| + |\mathcal{B}|$ maximum flows in G. This requires $O(|V|^2(|\mathcal{A}| + |\mathcal{B}|)\varphi(|V|))$ or $O(|V|^2|T|\varphi(|V|))$



$m' = (2,2,2,0) \in P(a) \cap P(b)$

$A = \{v_1, v_2, v_3, v_4\}, B = \emptyset$

$m = (2,2,2,2)$

$P_1 = (v_1, v_5, v_6, v_2)$
$P_2 = (v_2, v_6, v_3)$
$P_3 = (v_4, v_5, v_6, v_3)$
$P_4 = (v_1, v_5, v_4)$

FIG. 3.

operations.

Thus, the overall complexity of the algorithm is $O(|V|^2|T|\varphi(|V|) + \varphi(|T||V|))$. If one uses an MFMC algorithm of complexity $\varphi(n) = O(n^3)$, this gives an $O(|V|^5|T|)$ upper bound for the running time of the algorithm (to be compared with the complexity $O(|V|^6|T|^3)$ of the algorithm in [6]).

6. Node demand problems. The reader might have a feeling that invoking the locking theorem in the proof above contains a seemingly unnecessary twist. In fact, we originally tried to prove Theorem 4.3 by using the following more natural and direct approach, but Figure 3 will show why our attempt failed.

Recall that the polymatroid intersection theorem ensured the existence of a maximum even vector m' in $P(a) \cap P(b)$ for which $m'(T)/2$ is precisely $\tau(G,H)$. Theorem 4.3 would follow if there existed a system of H-feasible paths so that each $t \in T$ is the end node of precisely $m'(t)$ of them. Unfortunately, such a system need not always exist, as is shown by the figure.

Demand graph H is defined by the partitions $\mathcal{A} := \{\{v_1, v_4\}, \{v_2\}, \{v_3\}\}$ and $\mathcal{B} := \{\{v_1, v_3\}, \{v_2, v_4\}\}$. Here $\{\{v_1, v_4, v_5\}, \{v_2\}, \{v_3\}\}$ is an admissible subpartition of value 3; that is, the maximum $\mu(G,H)$ is at most 3. On the other hand, there are three H-admissible edge-disjoint paths in G, namely, $P_1 := (v_1, v_5, v_6, v_2), P_2 := (v_2, v_6, v_3), P_3 := (v_3, v_6, v_5, v_4)$. Hence the value of the primal and dual optima is 3. It can easily be checked that this system of paths is the *only* optimal solution. The bad thing is that two nodes (v_1 and v_4) are the end nodes of just one path (that is, an odd number of them). Therefore, there is no hope to obtain these paths by first determining an optimal *even vector* m' in the intersection of the two polymatroids in question and then finding H-admissible paths so that each node $t \in T$ is the end node of $m'(t)$ of them. Furthermore, one must insist on the evenness of m' since Theorem 2.2 is true only for such vectors.

(Incidentally, vector $m' := (2,2,2,0)$ is an optimal element of the polymatroid intersection and $\{A := T, B := \emptyset\}$ is a bipartition of T satisfying (4.2). Vector m arising in the proof is $m := (2,2,2,2)$. When Theorem 2.3 is applied to this m we obtain a family $\mathcal{F}$ of four paths, namely, $P_1 := (v_1, v_5, v_6, v_2), P_2 := (v_2, v_6, v_3), P_3 := (v_3, v_6, v_5, v_4), P_4 := (v_1, v_5, v_4)$. Among these paths P_4 is the only non-H-admissible, and we obtain P_1, P_2, P_3 as an optimal solution to the maximization problem.)

Although this direct approach to the maximization problem did not prove successful, it led us to the following problem to be considered for its own sake.

Let $G = (V,E)$ be a graph $H = (T,F)$, a demand graph with $T \subseteq V$. Moreover, let $m : T \to \mathbf{Z}_+$ be a *demand* function. The *node demand problem* consists of finding a system of H-admissible paths so that each terminal t is the end node of precisely $m(t)$ paths. We call the problem and also the vector m *feasible* when such a solution exists.

The node demand problem is called *Eulerian* if it is inner Eulerian and $m(t)+d(t)$ is even for each $t \in T$. We call a demand graph H *two-covered* (*one-covered*) if every node $t \in T$ belongs to at most two (exactly one) maximal stable sets of H. Note that bistable graphs are always two-covered but a five-element circuit, for example, is two-covered and not bistable. It can be shown that a graph H is two-covered if and only if H is the complement of the line graph of a triangle-free graph.

THEOREM 6.1. *Suppose that the node demand problem defined by* (G, H, m) *is Eulerian and* H *is two-covered. Then it is feasible if and only if the following condition holds:*

$$m(S) - m(X \cap T - S) \le d(X) \tag{6.1}$$

for every $X \subseteq V$ *and* $S \subseteq X \cap T$ *where* S *is stable in* H.

Proof. Since S is stable in H, in a solution to the node demand problem each path with an end node in S has the other end node in $T - S$. Among these $m(S)$ paths at most $m(X \cap T - S)$ may end in $X - S$, and hence at least $m(S) - m(X \cap T - S)$ must end outside X, from which (6.1) follows.

To prove the sufficiency first observe that the family $\mathcal{L}$ of maximal stable sets of H is 3-cross-free. Indeed, for a contradiction, let S_1, S_2, S_3 be maximal stable sets of H which are pairwise crossing. Since H is two-covered, $S_1 \cap S_2 \cap S_3 = \emptyset$ and there are distinct elements $a \in S_1 \cap S_2, b \in S_2 \cap S_3, c \in S_3 \cap S_1$. Now $\{a, b, c\}$ is stable and a maximal stable set S containing a, b, c is distinct from each S_i. But then the element a (and b, c, as well) would belong to more than two maximal stable sets, contradicting that H is two-covered.

CLAIM 4. $\lambda_m(S) = m(S)$ *for any stable set* S *of* H.

Proof. Recall that $\lambda_m(S)$ was defined to be the maximum number of edge-disjoint paths connecting S and $T - S$ so that each $x \in T$ is the end node of at most $m(x)$ of them. By a version of the Menger theorem $\lambda_m(S) = \min(d(X) + m(S - X) + m(T - S - X) : X \subseteq V)$. (Indeed, apply the edge-disjoint undirected version of the Menger theorem to the graph arising from G by adding two new nodes s, t so that s (respectively, t) is connected to each node x in S (respectively, in $T - S$) by $m(x)$ new parallel edges.)

If X denotes the set where the minimum is attained, then, by (6.1), we have $\lambda_m(S) = d(X) + m(S - X) + m(X \cap T - S) \ge m(S \cap X) - m(X \cap T - S) + m(S - X) + m(X \cap T - S) = m(S)$, and the claim follows. □

Apply Theorem 2.3 to $G, m, \mathcal{L}$ and consider the path system $\mathcal{F}$ provided by the theorem (where $\mathcal{L}$ is the collection of maximal stable sets of H).

CLAIM 5. $\mathcal{F}$ *is a solution to the node demand problem.*

Proof. Let S be an element of $\mathcal{L}$, that is, a maximal stable set of H. Since $\mathcal{F}$ locks S, $\mathcal{F}$ contains $\lambda_m(S) = m(S)$ paths connecting S and $T - S$. This shows that each node x in S is the end node of precisely $m(x)$ members of $\mathcal{F}$ and that each path in $\mathcal{F}$ having an end node in S must have the other end node in $T - S$.

Because every node x of H belongs to a maximal stable set of H, x is the end node of precisely $m(x)$ members of $\mathcal{F}$. Moreover, since every pair of nonadjacent nodes x, y of H belongs to a maximal stable set of H, no path in $\mathcal{F}$ may connect x and y; that is, $\mathcal{F}$ consists of H-feasible paths.

Remark. The condition in Theorem 6.1 may be formulated in an equivalent form. By taking $S := X \cap Z$ in (6.1), we see that (6.1) implies

$$m(X \cap Z) - m(X \cap T - Z) \le d(X) \tag{6.1$'$}$$

for every $X \subseteq V$ and every maximal stable set Z of H. Conversely, we claim that (6.1) follows from (6.1′). Indeed, let Z be a maximal stable set of H including S. Then $m(S) \leq m(X \cap Z) \leq d(X) + m(X \cap T - Z) \leq d(X) + m(X \cap T - S)$, and (6.1) follows.

Equation (6.1′) has the advantage that there are only a few maximal stable sets in a two-covered graph (at most $2|T|$). On the other hand, in the proof above it is slightly easier to work with (6.1).

REFERENCES

[1] B. V. Cherkasky, *A solution of a problem of multicommodity flows in a network*, Ekonom.-Mat. Metody, 13 (1977), pp. 143–151. (In Russian.)

[2] J. Edmonds, *Submodular functions, matroids, and certain polyhedra*, in Combinatorial Structures and Their Applications, R. Guy, H. Hanani, N. Sauer, and J. Schönheim, eds., Gordon and Breach, New York, 1970, pp. 69–87.

[3] C. A. J. Hurkens, A. Schrijver, and É. Tardos, *On fractional multicommodity flows and distance functions*, Discrete Math., 73 (1988), pp. 99–109.

[4] A. V. Karzanov, *Combinatorial methods to solve cut-determined multiflow problems*, in Combinatorial Methods for Flow Problems, Institute for System Studies, Moscow, Issue 3, 1979, pp. 6–69. (In Russian.)

[5] A. V. Karzanov, *A generalized MFMC-property and multicommodity cut problems*, in Finite and Infinite Sets, A. Hajnal and V. T. Sós, eds., Proc. 6th Hungarian Combinatorial Coll., Eger, North–Holland, Amsterdam, The Netherlands, 1984, pp. 443–486.

[6] A. V. Karzanov, *On multicommodity flow problems with integer-valued optimal solutions*, Dokl. Akad. Nauk UzSSR, 280 (1985). English translation: Soviet Math. Dokl., 31 (1988), pp. 151–154.

[7] A. V. Karzanov, *A class of maximum multicommodity flows with integer-valued optimal solutions*, in Modeling and Optimization in Complex Structure Systems, Omsk State University, Omsk, 1987, pp. 103–121. (In Russian.) English translation: Amer. Math. Soc. Translations, Ser. 2, 158 (1994), pp. 81–99.

[8] A. V. Karzanov and M. V. Lomonosov, *Systems of flows in undirected networks*, in Mathematical Programming, O.I. Larichev, ed., Institute for System Studies, Moscow, Issue 1, 1978, pp. 59–66. (In Russian.)

[9] A. V. Karzanov and P. A. Pevzner, *A complete description of the class of cut-nondetermined maximum multicommodity flow problems*, in Combinatorial Methods for Flow Problems, Institute for System Studies, Moscow, Issue 3, 1979, pp. 70–81. (In Russian.)

[10] M. V. Lomonosov, *Combinatorial approaches to multiflow problems*, Discrete Appl. Math., 11 (1985), pp. 1–93.

[11] M. V. Lomonosov, *Combinatorial approaches to multiflow problems*, Graph Theory Newsletters, 9 (1979), p. 4.

[12] L. Lovász, *On some connectivity properties of Eulerian graphs*, Acta Mat. Akad. Sci. Hungaricae, 28 (1976), pp. 129–138.

[13] B. Rothschild and A. Whinston, *On two-commodity network flows*, Oper. Res., 14 (1966), pp. 377–387.

[14] P. Schönsleben, *Ganzzahlige Polymatroid Intersections Algorithmen*, Ph.D. thesis, Eidgenössischen Techn. Hochschule, Zürich, Switzerland, 1980.

SIAM J. DISCRETE MATH.
Vol. 10, No. 2, pp. 171–179, May 1997

WORST CASE LENGTH OF NEAREST NEIGHBOR TOURS FOR THE EUCLIDEAN TRAVELING SALESMAN PROBLEM*

L. TASSIULAS†

Abstract. The worst case length of a tour for the Euclidean traveling salesman problem produced by the nearest neighbor (NN) heuristic is studied in this paper. Nearest neighbor tours for a set of arbitrarily located points in the d-dimensional unit cube are considered. A technique is developed for bounding the worst case length of a tour. It is based on identifying sequences of *coverings* of $[0,1]^d$. Each covering $\mathcal{P}_k$ consists of sets C_i, with diameter bounded by the *diameter* $D(\mathcal{P}_k)$ of the covering. For every sequence of coverings a bound is obtained that depends on the cardinality of the coverings and their diameters. The task of bounding the worst case length of an NN tour is reduced to finding appropriate sequences of coverings. Using coverings produced by the rectangular lattice with appropriately shrinking diameter, it is shown that the worst case length of an NN tour through N points in $[0,1]^d$ is bounded by $[d\sqrt{d}/(d-1)]N^{(d-1)/d} + o(N^{(d-1)/d})$. For the unit square the tighter bound $2.482\sqrt{N} + o(\sqrt{N})$ is obtained using regular hexagonal lattice coverings.

Key words. Euclidean traveling salesman problem, nearest neighbor tours, worst case analysis

AMS subject classifications. 68, 90

PII. S0895480194278246

1. Introduction.

Consider a set $V = \{x_1, \ldots, x_N\}$ of points in $[0,1]^d$. Let $G = (V, E)$ be the complete graph with vertex set V. The length of edge (x_k, x_l) is the Euclidean distance $|x_k - x_l|$ between x_k and x_l. Let $T(V)$ be the set of the tours for graph G. The tours of G are in one-to-one correspondence with the permutations of the vertices. A tour $x_{i_1}, x_{i_2}, \ldots, x_{i_N}$ with starting point x_{i_1} will be denoted by $(i_1, i_2, \ldots, i_N)$. The length of tour $t = (i_1, i_2, \ldots, i_N)$ is equal to the sum of the lengths of the edges of the tour; that is,

$$L(t) = \sum_{k=1}^{N} |x_{i_k} - x_{i_{k+1}}|,$$

where by convention $x_{i_{N+1}} = x_{i_1}$. The Euclidean traveling salesman problem (TSP) is to find the minimum length tour through the set of points V.

The TSP is one of the most heavily studied problems of combinatorial optimization [4, 8]. In general graphs where the length of the edges may be arbitrary the TSP was among the first problems shown to be NP-complete (see Karp [7]). The Euclidean TSP also has been shown to be NP-complete (see Papadimitriou [10]). There has been a lot of work on heuristics and approximate algorithms with guaranteed performance [5].

A popular heuristic for the Euclidean TSP is the nearest neighbor (NN) algorithm. According to this the tour is derived by selecting an arbitrary initial point x_{i_1} and visiting successively from point x_{i_k} the point $x_{i_{k+1}}$, which is the closest to x_{i_k}, among

* Received by the editors December 5, 1994; accepted for publication (in revised form) March 15, 1996. This research was supported in part by the National Science Foundation under grants NCR-9211417 and NCR-9406415, by the Air Force Office of Scientific Research under grant 95-1-0061, and by the New York State Center for Advanced Technology in Telecommunications, Polytechnic University.

http://www.siam.org/journals/sidma/10-2/27824.html

† Department of Electrical Engineering and Institute for Systems Research, University of Maryland, College Park, MD 20742 (leandros@isr.umd.edu).

those that have not been visited yet. Hence any tour $t = (i_1, i_2, \ldots, i_N)$ produced by the NN heuristic satisfies the property

$$|x_{i_{k+1}} - x_{i_k}| = \min_{j=k+1,\ldots,N} |x_{i_k} - x_{i_j}|, \ k = 1, \ldots, N-1. \tag{1}$$

Also, any tour satisfying (1) can be produced by the NN heuristic if the starting point is selected accordingly. Any tour that satisfies property (1) will be called an NN tour throughout the rest of the paper. Denote by $NN(V)$ the set of all NN tours that correspond to the set of points V. The objective of this paper is to study the worst case length of an NN tour over all configurations $V = \{x_1, \ldots, x_N\}$ of N points in the d-dimensional unit cube $[0,1]^d$. This is defined as

$$L_N = \sup_{V \subset [0,1]^d, |V|=N} \max_{t \in NN(V)} L(t).$$

One way to assess the performance of the NN heuristic is to compare L_N with the length of the worst case minimum length tour

$$P_N = \sup_{V \subset [0,1]^d, |V|=N} \min_{t \in T(V)} L(t).$$

There are several studies on obtaining bounds for P_N. Steele [11] contains a detailed account of related results. Few [3] obtained an upper bound on P_N for the general d-dimensional case; that is,

$$P_N \leq d\{2(d-1)\}^{(1-d)/2d} N^{(d-1)/d} + o(N^{1-2/d}).$$

This was further improved for large d by Moran [9], while Karloff [6] improved the upper bound for $d = 2$ by showing that $P_N \leq 0.984\sqrt{2}\sqrt{N} + c$. For the two-dimensional case, Supowit, Reingold, and Plaisted [12] proved that

$$\left(\frac{4}{3}\right)^{1/4} \sqrt{N} - o(\sqrt{N}) \leq P_N.$$

The performance of the NN heuristic has been previously studied for the TSP in general graphs as well as in more special cases of graphs that satisfy certain constraints. Johnson and Papadimitriou [5] review related work.

Upper bounds on L_N are obtained in this paper for the Euclidean TSP. From these bounds and well-known lower bounds on P_N it follows that for the Euclidean TSP in the unit square, the ratio L_N/P_N is bounded asymptotically by 2.3095 or, more precisely, that for every $\epsilon > 0$, there exists $N(\epsilon)$ such that $L_N/P_N \leq 2.3095 + \epsilon$ for $N > N(\epsilon)$. Similar results follow for NN tours in higher dimensions from the corresponding bounds on L_N in higher dimensions.

The rest of the paper is organized as follows. In section 2 the technique for bounding L_N using coverings of $[0,1]^d$ is presented and the main bounding theorem is obtained. In section 3 this technique is applied with coverings derived from the regular rectangular lattice and a bound for NN tours in d dimensions is obtained. In section 4 the bound is tightened for the unit square using regular hexagonal coverings. Some further points are discussed in section 5.

2. The general bound. The diameter $D(C)$ of a subset C of R^d is defined as

$$D(C) = \sup_{x,y \in C} |x - y|.$$

The essential property of an NN tour used in the derivation of the bounds in this paper is captured in the following lemma.

LEMMA 2.1. *For any subset C of R^d and tour $t = (i_1, i_2, \dots, i_N)$ produced by the NN heuristic, there is at most one vertex $x_{i_k} \in C$ such that*

$$|x_{i_k} - x_{i_{k+1}}| > D(C).$$

Proof. By contradiction, assume there are two vertices x_{i_l}, x_{i_m} in C such that $|x_{i_l} - x_{i_{l+1}}| > D(C)$ and $|x_{i_m} - x_{i_{m+1}}| > D(C)$. Without loss of generality assume that $l < m$. Then

$$|x_{i_m} - x_{i_l}| \geq \min_{j=l+1,\dots,n} |x_{i_j} - x_{i_l}| = |x_{i_{l+1}} - x_{i_l}| = |x_{i_l} - x_{i_{l+1}}| > D(C) \geq |x_{i_m} - x_{i_l}|,$$

which contradicts from (1) the assumption that t is an NN tour. □

Lemma 2.1 will be used to derive the main bounding theorem in the following, after some preliminary definitions.

A *covering* $\mathcal{P}$ of a set $A \subset R^d$ is defined to be any collection of subsets of R^d, $\mathcal{P} = \{C_l : l = 1, \dots, P\}$, $C_l \subset R^d,\ l = 1, \dots, P$, with the property $\cup_{l=1}^{P} C_l \supseteq A$. The sets that constitute the covering will be called *cells* of the covering in the following. The diameter $D(\mathcal{P})$ of the covering $\mathcal{P}$ is defined as

$$D(\mathcal{P}) = \max_{l=1,\dots,P} D(C_l).$$

The cardinality P of covering $\mathcal{P}$ will be denoted as $|\mathcal{P}|$.

Note that for every covering, a bound on an NN tour can be obtained easily using Lemma 2.1. In any cell there can be at most one point with an adjacent edge of the tour that has length greater than the cell diameter. Hence, at most $|\mathcal{P}|$ edges of an NN tour will have length larger than the diameter of the covering, while the length of all the other edges will be smaller than $D(\mathcal{P})$. Therefore,

$$L_N \leq (N - |\mathcal{P}|) D(\mathcal{P}) + |\mathcal{P}| D(A), \tag{2}$$

where the fact that the length of any edge of an NN tour will be less than $D(A)$ has been used. By considering sequences of coverings instead of a single covering, bounds tighter than (2) can be obtained. In the rest of the paper by "covering" we will mean the covering of $[0,1]^d$.

Consider sequences of coverings $\mathcal{P}_m,\ m = 1, \dots, M$ with decreasing diameter, where

$$D(\mathcal{P}_m) \geq D(\mathcal{P}_{m+1}),\ m = 1, \dots, M-1.$$

The following theorem holds.

THEOREM 2.2. *The worst case length of an NN tour is bounded as follows:*

$$L_N \leq N D(\mathcal{P}_M) + \sum_{m=2}^{M} |\mathcal{P}_m| (D(\mathcal{P}_{m-1}) - D(\mathcal{P}_m)) + |\mathcal{P}_1| (D(A) - D(\mathcal{P}_1)). \tag{3}$$

Proof. It is shown that for an arbitrary tour $t = (i_1, \dots, i_N)$,

$$L(t) \leq ND(\mathcal{P}_M) + \sum_{m=2}^{M} |\mathcal{P}_m|(D(\mathcal{P}_{m-1}) - D(\mathcal{P}_m)) + |\mathcal{P}_1|(D(A) - D(\mathcal{P}_1)). \tag{4}$$

Consider the increasing sequence of subsets of vertices V_m, $m = 1, \dots, M$ defined as follows:

$$V_m = \{x_{i_k} : x_{i_k} \in V, |x_{i_k} - x_{i_{k+1}}| > D(\mathcal{P}_m)\}.$$

Note that the sets $V - V_M,\ V_M - V_{M-1},\ V_{M-1} - V_{M-2}, \dots, V_2 - V_1, V_1$ constitute a partition of V. Therefore, the length of tour t can be written as follows:

$$L(t) = \sum_{x_{i_k} \in (V - V_M)} |x_{i_k} - x_{i_{k+1}}| + \sum_{m=2}^{M} \sum_{x_{i_k} \in (V_m - V_{m-1})} |x_{i_k} - x_{i_{k+1}}| + \sum_{x_{i_k} \in V_1} |x_{i_k} - x_{i_{k+1}}|. \tag{5}$$

By the definition of the sets V_i,

$$|x_{i_k} - x_{i_{k+1}}| \leq D(\mathcal{P}_M),\ x_{i_k} \in (V - V_M), \tag{6}$$

$$|x_{i_k} - x_{i_{k+1}}| \leq D(\mathcal{P}_{m-1}),\ x_{i_k} \in (V_m - V_{m-1}),\ m = 2, 3, \dots, M, \tag{7}$$

$$|x_{i_k} - x_{i_{k+1}}| \leq D(A),\ x_{i_k} \in V_1. \tag{8}$$

By substituting from equations (6), (7), and (8) to equation (5), we get

$$L(t) \leq |V - V_M| D(\mathcal{P}_M) + \sum_{m=2}^{M} |V_m - V_{m-1}| D(\mathcal{P}_{m-1}) + |V_1| D(A). \tag{9}$$

Since $V_1, V_2, \dots, V_M, V$ is an increasing sequence of sets ($V_m \subseteq V_{m+1}$), we have $|V_m - V_{m-1}| = |V_m| - |V_{m-1}|$, $m = 2, \dots, M$, and substituting in (9) we get

$$L(t) \leq (|V| - |V_M|) D(\mathcal{P}_M) + \sum_{m=2}^{M} (|V_m| - |V_{m-1}|) D(\mathcal{P}_{m-1}) + |V_1| D(A). \tag{10}$$

By rearranging the sum in the right-hand side of (10), we get

$$L(t) \leq |V| D(\mathcal{P}_M) + \sum_{m=2}^{M} |V_m|(D(\mathcal{P}_{m-1}) - D(\mathcal{P}_m)) + |V_1|(D(A) - D(\mathcal{P}_1)). \tag{11}$$

Note that relationship (11) holds for any TSP tour. The fact that t is an NN tour is now used to bound $|V_m|$. From Lemma 2.1 we have that any cell C of covering $\mathcal{P}_m$ can contain at most one point x_{i_k}, such that $|x_{i_k} - x_{i_{k+1}}| > D(C)$. Therefore, each cell of $\mathcal{P}_m$ can contribute at most one point to the set V_m; hence

$$|V_m| \leq |\mathcal{P}_m|,\ m = 1, \dots, M. \tag{12}$$

Substituting in inequality (11) from (12) we get (4). □

In the next two sections it is shown how Theorem 2.2 can be applied to specific coverings to get bounds on L_N.

3. Bounds from rectangular lattice coverings. In this section a bound on L_N is obtained using the coverings implied by the rectangular lattice. Consider the sequence of coverings $\mathcal{P}_k,\ k = 1, \dots, M$, where

$$\mathcal{P}_k = \{C_{l_1 l_2 \dots l_d} : l_i = 0, 1, \dots, k-1,\ i = 1, \dots, d\}$$

and

$$C_{l_1 l_2 \dots l_d} = \left\{ \left(\frac{l_1}{k} + x_1, \frac{l_2}{k} + x_2, \dots, \frac{l_d}{k} + x_d \right) :\ 0 \le x_i < \frac{1}{k},\ i = 1, \dots, d \right\}.$$

That is, the cells of the covering are d-dimensional cubes with edge length $1/k$. By applying Theorem 2.2 with the sequence of coverings above, we have the following.

THEOREM 3.1. *The worst case length of a tour through N points in $[0,1]^d$ produced by the NN heuristic is bounded as follows:*

(13)
$$L_N \le \sum_{m=1}^{d-1} \frac{(d-m+1)\sqrt{d}}{d-m} N^{(d-m)/d} + \ln(N^{1/d} - 1) + 1 + \frac{1}{N^{1/d} - 1} - \sqrt{d} \sum_{m=1}^{d-1} \frac{1}{d-m}.$$

Proof. Note that the diameter of all cells in covering $\mathcal{P}_k$ is equal to $\sqrt{d}/k$; therefore,

$$D(\mathcal{P}_k) = \frac{\sqrt{d}}{k}$$

and also

$$|\mathcal{P}_k| = P_k = k^d.$$

By applying Theorem 2.2 to this covering, we get

$$\begin{aligned} L_N &\le N\sqrt{d}\frac{1}{M} + \sum_{k=2}^{M} k^d \left(\frac{\sqrt{d}}{k-1} - \frac{\sqrt{d}}{k} \right) \\ &= N\sqrt{d}\frac{1}{M} + \sqrt{d} \sum_{k=2}^{M} \left(\frac{k^{d-2}}{k-1} + k^{d-2} \right). \end{aligned}$$

Using the formula for the sum of a geometric series, we get

$$L_N \le N\sqrt{d}\frac{1}{M} + \sqrt{d} \sum_{k=2}^{M} \left(k^{d-2} + k^{d-3} + \cdots + 1 + \frac{1}{k-1} \right). \tag{14}$$

Substituting in (14) using the well-known bounds (see [2]),

$$\sum_{k=m}^{n} f(k) \le \int_{m-1}^{n} f(x)dx$$

for the sums in the parentheses in (14), and after some calculations we get

$$L_N \le \frac{N\sqrt{d}}{M} + \sum_{m=1}^{d-1} \frac{\sqrt{d}}{d-m} M^{d-m} + \ln(M-1) - \sqrt{d} \sum_{m=1}^{d-1} \frac{1}{d-m} + 1. \tag{15}$$

Inequality (15) holds for all values of M. For $M = \lfloor N^{1/d} \rfloor$, inequality (15) becomes

$$(16)\, L_N \leq \frac{N\sqrt{d}}{\lfloor N^{1/d} \rfloor} + \sum_{m=1}^{d-1} \frac{\sqrt{d}}{d-m} \lfloor N^{1/d} \rfloor^{d-m} + \ln(\lfloor N^{1/d} \rfloor - 1) - \sqrt{d} \sum_{m=1}^{d-1} \frac{1}{d-m} + 1.$$

By replacing the floors in (16) such that the inequality remains true, we get

$$(17)\, L_N \leq \frac{N\sqrt{d}}{N^{1/d} - 1} + \sum_{m=1}^{d-1} \frac{\sqrt{d}}{d-m} N^{(d-m)/d} + \ln(N^{1/d} - 1) - \sqrt{d} \sum_{m=1}^{d-1} \frac{1}{d-m} + 1.$$

By using the formula for the sum of a geometric series in the term $N\sqrt{d}/(N^{1/d} - 1)$, we finally get

$$\begin{aligned} L_N \leq{} & \sum_{m=1}^{d-1} \frac{(d-m+1)\sqrt{d}}{d-m} N^{(d-m)/d} \\ & + \frac{1}{N^{1/d} - 1} + \ln(N^{1/d} - 1) - \sqrt{d} \sum_{m=1}^{d-1} \frac{1}{d-m} + 1, \end{aligned}$$

and the proof is complete. □

Note that the higher-order term of the bound in Theorem 3.1 is $[d\sqrt{d}/(d-1)]N^{(d-1)/d}$. The bound in Theorem 2.2 depends on the type of coverings used in the derivation. By selecting the appropriate type of cells in the coverings, the derived bound can be tightened, as is shown in the following for the unit square.

4. Tighter bounds for the unit square using the regular hexagonal lattice. Consider coverings of the unit square using the hexagonal lattice. The covering $\mathcal{P}_k$ consists of hexagons with diameter $2/\sqrt{3}k$, arranged as depicted in Figure 1. Hence the diameter of the covering is

$$(18) \qquad D(\mathcal{P}_k) = 2/(\sqrt{3}k).$$

By counting the cells in the covering carefully we can verify that

$$(19) \qquad |\mathcal{P}_k| \leq (2k+1)\frac{k}{\sqrt{3}} + 3k + 1.$$

Considering the sequence $\mathcal{P}_k$, $k = 1, \ldots, M$ of coverings $\mathcal{P}_k$ as above and using Theorem 2.2, we can obtain the following.

THEOREM 4.1. *The worst case length of a tour through N points in $[0,1]^2$ produced by the NN heuristic is bounded as follows:*

$$(20) \qquad L_N \leq 2^{5/2} 3^{-3/4} \sqrt{N} + \frac{10\sqrt{3}+2}{3\sqrt{3}} \ln\left(\left(\frac{3}{4}\right)^{1/4} \sqrt{N}\right) + \frac{4\sqrt{3}+5}{3}.$$

Proof. Applying Theorem 2.2 with the hexagonal coverings $\mathcal{P}_k$, $k = 1, \ldots, M$ and using (18) and (19), we obtain

$$(21)\, L_N \leq \frac{2N}{\sqrt{3}M} + \sum_{m=2}^{M} \left(\frac{2}{\sqrt{3}} m^2 + \frac{1+3\sqrt{3}}{3} m + 1\right)\left(\frac{2}{\sqrt{3}(m-1)} - \frac{2}{\sqrt{3}m}\right) + 3.$$

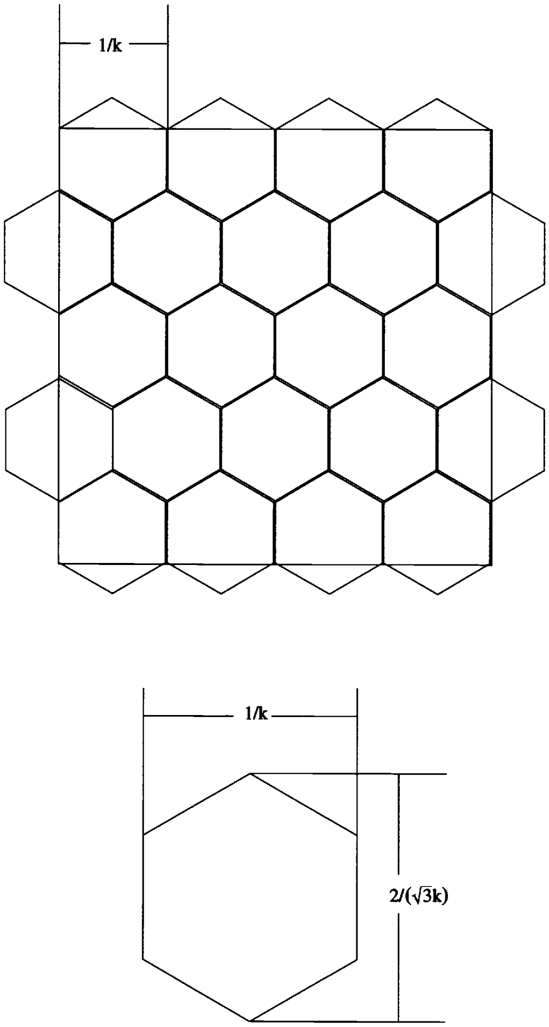


FIG. 1. *The unit square covered by a regular hexagonal covering and the cell of the covering are depicted.*

By doing some calculations in (21), we get

$$L_N \leq \frac{2N}{\sqrt{3}M} + \sum_{m=2}^{M} \frac{2}{\sqrt{3}} m^2 \frac{2}{\sqrt{3}m(m-1)}$$
$$+ \sum_{m=2}^{M} \frac{1+3\sqrt{3}}{3} \cdot \frac{2m}{\sqrt{3}m(m-1)} + \sum_{m=2}^{M} \frac{2}{\sqrt{3}m(m-1)} + 3,$$

from which we finally get

$$L_N \leq \frac{2N}{\sqrt{3}M} + \frac{4}{3}(M-2) + \sum_{m=2}^{M} \frac{10\sqrt{3}+2}{3\sqrt{3}(m-1)} + \sum_{m=2}^{M} \frac{2}{\sqrt{3}m(m-1)} + 3. \tag{22}$$

Substituting in (22) using the bounds

$$\sum_{k=m}^{n} f(k) \leq \int_{m-1}^{n} f(x)dx$$

for the summations, we get

$$L_N \leq \frac{2N}{\sqrt{3}M} + \frac{4}{3}(M-2) + \frac{10\sqrt{3}+2}{3\sqrt{3}} \ln(M-1) + \frac{4}{\sqrt{3}} + 3. \tag{23}$$

By selecting $M = \lceil (3/4)^{1/4}\sqrt{N} \rceil$, equation (23) becomes

$$L_N \leq \frac{2N}{\sqrt{3}\lceil (3/4)^{1/4}\sqrt{N}\rceil} + \frac{4}{3}(\lceil (3/4)^{1/4}\sqrt{N}\rceil - 2) + \frac{10\sqrt{3}+2}{3\sqrt{3}} \ln(\lceil (3/4)^{1/4}\sqrt{N}\rceil - 1) + \frac{4}{\sqrt{3}} + 3. \tag{24}$$

Replacing the ceilings in equation (24) such that the inequality remains true and after some calculations, we get

$$L_N \leq 2^{3/2}3^{-3/4}\sqrt{N} + \frac{4}{3}\left(\left(\frac{3}{4}\right)^{1/4}\sqrt{N} - 1\right) + \frac{10\sqrt{3}+2}{3\sqrt{3}} \ln\left(\left(\frac{3}{4}\right)^{1/4}\sqrt{N}\right) + \frac{4}{\sqrt{3}} + 3, \tag{25}$$

from which the theorem follows after simple calculations. □

Note that the highest-order term of the bound of Theorem 3.1 for the two-dimensional case is $2.84\sqrt{N}$, while the highest-order term of the bound of Theorem 4.1 is equal to $2.482\sqrt{N}$.

5. Discussion. A methodology for bounding the length of NN tours in Euclidean TSPs using coverings of $[0,1]^d$ was presented in this paper. The general bound in section 2 is proportional to both the diameter of the covering and its cardinality. Hence, in order to obtain good bounds, it is important to find coverings with small diameter and as small cardinalities as possible. In two dimensions, hexagonal coverings achieve a better trade-off between cardinality and diameter than rectangular coverings, and consequently the bound that was obtained using these coverings in section 4 is better than the one obtained by using rectangular coverings. In fact, the hexagonal covering is the one that achieves the optimal trade-off between diameter and cardinality in two dimensions, as is mentioned in the book of Conway and Sloane [1], where coverings and their properties are studied extensively.

REFERENCES

[1] J. H. Conway and N. J. A. Sloane, *Sphere Packings Lattices and Groups*, Springer-Verlag, New York, Berlin, 1993.

[2] T.H. Corman, C.E. Leiserson, and R.L. Rivest, *Introduction to Algorithms*, McGraw–Hill, New York, 1990.

[3] L. Few, *The shortest path and the shortest road through n points in a region*, Mathematika, 2 (1955), pp. 141–144.

[4] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA, 1979.

[5] D. S. Johnson and C. H. Papadimitriou, *Performance guarantees for heuristics*, in The Traveling Salesman Problem: A Guided Tour in Combinatorial Optimization, E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, eds., John Wiley, New York, 1985, pp. 145–180.

[6] H. Karloff, *How long can a Euclidean traveling salesman tour be?* SIAM J. Discrete Math., 2 (1989), pp. 91–99.

[7] R. M. Karp, *Reducibility among combinatorial problems*, in Complexity of Computer Computations, R. E. Miller and J. W. Thacher, eds., Plenum, New York, 1972, pp. 85–103.

[8] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, eds., *The Travelling Salesman Problem*, John Wiley, New York, 1985.

[9] S. Moran, *On the length of optimal TSP circuits in sets of bounded diameter*, J. Combin. Theory Ser. B, 37 (1984), pp. 113–141.

[10] C. H. Papadimitriou, *The Euclidean traveling salesman problem is NP-complete*, Theoret. Computer Sci., 4 (1977), pp. 237–244.

[11] J. M. Steele, *Probabilistic and worst case analyses of classical problems of combinatorial optimization in Euclidean space*, Math. Oper. Res., 15(4) (1990), pp. 749–770.

[12] K. J. Supowit, E. M. Reingold, and D. A. Plaisted, *The traveling salesman problem and minimum matching in the unit square*, SIAM J. Comput., 12 (1983), pp. 144–156.

SIAM J. Discrete Math.
Vol. 10, No. 2, pp. 180–189, May 1997

A MIN-MAX THEOREM FOR A CONSTRAINED MATCHING PROBLEM[*]

ANDREAS HEFNER[†]

Abstract. The following constrained matching problem arises in the area of manpower scheduling. Consider an undirected graph $G = (V, E)$ and a digraph $D = (V, A)$. A *master/slave-matching (MS-matching) in G with respect to D* is a matching in G such that for each arc $(u, v) \in A$ for which the node u is matched, the node v is matched too. The problem is to find an MS-matching of maximum cardinality. This paper addresses the special case where G is bipartite with bipartition $V = W \cup U$ and every (weakly) connected component of D is either an isolated node or two nodes in U which are joined by a single arc. The polyhedral structure of this special case is investigated and a min-max theorem which characterizes the cardinality of a maximum MS-matching in terms of the weight of a special node cover is derived. This min-max theorem includes as a special case the theorem of König.

Key words. constrained matching problem, polyhedral combinatorics, min-max relations

AMS subject classifications. 05C70, 90C10

PII. S0895480195280538

1. Introduction.

Let $G = (V, E)$ be an undirected graph. A *matching* in G is a subset of the edges such that no two edges share the same node. The *matching problem* is to find a matching of maximum cardinality. A *node cover* in G is a subset C of the nodes such that each edge is incident with at least one node in C. The *node cover problem* is to find a node cover of minimum cardinality.

Let $D = (V, A)$ be a digraph with the same node set as G. If (u, v) is an arc in A, we say that v is a *master* of u and u is a *slave* of v. A *master/slave-matching* (*MS-matching* for short) *in G with respect to D* is a matching in G with the property that if $(u, v) \in A$ and u is matched, then so is v (see Figure 1 for an example of an MS-matching). The *(unweighted) MS-matching problem* is the problem of finding an MS-matching of maximum cardinality.

Hefner and Kleinschmidt [5] encountered this problem in practice when they designed a manpower scheduling system for some printing works. They showed that the MS-matching problem is $\mathcal{NP}$-hard but is solvable in polynomial time if every (weakly) connected component of D has size at most two, even if edge weights are present and the problem is to find an MS-matching of maximum weight.

In this paper we study the MS-matching problem for instances where G is bipartite with bipartition $V = W \cup U$ and each connected component of D is either an isolated node or two nodes in U which are joined by a single arc. This special case was of particular interest in the application mentioned by Hefner and Kleinschmidt [5]. In this case D induces a partial function $f : U \to U$ with $f(u) = v$ if and only if $(u, v) \in A$. The domain of f, denoted by $\mathrm{dom}(f)$, is the set of slaves and the range of f, denoted by $\mathrm{ran}(f)$, is the set of masters. Note that f is injective, $f(u) \neq u$ for each $u \in \mathrm{dom}(f)$ and $\mathrm{dom}(f) \cap \mathrm{ran}(f) = \emptyset$. We call a partial function with these

[*] Received by the editors January 25, 1995; accepted for publication (in revised form) April 1, 1996. This research was supported by grant 03-KL7PAS-6 of the German Federal Ministry of Research and Technology and grant I-0309-146.06/93 of the German-Israeli-Foundation for Scientific Research and Development.

http://www.siam.org/journals/sidma/10-2/28053.html

[†] Department of Business Administration and Economics, University of Passau, 94030 Passau, Germany (hefner@winf.uni-passau.de).

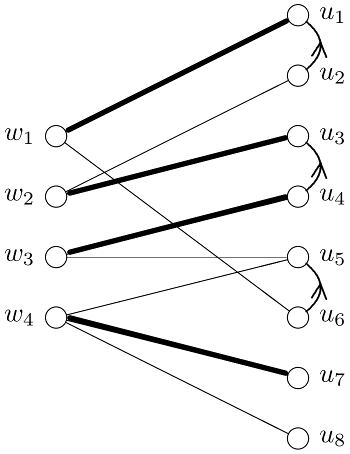


FIG. 1. *An MS-matching.*

properties a *dependence function* on U. Conversely, a bipartite graph together with a dependence function f defines an instance of the MS-matching problem which satisfies the above restrictions. Thus an MS-matching in G with respect to f is a matching M in G such that for each matched node $u \in \text{dom}(f)$ the node $f(u)$ is matched, too. If the dependence function f is clear from the context, then we will sometimes say that *M is an MS-matching in G* instead of saying that *M is an MS-matching in G with respect to f.*

Let G be a bipartite graph and f be a dependence function. The main result of this paper will be a min-max theorem which characterizes the cardinality of a maximum MS-matching in G with respect to f. In combinatorial optimization a lot of min-max relations are known. One of the most well known such relations is the following theorem of König [7].

THEOREM 1.1. *In a bipartite graph a maximum matching and a minimum node cover have the same cardinality.*

There are at least two reasons why such min-max theorems are useful. The first is that they often establish that an optimization problem is in $\mathcal{NP} \cap$ *co-*$\mathcal{NP}$ (see [1] for a general reference to the theory of $\mathcal{NP}$-completeness). Consider the following decision version of the matching problem: given a graph and a positive integer k, does there exist a matching of cardinality at least k? It is easy to see that the problem is in $\mathcal{NP}$ since we need only guess k edges and check in polynomial time that no two of them share a common node. On the other side the theorem of König enables us to show that the problem is also in *co-*$\mathcal{NP}$ if the graph is bipartite. A node cover of cardinality $k-1$ is a proof of polynomial length that a graph has no matching of cardinality k. Since for virtually all problems in $\mathcal{NP} \cap$ *co-*$\mathcal{NP}$ polynomial time algorithms are known, many people believe that $\mathcal{P} = \mathcal{NP} \cap$ *co-*$\mathcal{NP}$. Thus showing that a problem is in $\mathcal{NP} \cap$ *co-*$\mathcal{NP}$ provides a strong indication that the problem is solvable in polynomial time.

A second reason why min-max theorems are useful is that they provide stopping criteria for optimization algorithms. If M is a matching in a bipartite graph G and C is a node cover in G with $|C| = |M|$, then both M and C are optimal.

A general approach for obtaining min-max relations is offered by polyhedral combinatorics (see [9] for an excellent survey on polyhedral combinatorics).

To obtain a min-max theorem for MS-matchings we define objects which are dual to MS-matchings. For a vector $y \in \mathbf{Z}_+^{|V|}$ and an edge $e = [w,u] \in E$ we say that y *covers e exactly $y_w + y_u$ times* (the symbol $\mathbf{Z}_+$ denotes the set of nonnegative integers). An *MS-node cover in G with respect to f* is a vector $y \in \mathbf{Z}_+^{|V|}$ with the following two properties:

1. Every edge $[w,u] \in E$ with $u \in U \setminus \operatorname{ran}(f)$ is covered by y at least once.
2. Every edge $[w,u] \in E$ with $u \in \operatorname{ran}(f)$ is covered by y at least $1 + y_{f^{-1}(u)}$ times.

The *weight* w of an MS-node cover y is the sum over all components which do not correspond to slaves; i.e., $w = y(V \setminus \operatorname{dom}(f)) := \sum_{v \in V \setminus \operatorname{dom}(f)} y_v$.

In the next section we introduce two polyhedra $\operatorname{FMSM}(G,f)$ and $\operatorname{MSC}(G,f)$ which are dual to each other. The integral points of $\operatorname{FMSM}(G,f)$ are the incidence vectors of MS-matchings in G while the integral points of $\operatorname{MSC}(G,f)$ are MS-node covers in G. The subject of section 3 is the polytope $\operatorname{FMSM}(G,f)$. We characterize the vertices of $\operatorname{FMSM}(G,f)$ and show that the linear program max $\sum_{e \in E} x_e$, $x \in \operatorname{FMSM}(G,f)$ always has an integral optimal solution. The proof of the latter statement will provide a polynomial time algorithm for solving the (unweighted) MS-matching problem which is based on linear programming. In section 4 we study the polyhedron $\operatorname{MSC}(G,f)$ and prove that all its vertices are integral. Based on the results in sections 3 and 4 we will show in section 5 that in a bipartite graph with a dependence function the cardinality of a maximum MS-matching and the weight of a minimum MS-node cover are the same.

We conclude this section by introducing some notation which will be used in the subsequent sections. By $\mathbf{R}$ ($\mathbf{R}_+$, $\mathbf{R}_-$) we denote the set of real (nonnegative real, nonpositive real) numbers. Let G be an undirected graph. For a node v we denote the set of edges incident with v by $\delta(v)$. The *incidence vector* of a subset $F \subseteq E$ is the vector $\chi^F \in \{0,1\}^{|E|}$ defined by

$$\chi_e^F := \begin{cases} 1, & e \in F, \\ 0, & e \notin F. \end{cases}$$

The vector which has all components equal to zero is denoted by $\mathbf{0}$. Analogously, we define the vector $\mathbf{1}$. For the ith unit vector we write e_i. Let M and N be a set of row and column indices, respectively. For a vector $x = (x_i)_{i \in M}$ and a subset $I \subseteq M$ we define $x(I) := \sum_{i \in I} x_i$. Let $C = (C_{ij})_{\substack{i \in M \\ j \in N}}$ be an $|M| \times |N|$-matrix. For any $I \subseteq M$ and $J \subseteq N$ we will write C_{IJ} for the submatrix of C whose rows and columns are indexed by I and J, respectively. We write $C_{I\cdot}$ instead of C_{IN} and $C_{\cdot J}$ instead of C_{MJ}. If I or J is a singleton, we omit the braces (for example, for the ith row of C we write $C_{i\cdot}$ instead of $C_{\{i\}\cdot}$). We adopt the same notation for subvectors. For example, if $x = (x_i)_{i \in M}$ is a vector and $I \subseteq M$, we write x_I for the subvector whose components are indexed by I.

2. Polyhedra associated with the MS-matching problem. Let $G = (W, U, E)$ be a bipartite graph and f be a dependence function on U. The *MS-matching polytope* $\operatorname{MSM}(G,f)$ is the convex hull of incidence vectors of MS-matchings

$$\operatorname{MSM}(G,f) = \operatorname{conv}\{\chi^M | M \text{ is an MS-matching in } G \text{ with respect to } f\}.$$

The MS-matching problem can be written as an integer (binary) linear program:

$$\max\ \mathbf{1}^T x,$$

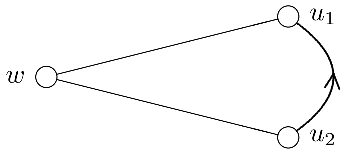


FIG. 2. *A graph G and a dependence function f for which FMSM(G, f) has fractional vertices.*

$$x(\delta(v)) \leq 1 \quad \text{for all } v \in V \setminus \text{dom}(f), \tag{1}$$

$$x(\delta(v)) - x(\delta(f(v))) \leq 0 \quad \text{for all } v \in \text{dom}(f), \tag{2}$$

$$x \in \{0,1\}^{|E|}. \tag{3}$$

Inequality (1) says that each node other than a slave is incident with at most one matched edge. Inequality (2) says that the number of matched edges incident with a slave v is not greater than the number of matched edges incident with its master $f(v)$. Note that the inequalities $x(\delta(v)) \leq 1$ for all $v \in \text{dom}(f)$ follow from the inequalities (1) and (2). Hence the MS-matching polytope can be written as

$$\text{MSM}(G, f) = \text{conv}\{x \in \{0,1\}^{|E|} \mid x \text{ satisfies (1), (2)}\}.$$

If we replace the integrality condition (3) by the weaker condition

$$x_e \geq 0 \quad \text{for all } e \in E, \tag{4}$$

then we obtain the *fractional MS-matching polytope*

$$\text{FMSM}(G, f) = \{x \in \mathbf{R}^{|E|} \mid x \text{ satisfies (1), (2), (4)}\}.$$

As demonstrated by the following example this polytope may have nonintegral vertices.

Example 2.1. Consider the graph G and the dependence function f of Figure 2. The vector $x \in \mathbf{R}^2$, $x_{wu_1} = x_{wu_2} = \frac{1}{2}$, is a nonintegral vertex of FMSM(G, f), since it is the unique optimal solution to max y_{wu_2}, $y \in$ FMSM(G, f).

The dual of the linear program max $\mathbf{1}^T x$, $x \in$ FMSM(G, f), is

$$\min \; y(V \setminus \text{dom}(f)) = \sum_{v \in V \setminus \text{dom}(f)} y_v,$$

$$y_w + y_u \geq 1 \quad \text{for all } [w, u] \in E,\ u \in U \setminus \text{ran}(f), \tag{5}$$

$$y_w + y_u - y_{f^{-1}(u)} \geq 1 \quad \text{for all } [w, u] \in E,\ u \in \text{ran}(f), \tag{6}$$

$$y_v \geq 0 \quad \text{for all } v \in V. \tag{7}$$

It is easy to see that the integral solutions to this linear program are MS-node covers in G with respect to f. Therefore we call the polyhedron

$$\text{MSC}(G, f) = \{y \in \mathbf{R}^{|V|} \mid y \text{ satisfies (5), (6), (7)}\}$$

the *MS-node cover polyhedron*. We will see in section 4 that all the vertices of MSC(G, f) are integral.

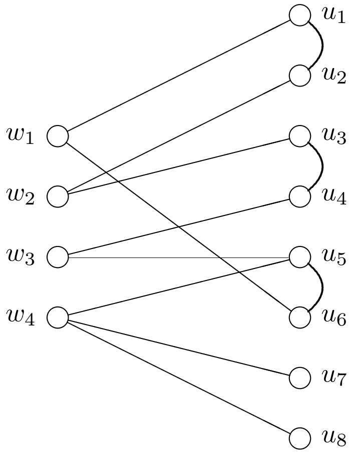


FIG. 3. *The auxiliary graph corresponding to the graph and dependence function of Figure* 1.

3. Odd path cycles and the polytope FMSM(G, f). Now we consider the polytope FMSM(G, f). To study its vertices we introduce an auxiliary graph.

The *auxiliary graph* corresponding to $G = (V, E)$ and f is the graph $H = (V, F)$ where $F = E \cup \{ [v, f(v)] \mid v \in \mathrm{dom}(f)\}$. The edges in $F \setminus E$ will be called *artificial edges*.

An *odd path cycle* of G is a subset $C \subseteq E$ which can be extended to an odd cycle in H by adding only artificial edges.

Hence an odd path cycle of G consists of an odd number of node disjoint paths which have even length. The endnodes of these paths are nodes in U.

Example 3.1. Consider the graph G and the dependence function f of Figure 1. The auxiliary graph H corresponding to G and f is shown in Figure 3. The edge set $C = \{[w_1, u_1], [w_1, u_6], [w_2, u_2], [w_2, u_3], [w_3, u_4], [w_3, u_5]\}$ is an odd path cycle of G since $C \cup \{[u_1, u_2], [u_3, u_4], [u_5, u_6]\}$ is an odd cycle in H.

The following theorem characterizes the vertices of FMSM(G, f).

THEOREM 3.2. *Let $x \in$ FMSM(G, f). If x is a vertex of FMSM(G, f) then $x_e \in \{0, \frac{1}{2}, 1\}$ for all $e \in E$ and the edges e for which $x_e = \frac{1}{2}$ form node disjoint odd path cycles in G.*

Proof. The proof of this theorem is a slight modification of an argument of Grötschel [2, p. 52] given for the 2-matching polytope. Let $\bar{x}$ be an arbitrary vertex of FMSM(G, f). We introduce slack variables $\bar{y}_v$ for each $v \in V$ defined by

$$\bar{y}_v = \begin{cases} 1 - \bar{x}(\delta(v)), & v \in V \setminus \mathrm{dom}(f), \\ \bar{x}(\delta(f(v))) - \bar{x}(\delta(v)), & v \in \mathrm{dom}(f). \end{cases}$$

By definition the vector $\binom{\bar{x}}{\bar{y}}$ satisfies the following equations:

$$x(\delta(v)) + y_v = 1 \quad \text{for all } v \in V \setminus \mathrm{dom}(f), \tag{8}$$

$$x(\delta(v)) - x(\delta(f(v))) + y_v = 0 \quad \text{for all } v \in \mathrm{dom}(f), \tag{9}$$

$$x, y \geq \mathbf{0}. \tag{10}$$

If we add for a node $v \in \mathrm{dom}(f)$ equation (8) for $f(v)$ to equation (9) for v, then equations (9) become

$$x(\delta(v)) + y_{f(v)} + y_v = 1 \quad \text{for all } v \in \operatorname{dom}(f). \tag{11}$$

Then $\binom{\bar{x}}{\bar{y}}$ is a vertex of the polytope

$$\begin{aligned}&\{\tbinom{x}{y} \in \mathbf{R}^{|E|+|V|} \mid x \text{ satisfies (8), (11), (10)}\}\\ &= \{\tbinom{x}{y} \in \mathbf{R}^{|E|+|V|} \mid C\tbinom{x}{y} = \mathbf{1}, x \geq \mathbf{0}, y \geq \mathbf{0}\},\end{aligned}$$

where C is the $|V| \times (|E| + |V|)$ coefficient matrix given by equations (8) and (11). The submatrix $C_{\cdot E}$ of C is the node-edge incidence matrix of G. If $v \notin \operatorname{ran}(f)$ then the column vector $C_{\cdot v}$ is the vth unit vector e_v. For $v \in \operatorname{ran}(f)$ the vector $C_{\cdot v}$ is the vector $e_v + e_{f^{-1}(v)}$; i.e., $C_{\cdot v}$ is the incidence vector of the artificial edge $[v, f^{-1}(v)]$. Hence, the submatrix $C_{\cdot E \cup \operatorname{ran}(f)}$ is the node-edge incidence matrix of the auxiliary graph H corresponding to G and f. Let E' be the set of edges e for which $\bar{x}_e$ is nonintegral and V' be the set of nodes v for which $\bar{y}_v$ is nonintegral. We have to show that $\bar{x}_e = \frac{1}{2}$ for all $e \in E'$ and that E' is the union of node disjoint odd path cycles. Let C' be the matrix which we obtain if we delete all zero rows in the matrix $C_{\cdot E' \cup V'}$ and let p and q be the number of rows and columns of C', respectively. Now we have

$$C'_{\cdot E'}\bar{x}_{E'} + C'_{\cdot V'}\bar{y}_{V'} = \mathbf{1}. \tag{12}$$

Since the columns of C' are linearly independent, we have $p \geq q$. Let r be the number of entries 1 in C'. Each column of C' has at most two 1's; i.e., $r \leq 2q$. On the other hand, each row of C' has at least two 1's, since each component of $x_{E'}$ and $y_{V'}$ is nonintegral but the right-hand side of equation (12) is integral. This means that $r \geq 2p$ and $p \leq q$, and, consequently, $p = q$. Therefore, C' is a quadratic nonsingular 0–1-matrix with exactly two nonzeros in each row and each column; i.e., $V' \subseteq \operatorname{ran}(f)$. It is well known in combinatorial optimization [8] that such a matrix is the node-edge incidence matrix of odd cycles in H. The unique solution to system (12) is $\bar{x}_e = \frac{1}{2}$ for all $e \in E'$ and $\bar{y}_v = \frac{1}{2}$ for all $v \in V'$. Since the column vectors of the submatrix $C'_{\cdot V'}$ are the incidence vectors of artificial edges in H, the submatrix $C'_{\cdot E'}$ is the node-edge incidence matrix of node disjoint odd path cycles in G. □

This characterization of the vertices of FMSM(G, f) is very useful for the optimization. Assume that H is bipartite. In this case a polynomial method for solving the MS-matching problem is straightforward. Since H has no cycle of odd length, G cannot have an odd path cycle. From Theorem 3.2 it follows that FMSM(G, f) has no fractional vertices and hence FMSM(G, f) = MSM(G, f). Thus the problem can be reduced to finding an optimal vertex solution to the linear program $\mathbf{1}^T y$, $y \in$ FMSM(G, f). It is well known that Khachians ellipsoid method [6] can be used to solve this problem in polynomial time (see [3], [4]).

On the other hand, if H is not bipartite, then FMSM(G, f) may have fractional vertices as demonstrated in Example 2.1. Although fractional vertices of FMSM(G, f) cannot be interpreted as MS-matchings we can still optimize over FMSM(G, f) to obtain an optimal solution. This is shown by the following theorem.

THEOREM 3.3. *The linear program* max $\mathbf{1}^T y$, $y \in$ *FMSM*(G, f), *always has an integral optimal solution. Such an integral optimal solution can be constructed from any fractional optimal vertex solution.*

Proof. Let x be a fractional optimal vertex solution to max $\mathbf{1}^T y$, $y \in$ FMSM(G, f), and $E' = \{e \in E \mid x_e \notin \mathbf{Z}\}$. From Theorem 3.2 we know that the edges in E'

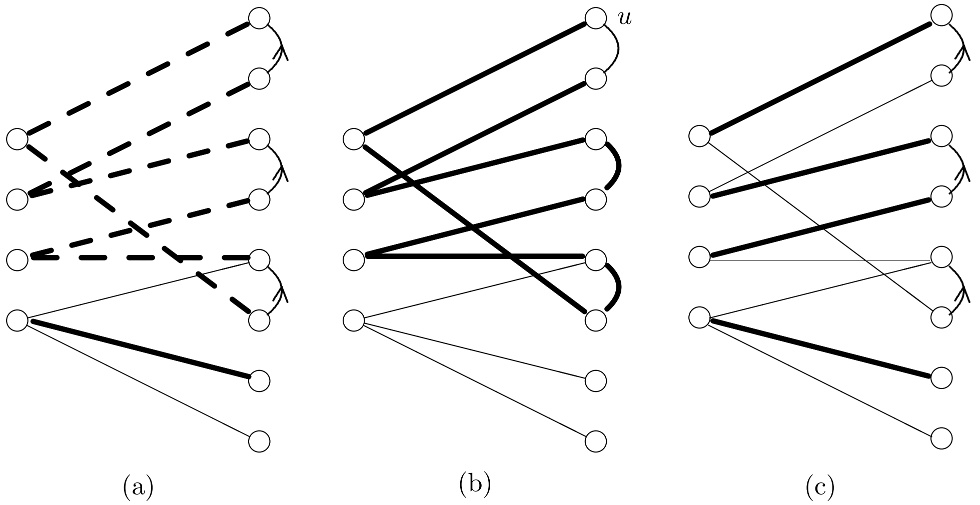


FIG. 4. (a) *A fractional optimal vertex solution. Solid thick edges correspond to variables with value* 1*, dashed thick edges to variables with value* $\frac{1}{2}$*, and thin edges to zero-valued variables.* (b) *The path* P*.* (c) *The integral optimal solution obtained from rounding.*

form node disjoint odd path cycles. Let us choose one odd path cycle C from E' and a master u which is the endnode of one of the paths in C. Now we can extend C to an odd cycle in the auxiliary graph H and drop the artificial edge which is incident to u. This yields a path P in H which has even length and the node u as one of its endnodes. Now we traverse the path P starting from node u. If a nonartificial edge e of P has odd rank, we round up x_e to 1; if e has even rank, we round down x_e to 0. The new solution x' resulting from this rounding procedure is again valid. Note that if we decrease the value $x(\delta(v))$ for a master v, we decrease the value $x(\delta(f^{-1}(v)))$ for its slave $f^{-1}(v)$, too; conversely, if we increase the value $x(\delta(v))$ for a slave v, we also increase the value $x(\delta(f(v)))$ for its master $f(v)$. Moreover since all the paths in the odd path cycle have even length we have rounded up as many variables as we have rounded down. Thus x' is in FMSM(G, f) and has the same objective value as x but strictly fewer fractional components. If we repeat this procedure for each odd path cycle, we obtain an integral optimal solution. □

As mentioned above, an optimal vertex solution to max $\mathbf{1}^T y$, $y \in$ FMSF(G, f) can be found in polynomial time. Since the rounding procedure described in the proof of Theorem 3.3 can obviously be done in $O(|E|)$ time, we have a polynomial algorithm for finding an MS-matching of maximum cardinality. (Alternatively, an optimal MS-matching can be found by reducing the problem to a nonbipartite matching problem; see [5].)

COROLLARY 3.4. *The MS-matching problem for a bipartite graph and a dependence function can be solved in polynomial time.*

Example 3.5. Consider again the MS-matching instance of Figure 1. A fractional optimal vertex solution is shown in Figure 4(a). The dashed thick edges correspond to fractional variables and form a single odd path cycle. Following the construction in the proof of Theorem 3.3 we choose $u = u_1$ and obtain the path P which is shown in Figure 4(b). By rounding up the variables corresponding to odd ranked edges of P and rounding down the variables corresponding to even ranked edges of P we obtain the integral optimal solution of Figure 4(c).

4. The polyhedron MSC($\boldsymbol{G, f}$). Next we turn to the polyhedron MSC(G, f).

THEOREM 4.1. *All vertices of the MS-node cover polyhedron MSC(G, f) are integral.*

Proof. Assume we had a nonintegral vertex y of MSC(G, f). We will show that y can be represented by a proper convex combination of two points $y^1, y^2 \in$ MSC(G, f). This will be a contradiction to the assumption that y is a vertex of MSC(G, f). For the definition of y^1 and y^2 we need four vectors $z, d^1, d^2, d \in \mathbf{R}^{|V|}$ defined by

$$z_v = \begin{cases} 1 - y_v, & v \in W, \\ y_v, & v \in U \setminus \operatorname{ran}(f), \\ y_v - y_{f^{-1}(v)}, & v \in \operatorname{ran}(f); \end{cases}$$

$$d_v^1 = \begin{cases} -1, & (v \in W \text{ and } z_v \in \mathbf{R}_+ \setminus \mathbf{Z}) \text{ or } (v \in U \text{ and } z_v \in \mathbf{R}_- \setminus \mathbf{Z}), \\ 1, & (v \in W \text{ and } z_v \in \mathbf{R}_- \setminus \mathbf{Z}) \text{ or } (v \in U \text{ and } z_v \in \mathbf{R}_+ \setminus \mathbf{Z}), \\ 0 & \text{otherwise;} \end{cases}$$

$$d_v^2 = \begin{cases} 1, & v \in \operatorname{ran}(f) \text{ and } z_{f^{-1}(v)} \notin \mathbf{Z}, \\ 0 & \text{otherwise;} \end{cases}$$

$$d = d^1 + d^2.$$

We define $y^1 = y + \epsilon d$ and $y^2 = y - \epsilon d$ where $\epsilon > 0$ is a sufficiently small real number. Then $y = \frac{1}{2}y^1 + \frac{1}{2}y^2$ is a convex combination of y^1 and y^2. To complete the proof we have to show that d is not the zero vector (i.e., the convex combination is proper) and y^1 and y^2 are valid points of MSC(G, f).

Let y_v be some nonintegral component of y. We distinguish three cases in order to show that d is not the zero vector:

Case 1. $v \in W$. Then $d_v = d_v^1 \in \{-1, 1\}$.

Case 2. $v \in U \setminus \operatorname{ran}(f)$. Then $z_v = y_v \in \mathbf{R}_+ \setminus \mathbf{Z}$; hence $d_v = d_v^1 = 1$.

Case 3. $v \in \operatorname{ran}(f)$. If $z_v = y_v - y_{f^{-1}(v)} \in \mathbf{Z}$, then $y_{f^{-1}(v)} \notin \mathbf{Z}$ since $y_v \notin \mathbf{Z}$. Thus $d_v = d_v^2 = 1$. If $z_v \in \mathbf{R}_+ \setminus \mathbf{Z}$ then $d_v = 1 + d_v^2 \geq 1$. Finally, if $z_v \in \mathbf{R}_- \setminus \mathbf{Z}$ then either $y_{f^{-1}(v)} \in \mathbf{Z}$ and $d_v = d_v^1 = -1$ or $y_{f^{-1}(v)} \notin \mathbf{Z}$ and $d_{f^{-1}(v)} = 1$ from Case 2.

In order to show that y^1 and y^2 are valid points of MSC(G, f) we have to check the inequalities (5), (6), and (7). If one of these inequalities is strict for y then it must be valid for both y^1 and y^2 since we have chosen ϵ sufficiently small. Therefore it is sufficient to check those inequalities which are satisfied by y with equality. In the following we only show that $y^1 \in$ MSC(G, f). The validity of y^2 is shown analogously.

Let $[w, u] \in E$, $u \in U \setminus \operatorname{ran}(f)$ and $y_w + y_u = 1$. Then $z_w = 1 - y_w = y_u = z_u$. We distinguish two cases:

Case 1. $z_w \in \mathbf{Z}$. Then $d_w = 0$ and $d_u = 0$; hence $y_w^1 = y_w$ and $y_u^1 = y_u$.

Case 2. $z_w \notin \mathbf{Z}$. Then $z_w \in \mathbf{R}_+ \setminus \mathbf{Z}$ since $z_w = y_u \geq 0$. Hence $d_w = -1$ and $d_u = 1$ and thus $y_w^1 + y_u^1 = y_w + \epsilon d_w + y_u + \epsilon d_u = 1$.

Let $[w, u] \in E$, $u \in \operatorname{ran}(f)$ and $y_w + y_u - y_{f^{-1}(u)} = 1$. Then $z_w = 1 - y_w = y_u - y_{f^{-1}(u)} = z_u$. We distinguish three cases:

Case 1. $z_w \in \mathbf{Z}$. Then $d_w = 0$; i.e., $y_w^1 = y_w$. Now if $z_{f^{-1}(u)} \in \mathbf{Z}$ then $d_u = 0$ and $d_{f^{-1}(u)} = 0$; hence $y_u^1 = y_u$ and $y_{f^{-1}(u)}^1 = y_{f^{-1}(u)}$. Otherwise $z_{f^{-1}(u)} = y_{f^{-1}(u)} \in \mathbf{R}_+ \setminus \mathbf{Z}$; i.e., $d_u = d_u^2 = 1$ and $d_{f^{-1}(u)} = d_{f^{-1}(u)}^1 = 1$. Hence $y_w^1 + y_u^1 - y_{f^{-1}(u)}^1 = y_w + (y_u + \epsilon) - (y_{f^{-1}(u)} + \epsilon) = 1$.

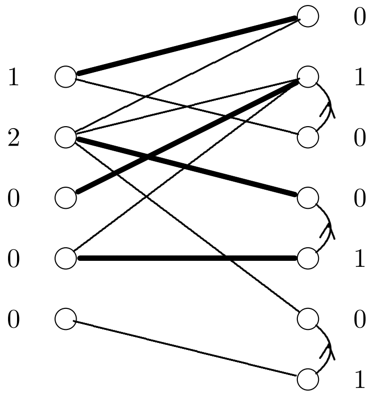


FIG. 5. *A maximum MS-matching and a minimum MS-node cover.*

Case 2. $z_w \in \mathbf{R}_+ \setminus \mathbf{Z}$. Then $d_w = -1$. Now if $z_{f^{-1}(u)} \in \mathbf{Z}$ then $d_u = 1$ and $d_{f^{-1}(u)} = 0$; hence $y^1_w + y^1_u - y^1_{f^{-1}(u)} = (y_w - \epsilon) + (y_u + \epsilon) - y_{f^{-1}(u)} = 1$. Otherwise $z_{f^{-1}(u)} = y_{f^{-1}(u)} \in \mathbf{R}_+ \setminus \mathbf{Z}$; i.e., $d_u = 2$ and $d_{f^{-1}(u)} = 1$. Hence $y^1_w + y^1_u - y^1_{f^{-1}(u)} = (y_w - \epsilon) + (y_u + 2\epsilon) - (y_{f^{-1}(u)} + \epsilon) = 1$.

Case 3. $z_w \in \mathbf{R}_- \setminus \mathbf{Z}$. Then $d_w = 1$. Now if $z_{f^{-1}(u)} \in \mathbf{Z}$ then $d_u = -1$ and $d_{f^{-1}(u)} = 0$; hence $y^1_w + y^1_u - y^1_{f^{-1}(u)} = (y_w + \epsilon) + (y_u - \epsilon) - y_{f(-1)(u)} = 1$. Otherwise $z_{f^{-1}(u)} = y_{f^{-1}(u)} \in \mathbf{R}_+ \setminus \mathbf{Z}$; i.e., $d_u = -1 + 1 = 0$ and $d_{f^{-1}(u)} = 1$. Thus $y^1_w + y^1_u - y^1_{f^{-1}(u)} = (y_w + \epsilon) + y_u - (y_{f^{-1}(u)} + \epsilon) = 1$.

Let $v \in V$ and $y_v = 0$. In order to show that $y^1_v = 0$ we show $d_v = 0$. Again we distinguish three cases.

Case 1. $v \in W$. Then $d_v = 0$ follows immediately from the definition of d.

Case 2. $v \in U \setminus \operatorname{ran}(f)$. Then $z_v = y_v = 0$ and hence $d_v = 0$.

Case 3. $v \in \operatorname{ran}(f)$. Then $z_v = -y_{f^{-1}(v)} \leq 0$. Now either $z_v \in \mathbf{Z}$ and $y_{f^{-1}(v)} \in \mathbf{Z}$ or $z_v \in \mathbf{R}_- \setminus \mathbf{Z}$ and $y_{f^{-1}(v)} \notin \mathbf{Z}$. In both cases we have $d_v = 0$. □

5. A min-max theorem.

Our main result is based on the theorems of sections 3 and 4.

THEOREM 5.1. *Let $G = (W, U, E)$ be a bipartite graph and f be a dependence function on U. The cardinality k of any MS-matching in G is no greater than the weight w of any MS-node cover in G. Furthermore the cardinality of a maximum MS-matching and the weight of a minimum MS-node cover are the same.*

Proof. Let k be the cardinality of an arbitrary MS-matching and w be the weight of an arbitrary MS-node cover. Then we have

$$\begin{aligned} k &\leq \max\,\{\mathbf{1}^T x \mid x \in \mathrm{MSM}(G,f)\} \\ &= \max\,\{\mathbf{1}^T x \mid x \in \mathrm{FMSM}(G,f)\} \\ &= \min\,\{y(V \setminus \operatorname{dom}(f)) \mid y \in \mathrm{MSC}(G,f)\} \\ &= \min\,\{y(V \setminus \operatorname{dom}(f)) \mid y \in \mathrm{MSC}(G,f), y \in \mathbf{Z}^{|V|}\} \\ &\leq w. \end{aligned}$$

The first and third equalities are proved in Theorems 3.3 and 4.1, respectively. The second equality follows from the duality theorem of linear programming. If k is the cardinality of a maximum MS-matching and w is the weight of a minimum MS-node cover, then the two inequalities are tight. This proves the theorem. $\square$

Example 5.2. In Figure 5 we see an MS-matching problem with an MS-matching of cardinality 4 and an MS-node cover of weight 4 showing that both are optimal.

Remark. Note that if the dependence function f is the partial function which is nowhere defined, then every MS-matching in a graph is an ordinary matching and every minimum MS-node cover is the incidence vector of a node cover. Thus the above min-max theorem can be considered as a generalization of the theorem of König.

REFERENCES

[1] M. R. Garey and D. S. Johnson, *Computers and Intractability*, Freeman, New York, 1979.

[2] M. Grötschel, *Polyedrische Charakterisierungen kombinatorischer Optimierungsprobleme*, Verlag Anton Hain, Meisenheim am Glan, 1977.

[3] M. Grötschel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica, 1 (1981), pp. 169–197.

[4] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer, Berlin, 1988.

[5] A. Hefner and P. Kleinschmidt, *A constrained matching problem*, Ann. Oper. Res., 57 (1995), pp. 135–145.

[6] L. G. Khachiyan, *A polynomial algorithm in linear programming*, Dokl. Akad. Nauk SSSR, 244 (1979), pp. 1093–1096 (in Russian); Soviet Math. Dokl., 20 (1979), pp. 191–194 (in English).

[7] D. König, *Gráfok és alkalmazásuk a determinánsok és halmazok elméletében*, Mathematikai és Természettudományi Értesitö, 34 (1916), pp. 104–119 (in Hungarian); *Über Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre*, Math. Ann., 77, pp. 453–465 (in German).

[8] M. W. Padberg, *On the facial structure of set packing polyhedra*, Math. Programming, 5 (1973), pp. 199–215.

[9] W. R. Pulleyblank, *Polyhedral combinatorics*, in Handbooks in Operations Research and Management Science, Vol. 1, G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, eds., North-Holland, Amsterdam, 1989, pp. 371–446.

SIAM J. Discrete Math.
Vol. 10, No. 2, pp. 190–200, May 1997

HYPERCUBES AND MULTICOMMODITY FLOWS*

B. YU†, J. CHERIYAN†, and P. E. HAXELL†

Abstract. The average degree of a subgraph H of the r-dimensional hypercube Q_r equals at most the maximum Hamming distance of any two nodes in H. A corollary is that the minimum number of edges to delete from Q_r such that any two nodes at Hamming distance ℓ are separated is $(r+1-\ell)2^{r-1}$. This corollary has applications to multicommodity flows.

Key words. cube graphs, compression, average degree, multicommodity flows, minimum cuts

AMS subject classifications. 68R10, 05C85, 90C27

PII. S089548019426560X

1. Introduction.

Let r be a positive integer. The r-dimensional hypercube Q_r is defined recursively in terms of the cartesian product of Q_{r-1} and the complete graph on two nodes K_2 as follows:

$$Q_1 = K_2,$$
$$Q_r = K_2 \times Q_{r-1}.$$

Alternatively, Q_r may be defined as a graph whose node set consists of 2^r r-dimensional boolean vectors (i.e., vectors with zero–one coordinates), where two nodes are adjacent whenever they differ in exactly one coordinate. Throughout the paper, a node of Q_r is identified with its boolean vector. We denote the coordinates of an r-dimensional boolean vector v by $v_1, v_2, \ldots, v_r$. For two boolean vectors v and w having the same dimension, their *Hamming distance*, denoted by $d(v,w)$, is defined to be the number of coordinates where they differ, $\sum_i |v_i - w_i|$. For a subgraph H of Q_r the *Hamming diameter*, denoted by $d(H)$, is defined to be the maximum over all pairs of nodes $v,\ w \in H$ of $d(v,w)$. The *average degree* of H equals twice the ratio of the number of edges of H to the number of nodes, $2|E(H)|/|V(H)|$.

Our main result follows in Theorem 1.1.

Theorem 1.1. *Let H be a subgraph of a hypercube. Then the average degree of H is at most $d(H)$, with equality holding if and only if H is a hypercube.*

This theorem has several interesting corollaries.

Corollary 1.2. *The minimum number of edges to delete from the hypercube Q_r such that any two nodes at Hamming distance ℓ ($\ell = 1, \ldots, r$) are separated is exactly $(r+1-\ell)2^{r-1}$.*

Corollary 1.3. *Let $\mathcal{F}$ be a monotone decreasing family of sets (i.e., if $S \in \mathcal{F}$, then every subset of S is also in $\mathcal{F}$) such that the union of any two sets in the family has cardinality at most ℓ. Then the average cardinality of a set in $\mathcal{F}$ is at most $\ell/2$, with equality holding if and only if $\mathcal{F}$ is the family of all subsets of a set of cardinality ℓ.*

Another consequence of Theorem 1.1 is a result for the existence of a monochromatic connected subgraph of Hamming diameter ℓ in a 2-edge coloring of the hypercube.

[*] Received by the editors March 30, 1994; accepted for publication (in revised form) April 9, 1996. The second and third authors were partially supported by NSERC research grants.
http://www.siam.org/journals/sidma/10-2/26560.html

[†] Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1 (byu@calum.csclub.uwaterloo.ca, jcheriyan@watdragon.uwaterloo.ca, pehaxell@math.uwaterloo.ca).

Corollary 1.4. *The smallest hypercube such that for any* 2*-coloring of its edges there exists a monochromatic connected component of Hamming diameter at least* ℓ *is* $Q_{2\ell-1}$.

Note that our main theorem and its corollaries give tight bounds. For each of the above results, weaker bounds are easy to obtain. We illustrate with weaker versions of Theorem 1.1 and Corollary 1.2.

Let the number of "ones" in an r-dimensional boolean vector v be denoted by $\#v$, i.e., $\#v = \sum_{i=1}^{r} v_i$. For a set of boolean vectors S, $\#S$ denotes $\sum_{v\in S} \#v$, i.e., the number of "ones" summed over all elements of S.

Lemma 1.5. *Let* H *be a subgraph of a hypercube. Then the average degree of* H *is at most* $2d(H)$.

Proof. Let the hypercube be r-dimensional. Consider the set of r-dimensional boolean vectors representing the nodes of Q_r, and assume that the zero vector is in H. Then for each vector $v \in H$, $\#v$ is at most $d(H)$, and so $\sum_{v\in H} \#v$ is at most $d(H)|V(H)|$.

The number of edges in H is at most $\sum_{v\in H} \#v$, since each edge can be associated with a "one" in a coordinate of one of its end nodes such that the other end node has a zero in that coordinate. Hence, the average degree, which equals $2|E(H)|/|V(H)|$, is at most $2\sum_{v\in H} \#v/|V(H)| \leq 2d(H)$. □

Corollary 1.6. *The minimum number of edges to delete from the hypercube* Q_r *such that any two nodes at Hamming distance* ℓ *(*$\ell = 1, \ldots, r$*) are separated is at least* $(r + 2 - 2\ell)2^{r-1}$.

Proof. Let $C \subseteq E$ be an edge set of minimum cardinality such that each connected component of $Q_r\backslash C$ has Hamming diameter $\leq \ell - 1$. By Lemma 1.5, each connected component D has average degree $\leq 2(\ell - 1)$; hence, the number of edges $|E(D)|$ is $\leq (\ell - 1)|V(D)|$. Summing over all connected components, the number of edges in $Q_r\backslash C$ is $\leq (\ell-1)2^r$. Hence, the number of edges in C is at least $|E(Q_r)| - (\ell-1)2^r = (r + 2 - 2\ell)2^{r-1}$. □

Corollaries 1.6 and 1.2 have applications to multicommodity flows.

The next section contains notation and definitions. Section 3 has the proof of the main theorem. Section 4 contains the proofs of the remaining corollaries as well as some basic results on hypercubes. The last section discusses multicommodity flows.

2. Preliminaries on hypercubes. Let B be the set of all r-dimensional boolean vectors, and let H be a fixed subset of B. Regarding the elements of B as the nodes of the r-dimensional hypercube Q_r, it is natural to associate with H the subgraph of Q_r induced by the nodes corresponding to H. We use H and its associated subgraph interchangeably. A *neighbor* of an element $v \in H$ is an element $w \in Q_r$ such that $d(v, w) = 1$. An H*-neighbor* of an element $v \in H$ is a neighbor of v that is in H. A neighbor w of v is called an *up neighbor* of v (respectively, a *down neighbor* of v) if $\#w > \#v$ ($\#w < \#v$). The *edge set* of $H, E(H)$ consists of all unordered pairs $\{u, v\}, u \in H, v \in H$ such that $d(u, v) = 1$. The *degree* of an element $v \in H$, $\deg_H(v)$ is the number of edges containing v, i.e., $\deg_H(v) = |\{w : \{v, w\} \in E(H)\}|$. The *average degree* of H is obviously $\sum_{v\in H} \deg_H(v)/|H|$.

For a subset S of H, let $E'(S)$ denote $E(H)\backslash E(H\backslash S)$; i.e., $E'(S)$ is the subset of $E(H)$ containing all pairs such that at least one element of the pair is in S.

An edge $\{v, w\}$ of Q_r whose adjacent nodes v and w differ in the ith coordinate ($i = 1, \ldots, r$) is called an i*-dimensional edge*; $\{v, w\}$ is said to have dimension i. An *aligned matching* of Q_r is a perfect matching whose edges all have the same dimension. The edges of Q_r can be partitioned into r aligned matchings, one per dimension, and

Q_{r+1} can be obtained by adding an aligned matching between two identical copies of Q_r.

A path P of Q_r, with end nodes, say u and v, is called a *shortest path* if it has the minimum number of edges among all paths of Q_r between u and v.

3. The proof of the main theorem. The compression of a set of boolean vectors is a well-known operation; our description follows [Bo 86]. For a set of boolean vectors H and its compressed set H', the diameter of H' is at most the diameter of H, while the average degree of H' is at least the average degree of H (the proof is given below). Therefore, it suffices to prove the main theorem for H'. Our proof uses induction on the number of nodes in H'. The induction step deletes a special subset S of nodes such that the ratio of the difference between the sum of the node degrees in H' and in $H'\backslash S$ to $|S|$ equals the Hamming diameter of H', i.e.,

$$\sum_{v\in H'} \deg_{H'}(v) - \sum_{v\in H'\backslash S} \deg_{H'\backslash S}(v) \quad = \quad d(H')|S|.$$

3.1. Compression. For a mapping $f : B \to B$ and $v \in B$, $f_j(v)$ denotes the jth coordinate of v's image; i.e., if $f(v) = w$, then $f_j(v) = w_j, j = 1, 2, \ldots, r$.

A *compression* on the ith coordinate P^i is a mapping $B \to B$ such that

$$P^i_j(v) = \begin{cases} v_j & \text{if } j \neq i, \\ 0 & \text{if } j = i. \end{cases}$$

P^i is called an *i-compression.* Intuitively, an i-compression "pushes" a boolean vector along an i-dimensional edge of Q_r. If $v_i = 0$, then note that $P^i(v) = v$.

Let H be a set of boolean vectors. The i-compression of H, $P^i(H)$ is defined to be $\{P^i(v)|v \in H\} \bigcup \{v \in H|P^i(v) \in H\}$. Intuitively, an i-compression of H pushes all of H along i-dimensional edges of Q_r such that no element of H is pushed into another. If $H = P^i(H)$, then H is called an *i-compressed set.* If H is an i-compressed set, then for any element $v \in H$ with $v_i = 1$, the down neighbor w of v with $w_i = 0$ is also in H.

Example. Let $H = \{001, 011, 100\}$. Then $d(H) = 3$ and $|E(H)| = 1$. We have $P^3(H) = \{000, 010, 100\}$, with $d(P^3(H)) = 2$ and $|E(P^3(H))| = 2$.

An i-compression of H obviously preserves the number of elements in H. The next lemma shows that it does not increase the diameter and does not decrease the number of edges.

LEMMA 3.1. *Let H be a set of boolean vectors, and let $P^i(H)$ be the i-compression of H. Then*

(i) $|H| = |P^i(H)|$,

(ii) $|E(H)| \leq |E(P^i(H))|$, *and*

(iii) $d(H) \geq d(P^i(H))$.

Proof. (i) The proof is obvious. (ii) We construct an injection from $E(H)$ to $E(P^i(H))$ as follows: an edge $\{x, y\} \in E(H)$ is mapped to

$$\begin{cases} \{P^i(x), P^i(y)\} & \text{if either } P^i(x) \notin H \text{ or } P^i(y) \notin H, \\ \{x, y\} & \text{otherwise.} \end{cases}$$

It is easy to check that this mapping is an injection. Then the conclusion follows. (iii) Let $x,\ y$ be a pair of vectors in $P^i(H)$ with $d(x, y) = d(P^i(H))$. Let $x' = x$ if $x \in H$, otherwise let x' be the vector in H with $P^i(x') = x$; similarly, let $y' = y$ if $y \in H$, otherwise let y' be the vector in H with $P^i(y') = y$.

If $x_i = y_i$, then it is easily seen that $d(x, y)$ in $P^i(H)$ equals $d(x', y')$ or $d(x', y') - 1$ in H and hence $d(x, y) \leq d(H)$. Otherwise, suppose that $x_i = 0$ and $y_i = 1$. Then observe that $y' = y$. Now, if $x \in H$, then $d(x, y)$ in $P^i(H)$ equals $d(x', y')$ in H, and hence $d(x, y) \leq d(H)$. Otherwise, if $x \notin H$, then observe that $P^i(y) \in H$ because $y \in P^i(H)$ and $y_i = 1$; hence, $d(x, y)$ in $P^i(H)$ equals $d(x', P^i(y))$ in H, so $d(x, y) \leq d(H)$. □

Observe that compressions on two different coordinates i and j commute, i.e., $P^i(P^j(H)) = P^j(P^i(H))$. Therefore, to compress a set on all coordinates, we may fix any ordering on the coordinates and apply i-compressions in the order. A set H is called a *compressed set* if it is compressed on all coordinates, namely, all down-neighbors of an element in H are also in H. The previous lemma obviously applies to the compressed set H' obtained from a set H.

3.2. The proof. Call a pair of elements $u,\ v \in H$ *antipodal* if $d(u, v) = d(H)$; i.e., the number of coordinates where u and v differ equals the Hamming diameter of H. For an antipodal pair $u,\ v$ in H, define Q_{uv} to be the subgraph of Q_r induced by nodes w such that for every coordinate i $(1 \leq i \leq r)$, where u and v agree, w also agrees with them. Note that Q_{uv} contains u and v and is a $d(H)$-dimensional hypercube. A coordinate i $(1 \leq i \leq r)$ is said to be *in* Q_{uv} if u and v differ in that coordinate. For an element $x \in H \cap Q_{uv}$, the antipodal element is the element $y \in Q_{uv}$ such that $d(x, y) = d(H)$.

LEMMA 3.2. *Let $u,\ v$ be an antipodal pair in H, and let Q_{uv} be the $d(H)$-dimensional hypercube induced by u and v.*

(i) *For every antipodal pair $x,\ y \in H \bigcap Q_{uv}$, every H-neighbor of x (or y) is contained in Q_{uv}.*

(ii) *Suppose that H is a compressed set of boolean vectors. Then for every coordinate i $(1 \leq i \leq r)$ that is not in Q_{uv} and for every antipodal pair $x,\ y \in H \cap Q_{uv}$, $x_i = y_i = 0$.*

Proof. (i) First, suppose that $x = u$ and $y = v$. Consider a neighbor w of v that is not in Q_{uv}; i.e., v and w agree on all coordinates except on one coordinate i, and i is not in Q_{uv}. Then u and w also differ in coordinate i, and, further, $u_j \neq w_j$ for each of the coordinates j in Q_{uv}. Hence, $d(u, w) = d(H) + 1$, and so w is not in H.

The proof for other antipodal pairs $x,\ y$ follows from the fact that $Q_{xy} = Q_{uv}$.

(ii) Since H is compressed, for every $x \in H$ all the down neighbors of x are in H. If $x, y \in H \cap Q_{uv}$, coordinate i is not in Q_{uv} and $x_i = 1$; then the down neighbor w of x that has $w_i = 0$ must be in H (by i-compression), and this is not possible since the antipodal element y of x would be at distance $d(x, y) + 1 = d(H) + 1$ from w. □

Denote by S the set of all elements x in $H \bigcap Q_{uv}$ such that there exists a y in $H \bigcap Q_{uv}$ with $d(x, y) = d(H)$; i.e., for every antipodal pair of elements $x,\ y$ in $H \bigcap Q_{uv}$, both x and y are in S.

LEMMA 3.3. *Let H be a compressed set of boolean vectors, and let $u,\ v$ be an antipodal pair in H. Let Q_{uv} and S be as above. Then $|E'(S)| = \#S$.*

Proof. For each edge $\{x, y\} \in E(H)$, if $\#x > \#y$ then orient it as $x \to y$, otherwise orient it as $y \to x$, i.e., orient each edge from its up end to its down end. Now note that for each $x \in S$, the number of edges oriented away from x exactly equals $\#x$, because for every coordinate j with $x_j = 1$ there exists an H-neighbor y that has $y_j = 0$, since H is compressed.

The proof is completed using the following claim.

CLAIM. For every edge $w \to x$ oriented into x $(x \in S)$, the up neighbor w of x is in S.

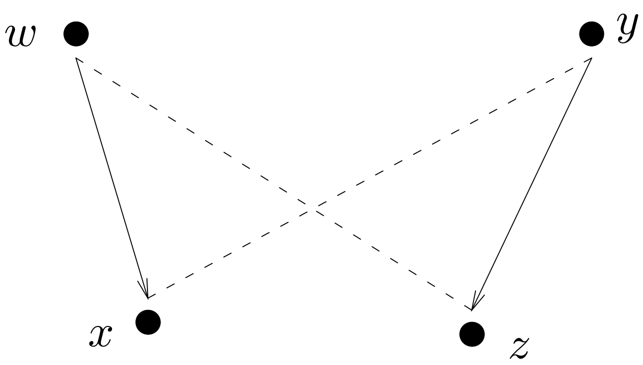


FIG. 1. *A key property of S: for every node $x \in S$ and every edge $w \to x$, w must be in S. Dashed lines indicate antipodal node pairs, i.e., $d(x, y) = d(w, z) = d(H)$.*

Suppose that w and x differ in the ith coordinate; so, $x_i = 0$ and $w_i = 1$ (see Figure 1). Since w is in Q_{uv} (by Lemma 3.2), the ith coordinate must be in Q_{uv}. Consider the antipodal element y of x. This element must be in S, and since $y_i \neq x_i$, $y_i = w_i = 1$. Let z be the down neighbor of y that differs only in the ith coordinate. Since H is compressed, z is in H, and by Lemma 3.2, z is in Q_{uv}. Now observe that $d(w, z) = d(H)$, because w and z differ in all coordinates of Q_{uv}. Hence, by the definition of S, both w and z are in S. This proves the claim and concludes the proof of the lemma. □

The proof of the main theorem follows. The subgraph H in our main theorem need not be connected; also, H need not be an induced subgraph.

Proof of Theorem 1.1. Both parts are proved together by induction on the number of nodes of H. Clearly, the theorem holds for a subgraph with one node.

Let H also denote the set of boolean vectors representing the nodes of the subgraph. Assume that H is a compressed set. This is without loss of generality since Lemma 3.1 shows that compression does not increase the diameter and does not decrease the average degree. In particular, if H has its average degree equal to its Hamming diameter before compression, then after compression either the equality holds or the first part of the theorem is violated.

Let u, v be an antipodal pair of vectors in H, and let Q_{uv} and S be as defined above.

Suppose that $H \backslash S \neq \emptyset$. For every $x \in S$ and every coordinate i not in Q_{uv}, x_i is zero, by Lemma 3.2. It follows that for every antipodal pair x, y in S, $\#x + \#y = d(H)$. Therefore $\#S = d(H)|S|/2$. Since $|E'(S)| = \#S$ (by Lemma 3.3), $|E'(S)| = d(H)|S|/2$. Now, consider $H \backslash S$ and apply the induction hypothesis to the induced subgraph to obtain

$$\frac{2|E(H\backslash S)|}{|V(H\backslash S)|} \leq d(H\backslash S) \leq d(H).$$

Since $E(H)$ is the disjoint union of $E'(S)$ and $E(H\backslash S)$,

$$(*) \quad 2|E(H)| = 2|E(H\backslash S)| + 2|E'(S)| \leq d(H)|V(H\backslash S)| + d(H)|S| = d(H)|V(H)|.$$

That is, the average degree of H is at most $d(H)$. This proves the first part of the theorem. If H has its average degree equal to its Hamming diameter, then inequality $(*)$

holds with equality throughout. Consequently, the average degree of $H\backslash S$ equals $d(H)$, which is at least $d(H\backslash S)$. By the induction hypothesis, $H\backslash S$ is a hypercube whose Hamming diameter equals $d(H)$. This is a contradiction since any hypercube that is strictly contained in H has Hamming diameter less than $d(H)$. Hence, in this case, the average degree of H is less than $d(H)$.

The other case is that $H\backslash S = \emptyset$. Then $H = S$ is a subgraph of Q_{uv}, and the average degree of H is at most the average degree of Q_{uv}, which equals $d(H)$. If H has its average degree equal to its Hamming diameter, then it is easily seen that $H = Q_{uv}$; i.e., H is a hypercube. This completes the proof. $\square$

4. More results on hypercubes. This section contains the proofs of Corollaries 1.2–1.4 and has two more results. Lemma 4.1 is needed to give the lower bound in Corollary 1.4. Lemma 4.3 below is needed to construct maximum multicommodity flows on hypercubes; see Lemma 5.4.

Proof of Corollary 1.2. The number of edges to delete from Q_r to separate any two nodes at Hamming distance ℓ is at least $(r+1-\ell)2^{r-1}$. The proof uses Theorem 1.1 and is similar to the proof of Corollary 1.6.

The bound on $|C|$ is tight: consider the case when each connected component of $Q_r\backslash C$ is a hypercube $Q_{\ell-1}$. $\square$

Remark. Consider the minimum number of edges to delete from Q_r such that any two nodes at Hamming distance ℓ or more are separated. The following observation shows that the bound of Corollary 1.2 applies here too.

If a connected subgraph of Q_r contains two nodes v and z with $d(v,z) = j$, then, for each $i = 1, 2, \ldots, j$, the subgraph has a pair of nodes x and y with $d(x,y) = i$.

Proof of Corollary 1.3. Represent each set S in $\mathcal{F}$ by its 0-1 incidence vector $v(S)$ and let H be the set of incidence vectors corresponding to $\mathcal{F}$. Observe that H is a compressed set since for every $v(S)$ in H, any set S' whose incidence vector is a down neighbor of $v(S)$ is in $\mathcal{F}$ (since $S' \subseteq S$ and $\mathcal{F}$ is monotone decreasing).

Clearly, the cardinality of a set $S \in \mathcal{F}$ equals $\#v(S)$. Further, H has Hamming diameter at most ℓ since the Hamming distance of any two vectors $v(S_1)$ and $v(S_2)$ equals the cardinality of the symmetric difference of S_1 and S_2, which is at most the cardinality of $S_1 \bigcup S_2$. Hence, the average cardinality of a set in $\mathcal{F}$ equals

$$\frac{(\sum_{v\in H} \#v)}{|V(H)|} = \frac{|E(H)|}{|V(H)|} \le \frac{\ell}{2},$$

where the last inequality follows from Theorem 1.1. If the average cardinality of a set in $\mathcal{F}$ equals $\ell/2$, then the average degree of H equals the Hamming diameter of H, so by Theorem 1.1 H is a hypercube. Then $\mathcal{F}$ is the family of all subsets of an ℓ-set. $\square$

LEMMA 4.1. *The edges of the* $2r$*-dimensional hypercube* Q_{2r} *can be colored with two colors such that every maximal monochromatic connected component is a hypercube* Q_r.

Proof. For $1 \le i \le r$, color the i-dimensional edges red, and color the remaining edges (of dimension j, $r < j \le 2r$) blue.

Take any maximal monochromatic connected component G, say, colored red. Let u be any node in G. Node u and all nodes of Q_{2r} whose coordinates differ from u only within the first r coordinates form a red Q_r, denoted by G'. Clearly, $G' \subseteq G$ by the maximality of G.

Moreover, for any edge incident with any node of G', if the edge is not in G', then it must have dimension greater than r, so the edge is colored blue. Hence, $G \subseteq G'$. Since $G = G'$, it follows that G is a Q_r. $\square$

Proof of Corollary 1.4. Let r be the smallest dimension such that for any 2-edge coloring of Q_r there exists a monochromatic connected component with Hamming diameter $\geq \ell$. Theorem 1.1 implies that $r = 2\ell - 1$. When $r = 2\ell - 2$, note that there exists a 2-coloring of Q_r such that each monochromatic connected component is a $Q_{\ell-1}$ by Lemma 4.1; therefore, no such component has Hamming diameter $\geq \ell$.

Now, consider $Q_r = Q_{2\ell-1}$. If a connected component formed by the edges colored red contains a pair of nodes whose Hamming distance is $\geq \ell$, then we are done. Otherwise, the edges colored blue form a multicut that separates all nodes at Hamming distance ℓ, so by Corollary 1.2 this multicut has cardinality $\geq (r + 1 - \ell)|V(Q_r)|/2 = (\ell)|V(Q_r)|/2$. Then the blue subgraph has a connected component whose average degree is $\geq \ell$, therefore Theorem 1.1 implies that this blue connected component has Hamming diameter $\geq \ell$. □

The proof of Lemma 4.3 uses the following observation.

LEMMA 4.2. *A shortest path in Q_r between two nodes v and w has $d(v, w)$ edges. Moreover, a path in Q_r is a shortest path if and only if each edge in the path has a distinct dimension.*

LEMMA 4.3. *The edges of Q_r can be partitioned among $2^r/2$ shortest paths of length r.*

Proof. The proof uses induction on odd r ($r = 1, 3, 5, \ldots$). Define a *shortest r-path partition* of Q_r, denoted S_r, to be a set of $2^r/2$ edge-disjoint shortest paths in Q_r of length r.

INDUCTION HYPOTHESIS. *If r is an odd positive integer, then Q_r has a shortest r-path partition S_r such that each node in Q_r is an end node of exactly one path in S_r.*

Clearly, the induction basis holds with $r = 1$.

Suppose that r is odd and satisfies the induction hypothesis. The induction proof is completed by constructing shortest path partitions for Q_{r+1} and Q_{r+2}. The construction for Q_{r+1} is given below; the construction for Q_{r+2} is similar and is illustrated in Figure 2.

Take two identical copies of Q_r, say Q_r and $\bar{Q}_r$, and form Q_{r+1} by adding an aligned matching between these two copies. Denote the corresponding node in $\bar{Q}_r$ of a node $v \in Q_r$ by $\bar{v}$ and the corresponding path in $\bar{Q}_r$ of a path $P \in S_r$ by $\bar{P}$. Let $\bar{S_r} = \{\bar{P} | P \in S_r\}$. For each path $P \in S_r$, let t_P and h_P denote the end nodes.

Extend each path $P \in S_r$ at node t_P by adding the edge $\{t_P, \bar{t}_P\}$, extend each path $\bar{P} \in \bar{S_r}$ at node $\bar{h}_P$ by adding the edge $\{\bar{h}_P, h_P\}$, and let S_{r+1} be the set of extended paths, i.e.,

$$S_{r+1} = \{P \cup \{t_P, \bar{t}_P\} \mid P \in S_r\} \quad \bigcup \quad \{\bar{P} \cup \{\bar{h}_P, h_P\} \mid \bar{P} \in \bar{S_r}\}.$$

Then each path in S_{r+1} is a shortest path of length $r + 1$, since each of its edges is in a different dimension, and any two paths in S_{r+1} are edge disjoint by construction. □

5. Multicommodity flows and the ratio of the minimum multicut to the maximum multiflow. The *multicommodity flow problem* consists of maximizing the sum of the flows of several commodities, each having its own source and sink, subject to flow conservation and capacity constraints. A *multicut* is defined to be an edge set whose removal from the network separates the source and sink of every commodity, and the multicut's capacity is defined to be the sum of the capacities of the edges in the set. The minimum capacity of a multicut provides an upper bound on the maximum value of a multicommodity flow. (Precise definitions are given below.)

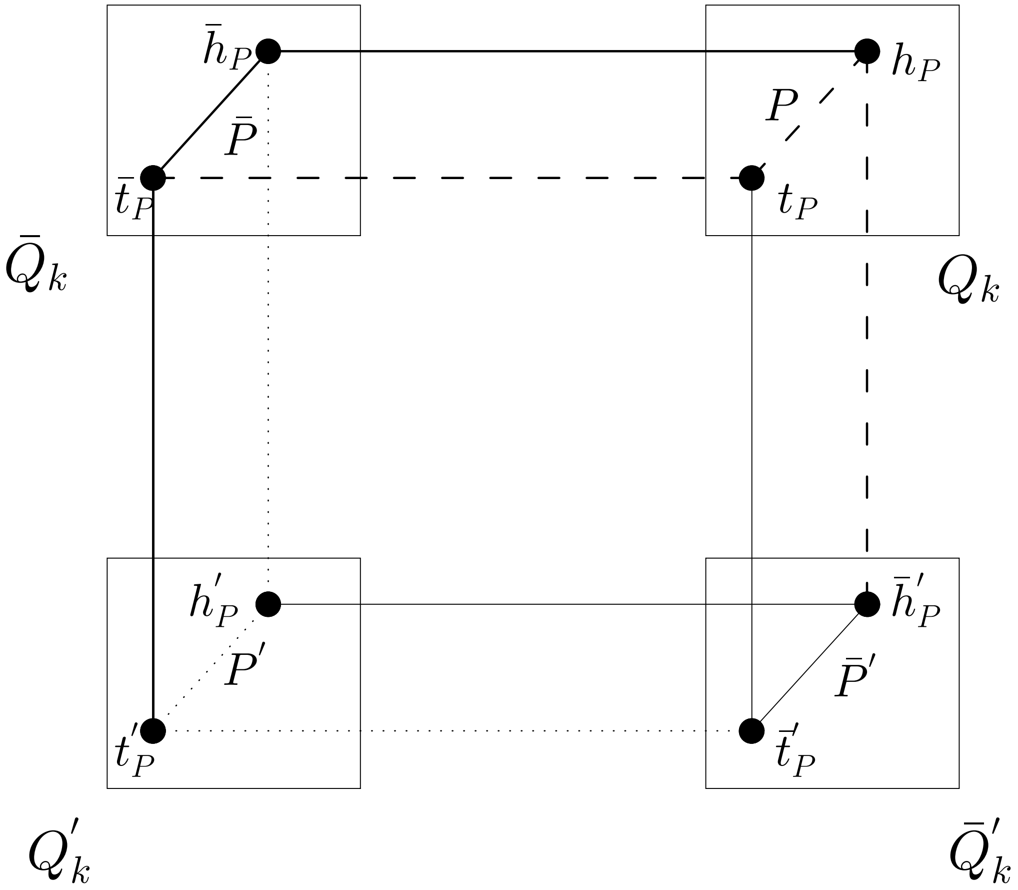


FIG. 2. *Partitioning the edges of Q_{r+2} into shortest paths of length $r+2$, using a partition of the edges of Q_r into shortest paths of length r.*

In the cases of 1-commodity and 2-commodity flows, the max-flow min-cut theorems of Ford and Fulkerson and Hu, respectively, show that the maximum value of the flow equals the minimum capacity of a (multi)cut. However, this equality does not hold for more than two commodities, and it is easy to construct examples showing that the ratio of the minimum capacity of a multicut to the maximum value of a multicommodity flow can be greater than one. (Throughout this section, we use the term *ratio* to mean the ratio just defined.) Building on recent pioneering work by Leighton and Rao [LR 88] and Klein et al. [KRAR 95], Garg, Vazirani, and Yannakakis [GVY 96] showed that the ratio is always $O(\log k)$, where k is the number of commodities. Garg, Vazirani, and Yannakakis considered whether the bound on the ratio is tight, i.e., whether there exist networks and source-sink pairs such that the ratio is $\Omega(\log k)$, and they succeeded in constructing such instances. Their construction is based on expander graphs. In fact, to the best of our knowledge, all known instances for showing an $\Omega(\log k)$ bound on the ratio are based on expanders.

We give a simple and explicit construction for a family of instances where an $\Omega(\log k)$ ratio is achieved. Here is the construction: for an even number r, take an r-dimensional hypercube Q_r and for every pair of nodes at distance $r/2$, specify a commodity with its source at one node of the pair and its sink at the other. Proposition 5.5 below shows that the ratio for the above instance is exactly $r/4 + 1/2$. Since the number of commodities k is at most $\binom{2^r}{2} \leq 2^{2r}$, the ratio is greater than $\log_2 k/8$.

Several recent papers prove approximate max-flow min-cut theorems for problems related to the multicut problem. The Garg–Vazirani–Yannakakis bound of $O(\log k)$ on the ratio of the minimum capacity of a multicut to the maximum value of a multicommodity flow is one such theorem. From the perspective of integer programming, such a theorem focuses on a standard integer programming formulation (IP) of the problem of interest (e.g., the multicut problem) and proves a bound on the integrality gap. By the *integrality gap* we mean the ratio of the optimal value of (IP) to the optimal value of the linear programming relaxation of (IP). Leighton and Rao [LR 88], based on earlier work by Shahrokhi and Matula [SM 90], study the sparsest cut problem and prove that the integrality gap is $O(\log k)$ (here, the number of commodities k is $\Omega(|V|^2)$). Also, [LR 88] gives instances of the sparsest cut problem on expander graphs where the integrality gap is $\Omega(\log k)$. Linial, London, and Rabinovich [LLR 95] show that the integrality gap for a natural generalization of the sparsest cut problem is at most the minimum distortion of an L^1-embedding of the metric $d(G, \ell)$, where G is the associated graph, $\ell : E(G) \to \mathbb{R}_+$ is a specific length function, and $d(G, \ell)$ is the metric of shortest paths distances w.r.t. ℓ on G. The minimum distortion is $O(\log |V|)$; see [LLR 95, Theorem 3.2]. (One consequence is that for instances of the sparsest cut problem on hypercubes, the integrality gap is $O(1)$.) Klein et al. [KPRT 94] and Even et al. [ENSS 95] give approximate max-flow min-cut theorems for a version of the multicut problem on directed graphs and for a Steiner version of the multicut problem. For more information, see the survey papers by Avis and Deza [AD 91], Shmoys [S 96], and Tardos [T 93].

Our results are stated for unit-capacity networks, i.e., networks such that every edge has capacity one. The results carry over to networks such that the capacity of every edge is at most a constant, i.e., $O(1)$.

5.1. Definitions and preliminary results. Let $G = (V, E)$ be an undirected graph, and let $u : E \to \mathbb{R}_+$ be a nonnegative real-valued capacity function on the edges. Let there be k node pairs $s_i,\ t_i,\ i = 1, \ldots, k$, where s_i specifies the source of the ith commodity and t_i specifies the sink. Let $\vec{G} = (V, \vec{E})$ be the digraph obtained from G by replacing each edge $\{v, w\}$ by oppositely oriented edges (v, w) and (w, v). Given a pair of distinguished nodes s and t in $\vec{G}$, an s-t flow f is a real-valued function on the edges of $\vec{G}$ that satisfies the flow conservation condition at every node $v \in V - \{s, t\}$, namely,

$$\sum_{(u,v)\in\Gamma_{in}(v)} f(u, v) \quad - \sum_{(v,w)\in\Gamma_{out}(v)} f(v, w) = 0,$$

where $\Gamma_{in}(v)$ is the set of edges going into v and $\Gamma_{out}(v)$ is the set of edges coming out of v. Note that f may be subject to additional constraints. The value of f, denoted $|f|$, is the net flow into t. A multicommodity flow consists of k s_i-t_i flows $f_1, \ldots, f_k$ on $\vec{G}$ such that for every undirected edge, the sum of the flows on it is at most its capacity, i.e.,

$$\sum_{i=1}^{k} f_i(v, w) + \sum_{i=1}^{k} f_i(w, v) \leq u(\{v, w\}) \qquad \forall \{v, w\} \in E.$$

The value of the multicommodity flow is defined to be the sum of the values of the flows $f_1, \dots, f_k$, $\sum_{i=1}^{k} |f_i|$.

The following result is well known; the proof is included for completeness.

LEMMA 5.1. *Let $G = (V, E)$, and s_i, t_i $(i = 1, \dots, k)$ be an instance of the unit-capacity multicommodity flow problem such that for $i = 1, \dots, k$, the distance in G between s_i and t_i is at least ℓ. Then the maximum value of a multicommodity flow is at most $|E|/\ell$.*

Proof. Let $f_1, \dots, f_k$ be a maximum multicommodity flow. Each f_i $(i = 1, \dots, k)$ can be decomposed into at most $|\vec{E}|$ path flows p_{ij} $(j = 1, \dots, |\vec{E}|)$ such that p_{ij} assigns a constant value $|p_{ij}|$ to each edge of some path from s_i to t_i in $\vec{G}$ and zero values to the other edges. Since every path between s_i and t_i $(i = 1, \dots, k)$ has at least ℓ edges, $\sum_{(v,w)\in\vec{E}} p_{ij}(v, w) \geq \ell |p_{ij}|$. Now, we have

$$\begin{aligned} |E| = \sum_{\{v,w\}\in E} u(\{v, w\}) &\geq \sum_{\{v,w\}\in E} \left(\sum_{i=1}^{k} f_i(v, w) + f_i(w, v) \right) \\ &= \sum_{(v,w)\in\vec{E}} \sum_{i,j} p_{ij}(v, w) \geq \sum_{i,j} |p_{ij}| \cdot \ell; \end{aligned}$$

therefore, the maximum value $\sum_i |f_i| = \sum_{i,j} |p_{ij}|$ is at most $|E|/\ell$. □

Recall the definition of a multicut: given G, u, and s_i, t_i $(i = 1, \dots, k)$ as above, it is an edge set $C \subseteq E$ such that in $G\backslash C$, for each $i = 1, \dots, k$, s_i and t_i are in different connected components; the capacity of the multicut, denoted $u(C)$, is $\sum_{\{v,w\}\in C} u(\{v, w\})$.

The following theorem is due to Garg, Vazirani, and Yannakakis.

THEOREM 5.2 (see [GVY 96]). *Let G, $u : E \to \mathbb{R}$, and s_i, t_i $(i = 1, \dots, k)$ be an instance of the multicommodity flow problem. Then the ratio of the minimum capacity of a multicut to the maximum value of a multicommodity flow is $O(\log k)$.*

5.2. Multicommodity flows on hypercubes. We first use a simple argument to construct a multicommodity flow instance on the hypercube Q_r such that the ratio is $\Omega(\log k)$, where k denotes the number of commodities. Then, we study a modified instance where the ratio can be determined *exactly* using Corollary 1.2 of our main theorem. Though the latter proof is not simple (since it uses Theorem 1.1 and Lemma 4.3), it does yield a higher ratio.

The proof of the next proposition follows easily from Corollary 1.6 and Lemma 5.1.

PROPOSITION 5.3. *For a number r that is a multiple of 4, take an r-dimensional hypercube Q_r and specify a commodity for every pair of nodes at distance $r/4$. For this instance, the ratio of the minimum capacity of a multicut to the maximum value of a multicommodity flow is at least $r/8 + 1/2$.*

The next result constructs a zero-one maximum multicommodity flow on a special but useful instance.

LEMMA 5.4. *Consider a unit-capacity multicommodity flow problem on Q_r, where a commodity is specified for every pair of nodes at distance ℓ. If r is an integer multiple of ℓ, then there exists a maximum multicommodity flow of value $|E(Q_r)|/\ell$ that is integral.*

Proof. The maximum value of the multicommodity flow is at most $|E(Q_r)|/\ell$, by Lemma 5.1. Now, Lemma 4.3 shows that $E(Q_r)$ can be partitioned into 2^{r-1} edge-disjoint shortest paths of length r. Since r/ℓ is an integer, each path in the partition can be split into r/ℓ shortest paths of length ℓ, giving a total of $E(Q_r)/\ell$ edge-disjoint shortest paths of length ℓ. These paths constitute an integral multicommodity flow of maximum value. $\square$

The next proposition follows from Corollary 1.2 and Lemma 5.4.

PROPOSITION 5.5. *For an even number r, take an r-dimensional hypercube Q_r, and specify a commodity for every pair of nodes at distance $r/2$. For this instance, the ratio of the minimum capacity of a multicut to the maximum value of a multicommodity flow is exactly $r/4 + 1/2$.*

REFERENCES

[AD 91] D. AVIS AND M. DEZA, *The cut cone, L^1 embeddability, complexity, and multicommodity flows*, Networks, 21 (1991), pp. 595–617.

[Bo 86] B. BOLLABÁS, *Combinatorics*, Cambridge University Press, Cambridge, UK, 1986.

[ENSS 95] G. EVEN, J. NAOR, B. SCHIEBER, AND M. SUDAN, *Approximating minimum feedback sets and multi-cuts in directed graphs*, in Proc. 4th I.P.C.O., E. Balas and J. Clausen, eds., LNCS 920, Springer-Verlag, Berlin, 1995.

[GVY 96] N. GARG, V. VAZIRANI, AND M. YANNAKAKIS, *Approximate max-flow min-(multi)cut theorems and their applications*, SIAM J. Comput., 25 (1996), pp. 235–251.

[KRAR 95] P. KLEIN, S. RAO, A. AGRAWAL, AND R. RAVI, *An approximate max-flow min-cut relation for undirected multicommodity flow, with applications*, Combinatorica, 15 (1995), pp. 187–202.

[KPRT 94] P. KLEIN, S. PLOTKIN, S. RAO, AND E. TARDOS, *Approximation Algorithms for Steiner and Directed Multicuts*, Tech. report ORIE TR-1119, Cornell University, Ithaca, NY, 1994.

[LR 88] F. T. LEIGHTON AND S. RAO, *An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms*, in Proc. 29th IEEE F.O.C.S., IEEE Press, Piscataway, NJ, 1988, pp. 422–431.

[LLR 95] N. LINIAL, E. LONDON, AND Y. RABINOVICH, *The geometry of graphs and some of its algorithmic applications*, Combinatorica, 15 (1995), pp. 215–245.

[SM 90] F. SHAHROKHI AND D. W. MATULA, *The maximum concurrent flow problem*, J. ACM, 37 (1990), pp. 318–334.

[S 96] D. B. SHMOYS, *Cut problems and their application to divide-and-conquer*, in Approximation Algorithms for NP-hard Problems, D. S. Hochbaum, ed., PWS Publishing Co., Boston, MA, 1996.

[T 93] E. TARDOS, *Approximate min-max theorems and fast approximation algorithms for multicommodity flow problems*, annotated bibliography, in Proc. of the Summer School on Combinatorial Optimization, Maastricht, The Netherlands, August 1993.

SIAM J. DISCRETE MATH.
Vol. 10, No. 2, pp. 201–208, May 1997

004

CONGESTION-FREE OPTIMAL ROUTINGS OF HYPERCUBE AUTOMORPHISMS[*]

MARK RAMRAS[†]

Abstract. We present an off-line method for routing a hypercube automorphism π in the minimum number of steps. The routing has the added virtue of being congestion-free. Our method is purely algebraic, and the routing is obtained easily from the standard representation of π as the product of a complementation and a bit permutation.

Key words. hypercube, routing, automorphism, BPC permutation

AMS subject classifications. 05C, 68

PII. S0895480194275175

Introduction. Among the permutations frequently routed on a hypercube network are bit permutations. For example, the bit-reversal permutation $x_1x_2\cdots x_n \mapsto x_nx_{n-1}\cdots x_1$ and the transpose permutation

$$x_1x_2\cdots x_n \mapsto x_{\lfloor \frac{n}{2}\rfloor+1}\cdots x_nx_1\cdots x_{\lfloor \frac{n}{2}\rfloor}$$

often arise in practice. Since these two exhibit worst-case performance for the greedy routing algorithm [Le, section 3.4.2], their routings are generally precomputed off-line. In this paper we present an off-line method for routing any hypercube automorphism π in the minimum possible number of steps. (By a hypercube automorphism, we mean a permutation of the nodes of the hypercube which preserves adjacency.) Hypercube automorphisms are precisely those permutations known in the literature as bit permute complement (BPC) permutations. Nassimi and Sahni [NS1] give an on-line algorithm for routing BPC permutations by means of a Beneš network. Thus their method takes $2n-1$ steps. Their algorithm is normal, which means that (a) in any given step only edges of one particular dimension are used (i.e., the algorithm is uniaxial) and (b) consecutive dimensions of edges are used in consecutive steps. Normal algorithms can be simulated efficiently on bounded-degree variations of the hypercube, such as the butterfly, shuffle-exchange graph and cube-connected cycles. In another paper [NS2], Nassimi and Sahni give an algorithm that routes any BPC permutation in the minimum possible number of steps. Latifi [La] adopts a different approach, allowing communication across a given edge in only one direction at a time (half-duplex communication). His method also routes in the minimum number of steps. However, with both methods, nodes generally store two messages at a time. With our method, the routing is not uniaxial, but it is congestion-free (at each step, each node contains exactly one message). Moreover, at each step at most two dimensions of edges are used. Liu and You [LY] give an algorithm for routing BPC permutations in $2n-1$ steps. Their routing is also congestion-free or, in their terminology, “conflict-free.” In fact, with their method, any n BPC permutations can be simultaneously routed in $2n-1$ steps, and for each permutation the routing is conflict-free. Furthermore, at each step, each edge of the hypercube transmits two messages, one in each direction. Our method, also congestion-free, routes each BPC

[*] Received by the editors October 3, 1994; accepted for publication (in revised form) April 9, 1996.

http://www.siam.org/journals/sidma/10-2/27517.html

[†] Department of Mathematics, Northeastern University, Boston, MA 02115 (mbramras@neu.edu).

permutation π in the minimum possible number of steps, namely the maximum, over all nodes x of the distance between x and $\pi(x)$.

1. Preliminaries. By Q_n we mean the n-dimensional hypercube. Γ will denote an arbitrary connected graph, and π will denote a permutation of $V(\Gamma)$, the nodes of Γ. By $d(x, y)$ we mean the distance in Γ between nodes x and y. We now recall some definitions from [R]:

[R, Definition 1.4] $k(\pi) = \max\ \{d(x, \pi(x)) \mid x \in V(\Gamma)\}$,
$\Delta = \{\pi \in \mathrm{Perm}(\Gamma) \mid k(\pi) = 1\}$,
[R, Definition 1.1] $t_\Delta(\pi) = \min\ \{t \mid \pi \in \Delta^t\}$,
where Δ^t is the set of all t-fold products of elements of Δ.

As explained in the introduction of [R], a representation of π as an element of Δ^t can be naturally identified with a t-step congestion-free routing of π, where by congestion-free we mean that at no time does any node contain more than one message. Thus $t_\Delta(\pi)$ is the minimum number of steps in a congestion-free routing of π. Clearly, any routing of π requires at least $k(\pi)$ steps (in a single step a message can stay put or else travel a distance of 1), and so $t_\Delta(\pi) \geq k(\pi)$. The purpose of this paper is to show that for π any automorphism of Q_n, the lower bound $k(\pi)$ is always achieved, and that minimum routings are easy to construct.

A few words about notation are necessary. We shall express permutations as products of cycles and denote by $(1, 2, \ldots, m)$ the cycle that maps i to $i + 1$ for $1 \leq i \leq m - 1$ and m to 1. We multiply cycles from right to left, so that cycle multiplication behaves exactly like composition of functions.

2. The main result.

Proposition 2.1. *If π is any automorphism of Q_n then $t_\Delta(\pi) = k(\pi)$. In fact, let $\Delta' = \Delta \cap \mathrm{Aut}\,(Q_n)$. Then there is a factorization of π of length $k(\pi)$ in which each factor $\alpha \in \Delta'$.*

We will prove this via a sequence of lemmas. First we recall some facts about the group $\mathrm{Aut}(Q_n)$. For a subset A of $\{1, 2, \ldots, n\}$, the complementation σ_A is the automorphism of Q_n defined by $\sigma_A(x_1, x_2, \ldots, x_n) = (y_1, y_2, \ldots, y_n)$ where

$$y_i = \begin{cases} x_i & \text{if } i \notin A, \\ \overline{x_i} = 1 + x_i \pmod 2 & \text{if } i \in A. \end{cases}$$

For a permutation θ of $\{1, 2, \ldots, n\}$, the automorphism ρ_θ of Q_n is defined by

$$\rho_\theta(x_1, x_2, \ldots, x_n) = x_{\theta(1)}, x_{\theta(2)}, \ldots, x_{\theta(n)}.$$

The mapping $\theta \mapsto \rho_\theta$ is an isomorphism between the symmetric group $\mathcal{S}_n$ and a subgroup of $\mathrm{Aut}(Q_n)$. Every automorphism of Q_n has a unique representation in the form $\sigma_A\, \rho_\theta$. (This is, essentially, the content of Exercises 3.11 and 3.12, pp. 743–744 of [Le].)

Remark. The group $\mathrm{Aut}(Q_n)$, known as the hyperoctahedral group, has been studied for other purposes: for example, Chen and Stanley [CS] answer the question of when a symmetry of Q_n has a fixed k-dimensional subcube, while Chen [C] computes the cycle index polynomial of this group.

Lemma 2.2. *For $2 \leq q \leq n$, and $\theta = (1, 2, \ldots, q)$,*

$$k(\rho_\theta) = t_\Delta(\rho_\theta) = \begin{cases} q & \textit{if } q \equiv 0 \pmod 2, \\ q - 1 & \textit{if } q \equiv 1 \pmod 2. \end{cases}$$

Proof. Let $\pi = \rho_\theta$. First suppose q is even, say, $q = 2m$. Let $x = ((10)^m 0^{n-q})$. Then $\pi(x) = (01)^m 0^{n-q}$, and so $d(x, \pi(x)) = 2m = q$, where d denotes Hamming distance. Thus $k(\rho_\theta) \geq q$. Now since $t_\Delta(\pi) \geq k(\pi)$, it suffices to show that $t_\Delta(\pi) \leq q$, i.e., that $\pi \in \Delta^q$. We do this by induction on m. For $m = 1$, we have

$$\rho_{(12)} = (\sigma_{\{1\}}) \cdot (\sigma_{\{1\}}\rho_{(12)}) \in \Delta^2.$$

For $m \geq 2$ we have

$$\rho_{(1,2,\ldots,2m)} = (\sigma_{\{2m\}}\rho_{(1,2m)}) \cdot (\sigma_{\{1\}}\rho_{(1,2m-1)}) \cdot \rho_{(1,2,\ldots,2(m-1))},$$

and since the first two factors each belong to Δ, the induction hypothesis for $m - 1$ guarantees that the right side of the equation belongs to Δ^q.

Now suppose q is odd, say, $q = 2m + 1$. With x as above we have $d(x, \pi(x)) = 2m = q - 1$, so $k(\pi) \geq q - 1$. As above, it suffices now to show that $t_\Delta(\pi) \leq q - 1$. For $m = 1$, $q = 3$ and

$$\rho_{(1,2,3)} = \rho_{(13)} \cdot \rho_{(12)} = (\sigma_{\{3\}}\rho_{(13)}) \cdot (\sigma_{\{1\}}\rho_{(12)}) \in \Delta^2.$$

For $m \geq 2$,

$$\rho_{(1,2,\ldots,2m+1)} = \rho_{(1,2m,2m+1)} \cdot \rho_{(1,2,\ldots,2m-1)}.$$

The first factor on the right side belongs to Δ^2 by the case $m = 1$ and, by our induction hypothesis, the second factor belongs to $\Delta^{2(m-1)}$. Thus $t_\Delta(\pi) \leq 2m = q - 1$. □

LEMMA 2.3. *If $A \subseteq \{1, 2, \ldots, m\}$ and $\theta = (1, 2, \ldots, m)$ then*

$$k(\sigma_A \, \rho_\theta) = \begin{cases} m & \textit{if } \mid A \mid \equiv m \pmod 2, \\ m - 1 & \textit{otherwise.} \end{cases}$$

Proof. Let $\pi = \sigma_A \, \rho_\theta$. Clearly, $k(\pi) \leq m$. Let $z = z_1 z_2 \cdots z_m 0 \cdots 0$ be defined by $z_1 = 0$ and, for $1 \leq i \leq m - 1$,

$$z_{i+1} = \begin{cases} z_i & \text{if } i \in A, \\ \overline{z_i} & \text{if } i \notin A. \end{cases}$$

Let $\pi(z) = y = y_1 y_2 \cdots y_m 00 \cdots 0$. Then for $1 \leq i \leq m - 1$,

$$y_i = \begin{cases} \overline{z_{i+1}} & \text{if } i \in A, \\ z_{i+1} & \text{if } i \notin A, \end{cases} \quad \text{and} \quad y_m = \begin{cases} \overline{z_1} & \text{if } m \in A, \\ z_1 = 0 & \text{if } m \notin A. \end{cases}$$

We claim that for $1 \leq i \leq m - 1, y_i = \overline{z_i}$. For if $i \in A$, then $y_i = \overline{z_i}$, while if $i \notin A$ then $y_i = z_{i+1} = \overline{z_i}$. Thus in either case we have $y_i = \overline{z_i}$. Hence $d(z, \pi(z)) = d(z, y) \geq m - 1$, and so $k(\pi) \geq m - 1$. Now $y_m = 1$ if $m \in A$ and 0 if $m \notin A$, while $z_m \equiv \mid \{j \leq m - 1 \mid j \notin A\} \mid \pmod 2$. But

$$\mid \{j \leq m - 1 \mid j \notin A\} \mid = \begin{cases} m - \mid A \mid & \text{if } m \in A, \\ m - \mid A \mid - 1 & \text{if } m \notin A. \end{cases}$$

Suppose now that $m - \mid A \mid \equiv 0 \pmod 2$. Then if $m \in A, z_m = 0$, while if $m \notin A, z_m = 1$. So in either case, $y_m = \overline{z_m}$. Thus $d(z, y) = m$. So if $m - \mid A \mid \equiv 0 \pmod 2$, $k(\pi) = m$.

Now assume that $m- \mid A \mid \equiv 1 \pmod 2$. We know that $m-1 \leq k(\pi) \leq m$. Suppose that $k(\pi) = m$. Then for some $x = x_1 \cdots x_n$, $d(x, \pi(x)) = m$. Let $\pi(x) = w = w_1 \cdots w_n$. As before, for $1 \leq i \leq m-1$,

$$(*) \qquad w_i = \begin{cases} \overline{x_{i+1}} & \text{if } i \in A, \\ x_{i+1} & \text{if } i \notin A, \end{cases} \quad \text{and} \quad w_m = \begin{cases} \overline{x_1} & \text{if } m \in A, \\ x_1 & \text{if } m \notin A. \end{cases}$$

Since $d(x, w) = m, w_i = \overline{x_i}$ for $1 \leq i \leq m$, so $w_i + x_i = 1$. Hence,

$$\sum_{i=1}^{m} w_i + \sum_{i=1}^{m} x_i = m.$$

So,

$$w_m + x_1 + \sum_{i=1}^{m-1} (w_i + x_{i+1}) = \sum_{i=1}^{m} w_i + x_i = m.$$

But by $(*)$,

$$w_i + x_{i+1} = \begin{cases} 1 & \text{if } i \in A, \\ 0 \text{ or } 2 & \text{if } i \notin A, \end{cases} \quad \text{and} \quad w_m + x_1 = \begin{cases} 1 & \text{if } m \in A, \\ 0 \text{ or } 2 & \text{if } m \notin A. \end{cases}$$

Hence, $m \equiv \mid A \mid \pmod 2$, contradicting the assumption that $m- \mid A \mid \equiv 1 \pmod 2$. So for all $x, d(x, \pi(x)) \leq m-1$. Hence, $k(\pi) = m-1$. □

Lemma 2.4. *Let θ and A be as in Lemma 2.3, and let $\pi = \sigma_A \rho_\theta$. Then $t_\Delta(\pi) = k(\pi)$.*

Proof. Suppose $A = \emptyset$. Then $\pi = \rho_{(1,2,\ldots,m)}$, and the result follows from Lemma 2.2. So assume that $A \neq \emptyset$. We argue by induction on m. If $m = 1$ then $\theta =$identity, $A = \{1\}$, and $\sigma_A \rho_\theta = \sigma_{\{1\}} \in \Delta$. Now let $m \geq 2$ and assume the result holds for $m-1$. Let i_1 be the least element of A. Since $(i_1, i_1+1, \ldots, m, 1, 2, \ldots, i_1-1) = (1, 2, \ldots, m)$, we may assume, with no loss of generality, that $i_1 = 1$. Now $\sigma_B \, \rho_\phi = \rho_\phi \, \sigma_{\phi^{-1}(B)}$ and

$$(1, 2, \ldots, m) = (1, m) \, (1, 2, \ldots, m-1).$$

Let $B = A - \{1\}$, $\phi = (1, m) = \phi^{-1}$, and $A' = \phi(B)$. Then we have

$$\pi = \sigma_A \rho_\theta = \Big(\sigma_{\{1\}} \, \sigma_B\Big) \Big(\rho_{(1,m)} \, \rho_{(1,2,\ldots,m-1)}\Big) = \Big(\sigma_{\{1\}} \, \rho_{(1,m)}\Big) \Big(\sigma_{A'} \, \rho_{(1,2,\ldots,m-1)}\Big).$$

Since $1 \notin B$, $m \notin A'$. Thus $A' \subseteq \{1, 2, \ldots, m-1\}$. So, by induction, $k(\sigma_{A'} \, \rho_{(1,2,\ldots,m-1)}) = t_\Delta(\sigma_{A'} \, \rho_{(1,2,\ldots,m-1)})$. By Lemma 2.3, this common value

$$= \begin{cases} m-1 & \text{if } \mid A' \mid \equiv m-1 \pmod 2, \\ m-2 & \text{otherwise.} \end{cases}$$

Now $|A'| = |A| - 1$. So $\mid A' \mid \equiv m-1 \pmod 2 \Leftrightarrow \mid A \mid \equiv m \pmod 2$. Thus,

$$k(\sigma_{A'} \, \rho_{(1,2,\ldots,m-1)}) = t_\Delta(\sigma_{A'} \, \rho_{(1,2,\ldots,m-1)}) = \begin{cases} m-1 & \text{if } \mid A \mid \equiv m \pmod 2, \\ m-2 & \text{otherwise.} \end{cases}$$

By Lemma 2.3,

$$k(\pi) = \begin{cases} m & \text{if } \mid A \mid \equiv m \pmod 2, \\ m-1 & \text{otherwise.} \end{cases}$$

So $t_\Delta(\pi) \leq k(\pi)$, and thus we have equality. □

Remark. Lemmas 2.3 and 2.4 hold for any m-cycle. Recall that elements a and b of a group G are called conjugate if $b = gag^{-1}$ for some $g \in G$. Now any two m-cycles are conjugate in the symmetric group $\mathcal{S}_n$. Since $\theta \mapsto \rho_\theta$ is an isomorphism of $\mathcal{S}_n$ into $\mathrm{Aut}(Q_n)$, conjugate permutations are mapped to conjugate automorphisms. But if π' is conjugate to π, then $k(\pi') = k(\pi)$. For let $\pi' = \alpha\pi\alpha^{-1}$. Then for any $x \in Q_n$,

$$d(x, \pi(x)) = d(\alpha(x), \alpha(\pi(x))) = d(\alpha(x), \alpha\pi\alpha^{-1}(\alpha(x))) = d(\alpha(x), \pi'(\alpha(x))),$$

where the first equality follows from the fact that α is an automorphism. Since α is onto, it follows that $k(\pi') = k(\pi)$. Hence, $\pi' \in \Delta \Leftrightarrow \pi \in \Delta$. Thus, if $\pi = \phi_t \cdots \phi_2\phi_1$ is a representation of π as an element of Δ^t, and we let $\phi'_i = \alpha\phi_i\alpha^{-1}$, then $\pi' = \phi'_t \cdots \phi'_2\phi'_1$ is a representation of π' as an element of Δ^t. So t_Δ takes the same value on conjugate automorphisms, and conjugate automorphisms have "conjugate" routings.

Definition 2.5. *For a permutation θ,* support $(\theta) = \{i \mid \theta\,(i) \neq i\}$.

Lemma 2.6. *Let $\theta \in \mathcal{S}_n = \mathrm{Perm}\{1, 2, \ldots, n\}$ and suppose $A \subseteq \mathrm{support}\,(\theta)$. Let $\pi = \sigma_A\,\rho_\theta$. Write θ as a product of disjoint cycles, $\theta = \theta_p \circ \theta_{p-1} \circ \cdots \circ \theta_1$. Let $A_i = A \cap \mathrm{support}\,(\theta_i)$ for $1 \leq i \leq p$. Then $\pi = (\sigma_{A_p}\rho_{\theta_p}) \cdots (\sigma_{A_1}\rho_{\theta_1})$ and*

$$t_\Delta(\pi) = k(\pi) = \sum_{i=1}^{p} k(\sigma_{A_i}\rho_{\theta_i})\,.$$

Proof. Since $A_i \subseteq \mathrm{support}(\theta_i)$ and for $i \neq j$, $\mathrm{support}\,(\theta_i) \cap \mathrm{support}\,(\theta_j) = \emptyset$, we have $A_i \cap \mathrm{support}\,(\theta_j) = \emptyset$. Hence, σ_{A_i} and ρ_{θ_j} commute. Thus $\pi = (\sigma_{A_p}\,\rho_{\theta_p}) \cdots (\sigma_{A_1}\,\rho_{\theta_1})$. By Lemma 2.4 and the preceding Remark, for $1 \leq i \leq p$, $t_\Delta(\sigma_{A_i}\rho_{\theta_i}) = k(\sigma_{A_i}\rho_{\theta_i})$. Now

$$t_\Delta(\pi) \leq \sum_{i=1}^{p} t_\Delta(\sigma_{A_i}\,\rho_{\theta_i}).$$

On the other hand, since $\mathrm{support}\,(\theta_i) \cap \mathrm{support}\,(\theta_j) = \emptyset$ for $i \neq j$, it follows that

$$k(\pi) = \sum_{i=1}^{p} k(\sigma_{A_i}\,\rho_{\theta_i}).$$

Hence, $t_\Delta(\pi) \leq k(\pi)$. Since the reverse inequality holds for all permutations, we have the desired equality. □

Corollary 2.7. *If $\theta \neq$identity then $\sigma_A\,\rho_\theta \in \Delta \Leftrightarrow A = \{i\}$ and $\theta = (i, j)$ for some $i \neq j$.*

Proof. $\sigma_A\,\rho_\theta \in \Delta \Leftrightarrow k(\sigma_A\,\rho_\theta) = 1$. If $A = \{i\}$ and $\theta = (i, j)$ for some $i \neq j$, then by Lemma 2.3, $k(\sigma_A\,\rho_\theta) = 1$. For the reverse implication, let $A = A' \cup B$, where $A' \subseteq \mathrm{support}\,(\theta)$ and $B = A \setminus A'$. Then $\sigma_A\,\rho_\theta = (\sigma_B)\,(\sigma_{A'}\rho_\theta)$ and so by Lemma 2.6

$$k(\sigma_A\,\rho_\theta) = k(\sigma_B) + k(\sigma_{A'}\,\rho_\theta) =\mid B \mid + k(\sigma_{A'}\,\rho_\theta).$$

Hence, $k(\sigma_A\,\rho_\theta) = 1 \Rightarrow \mid B \mid= 0$ and $k(\sigma_{A'}\,\rho_\theta) = 1$. By Lemma 2.3,

$$k(\sigma_{A'}\,\rho_\theta) = \begin{cases} m & \text{if } \mid A' \mid \equiv m \pmod{2}, \\ m-1 & \text{otherwise,} \end{cases}$$

where $m =\mid \text{support}\,(\theta) \mid$. Since $\theta \neq$ identity, $m \geq 2$. Therefore, since $k(\sigma_{A'}\,\rho_\theta) = 1$, $m \leq 2$, and so $m = 2$. Furthermore, $\mid A' \mid\neq$ (mod 2), so $\mid A' \mid= 1$. Since $B = \emptyset$, $A = A'$ and so A and θ have the desired form. □

We can now complete the proof of Proposition 2.1.

Proof of Proposition 2.1. Let $\pi = \sigma_C\,\rho_\theta$. Let $A = C \cap \text{support}\,(\theta)$ and $B = C \setminus A$. Then $\pi = \sigma_B(\sigma_A\,\rho_\theta)$. Since $B \cap \text{support}\,(\theta) = \emptyset$, it follows that $k(\pi) = k(\sigma_B) + k(\sigma_A\,\rho_\theta)$. On the other hand, $t_\Delta(\pi) \leq t_\Delta(\sigma_B) + t_\Delta(\sigma_A\,\rho_\theta)$. But $t_\Delta(\sigma_B) = \mid B \mid= k(\sigma_B)$ and, by Lemma 2.6, $t_\Delta(\sigma_A\,\rho_\theta) = k(\sigma_A\,\rho_\theta)$. So $t_\Delta(\pi) \leq k(\pi)$ and thus $t_\Delta(\pi) = k(\pi)$. □

Corollary 2.8. *Let π and θ be as in Lemma* 2.6. *Then an optimal routing of π can be achieved by routing the factors $\sigma_{A_i}\rho_{\theta_i}$ sequentially.*

Corollary 2.9. *For any $\pi \in \mathrm{Aut}(Q_n)$, there is a minimum routing of π such that at each step the number of dimensions of edges used is at most* 2.

Proof. By Proposition 2.1, π has a factorization in which each factor is either a $\sigma_{\{i\}}$ or a $\sigma_{\{i\}}\rho_{\{i,j\}}$. In the first case only edges of dimension i are used, while in the second, by Corollary 2.7, only edges of dimensions i and j are used. □

Examples. First we shall route the bit-reversal permutation on Q_6. This π_R is defined by $\pi_R(x_1x_2\cdots x_6) = x_6x_5\cdots x_1$. So $\pi = \rho_{(16)}\rho_{(25)}\rho_{(34)}$. Hence, the factorization

$$\pi_R = \Big(\Big(\sigma_{\{1\}}\Big)\Big(\sigma_{\{1\}}\rho_{(16)}\Big)\Big)\Big(\Big(\sigma_{\{2\}}\Big)\Big(\sigma_{\{2\}}\rho_{(25)}\Big)\Big)\Big(\Big(\sigma_{\{3\}}\Big)\Big(\sigma_{\{3\}}\rho_{(34)}\Big)\Big)$$

is a six-step routing for π_R. Geometrically, the steps alternate between 90° rotations and reflections through five-dimensional hyperplanes.

As a second example, we give a routing for the transpose permutation on Q_6 given by $\pi_T(x_1x_2\cdots x_6) = x_4x_5x_6x_1x_2x_3$. Thus $\pi_T = \rho_{(14)}\rho_{(25)}\rho_{(36)} = \rho_{(46)}\,\pi_R\,\rho_{(46)}^{-1}$.

Therefore, conjugating the factorization given above for π_R by $\rho_{(46)}$ we obtain a six-step routing for π_T:

$$\pi_T = \Big(\Big(\sigma_{\{1\}}\Big)\Big(\sigma_{\{1\}}\rho_{(14)}\Big)\Big)\Big(\Big(\sigma_{\{2\}}\Big)\Big(\sigma_{\{2\}}\rho_{(25)}\Big)\Big)\Big(\Big(\sigma_{\{3\}}\Big)\Big(\sigma_{\{3\}}\rho_{(36)}\Big)\Big). \qquad \square$$

Next, we give a recursive algorithm for routing any hypercube automorphism π. It is based on an alternate factorization of an m-cycle:

$$(1, 2, \ldots, m) = (1, 2, 3, \ldots m - 1)(m - 1, m).$$

First note that if $\pi = \sigma_C\rho_\theta$ then $\pi = \sigma_B\,(\sigma_A\,\rho_\theta)$, where $A = C \cap \text{support}\,(\theta)$ and $B = C \setminus A$. Write θ as a product of disjoint cycles $\theta = \theta_p\,\theta_{p-1}\,\cdots\,\theta_1$. For $1 \leq i \leq p$, let $A_i = A \cap \text{support}\,(\theta_i)$. Finally, let $B = \{b_1, b_2, \ldots, b_{|B|}\}$. Then

$$\pi = \Big(\sigma_{b_{|B|}}\cdots\sigma_{b_1}\Big)\Big(\sigma_{A_p}\rho_{\theta_p}\Big)\,\cdots\,\Big(\sigma_{A_1}\rho_{\theta_1}\Big).$$

Now a minimum-step routing for π is obtained by sequentially routing the factors $\sigma_{A_i}\rho_{\theta_i}$ and then the factors σ_{b_j}. Since the latter belong to Δ, it suffices to give an algorithm for routing $\sigma_A\rho_\theta$, where θ is an m-cycle and $A \subseteq \text{support}\,(\theta)$. For simplicity of notation, we assume that $\theta = (1, 2, \ldots, m)$.

Algorithm $\mathcal{A}(A, \theta, m)$.

Input: $m =$an integer ≥ 2, $\theta =$the m-cycle $(1, 2, \ldots, m)$, $A \subseteq \{1, 2, \ldots, m\}$.

Output: A factorization of $\pi = \sigma_A\rho_\theta$ into elements of Δ.

1. Case $m = 2$. If $A = \emptyset$, $\pi = (\sigma_1)(\sigma_1\rho_{(1,2)})$. If $A = \{i\}$, where $i \in \{1, 2\}$, then $\pi = \sigma_i\rho_{(1,2)}$, which is in Δ. If $A = \{1, 2\}$, then $\pi = (\sigma_1)(\sigma_2\rho_{(1,2)}) \in \Delta^2$.

2. Case $m \geq 3$.

$$\alpha_1 = \begin{cases} m & \text{if } m \in A, \\ m-1 & \text{if } m \notin A, \end{cases}$$

and

$$A' = \begin{cases} A \setminus \{m\} & \text{if } m \in A, \\ A \oplus \{1\} & \text{if } m \notin A, \end{cases}$$

where $\oplus$ denotes the symmetric difference operator, i.e., $A \oplus B = (A \setminus B) \cup (B \setminus A)$. Then the first factor of π is $\sigma_{\{\alpha_1\}}\rho_{(m-1,m)}$, an element of Δ. For the remaining factors, call $\mathcal{A}(A', \theta', m-1)$, where $\theta' = (1, 2, \ldots, m-1)$.

Proposition 2.10. *The output of Algorithm $\mathcal{A}(A, \theta, m)$ is a minimum-length factorization of π via elements of Δ, i.e., a minimum-step routing of π.*

Proof. The proof is by induction on m. The case $m = 2$ is clear. Now let $m \geq 3$ and assume the truth of the proposition for $m - 1$. First suppose that $m \in A$. Then $\alpha_1 = m$ and $A' = A \setminus \{m\}$. Now

$$\pi = \sigma_{A'}\sigma_{\{m\}}\rho_{(1,2,\ldots,m-1)}\rho_{(m-1,m)} = \left(\sigma_{A'}\rho_{(1,2,\ldots,m-1)}\right)\left(\sigma_{\{m\}}\rho_{(m-1,m)}\right)$$

and $\sigma_{\{m\}}\rho_{(m-1,m)} \in \Delta$. $A' \subseteq \{1, 2, \ldots m-1\}$, so by our induction hypothesis, Algorithm $\mathcal{A}(A', \theta', m-1)$ returns a minimum-length factorization of $\sigma_{A'}\rho_{(1,2,\ldots,m-1)}$ via elements of Δ.

It follows from Lemma 2.3 that $k(\pi) = 1 + k(\sigma_{A'}\rho_{(1,2,\ldots,m-1)})$. Hence, the resulting factorization of π via elements of Δ has minimum length.

Now suppose that $m \notin A$. Then $\alpha_1 = m - 1$ and $A' = A \oplus \{1\}$. We have

$$\pi = \sigma_{A'}\sigma_{\{1\}}\rho_{(1,2,\ldots,m-1)}\rho_{(m-1,m)} = \left(\sigma_{A'}\rho_{(1,2,\ldots,m-1)}\right)\left(\sigma_{\{m-1\}}\rho_{(m-1,m)}\right)$$

and $\sigma_{\{m-1\}}\rho_{(m-1,m)} \in \Delta$. Again, $A' \subseteq \{1, 2, \ldots m-1\}$, so by our induction hypothesis, Algorithm $\mathcal{A}(A', \theta', m-1)$ returns a minimum-length factorization of $\sigma_{A'}\rho_{(1,2,\ldots,m-1)}$ via elements of Δ, and, as before, the resulting factorization of π via elements of Δ has minimum length. □

We conclude with an analysis of the time complexity of our routing method. Let c denote the time it takes to compare two integers. Then the time it takes to decide whether an integer in $[m] = \{1, 2, \ldots, m\}$ belongs to a given subset A of $[m]$ is at most $c \cdot m$, and the time it takes to compute $A\Delta\{i\}$, where $i \in [m]$, is also at most $c \cdot m$. Then the total time needed to route $\pi = \sigma_A\rho_\theta \in \mathrm{Aut}(Q_n)$ is at most $2cn^2$, so it is quadratic in n. To see this, note that Algorithm $\mathcal{A}(A, \theta, m)$ calls itself $m - 2$ times. Each time, it computes α_1 and A'. Each requires at most cm time units. Thus the total time required for $\mathcal{A}(A, \theta, m)$ is at most $2cm^2$. Now the factorization of θ into disjoint cycles yields factors $\pi_i = \sigma_{A_i}\rho_{\theta_i}$, which are routed sequentially by $\mathcal{A}(A_i, \theta_i, m_i)$, where $\sum m_i =\mid \text{support}\,(\theta) \mid \leq n$. Hence, the total time needed to route π is at most $2c\sum m_i^2 \leq 2cn^2$.

REFERENCES

[C] W. Y. C. Chen, *Induced cycle structures of the hyperoctahedral group*, SIAM J. Discrete Math., 6 (1993), pp. 353–362.

[CS] W. Y. C. Chen and R. Stanley, *Derangements on the n-cube*, Discrete Math., 115 (1993), pp. 65–75.

[La] S. Latifi, *An effective approach to fast data permutations in hypercube multiprocessors*, Congr. Numer., 84 (1991), pp. 119–127.

[Le] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays · Trees · Hypercubes*, Morgan Kaufmann, San Mateo, CA, 1992.

[LY] Z. Liu and J.-H. You, *Conflict-free routing for BPC-permutations on synchronous hypercubes,* Parallel Comput., 19 (1993), pp. 323–342.

[NS1] D. Nassimi and S. Sahni, *A self-routing Beneš network and parallel permutation algorithms,* IEEE Trans. Comput., C-30, (1981), pp. 332–340.

[NS2] D. Nassimi and S. Sahni, *Optimal BPC permutations on a cube connected SIMD computer,* IEEE Trans. Comput., C-31, (1982), pp. 338–341.

[R] M. Ramras, *Routing permutations on a graph,* Networks, 23 (1993), pp. 391–398.

SIAM J. DISCRETE MATH.
Vol. 10, No. 2, pp. 209–222, May 1997

FINDING EVEN CYCLES EVEN FASTER[*]

RAPHAEL YUSTER[†] AND URI ZWICK[†]

Abstract. We describe efficient algorithms for finding even cycles in undirected graphs. Our main results are the following: (i) For every $k \geq 2$, there is an $O(V^2)$ time algorithm that decides whether an undirected graph $G = (V, E)$ contains a simple cycle of length $2k$, and finds one if it does. (ii) There is an $O(V^2)$ time algorithm that finds a shortest even cycle in an undirected graph $G = (V, E)$.

Key words. graph algorithms, cycles

AMS subject classifications. 05C85, 05C38, 68R10

PII. S0895480194274133

1. Introduction.

Throughout this work, the term *cycle* refers to a simple closed walk and the term *path* refers to a simple nonclosed walk. An even (odd) cycle is a cycle whose length is even (odd). An even (odd) path is a path whose length is even (odd).

The problems of finding cycles of a given length and of finding a shortest even and a shortest odd cycle in undirected and directed graphs are among the most basic and natural algorithmic graph problems. These problems have been considered by many researchers; see [10] for a survey.

In this work we consider (almost exclusively) the undirected versions of these problems. The directed versions of some of them are believed to be much harder. The problem "does a given directed graph $G = (V, E)$ contain a directed cycle of an even length?", for example, is not known to be in P, nor is it known to be NP-complete (see [9]). Though we do not shed any new light on the directed versions of the problems, we obtain surprisingly fast algorithms for some of the undirected versions.

Monien [7] presented an $O(VE)$ algorithm for finding all pairs of vertices that are connected by paths of length $k-1$, where $k \geq 2$ is a fixed integer. (Note that if k is part of the input, the problem is NP-hard.) A simple consequence of his algorithm is an $O(VE)$ algorithm for finding a cycle of length k, if one exists. In [1], an $O(M(V) \log V)$ algorithm is obtained for the same problem, where $M(n) = O(n^{2.376})$ is the complexity of Boolean matrix multiplication. This algorithm is more efficient when G is dense. Both algorithms work on directed as well as undirected graphs. In this work we show that if k is even and if the graph is undirected, then both these bounds can be improved. We obtain an $O(V^2)$ algorithm for finding cycles of a given even length in undirected graphs. An $O(V^2)$ algorithm for finding quadrilaterals (cycles of length four) is part of the folklore (cf. [8]) but all other cases are new. To obtain this $O(V^2)$ algorithm we utilize a combinatorial theorem of Bondy and Simonovits [4] that states,

[*] Received by the editors August 10, 1994; accepted for publication (in revised form) April 9, 1996. A preliminary version of this paper appeared in the Proceedings of the 21st International Colloquium on Automata, Languages and Programming, Jerusalem, Israel, 1994, pp. 532–543. This research was supported in part by the Basic Research Foundation administrated by the Israel Academy of Sciences and Humanities.

http://www.siam.org/journals/sidma/10-2/27413.html

[†] Department of Computer Science, School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel (raphy@math.tau.ac.il, zwick@math.tau.ac.il).

roughly, that dense enough undirected graphs contain many even cycles. We also prove a constructive version of their theorem.

The $O(V^2)$ algorithm for finding cycles with a given even length leads to the following strange state of affairs: deciding whether a given undirected graph contains a cycle of length, say, 100, is *asymptotically* faster than deciding, using any known algorithm, whether this graph contains a triangle (a cycle of length 3)! The term "asymptotically" above should be stressed, because our $O(V^2)$ bound, as well as Monien's $O(VE)$ bound, hides huge multiplicative factors that depend exponentially on k. This exponential dependence on k is probably unavoidable since the problem is NP-hard if k is part of the input.

A shortest cycle in a directed or undirected graph $G = (V, E)$ can be easily found in $O(VE)$ time by conducting a BFS (breadth first search) from each vertex. Itai and Rodeh [5] show that a shortest cycle can also be found in $O(M(V))$ time in the undirected case, and in $O(M(V) \log V)$ time in the directed case. They also notice that by halting the BFS conducted from each vertex in the $O(VE)$ algorithm when the first nontree edge is found (this implies an $O(V)$ running time for each BFS), an *almost* shortest cycle (a cycle whose length exceeds the length of a shortest cycle by at most 1) in an undirected graph can be found in $O(V^2)$ time.

Monien [6] described a sophisticated $O(V^2\alpha(V))$ algorithm for finding shortest even length cycles (SELCs) in undirected graphs, where $\alpha(n) = \alpha(n, n)$ is the functional inverse of Ackermann's function. His algorithm uses the fast union-find data structure. We describe an $O(V^2)$ algorithm for finding SELCs. Our algorithm is somewhat simpler, and it does not use any sophisticated data structure. At the heart of our algorithm lies a combinatorial lemma which is of interest in its own right. The lemma states that if C is a shortest even cycle in a graph, then there exists a vertex v on C from which the paths, on the cycle, to all the other vertices on the cycle are almost the shortest possible. In fact, each of these paths is of length at most 1 greater than the distance between the endpoints of the path.

We also describe a simple $O(M(V) \log V)$ algorithm for finding a shortest odd length cycle (SOLC) in an undirected graph $G = (V, E)$ and a simple $O(VE)$ algorithm for finding a SOLC in a directed or undirected graph $G = (V, E)$. Monien [7] described an $O(VE)$ algorithm for the undirected case.

This paper is organized as follows. In section 2 we present the algorithm for finding fixed-length even cycles in undirected graphs. In section 3 we investigate the combinatorial structure of SELCs. In section 4 we describe the algorithm for finding a SELC and prove its correctness. In section 5 we describe the simple algorithms for finding SOLCs in directed and undirected graphs. We end, in section 6, with some concluding remarks.

2. Finding even cycles of a given length. Throughout this section we use C_l to denote a cycle of length l. The main result of this section is the following theorem.

THEOREM 2.1. *For every $k \geq 2$, there is an $O((2k)! \cdot V^2)$ time algorithm that decides whether an undirected graph $G = (V, E)$ contains a C_{2k} and finds one if it does.*

We also obtain the following result, which is an algorithmic version of a result by Bondy and Simonovits [4].

THEOREM 2.2. *Let $l \geq 2$ be an integer and let $G = (V, E)$ be an undirected graph with $|E| \geq 100l \cdot |V|^{1+1/l}$. Then G contains a C_{2k} for every $k \in [l, l \cdot |V|^{1/l}]$. Furthermore, such a C_{2k} can be found in $O(k \cdot V^2)$ time. In particular, a cycle of length* exactly *$\lfloor l \cdot |V|^{1/l} \rfloor$ can be found in $O(V^{2+1/l})$ time.*

It is interesting to comment on the relationship between these two theorems. In *any* undirected graph $G = (V, E)$ and any $k \geq 2$, we can find a C_{2k}, if one exists, in $O((2k)! \cdot V^2)$ time. This running time is $O(V^2)$ for every *fixed* $k \geq 2$. The running time is exponential, however, if k is part of the input. If the graph $G = (V, E)$ is *dense* enough, i.e., if it contains $\Omega(V^{1+1/k})$ edges, then it does contain a C_{2k}, and such a C_{2k} can be found in $O(k \cdot V^2)$. Note that this is now polynomial in both V and k. In dense enough graphs, we can therefore find extremely long cycles efficiently. In a graph containing $\Omega(V^{3/2})$ edges, for example, we can find, in $O(V^{2.5})$ time, a cycle of length $\Theta(V^{1/2})$. This should be compared with the fact that the problem of deciding whether an undirected graph $G = (V, E)$ contains a cycle (or an even cycle) of length $\Omega(V^{1/2})$ is NP-hard.

The first ingredient used in the proofs of Theorems 2.1 and 2.2 is a combinatorial lemma of Bondy and Simonovits [4] (see also [3]). Their proof of the lemma is nonconstructive. By slightly altering their arguments we obtain a constructive version of their lemma which is required in the proof of Theorem 2.2. Before stating the lemma we need the following definition.

DEFINITION 2.3. *A coloring of the vertices of an undirected graph $G = (V, E)$ is said to be t-*periodic *if the endpoints of every path of length t are colored by the same color.*

Note that the coloring in the definition above is not required to be proper; i.e., adjacent vertices may be colored by the same color. We can now state the lemma of Bondy and Simonovits [4] and present an algorithmic proof of it.

LEMMA 2.4. *Let t be a positive integer, and let $G = (V, E)$ be a connected undirected graph with $|E| \geq 2t \cdot |V|$. Then any coloring of the vertices of G that uses at least three distinct colors is not t-periodic. Furthermore, if G is nonbipartite, then any coloring of the vertices of G that uses at least two distinct colors is not t-periodic. In both the bipartite and nonbipartite cases, two vertices of distinct colors and a path of length t connecting them can be found in $O(E)$ time.*

Proof. We begin by showing that G contains two adjacent vertices joined by two vertex-disjoint paths, each of length at least t, and that such a subgraph, called a Θ-*graph*, can be found in $O(E)$ time. It is easy to see that G contains a subgraph G' whose minimal degree is at least $2t$. Such a subgraph can be easily found in $O(E)$ time by sequentially removing from G vertices whose degrees are less than $2t$. Let $v_1, v_2, \ldots, v_m$ be a maximal path in G', i.e., a path that cannot be further extended. Such a path can be greedily constructed in $O(E)$ time. The vertex v_1 is then adjacent to at least $2t$ vertices $v_{i_1}, v_{i_2}, \ldots, v_{i_{2t}}$ on this path, where $2 = i_1 < i_2 < \cdots < i_{2t}$. The path $v_1, v_2, \ldots, v_{i_{2t}}$, along with the edges (v_1, v_{i_t}) and $(v_1, v_{i_{2t}})$, forms the desired Θ-graph.

The Θ-graph found contains three distinct cycles L_1, L_2, L_3 of lengths l_1, l_2, l_3, respectively, such that $l_1, l_2, l_3 > t$ and $l_1 + l_2 - l_3 = 2$. Every vertex v of the Θ-graph has at most four distinct paths of length t in the Θ-graph that start at v. We can easily check in $O(V)$ time whether, for each v, the endpoints of these paths are colored by the same color of v. If this is not the case, then we are done, since we have found two vertices colored by distinct colors and a path of length t connecting them.

Assume therefore that the Θ-graph is t-periodic. It is easy to see that if one of the cycles L_1, L_2, or L_3 is t^*-periodic, then the other cycles, and therefore the Θ-graph, must also be t^*-periodic. Let t^* be the smallest integer for which the Θ-graph is t^*-periodic. It follows that t^* is also the smallest period of the cycles L_1, L_2, or L_3, and as a consequence $t^* | l_1, l_2, l_3$. As $l_1 + l_2 - l_3 = 2$, we get that $t^* | 2$. Thus $t^* = 1$ or

$t^* = 2$, and the number of colors used to color the Θ-graph is at most 2.

Every vertex of G is connected by a simple path whose length is a multiple of t to a vertex of, say, L_1. If a vertex $v \in V$ is colored by color not appearing on L_1, then a simple path t whose endpoints are colored by distinct colors can be easily found in $O(V)$ time.

Finally, note that a 2-periodic coloring of a graph $G = (V, E)$ that uses two colors is necessarily a proper coloring. Any graph $G = (V, E)$ that has a 2-periodic coloring that uses only two colors must therefore be bipartite. □

The second ingredient used in the proof of Theorem 2.1 is the following result of Monien [7].

LEMMA 2.5. *There is an $O(k! \cdot E)$ time algorithm that, given a (directed or undirected) graph $G = (V, E)$, an integer $k \geq 2$, and a vertex $s \in V$, finds all vertices $v \in V$ connected to s by paths of length k, and exhibits one such path for each such v.*

The following are immediate consequences of Lemma 2.5.

COROLLARY 2.6. *Let $G = (V, E)$ be a (directed or undirected) graph and let $k \geq 3$ be an integer. There is an $O((k-1)! \cdot E)$ time algorithm that, given a vertex $s \in V$, decides whether there is a C_k that passes through s, and finds such a C_k if one exists.*

Proof. Find all the vertices connected to s by paths of length $k-1$ and check whether one of them is also connected to s by an edge. □

COROLLARY 2.7. *Let $G = (V, E)$ be a (directed or undirected) graph and let $k \geq 1$ be an integer. There is an $O((k+1)! \cdot E)$ time algorithm that, given two disjoint subsets A and B of vertices, determines whether there is a path of length k connecting a vertex from A and a vertex from B, and finds such a path if one exists.*

Proof. Assume that the graph is directed. (If not, replace each undirected edge by two anti-parallel directed edges.) Add a new vertex s and connect it to all the vertices of A. Now find all the vertices to which there are directed paths of length $k+1$ from s. □

Alon, Yuster, and Zwick [1] have recently described a $2^{O(k)} \cdot E \log V$ time algorithm for performing the task of Lemma 2.5 and $2^{O(k)} \cdot E$ *expected* time algorithms for the tasks of Corollaries 2.6 and 2.7. The dependency on k in the above complexity bounds can be improved, therefore, from $k!$ to $2^{O(k)}$ if randomization or an extra $\log V$ factor is allowed.

We are now ready to prove Theorem 2.1. We prove, in fact, the following slightly stronger result.

THEOREM 2.8. *Let $k > 1$ be a fixed integer. There is an $O((2k)! \cdot V)$ time algorithm that, given an undirected graph $G = (V, E)$ and a vertex $s \in V$, either verifies that s is not contained in any C_{2k} or finds a C_{2k} in G (not necessarily passing through s).*

Proof. The algorithm starts a BFS from the vertex s. For $v \in V$, let $d(v)$ be the distance between s and v in G. Let $L_i = \{v \mid d(v) = i\}$ be the set of vertices at level i of the BFS tree. At stage i the algorithm scans the adjacency lists of the vertices of L_i. During this scan, the algorithm keeps a count of the number of edges found so far inside L_i (an edge is inside L_i if both its endpoints are in L_i). Similarly, it keeps a count of the number of edges found so far between L_i and L_{i+1}. We use L'_{i+1} to denote the set of vertices of L_{i+1} that were already discovered by the search. The search is halted when one of the following conditions holds:

1. Stage $k-1$ has completed or the BFS has ended.
2. At least $4k \cdot |L_i|$ edges were found inside L_i.
3. At least $4k \cdot (|L_i| + |L'_{i+1}|)$ edges were found between L_i and L'_{i+1}.

Since the L_i's are disjoint, the total number of edges scanned before the search is halted is at most $12k\cdot|V|$. Hence, the search takes only $O(k\cdot V)$ time.

As in any BFS, when a vertex $v \in L_i$ is discovered, we let $\pi(v)$ be the vertex in L_{i-1} that discovered it. In such a way a shortest path tree rooted at s and consisting of all discovered vertices is maintained.

The algorithm continues in one of three possible ways, according to the condition that caused the BFS to halt.

Case 1. The BFS is halted because stage $k-1$ has completed.

In this case, the first $k+1$ levels $L_0, L_1, \ldots, L_k$ have all been discovered, and the subgraph G' induced by them (but not containing the edges inside L_k) contains at most $12k\cdot|V|$ edges. If s is on a C_{2k} then this C_{2k} is completely contained in G'. By Corollary 2.6, we can check whether such a cycle exists in $O((2k)!\cdot V)$ time.

Case 2. The BFS is halted because $4k\cdot|L_i|$ edges were found inside L_i for some $i < k$.

Stage i of the search is then left incomplete, but all the first $i{+}1$ levels $L_0, L_i, \ldots, L_i$ are already completely discovered. Consider the subgraph of G induced by L_i. This subgraph contains at least one connected component whose vertex set is $U \subseteq L_i$ and whose number of edges is at least $4k\cdot|U|$. Denote the subgraph composed of this connected component by H. Such a subgraph is easily found in $O(k\cdot L_i) = O(k\cdot V)$ time. Note that $|U| > 1$.

Assume at first that H is nonbipartite. (This is easily verified in $O(k\cdot V)$ time, since H contains $O(k\cdot V)$ edges.) Let c be the lowest common ancestor in the BFS tree of all the vertices in U. The vertex c is easily found in $O(k\cdot U) = O(k\cdot V)$ time in the following way: let $U_i = U$ and let $U_j = \{\pi(v) \mid v \in U_{j+1}\}$ for $j = i-1, i-2, \ldots$, until a U_h with $|U_h| = 1$ is reached. Then $U_h = \{c\}$. As $|U| > 1$, c must have at least two children in U_{h+1}. Let d be one of them. Let $X_1 \subset U$ be the descendents of d in U, and let $X_2 = U - X_1$. Color the vertices of X_1 red and the vertices of X_2 blue. By Lemma 2.4, the subgraph H cannot be $2(k-i+h)$-periodic (since it is nonbipartite, connected and colored by two distinct colors). There must therefore be a path of length $2(k-i+h)$ in H between a red vertex and a blue vertex. As explained in the proof of Lemma 2.4, we can find such a path p in $O(k\cdot U) \leq O(k\cdot V)$ time. (Such a path can also be found using Corollary 2.7, but the running time would be $O((2k)!\cdot V)$.) The path p can now be extended to a cycle of length $2k$ by adding the disjoint paths of the BFS tree from c to the two endpoints of p, each having length $i-h$. Note that this cycle contains s only if $c = s$.

Very similar actions are taken if H is bipartite. Let A and B be the vertex classes of H (i.e., A and B are disjoint, $A \cup B = U$, and all the edges in H are between A and B). Assume, without loss of generality, that $|A| > 1$. Let c be the lowest common ancestor in the BFS tree of all the vertices of A. The vertex c is found using the method described above. Assume again that c is in level h. As $|A| > 1$, c must have at least two children in level $h+1$. Let d be one of them. Let $X_1 \subset A$ be the descendents of d in A and let $X_2 = A - X_1$. Color the vertices of X_1 red, the vertices of X_2 blue, and the vertices of B green. By Lemma 2.4, the subgraph H cannot be $2(k-i+h)$-periodic, since it is connected and colored by three distinct colors. There must therefore be a path p of length $2(k-i+h)$ in H between two differently colored vertices. This path must be between a red vertex and a blue vertex, because any path of an even length that starts at a green vertex also ends at a green vertex. This path can again be found in $O(k\cdot V)$ time, and it can again be extended to a cycle of length $2k$.

Case 3. The BFS was halted because $4k\cdot(|L_i| + |L'_{i+1}|)$ edges were found between L_i and L'_{i+1}.

Find a connected subgraph H of the subgraph of G induced by L_i and L'_{i+1} with a vertex set U and with at least $4k{\cdot}|U|$ edges. Such a subgraph is easily found in $O(k{\cdot}V)$ time. Note that H is bipartite with vertex classes $A = U \cap L_i$ and $B = U \cap L'_{i+1}$. The algorithm can now proceed as in the previous case.

In any one of these three cases, the running time is $O((2k)! \cdot V)$. In fact, the running time of the algorithm in the second and third cases is only $O(k{\cdot}V)$. The only case in which a C_{2k} is not found by the algorithm is when no C_{2k} passes through s. This completes the proof of the theorem. □

Theorem 2.1 follows immediately from the above theorem. All we have to do is apply the algorithm described above from each vertex. We now turn to the proof of Theorem 2.2. The proof of Bondy and Simonovits actually shows that if $|E| \geq 100l \cdot |V|^{1+1/l}$ and $k \in [l, l \cdot |V|^{1/l}]$ then there *exists* a vertex $s \in V$ for which the algorithm of Theorem 2.8 stops *before* completing stage $k-1$. This immediately leads to the desired $O(k \cdot V^2)$ time algorithm. Theorem 2.8 has another interesting consequence.

THEOREM 2.9. *A C_{2k} in an undirected graph $G = (V, E)$ with $|E| \geq 101k \cdot |V|^{1+1/k}$ can be found in $O((2k)! {\cdot} V)$ expected time.*

Proof. Any graph on $|V|$ vertices and at least $100k{\cdot}|V|^{1+1/k}$ edges contains a C_{2k}. It follows immediately that the number of edges in a graph $G = (V, E)$ which are not contained in any C_{2k} is at most $100k \cdot |V|^{1+1/k}$. If $G = (V, E)$ contains at least $101k{\cdot}|V|^{1+1/k}$ edges, then a randomly chosen edge has a probability of at least $1/101$ of belonging to a C_{2k}. The randomized algorithm simply chooses a random edge and applies the algorithm of Theorem 2.8 to one of its endpoints. The expected number of applications before a desired C_{2k} is found is $O(1)$ and the expected running time is $O((2k)!{\cdot}V)$. □

3. The structure of shortest even length cycles. Let G be an undirected graph and let C be a SELC (shortest even length cycle) of it. Suppose the vertices on the cycle are consecutively labeled $v_0, v_1, \ldots, v_{2k-1}$. We denote by $d(x, y)$ the distance between two vertices x and y in G. Clearly, $d(v_0, v_i), d(v_0, v_{2k-i}) \leq i$ for every $1 \leq i \leq k$. If $d(v_0, v_i) = i$ and $d(v_0, v_{2k-i}) = i$, for every $1 \leq i \leq k$, then C, or some other SELC, can be easily found using a BFS from v_0. However, the paths on C between v_0 and v_i and between v_0 and v_{2k-i} are not necessarily shortest paths in G. As an example, consider K_4, the complete graph on four vertices. All the even cycles in K_4 are of length 4, but the distance between any two vertices is 1. It may be, therefore, that $d(v_0, v_i) < i$ or $d(v_0, d_{2k-i}) < i$ for some $1 \leq i \leq k$. It is not immediately clear how to find C, or any other SELC, in such a case.

The main result of this section is the following lemma that states that on every SELC C there is a vertex v_0 from which the paths, on C, to all the other vertices on C are *almost* shortest paths. An almost shortest path is a path whose length exceeds the length of a corresponding shortest path by at most one. Specifically, we have the following lemma.

LEMMA 3.1. *Let C be a SELC of G. Then the vertices on C can be consecutively labeled $v_0, v_1, \ldots, v_{2k-1}$, so that $i - 1 \leq d(v_0, v_i) \leq i$ and $i - 1 \leq d(v_0, v_{2k-i}) \leq i$ for every $1 \leq i \leq k$.*

This lemma is the cornerstone of the $O(V^2)$ algorithm for finding SELCs presented in the next section. We think, also, that this lemma is of interest in its own right. Before presenting a proof of Lemma 3.1, we present the following simple but useful lemma.

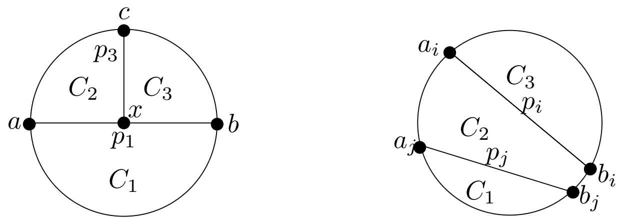


FIG. 3.1. *One of the cycles C_1, C_2, and C_3 is even.*

LEMMA 3.2. *If p_1 and p_2 are two distinct (but not necessarily disjoint) shortest paths in G between x and y, then C contains an even cycle whose length is at most $2d(x, y)$.*

Proof. Let $p_1 = (a_0, a_1, \ldots, a_{k-1}, a_k)$ and $p_2 = (b_0, b_1, \ldots, b_{k-1}, b_k)$ be two distinct shortest paths between $x = a_0 = b_0$ and $y = a_k = b_k$. Let $i \geq 0$ be the minimal index such that $a_i = b_i$ but $a_{i+1} \neq b_{i+1}$. Let j be the minimal index $j > i$ such that $a_j = b_j$. Then $(a_i, \ldots, a_j)$ and $(b_i, \ldots, b_j)$ are two shortest paths connecting a_i and a_j whose inner vertices are disjoint. We thus obtain a cycle of length $2(j - i) \leq 2k$. □

Proof of Lemma 3.1. Let H be a minimal subgraph of G (with respect to containment) containing C such that $d_H(x, y) = d(x, y)$ for every $x, y \in C$ ($d_H(x, y)$ denotes the distance between x and y in H). Let $e(H)$ be the edge set of H. If $H = C$, we are done. Otherwise, let $P = H \setminus e(C)$.

A path p whose two endpoints a and b are on C, but none of whose inner vertices are on C, that satisfies $|p| = d(a, b) < d_C(a, b)$, where $|p|$ is the length of p, is called an $a \sim b$ *shortcut*. Our first claim is that P is a collection of vertex disjoint shortcuts.

To see this, let P' be a connected component of P. The minimality of H implies that any edge of P' is contained in some shortcut. The component P' must therefore contain an $a \sim b$ shortcut p_1 for some $a, b \in C$. If P' is composed solely of this shortcut, we are done. Otherwise, let x be a vertex on p_1 incident to an edge e of P' which is not on p_1 (x may be a or b). The edge e is contained in some shortcut p_2. The shortcuts p_1 and p_2 meet only at x. If they had met in some other vertex y, a shorter even cycle would have existed, by Lemma 3.2, in the graph. Let p_3 be a portion of p_2 that connects x with some vertex c on C. Consider now the cycles C_1, C_2, and C_3 shown on the left of Fig. 3.1. Each of these cycles is of size less than $2k$. For C_1, this follows from the fact that $|p_1| < d_C(a, b)$. We show that $|C_2| < 2k$ as follows: let C_4 be the cycle comprised of p_1 with the part of C between a and b containing c. Since $|p_1| < d_C(a, b)$ we have that $|C_4| < 2k$. As p_3 is a shortest path between c and x, we get that $|C_2| \leq |C_4| < 2k$. The fact that $|C_3| < 2k$ follows from similar arguments. The sum of the lengths of these cycles is $2k + 2|p_1| + 2|p_3|$, which is even, and thus one of them must be even, contradicting the minimality of C. This contradiction shows that P' must simply be a shortcut.

We have shown that $P = \{p_1, \ldots, p_s\}$ is a set of disjoint shortcuts where $s \leq k$ (as each shortcut contains two vertices of C). We now claim that every two distinct shortcuts p_i and p_j must *cross* one another; i.e., each of the two paths on C between the endpoints of p_i contains an endpoint of p_j. See the left side of Fig. 3.2.

Assume, for contradiction, that the shortcuts p_i and p_j do not cross one another, as shown on the right side of Fig. 3.1. The length of each of the cycles C_1, C_2, and

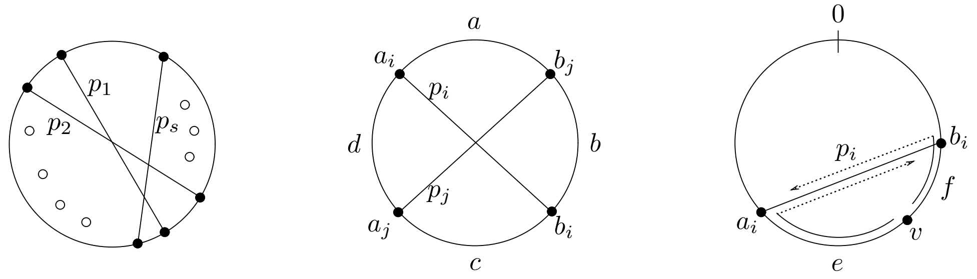


FIG. 3.2. *The shortcuts of P.*

C_3 there is less than $2k$. The sum of their lengths is $2k+2|p_i|+2|p_j|$, so one of them must be even, contradicting the minimality of C.

We have shown that the mutual position of p_i and p_j must be as shown in the middle of Fig. 3.2. Let a, b, c, d denote the four segments of C determined by the endpoints of these shortcuts. The minimality of C implies that $|p_i|+|a|+|b|$, $|p_i|+|c|+|d|$, $|p_j|+|b|+|c|$, and $|p_j|+|a|+|d|$ are all odd, since these are lengths of cycles smaller than $2k$. This, in turn, implies that $|p_i|+|p_j|+|a|+|c|$ and $|p_i|+|p_j|+|b|+|d|$ are even. These two expressions are the lengths of the "twisted" cycles a, p_i, c, p_j and b, p_i, d, p_j. As a consequence, these lengths are at least $2k$. In particular,

$$|p_i|+|p_j|+|a|+|c| \geq 2k = |a|+|b|+|c|+|d|. \tag{3.1}$$

Our third claim is that for any two vertices x, y on the cycle C there exists a shortest path between them that uses at most one shortcut. Consider a shortest path between x and y that contains at least two shortcuts. Let p_i and p_j be two consecutive shortcuts appearing on the path. Let c be the portion of the path that connects them, as shown again in the middle of Fig. 3.2. From (3.1), we get that $|p_i|+|c|+|p_j| \geq |b|+|c|+|d|$. We can therefore replace the portion p_i, c, p_j of the path by the path b, c, d without increasing the length. Continuing in this way, we can obtain a shortest path that uses at most one shortcut. In view of Lemma 3.2, a shortest path that uses more than one shortcut must connect two antipodal vertices, i.e., two vertices whose distance is k, on the cycle.

It is convenient at this point to fix a consecutive numbering $0, 1, \ldots, 2k-1$ of the vertices of the cycle C and to identify the vertices of C with their numbers. We let a_i and b_i, where $a_i < b_i$, be the two endpoints of the shortcut p_i. To every shortcut p_i we attach the following interval:

$$C_i = \left[\frac{a_i+b_i-|p_i|-1}{2}, \frac{a_i+b_i+|p_i|+1}{2}\right].$$

Both endpoints of this interval are integral since $b_i - a_i$ and $|p_i|$ have different parities; otherwise, C would not have been a SELC. As $|p_i| < b_i - a_i$, we get that $C_i \subseteq [a_i, b_i]$. The interval C_i corresponds to a subset of the vertices of C.

We claim that if $v \in C_i$, then for every vertex u on C, if a shortest path between v and u uses the shortcut p_i as its only shortcut, then the path between v and u along the cycle C is an almost shortest path between v and u. Recall that an almost shortest path between v and u is a path whose length is at most $d(v,u)+1$. To see this, suppose that $v \in C_i$ and that some shortest path from v to u uses p_i as its only shortcut. This shortest path must either go along portion e of the cycle C from v to a_i, then use p_i, and then go again along C, or go along portion f of the cycle C from v to b_i, then use p_i, and then go again along C. Both cases are shown on the right of Fig. 3.2. The definition of C_i implies, however, that

$$|e| = v - a_i \leq b_i - v + |p_i| + 1 = |f| + |p_i| + 1,$$

$$|f| = b_i - v \leq v - a_i + |p_i| + 1 = |e| + |p_i| + 1.$$

The path e, p_i can therefore be replaced by the path f, and the path f, p_i can be replaced by the path e while increasing the length by at most one, as required.

Our final task is to show that the intersection $\cap_{i=1}^{s} C_i$ of all these intervals is not empty. If $v_0 \in \cap_{i=1}^{s} C_i$, then the paths along C from v_0 to all other vertices on the cycle are almost shortest paths, as required. As all the C_i's are intervals, it is enough to show that any two of them intersect. Let C_i and C_j be two such intervals where $a_i < a_j$. The fact that p_i and p_j cross one another implies that $a_i < a_j < b_i < b_j$. To show that C_i and C_j intersect, we show that

$$\frac{a_j + b_j - |p_j| - 1}{2} \leq \frac{a_i + b_i + |p_i| + 1}{2}$$

and

$$\frac{a_i + b_i - |p_i| - 1}{2} \leq \frac{a_j + b_j + |p_j| + 1}{2}.$$

The first inequality is equivalent to $|p_i| + |p_j| + (2k - b_j + a_i) + (b_i - a_j) + 2 \geq 2k$. But $|p_i| + |p_j| + (2k - b_j + a_i) + (b_i - a_j)$ is the length of the twisted cycle a, p_i, c, p_j shown in the middle of Fig. 3.2. The length of this cycle is at least $2k$ by (3.1), proving the first inequality. The second inequality follows immediately from the fact that $a_i < a_j < b_i < b_j$. We have shown therefore that the intervals C_i and C_j, and therefore all the intervals, do intersect.

Any vertex $v_0 \in \cap_{i=1}^{s} C_i$ can play the role of v_0 in the statement of the lemma. This completes the proof of the lemma. $\square$

If a SELC C is edge disjoint from all other SELCs, then a sharp inequality holds in (3.1). This can be used to show that all the intervals $C'_i = [\frac{a_i+b_i-|p_i|+1}{2}, \frac{a_i+b_i+|p_i|-1}{2}]$ intersect. Every vertex v_0 in this intersection has the property that the shortest paths along the cycle C from v_0 to all other vertices are in fact shortest paths. The intersection $\cap_{i=0}^{s} C'_i$ may, however, be empty if C is not edge disjoint from all other SELCs.

Let $v_0, \ldots, v_{2k-1}$ be an ordering of C that satisfies the conditions of Lemma 3.1. In view of Lemma 3.2, it is impossible that $d(v_0, v_{k-1}) = d(v_0, v_{k+1}) = k-2$, because this yields two shortest paths of lengths $k-1$ from v_0 to v_k. We may therefore assume, without loss of generality, that $d(v_0, v_{k-1}) = k-1$. We call v_0 a *root* of C. If $d(v_0, v_k) = k$ we call C a cycle of *type one* with respect to (w.r.t.) v_0, and if $d(v_0, v_k) = k-1$ we call C a cycle of *type two* w.r.t. v_0. Every cycle of type two w.r.t. v_0 has a unique $0 < j < k$ such that $d(v_0, v_{2k-j}) = j$, and $d(v_0, v_{2k-j-1}) = j$. We call j the *index* of C w.r.t. the root v_0.

Finally, we note that if $v_0, \ldots, v_{2k-1}$ is an ordering of C that satisfies the conditions of Lemma 3.1, then $v_k, \ldots, v_{2k-1}, v_0, \ldots, v_{k-1}$ is also such an ordering; i.e., v_k can play the role of v_0.

4. An $O(V^2)$ algorithm for finding a shortest even cycle. Relying on Lemma 3.1, we obtain an $O(V^2)$ algorithm for finding a SELC in an undirected graph $G = (V, E)$. The algorithm starts a BFS from every vertex, but stops it as soon as an even cycle is detected. This ensures that the time spent in each such BFS is at most $O(V)$. We show that the shortest even cycle found in this way by the algorithm is indeed a SELC of the graph.

The BFS performed is an augmented version of the standard BFS capable of detecting even cycles. Let a be a vertex from which such an augmented BFS is performed (a is called the root of the BFS). For every vertex v, we record a set of four variables. The first two variables are standard; the other two are used to detect even cycles. These four variables are:

$d(v)$**:** the distance of v from a, i.e., the level of v in the BFS tree; $d(v) = \infty$ if v has not yet been discovered.

$\pi(v)$**:** the parent of v in the BFS tree; $\pi(v) = 0$ if $v = a$ or if v has not yet been discovered. If $\pi(v) \neq 0$ then $d(v) = d(\pi(v)) + 1$.

$\theta(v)$**:** the *match* of v; if $\theta(v) \neq 0$ then $\theta(v)$ is a vertex in the same level of v such that $(v, \theta(v)) \in E$. A vertex v is said to be *matched* if $\theta(v) \neq 0$. If v is matched then $\theta(v)$ will also be matched and $\theta(\theta(v)) = v$. The set of edges $\{(v, \theta(v)) \mid \theta(v) \neq 0\}$ is therefore a matching.

$\rho(v)$**:** the highest proper ancestor of v in the BFS tree that is matched. If v has no matched proper ancestors, then $\rho(v) = 0$.

We now describe how we process a vertex v that has been popped out of the BFS queue. Before we start scanning v's neighbors, we assume that $\rho(v)$, $d(v)$, and $\pi(v)$ are correctly set (v may or may not be matched at this point depending on whether it is adjacent to a vertex in its level that has been processed before it). The action taken for an edge (v, u) depends on the value of $d(u)$, $\theta(v)$, and $\theta(u)$ in the following way:

1. If $d(u) = d(v) - 1$, do nothing (this edge has been processed before, in its opposite direction).
2. If $d(u) = \infty$, let $d(u) \leftarrow d(v) + 1, \pi(u) \leftarrow v$ and enqueue u to the BFS queue.
3. If $d(u) = d(v) + 1$, halt the BFS since an even cycle was found. Let c be the lowest common ancestor, in the BFS tree, of v and u. Then the $c \sim v$ and $c \sim u$ tree paths and the edge (v, u) form an even cycle of length $2(d(v) + 1 - d(c))$. This cycle is shown in Fig. 4.1.
4. If $d(u) = d(v)$ and $\theta(v) = u$ (which also means that $\theta(u) = v$), do nothing (this edge has been processed before, in its opposite direction).
5. If $d(u) = d(v)$ and $\theta(v) \neq u$, and $\theta(v)$ and $\theta(u)$ are not both zero, halt the BFS since an even cycle was found. Assume, for example, that $\theta(v) = x \neq 0$. Let c be the lowest common ancestor, in the BFS tree, of x and u. The $c \sim x$ and $c \sim u$ tree paths and the edges $(x, v), (v, u)$ form an even cycle of length $2(d(v) + 1 - d(c))$. This cycle is shown in Fig. 4.1.
6. If $d(u) = d(v)$ and $\theta(v) = \theta(u) = 0$, test whether $\rho(v) = \rho(u)$. If they are equal, let $\theta(v) \leftarrow u$, $\theta(u) \leftarrow v$. If they are not equal, halt the BFS since an even cycle is found as follows. Assume, for example, that $\rho(v) = x \neq 0$ and let $y = \theta(x)$. Let c be the lowest common ancestor, in the BFS tree, of y and u. Then the $c \sim y$ tree path followed by the edge (y, x) followed by the $x \sim v$ tree path followed by the edge (v, u) followed by the $u \sim c$ tree path closes an even cycle of length $2(d(v) + 1 - d(c))$. This cycle is shown in Fig. 4.1. Note that this is a cycle (i.e., it is simple) since x is not an ancestor

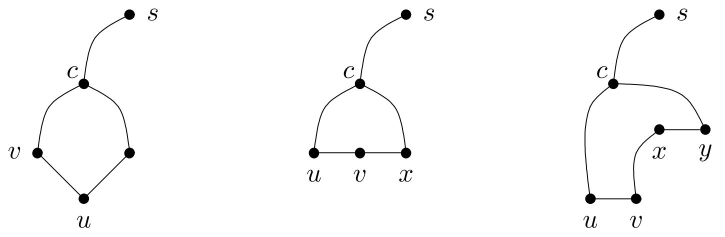


FIG. 4.1. *The even cycles detected by rules* 3*,* 5*, and* 6*.*

of u.

After we finish scanning all the neighbors of v, we rescan them to set $\rho(u)$ for every u that has become a child of v. We put $\rho(u) \leftarrow \rho(v)$ unless $\rho(v) = 0$ and $\theta(v) \neq 0$, in which case we put $\rho(u) \leftarrow v$. This completes the description of the algorithm.

THEOREM 4.1. *The augmented BFS scans no more than* $3|V|/2$ *edges and therefore runs in* $O(V)$ *time. Furthermore, if* C *is a SELC of length* $2k$ *and* v_0 *is a root of it, then an augmented BFS that starts from* v_0 *finds an even cycle of length* $2k$*.*

Proof. When the BFS halts (either because it has completed, or because an even cycle has been found), the only edges scanned are the BFS tree edges, the edges between matched vertices (these edges form a matching), and possibly an edge that closes an even cycle. There are at most $|V| - 1$ tree edges and at most $(|V| - 1)/2$ edges in the matching (the root of the BFS is never matched). The total number of edges scanned is therefore at most $3|V|/2$. The complexity claim is obvious because scanning an edge entails only a constant number of operations.

We now prove the second part of the theorem. Consider an augmented BFS that starts at a root v_0 of a SELC C. Note, according to the above six rules, that if the BFS halts while scanning the neighbors of a vertex v, the even cycle found has a length of at most $2(d(v) + 1)$.

Suppose that C is a SELC of type one w.r.t. v_0 (type-one and type-two SELCs were defined at the end of the previous section). Then v_{k-1} and v_{k+1} are both in level $k-1$ of the BFS. Suppose that v_{k+1} is processed after v_{k-1}. If an even cycle is found before the edge (v_{k+1}, v_k) is scanned, its length must be $2k$ (it cannot be shorter, of course). Otherwise, an even cycle of length $2k$ is found, using rule 3, when the edge (v_{k+1}, v_k) is scanned.

Suppose that C is a SELC of type two, with index $j = k - 1$ w.r.t. v_0. Then v_{k-1}, v_k, v_{k+1} are all in level $k - 1$ of the BFS. If an even cycle of length $2k$ is not found before processing the vertex v_k, such a cycle is found, using rule 5, when v_k is processed since it is adjacent to two vertices in its level.

Finally, suppose that C is a SELC of type two, with $j < k - 1$. Then both v_{k-1} and v_k are in level $k - 1$ of the BFS (and v_{k+1} is in level $k - 2$). We claim that $\rho(v_{k-1}) \neq \rho(v_k)$, and therefore an even cycle is found (using rule 6) when the edge (v_{k-1}, v_k) is scanned, if such a cycle were not found before. First, note that $\theta(v_{2k-j-1}) = v_{2k-j}$. (Both are in level j; there is an edge between them; and we did not halt at level j.) Second, note that $(v_0, v_1, \ldots, v_{k-1})$, $(v_0, v_{2k-1}, \ldots, v_{2k-j})$, and $(v_k, v_{k+1}, \ldots, v_{2k-j-1})$ are shortest paths in G (refer to Fig. 4.2). As these shortest paths connect vertices whose distance is less than k, they must be the unique shortest

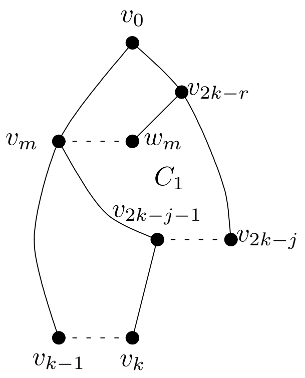


FIG. 4.2. *If* $\rho(v_k) = \rho(v_{k-1}) = v_m$ *then* $|C_1| = 2(j - r + 1) < 2k$.

paths between these vertices (cf. Lemma 3.2). These paths must therefore be tree paths; i.e., they must be contained in the BFS tree. It follows that v_{2k-j-1} is the ancestor of v_k at level j. Therefore, $\rho(v_k) \neq 0$. If $\rho(v_{k-1}) = 0$ we are done. Otherwise $\rho(v_{k-1}) = v_m$ where $1 \leq m < k-1$ since $v_0, v_1, \ldots, v_{k-2}$ are the proper ancestors of v_{k-1} (and v_0 is unmatched). Assume, for contradiction, that $\rho(v_k) = v_m$. Since v_{2k-j-1} is a matched ancestor of v_k, we have $m < j$, and v_m is an ancestor of v_{2k-j-1}. Let $w_m = \theta(v_m)$ be the match of v_m, and let v_{2k-r} be the lowest common ancestor, in the BFS tree, of w_m and v_{2k-j} (v_{2k-r} may be v_0). We obtain the following even cycle (cycle C_1 in Fig. 4.2) in G: $v_m \sim v_{2k-j-1} - v_{2k-j} \sim v_{2k-r} \sim w_m - v_m$, where $v_m \sim v_{2k-j-1}$, $v_{2k-j} \sim v_{2k-r}$, and $v_{2k-r} \sim w_m$ denote the tree paths between these vertices, and $v_{2k-j-1} - v_{2k-j}$, $w_m - v_m$ are the edges matching these vertices. The tree paths $v_m \sim v_{2k-j-1}$ and $v_m \sim v_{k-1}$ may coincide initially; this causes no problems. The cycle C_1 is indeed a cycle; i.e., it is simple, because it is composed of tree paths that cannot intersect one another. The length of C_1 is $2(j-r+1) \leq 2k-2$, contradicting the minimality of C. □

As a corollary of Theorem 4.1, we get that any graph containing more than $3(V-1)/2$ edges contains an even cycle. A simple example shows that this bound is the best possible. Just take any connected graph whose biconnected components are triangles. Furthermore, checking whether a graph contains an even cycle and exhibiting one if it does can be done in $O(V)$ time. Just perform *one* augmented BFS from an arbitrary vertex.

Finally, we point out that the result of this section is *not* implied by the results of section 2. We cannot afford checking, for $k = 1, 2, \ldots$, whether the graph contains a C_{2k} since the length of the smallest even cycle may be large.

5. Finding a shortest odd cycle in undirected and directed graphs.

Shortest odd length cycles (SOLCs) can be found in polynomial time in both directed and undirected graphs. Our objective in this section is to describe very simple, yet efficient, algorithms for both of these problems. Monien [6] obtained a simple $O(VE)$ time algorithm for finding SOLCs in undirected graphs. Using fast Boolean matrix multiplication algorithms we obtain an $O(M(V)\log V)$ algorithm for the same task. This algorithm is more efficient than Monien's algorithm for dense graphs.

THEOREM 5.1. *There is an $O(M(V)\log V)$ time algorithm that finds a shortest odd cycle in an undirected graph $G = (V, E)$.*

Proof. Let A be the adjacency matrix of G. We assume that G is nonbipartite since otherwise it contains no odd cycles. Recall that $A^k(i,i) = 1$ iff there is a closed walk of length k from i to itself (the multiplications used to obtain A^k are assumed to be Boolean). Since any closed walk of an odd length contains an odd cycle, the length of the SOLCs of G is the minimal odd k for which there exists an i such that $A^k(i,i) = 1$. Since G is undirected, $A^t(i,i) = 1$ implies $A^{t+2}(i,i) = 1$. We can therefore look for this minimal k using the following approach. Start computing $A, A^3, A^7, \ldots, A^{2^i-1}, \ldots$ until $A^{2^s-1}(i,i) = 1$ for some i. A binary search along the odd numbers between $2^{s-1} - 1$ and $2^s - 1$ can then be used to find k. The number of Boolean matrix multiplications required is clearly $O(\log V)$. A specific SOLC of length k can be found without increasing the complexity of the algorithm. $\square$

We turn our attention now to finding shortest odd cycles in directed graphs. Unlike in the undirected case, subpaths of SOLCs are not necessarily shortest paths, and therefore a simple BFS from every vertex does not suffice. Let $ed(u,v)$ be the length of a shortest even length directed walk from u to v. Similarly, let $od(u,v)$ be the length of a shortest odd length directed walk from u to v. If no odd (even) length walk exists we set $od(u,v) = \infty$ $(ed(u,v) = \infty)$. Note that the existence of a walk of length $ed(u,v)$ $(od(u,v))$ does not imply the existence of a simple walk of length $ed(u,v)$ $(od(u,v))$.

LEMMA 5.2. *If $C = (v_0, v_1, \ldots, v_{k-1})$ is a SOLC of a directed graph G, then $ed(v_0, v_{2i}) = 2i$ and $od(v_0, v_{2i-1}) = 2i - 1$ for every $1 \le i \le \frac{k-1}{2}$.*

Proof. Any closed walk of an odd length contains an odd cycle. There is an odd length closed walk containing v_0 whose length is $ed(v_0, v_{2i}) + k - 2i$. The minimality of C implies that $ed(v_0, v_{2i}) \ge 2i$. There is, however, a path of length $2i$ between v_0 and v_{2i}, and therefore $ed(v_0, v_{2i}) = 2i$. The second equality in the statement of the lemma follows using similar arguments. $\square$

Given a vertex s, we can easily compute $ed(s,v)$ and $od(s,v)$ for every $v \in V$ as follows. Construct a graph $G' = (V', E')$ where

$$V' = \{v_e, v_o \mid v \in V\},$$

$$E' = \{(x_e, y_o), (x_o, y_e) \mid (x,y) \in E\}.$$

The graph G' is a directed bipartite graph that contains an *even representative* v_e and an *odd representative* v_o for every vertex $v \in V$. It is easily seen that $ed(u,v) = d'(u_e, v_e)$ and that $od(u,v) = d'(u_e, v_o)$, for every $u, v \in V$, where $d'(u', v')$ denotes the distance between u' and v' in G'. By performing a BFS on G' from s_e, we can therefore find $ed(s,v)$ and $od(s,v)$, for every $v \in V$, in $O(E)$ time (we assume the graph contains no isolated vertices).

For every $s \in V$, we can find, in $O(E)$ time, a shortest odd length closed walk that contains s. We simply compute $oc(s) = \min\{ed(s,v) + 1 \mid (v,s) \in E\}$. If $oc(s) \neq \infty$, then a closed walk of length $oc(s)$, which is the minimal possible odd length, is easily found by tracing a shortest odd path from s to any vertex for which the minimum was achieved. The shortest odd length closed walk found in this way must be a SOLC. We thus obtain the following.

THEOREM 5.3. *A shortest odd length cycle in a directed graph $G = (V, E)$, if one exists, can be found in $O(VE)$ time.*

6. Concluding remarks. We have shown that interesting combinatorial properties of even cycles in undirected graphs lead to very efficient algorithms for finding even cycles of a given length and for finding shortest even cycles. Note that if the input graph is given using an adjacency matrix, then these $O(V^2)$ algorithms are optimal. It seems plausible to conjecture that $O(V^2)$ is the best possible bound, in terms of V, for these problems even if the adjacency lists of the graphs are given as input. Note that $O(V^2)$ time is currently the best known time even for finding quadrilaterals (C_4's).

Based on the results of this paper, Alon, Yuster, and Zwick [2] have recently obtained an $O(E^{4/3})$ algorithm for finding a C_4 in undirected graphs. More generally, a C_{4k} can be found in $O(E^{2-(\frac{1}{k}-\frac{1}{2k+1})})$ time. These algorithms are better than the $O(V^2)$ time algorithms presented here for relatively sparse graphs. It is interesting to note that the hardest cases for the C_4 problem, for example, are graphs with $\Theta(V^{3/2})$ edges.

Acknowledgment. The authors would like to thank Noga Alon for many helpful discussions.

REFERENCES

[1] N. Alon, R. Yuster, and U. Zwick, *Color-coding*, J. Assoc. Comput. Mach., 42 (1995), pp. 844–856. A preliminary version appeared in *Proc. 26th Annual ACM Symp. on Theory of Computing*, Montréal, Canada, 1994, pp. 326–335.

[2] N. Alon, R. Yuster, and U. Zwick, *Finding and counting given length cycles*, in Proc. 2nd European Symp. on Algorithms, Utrecht, the Netherlands, 1994, pp. 354–364; Algorithmica, to appear.

[3] B. Bollobás, *Extremal Graph Theory*, Academic Press, New York, 1978.

[4] J.A. Bondy and M. Simonovits, *Cycles of even length in graphs*, J. Combin. Theory, Ser. B, 16 (1974), pp. 97–105.

[5] A. Itai and M. Rodeh, *Finding a minimum circuit in a graph*, SIAM J. Comput., 7 (1978), pp. 413–423.

[6] B. Monien, *The complexity of determining a shortest cycle of even length*, Computing, 31 (1983), pp. 355–369.

[7] B. Monien, *How to find long paths efficiently*, Ann. Discrete Math., 25 (1985), pp. 239–254.

[8] D. Richards and A. L. Liestman, *Finding cycles of a given length*, Ann. Discrete Math., 27 (1985), pp. 249–256.

[9] C. Thomassen, *Even cycles in directed graphs*, European J. Combinatorics, 6 (1985), pp. 85–89.

[10] J. van Leeuwen, *Graph algorithms*, in Handbook of Theoretical Computer Science, Volume A, Algorithms and Complexity, J. van Leeuwen, ed., Elsevier and The MIT Press, 1990, Chap. 10, pp. 525–631.

SIAM J. DISCRETE MATH.
Vol. 10, No. 2, pp. 223–245, May 1997

006

LOAD BALANCING IN QUORUM SYSTEMS[*]

RON HOLZMAN[†], YOSI MARCUS[‡], AND DAVID PELEG[‡]

Abstract. This paper introduces and studies the question of balancing the load on processors participating in a given quorum system. Our proposed measure for the degree of balancing is the ratio between the load on the least frequently referenced element and on the most frequently used one.

We give some simple sufficient and necessary conditions for perfect balancing. We then look at the balancing properties of the common class of voting systems and prove that every voting system with odd total weight is perfectly balanced. (This holds, in fact, for the more general class of ordered systems.)

We also give some characterizations for the balancing ratio in the worst case. It is shown that for any quorum system with a universe of size n, the balancing ratio is no smaller than $1/(n-1)$, and this bound is the best possible. When restricting attention to nondominated coteries (NDCs), the bound becomes $2/\big(n-\log_2 n+o(\log n)\big)$, and there exists an NDC with ratio $2/\big(n-\log_2 n-o(\log n)\big)$.

Next, we study the interrelations between the two basic parameters of load balancing and quorum size. It turns out that the two size parameters suitable for our investigation are the size of the largest quorum and the *optimally weighted average quorum size* (OWAQS) of the system. For the class of ordered NDCs (for which perfect balancing is guaranteed), it is shown that over a universe of size n, some quorums of size $\lceil (n+1)/2 \rceil$ or more must exist (and this bound is the best possible). A similar lower bound holds for the OWAQS measure if we restrict attention to voting systems. For nonordered systems, perfect balancing can sometimes be achieved with much smaller quorums. A lower bound of $\Omega(\sqrt{n})$ is established for the maximal quorum size and the OWAQS of any perfectly balanced quorum system over n elements, and this bound is the best possible.

Finally, we turn to quorum systems that cannot be perfectly balanced, but have some balancing ratio $0 < \rho < 1$. For such systems we study the trade-offs between the required balancing ratio ρ and the quorum size it admits in the best case. It is easy to get an analogue of the result for perfect balancing, yielding a lower bound of $\sqrt{n\rho}$. We actually get a better estimate by a refinement of the argument.

Key words. quorum, coterie, intersecting hypergraphs, load balancing

AMS subject classifications. 68R05, 68Q22, 68M10, 05C65, 05099

PII. S0895480193260303

1. Introduction.

1.1. Motivation. *Quorum systems* serve as a basic tool providing a uniform and reliable way for achieving coordination between processes in a distributed system. Quorum systems are defined as follows. Suppose that the system is composed of n elements $u_1, \ldots, u_n$, taken from a universe U, representing sites, nodes, processors, or other abstract entities. A *set system* is a collection $\mathcal{S}$ of sets over the universe U. A set system $\mathcal{S}$ is said to satisfy the *quorum intersection* requirement if for every two sets S_i and S_j in $\mathcal{S}$, the intersection $S_i \cap S_j$ is not empty. A *quorum system* is a collection of sets that enjoys the quorum intersection property. The sets of $\mathcal{S}$ are referred to as the *quorums* of the system.

[*] Received by the editors December 15, 1993; accepted for publication (in revised form) April 23, 1996. The research of the third author was supported in part by a Walter and Elise Haas Career Development Award and by an Israel Science Foundation grant. Part of the work was done while he was visiting the IBM T. J. Watson Research Center.

http://www.siam.org/journals/sidma/10-2/26030.html

[†] Department of Mathematics, Technion–Israel Institute of Technology, Haifa 32000, Israel (holzman@techunix.technion.ac.il).

[‡] Department of Applied Mathematics and Computer Science, The Weizmann Institute of Science, Rehovot 76100, Israel (marcus@wisdom.weizmann.ac.il, peleg@wisdom.weizmann.ac.il).

Applications for quorum systems in distributed systems include control and management problems such as *mutual exclusion* (cf. [R86]), *name servers* (cf. [MV88]), and *replicated data management* (cf. [H84]). In all of these cases, the use of quorum systems is centered on the following basic idea. The application requires that certain information items be stored in the network in a reliable and consistent way. Storing the information at a single central site is problematic in case that site crashes. Storing the information at one particular *set of sites* may overcome this problem, but will prevent working in the system if a communication failure causes a partition in the network, since if users at different parts of the network continue working separately, the information can no longer be guaranteed to be consistent.

The conceptual solution based on quorum systems is to make use of a large *collection* of possible sets of sites in the system. Each such set forms a *quorum* in the sense that any query or update operation concerning the information at hand can be performed by accessing the elements of this single set alone, and the choice of the particular quorum to be used can be made arbitrarily (i.e., all quorums are equally adequate).

In particular, in order to perform an update to the information, the user selects one quorum S_i in the quorum system $\mathcal{S}$, and records the update in every one of the elements that compose S_i. Likewise, a potential consumer of this information may choose any quorum $S_j \in \mathcal{S}$, and query the elements of S_j for the needed information. Note that the consumer must query *each* of the elements of S_j in order to be certain of obtaining the latest version. The reason for this is that a sequence of k updates, performed by a number of different users, may make use of different quorums $S_{i_1}, \ldots, S_{i_k}$, and therefore the elements of a quorum S_j used in a subsequent query may contain different information. Specifically, if the element $x \in S_j$ does not belong to S_{i_k} then the information stored in it will not be the most recent one. Moreover, it is impossible to tell, just by inspecting the data stored at x, whether this is the last version. Luckily, since the intersection of every two quorums in a quorum system is not empty, the consumer is guaranteed to encounter at least one element that is able to supply the most up-to-date version (namely, the element at the intersection of S_j and S_{i_k}).

This type of solution is capable of withstanding crashes and network partitions (up to a point), due to the greater degree of freedom the user has in choosing the quorum. In particular, in the case of crashes, the consumer can choose a quorum that does not include the crashed elements, and in the case of a partition, it may still be possible for one part of the network to contain a complete quorum. (Of course, it is quite impossible for two disconnected parts of the system to both contain complete quorums!)

Considerable attention is given in the literature to a special type of quorum system called a *coterie* (see [GB85] and [IK90]). A *coterie* is a quorum system in which the quorums are not allowed to fully contain each other. A subclass of special interest is that of *nondominated coteries* (or *NDCs*), which are better than other coteries in terms of fault tolerance and communication cost. This subclass is defined as follows. Given two coteries $\mathcal{S}_1$ and $\mathcal{S}_2$ over the same universe U, we say that $\mathcal{S}_2$ *dominates* $\mathcal{S}_1$ if $\mathcal{S}_2 \neq \mathcal{S}_1$ and for every quorum $S \in \mathcal{S}_1$ there is a quorum $T \in \mathcal{S}_2$ such that $T \subseteq S$. An NDC is a coterie which is not dominated by any other coterie (see [GB85]).

1.2. Load balancing. There are many types of quorum systems, and many parameters of quorum systems affecting the applications using them. Such parameters include quorum sizes (affecting communication costs) and the number of quorums (affecting immunity to partitioning).

Of special interest are parameters for evaluating the distribution of workload over the system, and measuring the degree of balancing possible for a given quorum system. If all the users of the system prefer to use one particular quorum while possible (e.g., in a failure-free execution), then the elements participating in this quorum will be overloaded compared to others. So it makes sense to try to use a more uniform distribution for selecting the quorum to be accessed. Formally, given a quorum system $\mathcal{S} = \{S_1, \ldots, S_m\}$, a *quorum load vector* (QLV) is a vector $\mathbf{v} = (v_1, v_2, \ldots, v_m)$ expressing the distribution of relative loads placed on the quorums of $\mathcal{S}$. (That is, in a long sequence of quorum accesses, a v_i fraction of the accesses is directed at quorum S_i.)

This distribution induces an *access rate* for each element u_j, which is the sum of the access frequencies of the quorums it belongs to, $a_j = \sum_{u_j \in S_i} v_i$. Thus the *element load vector* (ELV) $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ induced by the QLV $\mathbf{v}$ expresses the relative loads placed on the elements of U when using the QLV $\mathbf{v}$.

Our proposed measure for the degree of balancing is the ratio between the rate of accesses to the least frequently used element in the quorum system and the rate of accesses to the most frequently used one. Formally, given a QLV $\mathbf{v}$ for $\mathcal{S}$ and the induced ELV $\mathbf{a}$, we let $\rho_{\mathcal{S},\mathbf{v}} = \min\{a_j\}/\max\{a_j\}$, and define the *balancing ratio* $\rho_{\mathcal{S}}$ of $\mathcal{S}$ as the maximum $\rho_{\mathcal{S},\mathbf{v}}$ over all QLVs $\mathbf{v}$. A system is said to be *perfectly balanced* if all the elements are accessed at the same rate, namely, $\rho_{\mathcal{S}} = 1$.

This paper focuses on issues related to balancing. In current technologies, a common and promising way to increase computing power is by connecting many fast processors together into compound systems. Quorum systems can be used for coordination in such systems. For small systems, the effect of the particular quorum system used on the communication cost is not significant. However, when the systems become larger, the importance of choosing a good quorum system may significantly increase. In particular, some quorum systems may be well adapted to the demand of load balancing, while for others, such a demand may impose heavy communication costs. Worse yet, certain types of quorum systems may be incapable of providing perfect or even partial balancing, regardless of the cost.

In this paper we introduce and address this issue, defining the fundamental notions and concepts relevant to load balancing, and developing some basic results on the balancing properties of a variety of quorum system classes.

Let us remark that to the best of our knowledge, currently existing systems do not address the issue of load balancing at all. Consequently, the quorum selection mechanisms used in existing systems typically do not provide such balancing, as they base the selection on some arbitrary choice, or worse, on a fixed search pattern, perpetuating the imbalance.

However, even though current quorum systems do not provide any means for balancing the load on the processors, it should be clear that there is no *inherent* reason that prevents them from doing so. In fact, given a desirable QLV $\mathbf{v}$ for the quorum system $\mathcal{S}$ at hand (i.e., a QLV $\mathbf{v}$ for which $\rho_{\mathcal{S},\mathbf{v}} = \rho_{\mathcal{S}}$), it is rather straightforward to develop a simple randomized protocol for quorum selection, based on interpreting $\mathbf{v}$ as a probability distribution over the quorums, and drawing a quorum at random according to $\mathbf{v}$. Such a protocol will in fact enforce an actual load distribution close to the optimal one, with high probability. For many natural quorum system classes (including most of the specific classes discussed in what follows), this protocol will also enjoy an efficient (and fast) distributed implementation.

1.3. Related work. Synchronization and coordination are central issues in the area of distributed systems. Many types of synchronization protocols rely on variants of quorum systems. In [H84] quorum intersection is defined between *read quorums* and *write quorums*, and also between other abstract types of quorums. In [MV88] aspects of distributed control are examined and lower bounds are presented for certain types of quorum systems. The issues of fault tolerance and availability of quorum systems are studied in [PW93]. For more on the applicability of quorum-based techniques in distributed systems, and on the examples mentioned above, the reader is referred to [H84, GB85] and the references therein. We are unaware of previous discussion of load balancing issues in the context of quorum systems in the literature.

Set systems in general (including intersecting hypergraphs in particular) were studied extensively in recent years (cf. [B86]). The terms *coterie* and *nondominated coterie* (NDC) are defined in [GB85], and many properties of coteries and NDCs are presented. Some interesting properties of NDCs are derived in [L73]. In [IK90] a relationship is established between coteries and boolean functions. Properties of coteries and NDCs are derived from properties of the appropriate functions.

1.4. Contributions. This paper focuses on a number of questions related to the issue of balancing the load on processors participating in a given quorum system.

We begin by giving some simple sufficient and necessary conditions for perfect balancing. (One trivial necessary condition is that the system is nonredundant; namely, that every element participates in some quorum.)

We then look at the balancing properties of the common class of voting systems. (A voting system is based on assigning a number of "votes" to each element of the universe; the votes induce a quorum system by taking as a quorum any collection of elements that holds a "minimal" majority of all the votes.) We define the class of ordered NDCs, which is an extension of voting systems, and prove that every ordered NDC is perfectly balanced. It follows, in particular, that every voting system with odd total number of votes is perfectly balanced.

Next we turn to characterizations for the balancing ratio in the worst case. We show that for any quorum system with a universe of size n, the balancing ratio is no smaller than $1/(n-1)$, and this bound is the best possible. When restricting attention to NDCs, the bound becomes $2/\big(n-\log_2 n+o(\log n)\big)$, and there exists an NDC with ratio $2/\big(n-\log_2 n-o(\log n)\big)$.

Next, we study the interrelationships between the two basic parameters of load balancing and quorum size. It turns out that the two size parameters suitable for our investigation are the size of the largest quorum and the *optimally weighted average quorum size* (OWAQS) of the system (corresponding to an optimal load vector).

For the class of ordered NDCs (for which perfect balancing is guaranteed), it is shown that over a universe of size n, some quorums of size $\lceil (n+1)/2 \rceil$ or more must exist (and this bound is the best possible). A similar lower bound holds for the OWAQS measure if we restrict attention to voting systems.

For nonordered systems, perfect balancing can sometimes be achieved with much smaller quorums. A lower bound of $\Omega(\sqrt{n})$ is established for the maximal quorum size and the OWAQS of any perfectly balanced quorum system over n elements, and this bound is the best possible.

Finally, we turn to quorum systems that cannot be perfectly balanced, but have some balancing ratio $0 < \rho < 1$. For such systems we study the trade-offs between the required balancing ratio ρ and the quorum size it admits in the best case. It is easy to get an analogue of the result for perfect balancing, yielding a lower bound of

$\sqrt{n\rho}$. We actually get a better estimate, by a refinement of the argument.

2. Basic notions.

Definition. *A* quorum system *is a pair* $(U,\mathcal{S})$*, where* U *is a nonempty finite set and* $\mathcal{S}$ *is a set of nonempty subsets of* U *such that the intersection of every two sets in* $\mathcal{S}$ *is nonempty. We refer to the set* U *as the* universe *and to the sets in* $\mathcal{S}$ *as the* quorums *of the system.*

It is sometimes convenient to represent a quorum system by a matrix of 0's and 1's.

Definition. *The* quorum matrix *of a quorum system* $(U,\mathcal{S})$ *is the* $m\times n$ *matrix* $\hat{\mathcal{S}}=(\hat{s}_{ij})$ *obtained as follows: the elements of* U *are enumerated as* $u_1,u_2,\ldots,u_n$*, the quorums in* $\mathcal{S}$ *are enumerated as* $S_1,S_2,\ldots,S_m$*, and*

$$\hat{s}_{ij}=\begin{cases}1 & \textit{if } u_j\in S_i,\\ 0 & \textit{otherwise.}\end{cases}$$

We shall usually be interested in quorum systems in which no quorum contains another, since in the case of containment the larger quorum is redundant for our purposes.

Definition. *A* coterie *is a quorum system in which no quorum contains another quorum.*

In order to describe and analyze a coterie, it is often convenient to refer to the set of subsets of the universe which contain some quorum. This is facilitated by the following definition.

Definition. *A* monotone quorum system *(MQS) is a quorum system* $(U,\mathcal{M})$ *such that* $S\in\mathcal{M}$ *and* $S\subseteq T\subseteq U$ *imply* $T\in\mathcal{M}$*. Given a coterie* $(U,\mathcal{S})$*, a* superquorum *is any subset of* U *that contains a quorum of* $\mathcal{S}$*. The* MQS generated by $(U,\mathcal{S})$ *is the collection of superquorums of* $(U,\mathcal{S})$*, namely, the system* $(U,\bar{\mathcal{S}})$*, where* $T\in\bar{\mathcal{S}}$ *if and only if* $T\supseteq S$ *for some* $S\in\mathcal{S}$*. Conversely, if we are given a MQS* $(U,\bar{\mathcal{S}})$ *then the coterie* $(U,\mathcal{S})$ *is determined uniquely (*$S\in\mathcal{S}$ *if and only if* $S\in\bar{\mathcal{S}}$ *and no proper subset of* S *is in* $\bar{\mathcal{S}}$*) and is called the* coterie derived from $(U,\bar{\mathcal{S}})$*.*

Example 2.1. Minimal Majority Coterie. Let $|U|=n$ and let $\bar{\mathcal{S}}=\left\{S\subseteq U\colon\ |S|>\frac{n}{2}\right\}$; that is, the superquorums are the sets containing a majority of elements. The coterie derived from $(U,\bar{\mathcal{S}})$ is that in which the quorums are all subsets of U of size $\left\lceil\frac{n+1}{2}\right\rceil$. □

Notation. When $U=\{u_1,u_2,\ldots,u_n\}$ and $x_1,x_2,\ldots,x_n$ are real numbers, we denote the *x-weight* of a subset $S\subseteq U$ by

$$x(S)=\sum_{u_j\in S}x_j.$$

Example 2.2. Voting Coterie. Let $U=\{u_1,u_2,\ldots,u_n\}$ and assume that to each $u_j\in U$ we assign a nonnegative integer w_j, called the *weight* of u_j. Then we define the MQS $\bar{\mathcal{S}}=\left\{S\subseteq U\colon\ w(S)>\frac{w(U)}{2}\right\}$. The coterie derived from $(U,\bar{\mathcal{S}})$ is that in which the quorums are those subsets of U which carry a majority of the total weight and are inclusion-minimal with respect to this property. A coterie $(U,\mathcal{S})$ obtained in this manner is called a *voting system*. Observe that the minimal majority coterie of Example 2.1 is a special case of a voting system, in which all weights are equal. □

Example 2.3. Star Coterie. Let $U=\{u_1,u_2,\ldots,u_n\}$ and let $\mathcal{S}$ consist of the $n-1$ quorums $\{u_1,u_2\},\{u_1,u_3\},\ldots,\{u_1,u_n\}$. Then $(U,\mathcal{S})$ is a coterie. We call such a coterie a *star*. Observe that a star is also a voting system (take $w_1 = n - 1$, $w_2 = \cdots = w_n = 1$). $\square$

Voting systems play a distinguished role in the study of quorum systems because of the natural and simple way in which they are specified. The defining weights also supply a ranking of the elements of U in terms of their importance for forming quorums. This notion is captured by the following definition.

Notation. Let $U = \{u_1, u_2, \dots, u_n\}$, and let $S \subseteq U$ with $u_i \notin S$, $u_j \in S$. We denote by S_j^i the *replacement set* $\bigl(S \setminus \{u_j\}\bigr) \cup \{u_i\}$.

DEFINITION. *Let $(U, \mathcal{S})$ be a coterie. We say that $(U, \mathcal{S})$ is* ordered *if it is possible to enumerate the elements of U as $u_1, u_2, \dots, u_n$ so that the following holds: if $1 \leq i < j \leq n$ and S is a superquorum with $u_i \notin S$, $u_j \in S$, then S_j^i is also a superquorum.*

Intuitively, the above property means that if $i < j$ then u_i is at least as useful as u_j for forming quorums. The reason that the definition refers to superquorums rather than quorums is that it may happen that S is a quorum but S_j^i is a nonminimal superquorum. It is straightforward to check, and we will do so now.

FACT 2.4. *Every voting system is ordered.*

Proof. This is proved by enumerating the elements so that $w_1 \geq w_2 \geq \cdots \geq w_n$. $\square$

The converse is known to be false; that is, there exist ordered coteries that cannot be obtained as a voting system [Os85]. There are also coteries that are not ordered, as witnessed by the following class of examples.

Example 2.5. FPP. Let U and $\mathcal{S}$ be the sets of points and lines, respectively, of a *finite projective plane* (see [H86]). We recall that in a finite projective plane of order q (abbreviated FPP(q)) there are n points and n lines, where $n = q^2 + q + 1$. Each line contains $q + 1$ points and there are $q + 1$ lines going through each point. Any two lines have exactly one point in common, and through any two points there is exactly one line. A FPP(q) is known to exist for every q which is a prime power. Clearly, if $(U, \mathcal{S})$ is a FPP(q), $q \geq 2$, then $(U, \mathcal{S})$ is not ordered, since no point can replace another in a line. $\square$

A special class of coteries arises from a concept of domination among coteries (see [GB85]).

DEFINITION. *Let $(U, \mathcal{S}_1)$ and $(U, \mathcal{S}_2)$ be coteries. We say that $(U, \mathcal{S}_2)$* dominates *$(U, \mathcal{S}_1)$ if $\mathcal{S}_2 \neq \mathcal{S}_1$ and for every quorum $S \in \mathcal{S}_1$ there is a quorum $T \in \mathcal{S}_2$ such that $T \subseteq S$. A* nondominated coterie *(NDC) is a coterie which is not dominated by any other coterie.*

The following fact (cf. Cor. 2.1 in [IK90]) can be used as a convenient alternative definition of an NDC.

PROPOSITION 2.6. *Let $(U, \mathcal{S})$ be a coterie. Then $(U, \mathcal{S})$ is an NDC if and only if for every partition of U into two parts S_1 and S_2, one of the S_i $(i = 1, 2)$ is a superquorum.*

We now record a simple but useful property of NDCs.

PROPOSITION 2.7. *Let $(U, \mathcal{S})$ be an NDC, and let $u \in U$ be in $\cup\mathcal{S}$ (that is, u belongs to at least one quorum). Then:*

(a) *There exist two quorums S and T such that $S \cap T = \{u\}$.*

(b) *If, moreover, $(U, \mathcal{S})$ is ordered with corresponding enumeration $u_1, u_2, \dots, u_n$ of U and $u = u_j$, then there are two quorums S and T such that $S \cap T = \{u_j\}$ and $S \cup T \supseteq \{u_1, \dots, u_j\}$.*

Proof. Let S be a quorum containing u. Applying the property given in Proposition 2.6 to the partition $S \setminus \{u\}$, $(U \setminus S) \cup \{u\}$, we conclude that $(U \setminus S) \cup \{u\}$ is a superquorum. Let T be a quorum contained in it. Then $S \cap T \subseteq \{u\}$, and since the intersection of two quorums is nonempty we have $S \cap T = \{u\}$, establishing part (a).

To prove part (b), assume that $i < j$ and $u_i \notin S \cup T$. By the property of an ordered coterie it follows that the replacement set T^i_j is a superquorum. This, however, is a contradiction since T^i_j is disjoint from S. □

Let us examine the above examples of coteries to see whether they are NDCs.

FACT 2.8.

(a) *The minimal majority coterie of Example* 2.1 *is an NDC if and only if* n *is odd.*

(b) *A sufficient condition for a voting system (Example* 2.2*) to be an NDC is that the total weight be odd.*

(c) *A star coterie (Example* 2.3*) is dominated.*

(d) *A finite projective plane FPP*(q) *(Example* 2.5*) is an NDC for* $q = 2$ *but is dominated for all* $q > 2$*.*

Proof. Parts (a) and (b) [GB85] are seen easily from Proposition 2.6. Part (c) follows since neither $\{u_1\}$ nor $\{u_2, \ldots, u_n\}$ is a superquorum. For Part (d) see [P70, C93]. □

We remark that despite Fact 2.8(d), FPP(q) satisfies the property of Proposition 2.7(a) for all q.

The central concept of this research deals with load balancing.

DEFINITION. *Let* $(U, \mathcal{S})$ *be a quorum system with quorum matrix* $\hat{\mathcal{S}} = (\hat{s}_{ij})$, $i = 1, \ldots, m$, $j = 1, \ldots, n$. *A* quorum load vector *(QLV) is a vector* $\mathbf{v} = (v_1, v_2, \ldots, v_m)$ *whose components are real nonnegative numbers (not all zero) expressing the relative loads that are to be placed on the quorums of* $\mathcal{S}$*. The* element load vector *(ELV) induced by the QLV* $\mathbf{v}$ *is the vector* $\mathbf{a} = \mathbf{a}(\mathcal{S}, \mathbf{v}) = (a_1, a_2, \ldots, a_n)$ *computed by* $\mathbf{a} = \mathbf{v}\hat{\mathcal{S}}$ *and expressing the relative loads placed on the elements of* U *when using the QLV* $\mathbf{v}$.

DEFINITION. *Let* $(U, \mathcal{S})$ *be a quorum system. Given a QLV* $\mathbf{v} = (v_1, v_2, \ldots, v_m)$ *which induces the ELV* $\mathbf{a} = (a_1, a_2, \ldots, a_n)$*, we define the* balancing ratio *for* $\mathcal{S}$ *and* $\mathbf{v}$ *as*

$$\rho_{\mathcal{S},\mathbf{v}} = \frac{\min_{j=1,\ldots,n}\{a_j\}}{\max_{j=1,\ldots,n}\{a_j\}}.$$

The balancing ratio *of* $(U, \mathcal{S})$ *is defined as*

$$\rho_{\mathcal{S}} = \max\{\rho_{\mathcal{S},\mathbf{v}} \colon \ \mathbf{v} \text{ is a QLV}\}.$$

A straightforward continuity and compactness argument shows that $\rho_{\mathcal{S}}$ is well defined. We have associated with each quorum system $(U, \mathcal{S})$ a parameter $0 \leq \rho_{\mathcal{S}} \leq 1$, which tells us how evenly we can spread the load among the elements of U if we are allowed to assign the relative loads to the quorums optimally. The higher the $\rho_{\mathcal{S}}$, the better behaved the quorum system is from the point of view of load balancing.

We note the following basic fact regarding the balancing ratio.

FACT 2.9. *If* $U \neq \cup\mathcal{S}$ *then* $\rho_{\mathcal{S}} = 0$.

Proof. If $U \neq \cup\mathcal{S}$, then there is some element $u_i \in U$ that does not participate in any quorum of $\mathcal{S}$. Hence, no matter which QLV $\mathbf{v}$ we choose, a_i will be zero, and thus the balancing ratio $\rho_{\mathcal{S},\mathbf{v}}$ will be zero too. □

Consequently, in studying the balancing ratio it is natural to make the assumption that each element appears in some quorum.

DEFINITION. *A quorum system* $(U, \mathcal{S})$ *is* nonredundant *if each element of* U *appears in some quorum*; *i.e.,* $U = \cup\mathcal{S}$.

Once this assumption holds, we have $\rho_{\mathcal{S}} > 0$. The most pleasing situation is when all element loads can be made equal; that is, $\rho_{\mathcal{S}} = 1$.

DEFINITION. *A quorum system* $(U, \mathcal{S})$ *is* perfectly balanced *if* $\rho_{\mathcal{S}} = 1$.

3. Perfect balancing. We begin with a simple sufficient condition for perfect balancing.

DEFINITION. *Let* $(U, \mathcal{S})$ *be a quorum system and let* $u \in U$. *The* degree *of* u *in* $\mathcal{S}$ *is* $d_{\mathcal{S}}(u) = \left|\{S \in \mathcal{S}\colon\ u \in S\}\right|$. *We say that* $(U, \mathcal{S})$ *is* regular *if all elements of* U *have the same degree in* $\mathcal{S}$.

PROPOSITION 3.1. *Every regular quorum system is perfectly balanced.*

Proof. The proposition is proved by assigning equal loads to all quorums. □

As an application of Proposition 3.1, we note that the minimal majority quorum systems of Example 2.1 and the FPP coterie of Example 2.5 are regular, and hence perfectly balanced. The star coterie (Example 2.3), on the other hand, is not perfectly balanced (when $n \geq 3$), since it can be seen that the load on the center of the star is the sum of the loads on the other elements.

In trying to determine when a given quorum system is perfectly balanced, the following characterization is useful.

PROPOSITION 3.2. *Let* $(U, \mathcal{S})$ *be a quorum system, with* $U = \{u_1, u_2, \ldots, u_n\}$. *Then* $(U, \mathcal{S})$ *is perfectly balanced if and only if there exists no* $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ *satisfying*

$$x(S) \geq 0 \text{ for all } S \in \mathcal{S}, \tag{1}$$

$$x(U) < 0. \tag{2}$$

(Recall the x-weight notation.)

Proof. The quorum system $(U, \mathcal{S})$ is perfectly balanced if there exists a real nonnegative vector $\mathbf{v}$ solving the equation system $\mathbf{v}\hat{\mathcal{S}} = \mathbf{1}$, where $\mathbf{1}$ denotes the n-dimensional vector of 1's. By the Minkowski–Farkas Lemma ([F01]; cf. [C83]), this is equivalent to the condition that the system of inequalities $\mathbf{x}\hat{\mathcal{S}}^{\top} \geq \mathbf{0}$, $\mathbf{x}\cdot\mathbf{1}^{\top} < 0$ has no solution. □

Our main result in this section is concerned with ordered NDCs. It will be derived from the following lemma.

LEMMA 3.3. *Let* $(U, \mathcal{S})$ *be a nonredundant NDC. Suppose that* $(U, \mathcal{S})$ *is ordered with corresponding enumeration* $u_1, u_2, \ldots, u_n$ *of* U. *Let* $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ *and* $\alpha \in \mathbb{R}$ *satisfy*

$$x(S) \geq \alpha \text{ for all } S \in \mathcal{S}, \tag{3}$$

$$x(U) \leq 2\alpha. \tag{4}$$

Then $x_j \geq 0$ *for* $j = 1, 2, \ldots, n$.

Proof. Suppose, for contradiction, that $x_j < 0$ for some j, and let J be the largest such j. By Proposition 2.7(b) there exist two quorums S and T such that $S \cap T = \{u_J\}$ and $S \cup T \supseteq \{u_1, \ldots, u_J\}$. By the choice of J, we have $x_i \geq 0$ for every $u_i \in U \setminus (S \cup T)$, and hence $x(U) - x(S \cup T) = x\big(U \setminus (S \cup T)\big) \geq 0$. Therefore, using (3) and $x_J < 0$, we get

$$x(U) \geq x(S \cup T) = x(S) + x(T) - x_J > 2\alpha,$$

which contradicts (4). □

THEOREM 3.4. *Every ordered nonredundant NDC is perfectly balanced.*

Proof. For the sake of contradiction, let $(U, \mathcal{S})$ have the properties stated, but fail to be perfectly balanced. By Proposition 3.2 there exists $\mathbf{x} \in \mathbb{R}^n$ satisfying (1) and (2). We may apply Lemma 3.3 with $\alpha = 0$ and conclude that $x_j \geq 0$ for $j = 1, 2, \ldots, n$. But this is inconsistent with (2). □

By Facts 2.4 and 2.8(b) we have the following corollary.

COROLLARY 3.5. *Every nonredundant voting system (Example* 2.2*) with odd total weight is perfectly balanced.* □

We remark that none of the assumptions made in Theorem 3.4 is superfluous. Indeed, the nonredundancy assumption is necessary for perfect balancing by Fact 2.9. If we drop the assumption of nondomination, the star coterie is an example that satisfies the other assumptions but not the conclusion. A class of examples indicating that the assumption of being ordered cannot be dispensed with will be presented in the following section (Example 4.3).

4. The balancing ratio in the worst case.

4.1. Characterization for the balancing ratio. The following proposition gives a dual formulation for the balancing ratio in the case when it is less than 1; it complements Proposition 3.2, which dealt with the case when the balancing ratio is 1.

We shall use the following notation: if $U = \{u_1, u_2, \ldots, u_n\}$ and $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ then

$$P = \{u_j \in U\colon\ x_j > 0\},$$
$$N = \{u_j \in U\colon\ x_j < 0\}.$$

The expressions $x(P)$ and $x(N)$ will be used following our x-weight notation.

PROPOSITION 4.1. *Let* $(U, \mathcal{S})$ *be a quorum system, with* $U = \{u_1, u_2, \ldots, u_n\}$ *and* $\rho_{\mathcal{S}} < 1$. *Then*

$$\rho_{\mathcal{S}} = \min\{x(P)\},$$

where the minimum is taken over all $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ *satisfying*

$$x(S) \geq 0 \quad \textit{for all} \;\; S \in \mathcal{S}, \tag{5}$$

$$x(N) = -1. \tag{6}$$

Proof. The balancing ratio $\rho_{\mathcal{S}}$ can be defined as the optimal value of ρ in the linear programming problem

$$\rho_{\mathcal{S}} = \max_{\mathbf{v}, \rho}\{\rho\},$$

subject to

$$\begin{aligned} \mathbf{v}\hat{\mathcal{S}} &\leq \mathbf{1}, \\ \boldsymbol{\rho} - \mathbf{v}\hat{\mathcal{S}} &\leq \mathbf{0}, \\ \mathbf{v} &\geq \mathbf{0}, \\ \rho &\geq 0, \end{aligned}$$

where $\hat{\mathcal{S}}$ is the quorum matrix, $\boldsymbol{\alpha}$ denotes (for $\alpha \in \mathbb{R}$) the vector of appropriate dimension with all components equal to α, vector inequalities are understood componentwise, and the maximum is taken over all QLVs $\mathbf{v}$ and $\rho \in \mathbb{R}$. Note that this formulation is equivalent to the definition of $\rho_{\mathcal{S}}$, since the QLV $\mathbf{v}$ can always be normalized so that the largest component of the induced ELV $\mathbf{a}$ becomes 1. By linear programming duality, we can express $\rho_{\mathcal{S}}$ in the form

$$\rho_{\mathcal{S}} = \min_{\mathbf{y},\mathbf{z}}\{y(U)\}$$

subject to

$$z(U) \geq 1, \tag{7}$$

$$y(S) - z(S) \geq 0, \text{ for all } \ S \in \mathcal{S}, \tag{8}$$

$$\mathbf{y} \geq \mathbf{0}, \tag{9}$$

$$\mathbf{z} \geq \mathbf{0}, \tag{10}$$

where the minimum is taken over all vectors $\mathbf{y}, \mathbf{z} \in \mathbb{R}^n$.

We begin by showing that there exists a vector $\mathbf{x}$ satisfying (5) and (6) and also $x(P) \leq \rho_{\mathcal{S}}$. The inequality $\rho_{\mathcal{S}} \geq \min\{x(P)\}$ then follows. Suppose now that $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ and $\mathbf{z} = (z_1, z_2, \ldots, z_n)$ satisfy (7)–(10), and yield the optimal value in the dual linear programming problem. We may assume that $z(U) = 1$, since we can achieve this by decreasing the values of the components of $\mathbf{z}$ without affecting the value of the solution or the validity of the constraints. Let $\mathbf{x} = \mathbf{y} - \mathbf{z}$, and let $N \subseteq U$ be defined with respect to $\mathbf{x}$ as in the statement of the proposition.

We observe first that $x(U) < 0$. Indeed,

$$x(U) = y(U) - z(U) = \rho_{\mathcal{S}} - 1$$

and $\rho_{\mathcal{S}} < 1$ by assumption. It follows in particular that $N \neq \emptyset$, and therefore $x(N) < 0$. On the other hand, by (9) and (10),

$$x(N) = \sum_{u_j \in N} y_j - z_j \geq -\sum_{u_j \in N} z_j \geq -\sum_{u_j \in U} z_j = -1.$$

Therefore, we can find a real number $\alpha \geq 1$ so that $\alpha x(N) = -1$; hence the vector $\alpha\mathbf{x}$ satisfies (6). It follows from (8) that $\mathbf{x}$, and hence also $\alpha\mathbf{x}$, satisfies (5). Thus $\alpha\mathbf{x}$ satisfies both (5) and (6). We have

$$\alpha x(P) = \alpha x(U) - \alpha x(N) = \alpha x(U) + 1 \leq x(U) + z(U) = y(U) = \rho_{\mathcal{S}}$$

(where the inequality relies on $x(U) < 0$ and $\alpha \geq 1$, and on (7)).

It remains to show, in the other direction, that any $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ which satisfies (5) and (6) has $x(P) \geq \rho_{\mathcal{S}}$. The inequality $\rho_{\mathcal{S}} \leq \min\{x(P)\}$ follows immediately. Let $\mathbf{x}$ be such a vector. We define the vectors $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ and $\mathbf{z} = (z_1, z_2, \ldots, z_n)$ by

$$y_j = \begin{cases} x_j & \text{if } \ x_j > 0, \\ 0 & \text{otherwise}, \end{cases}$$

$$z_j = \begin{cases} -x_j & \text{if } \ x_j < 0, \\ 0 & \text{otherwise}. \end{cases}$$

It can be checked that $\mathbf{y}$ and $\mathbf{z}$ satisfy (7)–(10). It follows that $y(U) \geq \rho_{\mathcal{S}}$. Since $y(U) = x(P)$, we are done. □

4.2. A lower bound for the balancing ratio. We now address the following question: within the class of all nonredundant quorum systems with a universe of size n, how low can the balancing ratio be in the worst case?

THEOREM 4.2. *Let $(U,\mathcal{S})$ be a nonredundant quorum system with $U=\{u_1,u_2,\dots,u_n\}$, $n\geq 2$. Then $\rho_{\mathcal{S}}\geq 1/(n-1)$. This bound is the best possible.*

Proof. We may assume that $\rho_{\mathcal{S}}<1$. By Proposition 4.1, we have to prove that any $\mathbf{x}=(x_1,x_2,\dots,x_n)\in\mathbb{R}^n$ which satisfies (5) and (6) has $x(P)\geq 1/(n-1)$. Since $x(N)=-1$ (by (6)) and $|N|\leq n-1$ (due to (5)), there exists some $u_j\in N$ with $x_j\leq -1/(n-1)$. Using the nonredundancy assumption, let S be a quorum with $u_j\in S$. Then

$$x(P)\geq x(S\cap P)\geq -x(S\cap N)\geq -x_j\geq 1/(n-1)$$

(where the second inequality is due to (5) again).

A candidate for attaining the worst case is the star coterie of Example 2.3, whose balancing ratio is easily seen to be $1/(n-1)$. □

4.3. A lower bound for the balancing ratio on NDCs. The worst case for the balancing ratio occurs for the star, which is a dominated coterie. What happens if we restrict attention to NDCs? The following construction, taken from [EL74], exhibits a low balancing ratio.

Example 4.3. Nucleus Coterie. Let $r\geq 2$ be an integer and let U be the disjoint union of the sets K and L, where $|K|=2r-2$ and $|L|=\binom{2r-2}{r-1}/2$. Let the elements of L be put in a one-to-one correspondence with the halvings of K. That is, to every unordered pair A,B of disjoint subsets of K of size $r-1$ each there corresponds an element $u_{A,B}$ of L. Let $\mathcal{S}$ consist of all sets of the form $A\cup\{u_{A,B}\}$ and $B\cup\{u_{A,B}\}$, where A,B is a halving of K, as well as all subsets of K of size r. It is easy to verify that $(U,\mathcal{S})$ is an NDC (using Proposition 2.6) and it is nonredundant. The number of elements is $n=2r-2+\binom{2r-2}{r-1}/2$. The balancing ratio is 1 when $r=2$ and is $\rho_{\mathcal{S}}=4/\binom{2r-2}{r-1}$ when $r\geq 3$. The latter can be verified by noting that (a) the QLV $\mathbf{v}$ assigning zero load to the quorums contained in K and load 1 to every other quorum satisfies $\rho_{\mathcal{S},\mathbf{v}}=4/\binom{2r-2}{r-1}$, and (b) the vector $\mathbf{x}=(x_1,x_2,\dots,x_n)$ defined by

$$x_j=\begin{cases}\frac{2}{(r-1)\binom{2r-2}{r-1}} & \text{if } u_j\in K,\\ -\frac{2}{\binom{2r-2}{r-1}} & \text{if } u_j\in L,\end{cases}$$

satisfies (5) and (6) and $x(P)=4/\binom{2r-2}{r-1}$. □

We observe that for $r=3$ the above construction gives a nonredundant NDC $(U,\mathcal{S})$ with $|U|=7$ which has balancing ratio $\rho_{\mathcal{S}}=2/3$. It is therefore an example showing that Theorem 3.4 does not remain true if the assumption of being ordered is removed. No such example with universe of size smaller than 7 exists. Indeed, for $n\leq 5$ it is known that every NDC is a voting system and hence ordered [GB85]. For $n=6$, an exhaustive search shows that all nonredundant NDCs are perfectly balanced.

For large r, the above construction gives almost the worst case as will be proved next.

THEOREM 4.4. *For every nonredundant NDC $(U,\mathcal{S})$ with $U=\{u_1,u_2,\dots,u_n\}$, $\rho_{\mathcal{S}}\geq 2/\bigl(n-\log_2 n+o(\log n)\bigr)$. Furthermore, there exists such an NDC $(U,\mathcal{S})$ with $\rho_{\mathcal{S}}=2/\bigl(n-\log_2 n-o(\log n)\bigr)$.*

Proof. Let $(U, \mathcal{S})$ satisfying the assumptions be given, and let us write

$$\rho_{\mathcal{S}} = \frac{2}{n-\alpha}$$

for a suitable real number α. We have to prove that $\alpha \geq \log_2 n - o(\log n)$.

Let $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ be a vector which satisfies (5) and (6) and has

$$x(P) = \frac{2}{n-\alpha}. \tag{11}$$

Given any $u_j \in N$ we can find, using Proposition 2.7(a), two quorums S_j and T_j such that $S_j \cap T_j = \{u_j\}$. We have then (relying on (5) for the second inequality)

$$\begin{aligned} x(P) &\geq x(S_j \cap P) + x(T_j \cap P) \\ &\geq -x(S_j \cap N) - x(T_j \cap N) \geq -2x_j. \end{aligned} \tag{12}$$

It follows now from (11) and (12) that

$$x_j \geq -\frac{1}{n-\alpha} \quad \text{for all} \quad u_j \in N. \tag{13}$$

Before continuing the proof, let us note that at this stage we could easily deduce that $\alpha \geq 2$. Indeed, in (12) it must be the case that $S_j \cap P$ and $T_j \cap P$ are nonempty (since both S_j and T_j contain $u_j \in N$, yet by (5) both $x(S_j), x(T_j) \geq 0$). As these sets are disjoint, we know that $|P| \geq 2$ and hence $|N| \leq n-2$. It follows by (6) that there exists $u_j \in N$ with $x_j \leq -1/(n-2)$. In view of (13), this implies that $\alpha \geq 2$. Thus we have a simple proof of the estimate $\rho_{\mathcal{S}} \geq 2/(n-2)$. In order to get the slightly better estimate stated in the theorem, some more work is needed.

Assume without loss of generality (w.l.o.g.) that $x_1 \geq x_2 \geq \cdots \geq x_n$. Split U into three disjoint parts by setting the boundary values

$$M_1 = -\frac{1}{\sqrt{\log_2 n}\,(n-\alpha)} \qquad \text{and} \qquad M_2 = -\frac{2}{3(n-\alpha)},$$

and defining

$$\begin{aligned} A &= \{u_1, u_2, \ldots, u_\ell\} = \Big\{u_j \in U\colon\ x_j \geq M_1\Big\}, \\ B &= \{u_{\ell+1}, u_{\ell+2}, \ldots, u_p\} = \Big\{u_j \in U\colon\ M_2 \leq x_j < M_1\Big\}, \\ C &= \{u_{p+1}, u_{p+2}, \ldots, u_n\} = \Big\{u_j \in U\colon\ x_j < M_2\Big\}. \end{aligned}$$

Note that $P \subseteq A$. Hence using these definitions plus (13) and (6), we can deduce

$$M_1\ell + M_2(p-\ell) - \frac{1}{n-\alpha}(n-p) \leq M_1 \cdot |P| + x(A \cap N) + x(B) + x(C) \leq x(N) = -1.$$

This can be rewritten as

$$\alpha \geq \frac{1}{3}p + \left(\frac{2}{3} - \frac{1}{\sqrt{\log_2 n}}\right)\ell. \tag{14}$$

For each $u_j \in C$, let us choose as above two quorums S_j and T_j such that $S_j \cap T_j = \{u_j\}$. Let us write $S'_j = S_j \setminus \{u_j\}$, $T'_j = T_j \setminus \{u_j\}$. We now establish some properties of these sets.

First, we claim that

$$S'_j, T'_j \subseteq A \cup B \quad \text{for} \quad j = p+1, \dots, n. \tag{15}$$

To see this, suppose for instance that $u_k \in S'_j \cap C$ for some $k \neq j$. Then we may sharpen (12) to get

$$x(P) \geq -2x_j - x_k > \frac{2}{n-\alpha},$$

which contradicts (11).

Second, we estimate the B portion of each set S'_j by

$$|S'_j \cap B| < \frac{2}{3}\sqrt{\log_2 n} \quad \text{for} \quad j = p+1, \dots, n. \tag{16}$$

This is seen again by sharpening (12) in the form

$$x(P) \geq -2x_j - x(S'_j \cap B) > \frac{4}{3(n-\alpha)} + \frac{|S'_j \cap B|}{\sqrt{\log_2 n}\ (n-\alpha)}$$

and comparing with (11).

Third, we argue that

$$S'_j \neq S'_k \quad \text{for} \quad j \neq k,\ p+1 \leq j,\ k \leq n. \tag{17}$$

Indeed, if $S'_j = S'_k$ then $S'_j \cap T'_k = \emptyset$ which implies, by (15), that $S_j \cap T_k = \emptyset$, in contradiction to the quorum intersection property.

It follows from (15)–(17), by considering the mapping $j \mapsto S'_j$, that

$$n - p \leq 2^{\ell} \sum_{i < \frac{2}{3}\sqrt{\log_2 n}} \binom{p-\ell}{i}. \tag{18}$$

Going back to (14) we see that if $p \geq 3\log_2 n$ we are done. So we assume that $p < 3\log_2 n$ and then obtain from (18) that

$$n - 3\log_2 n < 2^{\ell}(3\log_2 n)^{\frac{2}{3}\sqrt{\log_2 n}}.$$

Taking logarithms we get $\ell > \log_2 n - o(\log n)$. Using (14) and $p \geq \ell$ we have

$$\alpha \geq \left(1 - \frac{1}{\sqrt{\log_2 n}}\right)\ell > \left(1 - \frac{1}{\sqrt{\log_2 n}}\right)\bigl(\log_2 n - o(\log n)\bigr) = \log_2 n - o(\log n)$$

as required.

An example of a coterie nearly matching the bound is the nucleus coterie of Example 4.3, which for large r has $\rho_{\mathcal{S}} = 2/\bigl(n - \log_2 n - o(\log n)\bigr)$. □

We add two comments concerning Theorem 4.4 and its proof:

1. By some finer tuning of the proof it is possible to replace the $o(\log n)$ term by $\frac{4}{3}\sqrt{\log_2 n}$. Details are omitted.
2. The theorem remains true, with the same proof, if instead of an NDC we consider any quorum system having the property given in Proposition 2.7(a).

5. Load balancing and quorum size.

5.1. Measures for quorum size. In this section we study the extent of compatibility of two desirable goals: having a high balancing ratio and having small quorum sizes. The general theme will be that a high balancing ratio cannot be obtained with small quorum sizes.

Definition. *A quorum system is* r-uniform *if every quorum has* r *elements.*

If a quorum system is r-uniform then clearly we should use r as the parameter describing the quorum size. But for more general quorum systems, the question arises as to which parameter should be used for evaluating quorum sizes. Two conceivable parameters that do not serve our purposes well are the minimum quorum size and the average quorum size. This is illustrated by the following example.

Example 5.1. Wheel Coterie. Let $U = \{u_1, u_2, \dots, u_n\}$ and let $\mathcal{S}$ consist of the $n-1$ quorums $\{u_1, u_2\}, \{u_1, u_3\}, \dots, \{u_1, u_n\}$ and the additional quorum $\{u_2, u_3, \dots, u_n\}$. This differs from the star coterie (Example 2.3) only in the addition of the last quorum. It is easy to check that $(U, \mathcal{S})$ is perfectly balanced. Yet the minimum quorum size is 2 and the average quorum size is $3(n-1)/n$, both low numbers. We remark also that $(U, \mathcal{S})$ is a voting system and an NDC.

It turns out that two other parameters are more suitable for our investigation.

Definition. *Let* $(U, \mathcal{S})$ *be a quorum system. The* rank *of* $(U, \mathcal{S})$ *is defined as*

$$r_{\mathcal{S}} = \max\bigl\{|S|\colon\ S \in \mathcal{S}\bigr\}.$$

Definition. *Let* $(U, \mathcal{S})$ *be a quorum system with quorum matrix* $\hat{\mathcal{S}} = (\hat{s}_{ij})$, $i = 1, \dots, m$, $j = 1, \dots, n$. *Let* $\mathbf{v} = (v_1, v_2, \dots, v_m)$ *be a QLV. The* weighted average quorum size *(WAQS) of* $(U, \mathcal{S})$ *corresponding to* $\mathbf{v}$ *is*

$$g_{\mathcal{S},\mathbf{v}} = \frac{1}{\sum_{i=1}^{m} v_i} \sum_{i=1}^{m} v_i |S_i| = \frac{\sum_{j=1}^{n} a_j}{\sum_{i=1}^{m} v_i},$$

where $\mathbf{a} = (a_1, a_2, \dots, a_n)$ *is the ELV induced by* $\mathbf{v}$, *that is,* $\mathbf{a} = \mathbf{v}\hat{\mathcal{S}}$. *In the case when* $\mathbf{v}$ *is an optimizing QLV (that is,* $\rho_{\mathcal{S},\mathbf{v}} = \rho_{\mathcal{S}}$*), we refer to* $g_{\mathcal{S},\mathbf{v}}$ *as an* optimally weighted average quorum size *(OWAQS).*

As an illustration, let us apply these notions to the wheel coterie of Example 5.1. The rank there is $n-1$. The unique (up to proportionality) optimizing QLV is $\mathbf{v} = (1, 1, \dots, 1, n-2)$, which gives the OWAQS $g_{\mathcal{S},\mathbf{v}} = n(n-1)/(2n-3)$, which is slightly more than $n/2$.

In our context of load balancing, it seems that the notion of an OWAQS is the suitable way to measure quorum size. The rank is also interesting as a worst case measure. If the quorum system is r-uniform then all approaches give r as the answer. In general, the WAQS and even the OWAQS are not unique, as they depend on $\mathbf{v}$. Clearly, for every QLV $\mathbf{v}$ we have $g_{\mathcal{S},\mathbf{v}} \le r_{\mathcal{S}}$.

5.2. Quorum size bounds for ordered NDCs. In the first part of our analysis we shall focus on ordered nonredundant NDCs. This is a natural class of quorum systems for which we know that perfect balancing is guaranteed (Theorem 3.4). So it is interesting to ask what quorum sizes this class admits, or more precisely, how low we can make the rank and the OWAQS within this class.

Theorem 5.2. *Let* $(U, \mathcal{S})$ *be an ordered nonredundant NDC with universe of size* n. *Then* $r_{\mathcal{S}} \ge \bigl\lceil (n+1)/2 \bigr\rceil$. *This bound is the best possible.*

Proof. Let $u_1, u_2, \ldots, u_n$ be an enumeration of U with respect to which $(U, \mathcal{S})$ is ordered. Applying Proposition 2.7(b) with $u = u_n$, we obtain two quorums S and T such that $S \cap T = \{u_n\}$ and $S \cup T = U$. Then $|S| + |T| = n + 1$, so at least one of them has size $\geq \lceil (n+1)/2 \rceil$.

For odd n, the optimality of the bound is shown by the minimal majority coterie of Example 2.1. For even n this is shown by a slight modification of that example. $\square$

We note that no assumption of the theorem is redundant. The nucleus coterie of Example 4.3 is an r-uniform nonredundant NDC with $r \sim \frac{1}{2} \log_2 n$. The star (Example 2.3) is a 2-uniform ordered nonredundant coterie. If the nonredundancy assumption is removed then n may be made arbitrarily large without affecting anything else.

A similar lower bound on the OWAQS holds if we restrict attention to voting systems, a subclass of ordered coteries.

THEOREM 5.3. *Let $(U, \mathcal{S})$ be a perfectly balanced voting system with universe of size n. Then for every optimizing QLV $\mathbf{v}$, the OWAQS is greater than $n/2$.*

Proof. Let $\hat{\mathcal{S}} = (\hat{s}_{ij})$, $i = 1, \ldots, m$, $j = 1, \ldots, n$, be the quorum matrix, and let $\mathbf{v} = (v_1, v_2, \ldots, v_m)$ be a QLV such that $\rho_{\mathcal{S},\mathbf{v}} = 1$. Then the ELV induced by $\mathbf{v}$ is $\mathbf{a} = \mathbf{v}\hat{\mathcal{S}} = (a_1, a_2, \ldots, a_n)$ with all a_j equal, say, to the common value a. Let $\mathbf{w}^\top$ be a column vector whose components $w_1, w_2, \ldots, w_n$ are weights which determine the voting system $(U, \mathcal{S})$. Then it follows from the definition of a voting system that every component of $\hat{\mathcal{S}}\mathbf{w}^\top$ is greater than $w(U)/2$. Therefore

$$\mathbf{v}\hat{\mathcal{S}}\mathbf{w}^\top > \frac{w(U)}{2} \sum_{i=1}^{m} v_i. \tag{19}$$

On the other hand, since every component of $\mathbf{v}\hat{\mathcal{S}}$ equals a, we have

$$\mathbf{v}\hat{\mathcal{S}}\mathbf{w}^\top = a w(U). \tag{20}$$

Combining (19) and (20) we get $a > \frac{1}{2} \sum_{i=1}^{m} v_i$. Therefore

$$g_{\mathcal{S},\mathbf{v}} = \frac{\sum_{j=1}^{n} a_j}{\sum_{i=1}^{m} v_i} = \frac{na}{\sum_{i=1}^{m} v_i} > \frac{n}{2}. \qquad \square$$

Comparing the last two theorems, it is natural to ask whether the (stronger) conclusion of Theorem 5.3 holds under the conditions of Theorem 5.2. The question involves the class of ordered NDCs that are not voting systems (and therefore Theorem 5.3 does not apply to them). It is not easy to construct examples for this class, but this has been done: two such examples with universe of size 13 are given in [Os85]. In the following theorem we show not only that there is a member of this class for which the conclusion of Theorem 5.3 fails, but that it fails for every member of this class.

THEOREM 5.4. *Let $(U, \mathcal{S})$ be an ordered nonredundant NDC with universe of size n. Suppose further that $(U, \mathcal{S})$ is not a voting system. Then there exists an optimizing QLV $\mathbf{v}$ whose OWAQS is equal to $n/2$.*

Proof. Let $(U, \mathcal{S})$ satisfy the assumptions of the theorem and assume that $(U, \mathcal{S})$ is ordered with corresponding enumeration $u_1, u_2, \ldots, u_n$ of U.

As the first step in the proof, we claim that there is no pair (x, α), where $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ and α is a real number, such that

$$x(S) > \alpha > x(U \setminus S) \quad \text{for all} \quad S \in \mathcal{S}. \tag{21}$$

To prove the claim, suppose that such $\mathbf{x}$ and α exist. Then we may change the value of α, if necessary, to be $x(U)/2$, and (21) will still hold. Indeed, $x(S) > x(U \setminus S)$ implies that $x(S) > x(U)/2 > x(U \setminus S)$. So we shall assume that $\alpha = x(U)/2$. Applying Lemma 3.3 we deduce that $x_j \geq 0$ for $j = 1, 2, \ldots, n$. Now, let T be any subset of U. If T is a superquorum, say $T \supseteq S \in \mathcal{S}$, then it follows from (21) and the nonnegativity of the components of $\mathbf{x}$ that $x(T) \geq x(S) > \alpha$. If T is not a superquorum, then it follows from Proposition 2.6 that $U \setminus T$ is a superquorum, and therefore $x(T) = x(U) - x(U \setminus T) < x(U) - \alpha = \alpha$. We have shown that $\bar{\mathcal{S}} = \{T \subseteq U\colon\ x(T) > \alpha\}$. This indicates that $(U, \mathcal{S})$ is a voting system (strictly speaking, our definition of a voting system requires the weights to be integers, but this can be arranged by taking good enough rational approximations of the x_j's and clearing denominators). As this contradicts our assumption, we have proved the claim.

Let $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m \in \{0,1\}^n$ be the characteristic vectors of the quorums $S_1, S_2, \ldots, S_m$ ($\mathcal{S} = \{S_1, \ldots, S_m\}$). Let

$$Y = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m\},$$
$$Z = \{\mathbf{1} - \mathbf{y}_1, \mathbf{1} - \mathbf{y}_2, \ldots, \mathbf{1} - \mathbf{y}_m\},$$

where $\mathbf{1}$ is the all-1 n-dimensional vector. The claim asserts that there is no hyperplane that separates the points of Y from those of Z. It follows that

$$A = \operatorname{conv}(Y) \cap \operatorname{conv}(Z) \neq \emptyset,$$

where $\operatorname{conv}(X)$ denotes the convex closure of X. The set A is convex and symmetric about $\frac{\mathbf{1}}{\mathbf{2}}$ (that is, $\mathbf{a} \in A$ implies $\mathbf{1} - \mathbf{a} \in A$). Hence $\frac{\mathbf{1}}{\mathbf{2}} \in A$, and in particular $\frac{\mathbf{1}}{\mathbf{2}} \in \operatorname{conv}(Y)$. The latter means that there exists a QLV $\mathbf{v} = (v_1, v_2, \ldots, v_m)$ with $\sum_{i=1}^m v_i = 1$ which induces the ELV $\mathbf{a} = \frac{\mathbf{1}}{\mathbf{2}}$. For this $\mathbf{v}$ we get

$$g_{\mathcal{S},\mathbf{v}} = \frac{\sum_{j=1}^n a_j}{\sum_{i=1}^m v_i} = \frac{n \cdot \frac{1}{2}}{1} = \frac{n}{2}. \qquad \square$$

Theorems 5.3 and 5.4 yield the following characterization of voting systems within ordered NDCs.

Corollary 5.5. *Let $(U, \mathcal{S})$ be an ordered nonredundant NDC with universe of size n. Then the following are equivalent:*

(a) *$(U, \mathcal{S})$ is a voting system.*

(b) *Every OWAQS is greater than $n/2$.*

(c) *No OWAQS is equal to $n/2$.*

Before leaving the ordered world, we want to mention without details two examples that we have constructed:

1. An ordered coterie which is perfectly balanced but whose rank is less than $n/2$. (This shows that the nondomination assumption in Theorems 5.2 and 5.4 cannot be removed, even if we add the assumption of perfect balancing. It also shows that relaxing "voting system" to "ordered" in Theorem 5.3 admits examples where the theorem's conclusion fails in a more essential sense than indicated by Theorem 5.4.)
2. A quorum system satisfying all the assumptions of Theorem 5.4 for which there is an OWAQS which is less than $n/2$. (This shows that the existential quantifier in the theorem's conclusion cannot be made universal.)

5.3. Quorum size bounds for (nonordered) perfectly balanced quorum systems. The foregoing theorems indicate that certain methods for constructing quorum systems or certain properties of quorum systems which guarantee perfect balancing are costly in terms of quorum size. But perfect balancing can be achieved with considerably smaller quorums. Indeed, a FPP(q) (Example 2.5) is $(q+1)$-uniform and has a universe of size $n = q^2 + q + 1$, so its rank is roughly $\sqrt{n}$. It is perfectly balanced by Proposition 3.1.

Our next goal is to prove the optimality (in terms of quorum size) of the finite projective planes among all perfectly balanced quorum systems. For this purpose, we first review some known concepts and results on fractional matchings in hypergraphs. We express them using the terminology of the current paper.

DEFINITION. *Let $(U, \mathcal{S})$ be a quorum system with quorum matrix $\hat{\mathcal{S}} = (\hat{s}_{ij})$, $i = 1, \ldots, m$, $j = 1, \ldots, n$. A* fractional matching *in $(U, \mathcal{S})$ is a QLV $\mathbf{v} = (v_1, v_2, \ldots, v_m)$ such that the induced ELV $\mathbf{a} = \mathbf{v}\hat{\mathcal{S}} = (a_1, a_2, \ldots, a_n)$ satisfies $a_j \leq 1$, $j = 1, \ldots, n$. The* size *of a fractional matching $\mathbf{v} = (v_1, v_2, \ldots, v_m)$ is defined as*

$$|\mathbf{v}| = \sum_{i=1}^{m} v_i.$$

The fractional matching number *of $(U, \mathcal{S})$ is defined as*

$$\nu^*_{\mathcal{S}} = \max\bigl\{|\mathbf{v}|\colon \mathbf{v} \text{ is a fractional matching in } (U, \mathcal{S})\bigr\}.$$

It is easy to deduce the following from the quorum intersection property.

PROPOSITION 5.6. *Let $(U, \mathcal{S})$ be a quorum system. Then for every quorum $S \in \mathcal{S}$ we have $\nu^*_{\mathcal{S}} \leq |S|$. As a consequence, $\nu^*_{\mathcal{S}} \leq g_{\mathcal{S},\mathbf{v}}$ for every WAQS $g_{\mathcal{S},\mathbf{v}}$.*

The following finer estimate for $\nu^*_{\mathcal{S}}$ is due to Füredi.

PROPOSITION 5.7 (see [F81]). *Let $(U, \mathcal{S})$ be a quorum system of rank $r_{\mathcal{S}} = r$. Then $\nu^*_{\mathcal{S}} \leq r - 1 + 1/r$.*

A FPP(r–1), if it exists, is an r-uniform quorum system with universe of size $r^2 - r + 1$ and fractional matching number $r - 1 + 1/r$. Thus Füredi's bound is attained for those values of r such that a FPP(r–1) exists. The following corollary of Proposition 5.7 had been proved earlier by Lovász.

PROPOSITION 5.8 (see [L75]). *Let $(U, \mathcal{S})$ be an r-uniform, regular quorum system. Then $|U| \leq r^2 - r + 1$.*

Note that this bound too is attained for those values of r such that a FPP(r–1) exists.

We now return to our investigation of quorum size in perfectly balanced quorum systems.

THEOREM 5.9. *Let $(U, \mathcal{S})$ be a perfectly balanced quorum system with $|U| = n$. Then every OWAQS is at least $\sqrt{n}$.*

Proof. Let $\mathbf{v} = (v_1, v_2, \ldots, v_m)$ be a QLV with $\rho_{\mathcal{S},\mathbf{v}} = 1$. Then the ELV induced by $\mathbf{v}$ is $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ with all a_j equal. By a suitable normalization, which does not affect $g_{\mathcal{S},\mathbf{v}}$, we may assume that $a_1 = a_2 = \cdots = a_n = 1$. With this assumption, $\mathbf{v}$ is a fractional matching. We have

$$g_{\mathcal{S},\mathbf{v}} = \frac{\sum_{j=1}^{n} a_j}{\sum_{i=1}^{m} v_i} = \frac{n}{|\mathbf{v}|} \geq \frac{n}{\nu^*_{\mathcal{S}}},$$

and therefore

$$n \leq g_{\mathcal{S},\mathbf{v}}\nu^*_{\mathcal{S}}. \tag{22}$$

Using Proposition 5.6 this implies

$$n \leq g^2_{\mathcal{S},\mathbf{V}},$$

which yields the desired lower bound on $g_{\mathcal{S},\mathbf{V}}$. ◻

The foregoing theorem establishes the asymptotic optimality (in terms of OWAQS) of the finite projective planes among all perfectly balanced quorum systems. We can get exact optimality in terms of the rank, as follows.

THEOREM 5.10. *Let $(U, \mathcal{S})$ be a perfectly balanced quorum system of rank $r_{\mathcal{S}} = r$. Then $|U| \leq r^2 - r + 1$.*

Proof. We obtain (22) as in the proof of the previous theorem. Then, from $g_{\mathcal{S},\mathbf{V}} \leq r$ and Proposition 5.7, we get $|U| = n \leq r(r - 1 + 1/r) = r^2 - r + 1$. ◻

The last theorem is seen to be a generalization of the result of Lovász (Proposition 5.8): the uniformity assumption is dispensed with, as the rank suffices, and the regularity assumption is relaxed to perfect balancing.

5.4. Size-balancing trade-offs for unbalanced quorum systems. We have seen that if we insist on perfect balancing then the best we can do is to use quorums of size $\sim \sqrt{n}$. What if we relax perfect balancing and are willing to accept a balancing ratio not worse than some number ρ, $0 < \rho < 1$? Is there a trade-off between the required level ρ and the quorum size it admits in the best case?

It is easy to get an analogue of Theorem 5.9 (or 5.10) by observing that when $\rho_{\mathcal{S},\mathbf{V}} \geq \rho$ one obtains an adaptation of (22) in the form $n\rho \leq g_{\mathcal{S},\mathbf{V}}\nu^*_{\mathcal{S}}$. From this it follows that $g_{\mathcal{S},\mathbf{V}} \geq \sqrt{n\rho}$. We shall get a better estimate by a refinement of the argument, based on the following lemma.

LEMMA 5.11. *Let $\rho, a_1, a_2, \ldots, a_n$ be real numbers such that $0 < \rho \leq 1$ and $\rho \leq a_j \leq 1$ for $j = 1, \ldots, n$. Then*

$$\frac{\sum_{j=1}^n a_j^2}{\left(\sum_{j=1}^n a_j\right)^2} \leq \frac{(1+\rho)^2}{4n\rho}.$$

Proof. Given ρ and n, consider the problem of maximizing

$$f(a_1, a_2, \ldots, a_n) = \frac{\sum_{j=1}^n a_j^2}{\left(\sum_{j=1}^n a_j\right)^2}$$

subject to $\rho \leq a_j \leq 1$, $j = 1, \ldots, n$. For any $1 \leq i \leq n$ we have

$$\frac{\partial f}{\partial a_i} = \frac{2\sum_{\substack{j=1 \\ j \neq i}}^n (a_i - a_j)a_j}{\left(\sum_{j=1}^n a_j\right)^3}.$$

Since the numerator in the above expression is an increasing function of a_i, it follows that the maximum under consideration is attained when $a_i = \rho$ or $a_i = 1$. Indeed, if $\rho < a_i < 1$ and $\frac{\partial f}{\partial a_i} = 0$, then $\frac{\partial f}{\partial a_i}$ is negative for smaller values of a_i and positive for larger values of a_i, so we are looking at a minimum of f as a function of a_i.

Thus, it suffices to consider points $(a_1, a_2, \ldots, a_n)$ where k of the a_j's equal ρ and the other $n - k$ equal 1. Letting $x = k/n$ we have for such points

$$f(a_1, a_2, \ldots, a_n) = \frac{\rho^2 k + n - k}{(\rho k + n - k)^2} = \frac{1 - (1-\rho^2)x}{n\big(1 - (1-\rho)x\big)^2}.$$

One can show by elementary analysis that this expression is maximized in the interval $0 \leq x \leq 1$ when $x = 1/(1+\rho)$, and attains there the value $(1+\rho)^2/4n\rho$. $\square$

THEOREM 5.12. *Let $(U, \mathcal{S})$ be a quorum system with $|U| = n$. Let $0 < \rho \leq 1$ and let $\mathbf{v}$ be a QLV such that $\rho_{\mathcal{S},\mathbf{v}} \geq \rho$. Then*

$$g_{\mathcal{S},\mathbf{v}} \geq \frac{2\sqrt{n\rho}}{1+\rho}.$$

Proof. Let $\hat{\mathcal{S}} = (\hat{s}_{ij})$, $i = 1, \ldots, m$, $j = 1, \ldots, n$, be the quorum matrix. By a normalization which does not affect $\rho_{\mathcal{S},\mathbf{v}}$ or $g_{\mathcal{S},\mathbf{v}}$, we may assume that the ELV $\mathbf{a} = \mathbf{v}\hat{\mathcal{S}} = (a_1, a_2, \ldots, a_n)$ induced by $\mathbf{v}$ satisfies $\rho \leq a_j \leq 1$, $j = 1, \ldots, n$. We observe that by the quorum intersection property we have $\hat{\mathcal{S}}\hat{\mathcal{S}}^\top \geq \hat{1}$, where $\hat{1}$ denotes the $m \times m$ all-1 matrix, and the inequality holds entry-by-entry. Therefore,

$$\sum_{j=1}^{n} a_j^2 = \mathbf{a} \cdot \mathbf{a}^\top = \mathbf{v}\hat{\mathcal{S}}\hat{\mathcal{S}}^\top \mathbf{v}^\top \geq \mathbf{v}\hat{1}\mathbf{v}^\top = |\mathbf{v}|^2.$$

Using this and Lemma 5.11 we have

$$g_{\mathcal{S},\mathbf{v}}^2 = \frac{\left(\sum_{j=1}^n a_j\right)^2}{|\mathbf{v}|^2} \geq \frac{\left(\sum_{j=1}^n a_j\right)^2}{\sum_{j=1}^n a_j^2} \geq \frac{4n\rho}{(1+\rho)^2}.$$

Upon taking square roots we obtain the required result. $\square$

We now describe a construction showing that the bound given in Theorem 5.12 is rather tight.

Example 5.13. Ext-FPP. Let $0 < \rho < \frac{1}{2}$ and let r be a positive integer such that a FPP(r–1) exists. Let P and $\mathcal{L}$ be the sets of points and lines, respectively, of a FPP(r–1). Let K be a set of size $\left[(1-2\rho)/\rho\right](r^2 - r + 1)$, disjoint from P, and let $\mathcal{M}$ be the set of all subsets of K of size $(1-2\rho)r$. (We ignore adjustments that need to be made when these numbers are not integers. The effect of such adjustments is negligible when r is large.) Let $U = P \cup K$ and let $\mathcal{S}$ consist of all sets of the form $L \cup M$, where $L \in \mathcal{L}$ and $M \in \mathcal{M}$. Then $\mathcal{S}$ satisfies the quorum intersection requirement, because any two lines in $\mathcal{L}$ intersect. Since P has $r^2 - r + 1$ points and each line in $\mathcal{L}$ contains r points, we see that $|U| = n = [(1-\rho)/\rho]\,(r^2 - r + 1)$ and each quorum in $\mathcal{S}$ has size $2(1-\rho)r$.

Let $\mathbf{v}$ be a QLV assigning equal load to all the quorums in $\mathcal{S}$. Then it can be verified that $\rho_{\mathcal{S},\mathbf{v}} = \rho$. Indeed, it follows from considerations of symmetry that the induced ELV is constant over K and over P, and the ratio between the two constants can be computed as

$$\frac{|M| \cdot |P|}{|L| \cdot |K|} = \frac{(1-2\rho)r \cdot (r^2 - r + 1)}{r \cdot \frac{1-2\rho}{\rho}(r^2 - r + 1)} = \rho$$

(here $M \in \mathcal{M}$ and $L \in \mathcal{L}$). To evaluate the performance of this construction, we have to compare

$$g_{\mathcal{S},\mathbf{v}} = 2(1-\rho)r$$

with the bound of Theorem 5.12:

$$\frac{2\sqrt{n\rho}}{1+\rho} = \frac{2\sqrt{1-\rho}\sqrt{r^2 - r + 1}}{1+\rho}.$$

It is readily seen that the ratio between the two quantities approaches 1 as $\rho \to 0$ and $r \to \infty$. The ratio is in general less than $\bigl(1+\rho/2)(1+1/(2r)\bigr)$. $\square$

The theorem and the construction delineate with a good degree of precision a trade-off between the required level of balancing ρ (when $0 < \rho < \frac{1}{2}$) and the quorum size it admits in the best case. We remark that we do not know how to handle profitably the case when $\frac{1}{2} \leq \rho < 1$: if the required level of balancing is in this interval, the construction with smallest quorum size that we know is the same as for perfect balancing (namely, the finite projective plane).

5.5. Size-balancing trade-offs for NDCs. In view of the distinguished role played by NDCs among quorum systems, it is interesting to investigate the relation between the level of balancing and the quorum size within this special class. We start by describing a construction, borrowed from [EL74], of an NDC with quorums of size $O(\sqrt{n})$ which is, as we shall show, perfectly balanced.

The method of construction is inductive. In the inductive step, we are given an $(r-1)$-uniform quorum system $(U', \mathcal{S}')$. We take a set R of size r, disjoint from U', and form the new universe $U = U' \cup R$. We define the collection $\mathcal{S}$ by: $S \in \mathcal{S}$ if and only if $S = S' \cup \{u\}$ for some $S' \in \mathcal{S}'$ and $u \in R$, or $S = R$. We thus obtain a new system $(U, \mathcal{S})$.

PROPOSITION 5.14. *Let $(U, \mathcal{S})$ be obtained from the quorum system $(U', \mathcal{S}')$ as above. Then:*

(a) $(U, \mathcal{S})$ *is an r-uniform quorum system.*

(b) *If $(U', \mathcal{S}')$ is an NDC then so is $(U, \mathcal{S})$.*

(c) *If $(U', \mathcal{S}')$ is perfectly balanced then so is $(U, \mathcal{S})$.*

Proof. Part (a) is straightforward. Part (b) can be verified using Proposition 2.6. Indeed, let S_1, S_2 be a partition of U. Since $(U', \mathcal{S}')$ is an NDC and S_1', S_2' is a partition of U' (where $S_i' = S_i \cap U'$), we may assume that S_1', say, contains some $S' \in \mathcal{S}'$. Then, if $S_1 \cap R \neq \emptyset$ we conclude that S_1 contains a quorum of the form $S' \cup \{u\}$; if, on the other hand, $S_1 \cap R = \emptyset$ then S_2 contains the quorum R.

To prove part (c), let $\mathbf{v}'$ be a QLV for $(U', \mathcal{S}')$ which induces a load of 1 on each element of U'. Let $\mathbf{v}$ be the QLV for $(U, \mathcal{S})$ defined by: the load of $S' \cup \{u\}$, where $S' \in \mathcal{S}'$ and $u \in R$, is $1/r$ of the load of S' in $\mathbf{v}'$, and the load of R is $1 - |\mathbf{v}'|/r$ (this quantity is positive by virtue of Proposition 5.6). Then $\mathbf{v}$ induces a load of 1 on each element of U. $\square$

Example 5.15. Triangular. Let $(U^r, \mathcal{S}^r)$ be an r-uniform quorum system obtained by successive applications of the inductive step described above, starting from a system of one element. We call $(U^r, \mathcal{S}^r)$ a *triangular system*. It follows from Proposition 5.14 that $(U^r, \mathcal{S}^r)$ is an NDC and is perfectly balanced. The size of its universe is $|U^r| = n = (r+1)r/2$. $\square$

We observe that the quorum size achieved in the above construction is about $\sqrt{2n}$ and is thus within a multiplicative constant factor of the lower bound of $\sqrt{n}$ given in Theorem 5.9 (for all perfectly balanced quorum systems, not just NDCs). It seems plausible that the lower bound can be improved for the class of NDCs, but we are unable to do this. On the other hand, we can achieve a (very) slight improvement on the construction. Let

$$n(r) = \max\bigl\{|U|\colon\ (U, \mathcal{S}) \text{ is an } r\text{-uniform, perfectly balanced NDC}\bigr\}.$$

Then the above construction gives $n(r) \geq (r+1)r/2$. For $r = 3$ this becomes $n(3) \geq 6$, but the Fano plane (FPP(2)) shows that $n(3) \geq 7$ (in fact, we can deduce from Theorem 5.10 that $n(3) = 7$). When the inductive method described above is applied starting from the Fano plane, we obtain that $n(r) \geq (r+1)r/2+1$ for $r \geq 3$. In order to introduce a further improvement we need the following definition and easy facts.

DEFINITION. *Let $(U,\mathcal{S})$ be a quorum system, with $U = \{u_1, u_2, \ldots, u_n\}$. Let $(U_j, \mathcal{S}_j)$, $j = 1, \ldots, n$, be quorum systems, with the U_j's pairwise disjoint. The* composite quorum system *(CQS) formed by substituting $(U_j, \mathcal{S}_j)$, $j = 1, \ldots, n$, for the elements of $(U,\mathcal{S})$, denoted $CQS(\mathcal{S}, \{\mathcal{S}_j\})$, has as its universe $\bigcup_{j=1}^{n} U_j$ and as its quorums all sets obtained as follows: take any $S = \{u_{j_1}, u_{j_2}, \ldots, u_{j_k}\} \in \mathcal{S}$ and for each j_i, $i = 1, \ldots, k$, take any $S_{j_i} \in \mathcal{S}_{j_i}$, and form the (composite) quorum $\bigcup_{i=1}^{k} S_{j_i}$.*

PROPOSITION 5.16.

(a) *If $(U,\mathcal{S})$ is r-uniform and each $(U_j, \mathcal{S}_j)$ is s-uniform, then the CQS is rs-uniform.*

(b) *If $(U,\mathcal{S})$ is uniform and regular, and all of the $(U_j, \mathcal{S}_j)$ are regular with the same common degree and the same number of quorums, then the CQS is regular.*

(c) *If $(U,\mathcal{S})$ and each of the $(U_j, \mathcal{S}_j)$ are NDCs, then the CQS is an NDC.*

Now, consider the CQS formed by substituting seven copies of the Fano plane for the seven points of a Fano plane. By Proposition 5.16, this is a 9-uniform, regular (hence perfectly balanced) NDC. This shows that $n(9) \geq 49$, whereas $(r+1)r/2 = 45$ for $r = 9$. When the inductive method is applied successively starting from this CQS, we obtain that $n(r) \geq (r+1)r/2+4$ for $r \geq 9$.

CONJECTURE 5.17. *$n(r) = (r+1)r/2 + O(1)$.*

We observe that if $(U,\mathcal{S})$ is an r-uniform, perfectly balanced quorum system with $|U| = n$, then $\nu_{\mathcal{S}}^* = n/r$. Thus, the above conjecture can be reformulated as saying that if $(U,\mathcal{S})$ is an r-uniform, perfectly balanced NDC then $\nu_{\mathcal{S}}^* \leq (r+1)/2 + O(1/r)$. We believe, in fact, that this holds even without the assumptions of uniformity and perfect balancing. That is, we believe that the assumption of nondomination alone should permit the following improvement on Füredi's bound (Proposition 5.7).

CONJECTURE 5.18. *Let $(U,\mathcal{S})$ be an NDC of rank $r_{\mathcal{S}} = r$. Then $\nu_{\mathcal{S}}^* \leq (r+1)/2 + O(1/r)$.*

If we do not insist on perfect balancing, but continue to consider only NDCs, how low can we make the quorum size? It follows from a more general result in [T85] that any nonredundant NDC having rank r has universe of size smaller than $\binom{2r}{r}$. Recalling the nucleus coterie of Example 4.3, where the size of the universe is larger than $\binom{2r-2}{r-1}/2$, we see that Tuza's bound is within a multiplicative constant factor of being best possible. Stating the result differently, we can say that the smallest possible rank among all nonredundant NDCs with universe of size n is $\frac{1}{2}\log_2 n + \frac{1}{4}\log_2\log_2 n + O(1)$.

Suppose we require some level of balancing; that is, we consider NDCs with balancing ratio not worse than some number ρ, $0 < \rho < 1$. How low can we make the quorum size then? We are unable to improve on the lower bound given in Theorem 5.12 (which is not restricted to NDCs). A construction that attempts to approach that lower bound using NDCs is given next. It is not as good as the one (using dominated coteries) given by the Ext-FPP coterie of Example 5.13.

Example 5.19. CQS (Triangular, Nucleus). Let $(U^r, \mathcal{S}^r)$ be an r-uniform triangular NDC with $|U^r| = (r+1)r/2$ which is perfectly balanced, as in Example 5.15. Let $(U_s, \mathcal{S}_s)$ be an s-uniform nucleus NDC with $|U_s| = 2s-2+\binom{2s-2}{s-1}/2$ and $\rho_{\mathcal{S}_s} = 4/\binom{2s-2}{s-1}$, as in Example 4.3. Let $(U_s^r, \mathcal{S}_s^r)$ be the CQS formed by substituting $(r+1)r/2$ copies of $(U_s, \mathcal{S}_s)$ for the elements of $(U^r, \mathcal{S}^r)$. It follows from Proposition 5.16 that $(U_s^r, \mathcal{S}_s^r)$ is an rs-uniform NDC. It is easy to see that $\rho_{\mathcal{S}_s^r} = \rho_{\mathcal{S}_s}$. A rough computation shows that in terms of the size n of the universe and the balancing ratio

Table 1

Quorum systems	Hypergraph theory	Game theory
element	vertex	player
universe	vertex-set	player-set
quorum	hyperedge	minimal winning coalition
superquorum	nonindependent set of vertices	winning coalition
quorum system	intersecting hypergraph	proper simple game
monotone quorum system	intersecting filter	monotone proper simple game
coterie	intersecting antichain	(minimal winning coalitions of a) monotone proper simple game
nondominated coterie	3-chromatic intersecting antichain	(minimal winning coalitions of a) constant-sum game
rank	rank	size of a largest minimal winning coalition
voting system	threshold	weighted majority game
ordered	shifted	having a complete desirability relation
nonredundant	no isolated vertices	without dummies
quorum load vector	fractional matching (up to normalization)	weight assignment to the minimal winning coalitions
perfectly balanced	quasi-regularizable	having a balanced collection of minimal winning coalitions

ρ, the quorums of the CQS have size $\sim \sqrt{n\rho} \cdot \frac{1}{2} \log_2 \frac{1}{\rho}$. The ratio of this to the lower bound of Theorem 5.12 is $O\big(\log \frac{1}{\rho}\big)$. □

Appendix: A polyglot dictionary. The motivation for the research reported in this paper came from computer science, but the concepts involved have also been studied in other areas of science, under various interpretations and using various systems of terminology. To help overcome the language barriers, we thought it useful to provide here translations of the concepts into the languages of two other areas: hypergraph theory and game theory. Our little dictionary (Table 1) is only schematic, and for more information we refer the reader to books such as [B89] and [Ow82]. We should also mention that, due to scope limitations, our dictionary leaves out several other areas in which these concepts have come up. These include Boolean functions theory, reliability theory, neural networks, percolation theory, etc.

As an illustration of the possible appeal of our work to researchers in other areas, we rephrase Theorem 3.4 in game-theoretic terms and Conjecture 5.18 in hypergraph terms.

Theorem 3.4*. *Let G be a constant-sum game without dummies, having a complete desirability relation. Then G has a balanced collection of minimal winning coalitions.*

Conjecture 5.18*. *Let $\mathcal{H}$ be a 3-chromatic intersecting hypergraph of rank r. Then $\nu^*_{\mathcal{H}} \leq (r+1)/2 + O(1/r)$.*

Acknowledgment. We thank the anonymous referees for their helpful comments.

REFERENCES

[B89] C. Berge, *Hypergraphs*, North-Holland, Amsterdam, 1989.

[B86] B. Bollobás, *Combinatorics*, Cambridge University Press, Cambridge, UK, 1986.

[C83] V. Chvátal, *Linear Programming*, W.H. Freeman, New York, 1983.

[C93] H. Cohen, *Quorum Systems: Domination, Fault-Tolerance, Balancing*, MS Thesis, The Weizmann Institute of Science, Rehovot, Israel, 1993.

[EL74] P. Erdős and L. Lovász, *Problems and results on* 3*-chromatic hypergraphs and some related questions*, in Infinite and Finite Sets, Colloq. Math. Soc. János Bolyai 10, North-Holland, Amsterdam, 1974, pp. 609–627.

[F01] J. Farkas, *Uber die Theorie der Einfachen Ungleichungen*, J. Reine Angew. Math., 124 (1901), pp. 1–27.

[F81] Z. Füredi, *Maximum degree and fractional matchings in uniform hypergraphs*, Combinatorica, 1 (1981), pp. 155–162.

[GB85] H. Garcia-Molina and D. Barbara, *How to assign votes in a distributed system*, J. Assoc. Comput. Mach., 32 (1985), pp. 841–860.

[H86] M. Hall, *Combinatorial Theory*, John Wiley, New York, 1986.

[H84] M. P. Herlihy, *Replication methods for abstract data types*, Ph.D. thesis, MIT, Cambridge, MA, 1984.

[IK90] T. Ibaraki and T. Kameda, *Theory of coteries*, Tech. Rep. CSS/LCCR TR90-09, Simon Fraser University, Burnaby, BC, Canada, 1990.

[L73] L. Lovász, *Coverings and colorings of hypergraphs*, in Proc. 4th Southeastern Conf. on Combinatorics, Graph Theory and Computing, Florida Atlantic University, Utilitas Mathematica, Winnipeg, Canada, 1973, pp. 47–56.

[L75] L. Lovász, *On the minimax theorems of combinatorics*, Mat. Lapok, 26 (1975), pp. 209–264 (in Hungarian).

[MV88] S. J. Mullender and P.M.B. Vitányi, *Distributed match-making*, Algorithmica, 3 (1988), pp. 367–391.

[Os85] A. Ostmann, *Decisions by players of comparable strength*, Z. Nationalökonom., 45 (1985), pp. 267–284.

[Ow82] G. Owen, *Game Theory*, Academic Press, Boston, MA, 1982.

[PW93] D. Peleg and A. Wool, *The availability of quorum systems*, Tech. Rep. CS93-17, The Weizmann Institute of Science, Rehovot, Israel, 1993.

[P70] J. Pelikán, *Properties of balanced incomplete block designs*, in Combinatorial Theory and Its Applications, Colloq. Math. Soc. János Bolyai 4, North-Holland, Amsterdam, 1970, pp. 869–889.

[R86] M. Raynal, *Algorithms for mutual exclusion*, MIT Press, Cambridge, MA, 1986.

[T85] Zs. Tuza, *Critical hypergraphs and intersecting set-pair systems*, J. Combin. Theory Ser. B, 39 (1985), pp. 134–145.

SIAM J. DISCRETE MATH.
Vol. 10, No. 2, pp. 246–269, May 1997

007

IMAGES AND PREIMAGES IN RANDOM MAPPINGS[*]

MICHAEL DRMOTA[†] AND MICHÈLE SORIA[‡]

Abstract. We present a general theorem that can be used to identify the limiting distribution for a class of combinatorial schemata. For example, many parameters in random mappings can be covered in this way. In particular, we can derive the limiting distribution of those points with a given number of total predecessors.

Key words. random mappings, combinatorial constructions, limiting distributions

AMS subject classifications. 05A16, 60F

PII. S0895480194268421

1. Introduction.

By a random mapping $\varphi \in \mathcal{F}_n \subseteq \mathcal{F} = \bigcup_{n \geq 0} \mathcal{F}_n$ we mean an arbitrary mapping $\varphi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ such that every mapping has equal probability n^{-n}. The main purpose of this paper is to obtain limit theorems, when n tends to infinity, for special parameters in random mappings, e.g., for the number of image points. Since every random mapping $\varphi \in \mathcal{F}_n$ has equal probability it suffices to count the number of random mappings $\varphi \in \mathcal{F}_n$ satisfying a special property, e.g., that the number of image points equals k. By dividing this number by n^n we get the probability of interest. In order to get the limit distribution for $n \to \infty$ it is not necessary to know the exact value. We just have to evaluate these numbers asymptotically. We shall show that this can be done by a singularity analysis of a proper bivariate generating function.

It should be noted that some of our limit distributions on random mappings are well known (compare with [4, 16]). But our main goal is to provide a general method to derive such limit theorems. In particular, we use bivariate generating functions and singularity analysis. Especially we are able to characterize the (up to now unknown) limit distribution of the number of those points with a fixed number of total predecessors. It is a Gaussian distribution.

Our basic combinatorial concept is that of labelled combinatorial constructions and the relation to exponential generating functions. A big advantage of such combinatorial constructions is that we can mark a parameter in the constructions which directly leads to a bivariate generating function for the number of objects according to their size and the value of the parameter of interest.

2. Marking in random mappings.

Every mapping $\varphi \in \mathcal{F}_n$ can be identified with its functional graph G_φ where $V(G_\varphi) = \{1, \ldots, n\}$ and $E(G_\varphi) = \{(i, \varphi(i)) \,|\, 1 \leq i \leq n\}$. It is obvious that each component of G_φ consists of a cycle (at least of a loop), and every cyclic point is the root of (labelled) tree (see Figure 1).

Hence we can interpret a mapping $\varphi \in \mathcal{F}$ as a set of cycles of trees. Furthermore, since there is no restriction on their structure, the trees (usually known as Cayley trees) can be recursively described as a root followed by a set of trees:

$$\mathcal{F} = \texttt{set}(\texttt{cycle}(\mathcal{T})), \tag{1}$$

$$\mathcal{T} = \circ \cdot \texttt{set}(\mathcal{T}). \tag{2}$$

[*] Received by the editors May 25, 1994; accepted for publication (in revised form) April 29, 1996. http://www.siam.org/journals/sidma/10-2/26842.html

[†] Department of Discrete Mathematics, Technical University of Vienna, Wiedner Hauptstrasse 8–10, A-1040 Vienna, Austria (michael.drmota@tuwien.ac.at).

[‡] LITP, University Paris 6, Place Jussieu, F-75000 Paris, France (soria@litp.ibp.fr).

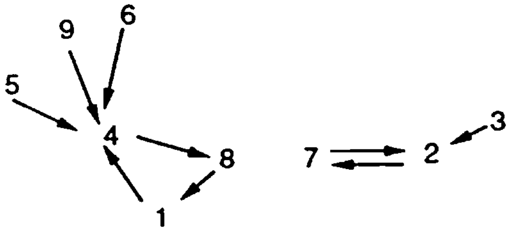


FIG. 1.

Both structures $\mathcal{F}$ and $\mathcal{T}$ fit into the concept of (labelled) combinatorial structures synthesized by Flajolet [11] (see also [17]). Let us give a short description of such structures.

Let C be a combinatorial structure of (labelled) elements, $|c|$ denote the size of $c \in C$, and $c_n = |\{c \in C \mid |c| = n\}|$ denote the number of elements of size n. (Labelled means that there are always $n!$ "isomorphic" elements $c \in C$ of size n which differ by their labels $1, \ldots, n$.) Furthermore we associate a (labelled) combinatorial structure C with the (exponential) generating function

$$\hat{c}(x) = \sum_{c \in C} \frac{x^{|c|}}{(|c|)!} = \sum_{n \geq 0} \frac{c_n}{n!} x^n. \tag{3}$$

The advantage of these generating functions is that there is a correspondence between special combinatorial constructions and special operations with the corresponding generating functions. For example, if the (labelled) combinatorial structure C is the product $C_1 \cdot C_2$, then

$$\hat{c}(x) = \hat{c}_1(x)\hat{c}_2(x). \tag{4}$$

Note that $C_1 \cdot C_2$ is not the set theoretic cartesian product because you have to transform the labelling $\{1, \ldots, k\}$ of $c_1 \in C_1$ and the labelling $\{1, \ldots, n-k\}$ of $c_2 \in C_2$ to a labelling $\{1, \ldots, n\}$ of $c \in C_1 \cdot C_2$. Since there are $k!$ "isomorphic" elements in C_1 and $(n-k)!$ "isomorphic" elements in C_2 we have

$$\frac{c_n}{n!} = \sum_{k=0}^{n} \frac{c_{1,k}}{k!} \frac{c_{2,n-k}}{(n-k)!}$$

according to (4). Hence, if $C = \texttt{set}(C_1)$ then we get

$$\hat{c}(x) = \sum_{m \geq 0} \frac{1}{m!} \hat{c}_1^m(x) = e^{\hat{c}_1(x)}, \tag{5}$$

and if $C = \texttt{cycle}(C_1)$ then we have

$$\hat{c}(x) = \sum_{m \geq 1} \frac{1}{m} \hat{c}_1^m(x) = \log \frac{1}{1 - \hat{c}_1(x)}. \tag{6}$$

Applying this concept to random mappings, the (exponential) generating function

$$\hat{f}(x) = \sum_{n \geq 0} \frac{n^n}{n!} x^n \tag{7}$$

satisfies

$$\hat{f}(x) = \exp\left(\log \frac{1}{1-\hat{t}(x)}\right) = \frac{1}{1-\hat{t}(x)}, \tag{8}$$

where $\hat{t}(x)$, the generating function for Cayley trees, is given by

$$\hat{t}(x) = xe^{\hat{t}(x)}. \tag{9}$$

A big advantage of such combinatorial constructions is that we can formally mark a parameter in the constructions by a symbol like $[u]$. And this marking directly leads to bivariate generating function for the number of objects according to their size and the value of the parameter of interest.

For example, if we are interested in the number of trees in graphs of random mappings we have to mark the trees in the combinatorial construction

$$\mathcal{F} = \texttt{set}(\texttt{cycle}([u]\mathcal{T})).$$

Formally this leads to

$$\hat{f}(x,u) = \exp\left(\log \frac{1}{1-u\hat{t}(x)}\right) = \frac{1}{1-u\hat{t}(x)},$$

which is exactly the generating function $\hat{f}(x,u) = \sum f_{nk} \frac{x^n}{n!} u^k$ of the numbers f_{nk} of random mappings with k trees in the graph representation. (Note that the number of trees is exactly the number of cyclic points.)

Or if we are interested in the number of components, we have to mark

$$\mathcal{F} = \texttt{set}([u]\texttt{cycle}(\mathcal{T}))$$

and get

$$\hat{f}(x,u) = \exp\left(u \log \frac{1}{1-\hat{t}(x)}\right) = \frac{1}{(1-\hat{t}(x))^u}.$$

Next we will use this marking method to describe special parameters related to image and preimage points.

Points at distance d to a cycle. First we will discuss preimages of cyclic points. For this purpose let Y_φ denote the cyclic points of a random mapping $\varphi \in \mathcal{F}$ and $d \geq 1$. Specifically, we are interested in $\varphi^{-d}(Y_\varphi) \setminus \varphi^{-d+1}(Y_\varphi)$, i.e., noncyclic points at distance d to the cyclic points.

Lemma 1. *Let*

$$A_0(x,u) = \frac{1}{1-u}$$

and

$$A_{d+1}(x,u) = A_d(x, xe^u)$$

for $d \geq 1$. Then

$$\hat{f}(x,u) = A_d(x, u\hat{t}(x)) \tag{10}$$

is the (exponential) generating function of random mappings where points contained in $\varphi^{-d}(Y_\varphi) \setminus \varphi^{-d+1}(Y_\varphi)$ *are marked.*

Proof. Let $\hat{t}_d(x,u)$ denote the (exponential) generating function of labelled rooted trees where nodes of distance $d \geq 0$ from the root are marked. Obviously we have

$$\begin{aligned} \hat{t}_0(x,u) &= u\hat{t}(x) \qquad \text{and} \\ \hat{t}_{d+1}(x,u) &= xe^{\hat{t}_d(x,u)} \quad \text{for } d \geq 0, \end{aligned}$$

which directly leads to

$$\hat{f}(x,u) = \frac{1}{1-\hat{t}_d(x,u)} = A_d(x, u\hat{t}(x)). \qquad \square$$

Points with in-degree r. Another interesting parameter is the number of points ν with $|\varphi^{-1}(\{\nu\})| = r$, where $r \geq 0$ is a fixed integer.

Lemma 2. *Let* $\hat{p}_r(x,u)$ *denote the solution of*

$$\hat{p}_r(x,u) = xe^{\hat{p}_r(x,u)} + (u-1)x\frac{\hat{p}_r(x,u)^r}{r!}. \tag{11}$$

Then

$$\hat{f}(x,u) = \frac{1}{1-\left(xe^{\hat{p}_r(x,u)} + (u-1)x\frac{\hat{p}_r(x,u)^{r-1}}{(r-1)!}\right)} \tag{12}$$

is the (exponential) generating function of random mappings where points ν *with* $|\varphi^{-1}(\{\nu\})| = r$ *are marked.*

Proof. According to the recursive structure of Cayley trees $\mathcal{T} = \circ \cdot \mathtt{set}(\mathcal{T})$, the nodes with in-degree r are those followed by r subtrees. Hence the bivariate generating function for trees with variable u marking nodes with in-degree r satisfies

$$\begin{aligned} \hat{p}_r(x,u) &= x\sum_{m\neq r}\frac{\hat{p}_r(x,u)^m}{m!} + ux\frac{\hat{p}_r(x,u)^r}{r!} \\ &= xe^{\hat{p}_r(x,u)} + (u-1)x\frac{\hat{p}_r(x,u)^r}{r!}. \end{aligned}$$

Now a cyclic point in the functional graph of a random mapping has in-degree r if and only if it has in-degree $r-1$ in the corresponding trees. This proves (12). $\square$

Notice that the expression of the bivariate generating function is simpler if we neglect the edges between cyclic points (i.e., cyclic points are marked if they have in-degree $r+1$, and noncyclic points are marked if they have in-degree r). Actually, we then consider sequences of Cayley trees instead of random mappings.

Lemma 2′. *The (exponential) generating function of sequences of Cayley trees, where marked nodes are those with in-degree* r*, is*

$$\hat{f}(x,u) = \frac{1}{1-\hat{p}_r(x,u)}\,, \tag{13}$$

where $\hat{p}_r(x,u)$ *is the same as in Lemma* 2.

Points with r*-antecedents.* Finally, we want to count those points where the total number of preimages equals $r \geq 0$.

Lemma 3. *Let $\hat{a}_r(x,u)$ denote the solution of*

$$\hat{a}_r(x,u) = xe^{\hat{a}_r(x,u)} + (u-1)t_r x^r, \tag{14}$$

where $t_r = \frac{r^{r-1}}{r!}$. Then

$$\hat{f}(x,u) = \frac{1}{1-\hat{a}_r(x,u)+(u-1)t_r x^r} \exp\left(\frac{x^r}{r}\sum_{m=0}^{r}\frac{r^{r-m}}{(r-m)!}(u^m-1)\right) \tag{15}$$

is the (exponential) generating function of random mappings where points ν with

$$\left|\bigcup_{d\ge 0}\varphi^{-d}(\{\nu\})\right| = r$$

are marked.

Proof. As in the proof of Lemma 2 we first mark the nodes with the total number of preimages r in Cayley trees: a node is marked if and only if it is the root of a tree of total size r (the root is considered to be its own preimage). Hence we get (14), where t_r is the coefficient of x^r in the series expansion

$$\hat{t}(x) = \sum_{n\ge 0} t_n x^n.$$

Since $\hat{t}(x)$ satisfies the functional equation (9), using Lagrange's inversion theorem we get

$$t_n = \frac{1}{n}[y^{n-1}]e^{yn} = \frac{n^{n-1}}{n!}.$$

For cyclic points in a random mapping all points in the corresponding component (of the functional graph) are preimages. Hence for a component with m cyclic points, the bivariate generating function, with u marking the number of points having r preimages, is

$$\frac{\left(xe^{\hat{a}_r(x,u)}\right)^m}{m} + \frac{1}{m}(u^m-1)x^r[v^r]\hat{t}(v)^m,$$

where

$$[v^r]\hat{t}(v)^m = \frac{m}{r}[y^{r-m}]e^{yr} = \frac{mr^{r-m}}{r(r-m)!}$$

is the number of forests composed with m components and of total size r. This directly gives (15). □

dth iterate points. It is also interesting to consider $\varphi^d(\{1,\dots,n\})$, the dth iterate image points.

Lemma 4. *Set*

$$h_0(x) = 0 \qquad \textit{and}$$
$$h_{i+1}(x) = xe^{h_i(x)} \quad \textit{for } i \ge 0,$$

and let $\hat{i}_d(x,u)$ be the solution of

$$\hat{i}_d(x,u) = xue^{\hat{i}_d(x,u)} - (u-1)h_d(x). \tag{16}$$

Then

$$\hat{f}(x,u) = \frac{1}{1-\hat{i}_d(x,u)-(u-1)h_d(x)} \tag{17}$$

is the (exponential) generating function of random mappings where points $\nu \in \varphi^d(\{1,\ldots,n\})$ are marked.

Proof. Clearly, $h_d(x)$ is the (exponential) generating function of Cayley trees with height $< d$. In Cayley trees, the dth iterate image points are points at distance $\geq d$ from a leaf. Hence, $\hat{i}_d(x,u)$, the bivariate generating function of trees where nodes having a leaf at distance $\geq d$ are marked, satisfies (16). For random mappings, since all cyclic points are dth iterate image points we get

$$\hat{f}(x,u) = \frac{1}{1-uxe^{\hat{i}_d(x,u)}},$$

which leads to (17). □

Here again, as in the case of points with in-degree r, the expression of the bivariate generating function is simpler if we neglect the edges between cyclic points.

LEMMA 4′. *The (exponential) generating function of random mappings, where the marked points are those at distance $\geq d$ from a leaf of their own subtree, is*

$$\hat{f}(x,u) = \frac{1}{1-\hat{i}_d(x,u)}, \tag{18}$$

where $\hat{i}_d(x,u)$ is the same as in Lemma 4.

Direct dth iterate points. The most difficult example (from the combinatorial point of view) is the case of dth iterate image points of nonimage points $\varphi^d(\{1,\ldots,n\} \setminus \varphi(\{1,\ldots,n\}))$. In other words we will count those nodes that are connected by a (directed) path of length d to a nonimage point. Nevertheless, there is a rather easy subcase where edges between cyclic points are neglected; i.e., the problem can be reduced to a problem inside trees. Although this is only a very small change, it will turn out that the corresponding limiting distributions differ. (And it will also be the case for Lemmas 2 and 4 versus Lemmas 2′ and 4′.)

LEMMA 5. *Set*

$$\begin{aligned} c_0(x,y) &= x \qquad \textit{and} \\ c_{i+1}(x,y) &= x\left(e^y - e^{y-c_i(x,y)}\right) \quad \textit{for } i \geq 0, \end{aligned}$$

and let $\hat{l}_d(x,u)$ be the solution of

$$\hat{l}_d(x,u) = xe^{\hat{l}_d(x,u)} + (u-1)c_d(x,\hat{l}_d(x,u)). \tag{19}$$

Then

$$\tilde{f}(x,u) = \exp\left(\log\frac{1}{1-\hat{l}_d(x,u)}\right) = \frac{1}{1-\hat{l}_d(x,u)} \tag{20}$$

is the (exponential) generating function of random mappings, where marked points are those connected to a leaf by a path of length d which does not contain cyclic edges. (Note that the root of a tree of size 1 *is also a leaf.)*

Proof. Let $\hat{l}_d(x,u)$ denote the generating function of Cayley trees where nodes having a leaf at distance d are marked. The generating function where leaves are marked is $\hat{l}_0(x,u) = ux + x(e^{l_0(x,u)} - 1)$ and, inductively, $\hat{l}_d(x,u) = y_d(x,u) + xe^{\hat{l}_d(x,u)-y_{d-1}(x,u)}$, where $y_d(x,u)$ is the generating function for trees with a leaf at distance d to the root: $y_d(x,u) = uc_d(x,\hat{l}_d(x,u))$, and for $i = 1,\ldots,(d-1)$, $y_i(x,u) = c_i(x,\hat{l}_d(x,u))$. Since $xe^{\hat{l}_d(x,u)-y_{d-1}(x,u)} = xe^{\hat{l}_d(x,u)} - c_d(x,\hat{l}_d(x,u))$ represents the trees such that the root is not at distance d to a leaf, $\hat{l}_d(x,u)$ satisfies (19) and hence (20). □

LEMMA 5′. *Let $c_i(x,y)$ and $\hat{l}_d(x,u)$ be defined as in Lemma* 5 *and $\hat{f}_d(x,u)$ be the (exponential) generating function of random mappings where points $\nu \in \varphi^d(\{1,\ldots,n\}\setminus \varphi(\{1,\ldots,n\}))$ are marked.*

For $d=0$ and $d=1$ we have

$$\hat{f}_0(x,u) = \frac{1}{1-xe^{\hat{l}_0(x,u)}} \quad \textit{and} \quad \hat{f}_1(x,u) = \frac{1}{1-\hat{l}_1(x,u)}. \tag{21}$$

For $d=2$ set $y_1(x,u) = c_1(x,\hat{l}_2(x,u))$, $y_2(x,u) = uc_2(x,\hat{l}_2(x,u))$, and

$$y_{12}(x,u) = ux\left(e^{\hat{l}_2(x,u)} - e^{\hat{l}_2(x,u)-x} - e^{\hat{l}_2(x,u)-y_1(x,u)} + e^{\hat{l}_2(x,u)-y_1(x,u)-x}\right).$$

Then

$$\hat{f}_2(x,u) = \frac{1}{1-\hat{l}_2(x,u)-(u-1)\left(y_1(x,u)(1-y_2(x,u)) - y_{12}(x,u)(1-\hat{l}_2(x,u))\right)}. \tag{22}$$

Proof. The results for $\hat{f}_0(x,u)$ and $\hat{f}_1(x,u)$ are obvious: for $d=0$, cyclic points are not leaves of random mappings and, for $d=1$, edges inside cycles are of no importance. For $d=2$, the situation gets more delicate: in addition to the interpretations of $y_1(x,u)$ and $y_2(x,u)$ observe that $y_{12}(x,u)$ corresponds to those trees having both a leaf at distance 1 to the root and a leaf at distance 2 to the root. (For the sake of shortness we will use the terms y_1-tree (respectively, y_2-tree) for a tree with a leaf at distance 1 (respectively, 2) to the root.) Set

$$\begin{aligned} w &= u(y_1 - y_{12}) + (\hat{l}_2 - y_1 - y_2 + y_{12}) \qquad \text{and} \\ s &= uy_{12} + (y_2 - y_{12}). \end{aligned}$$

Then w corresponds to also marking y_1-trees that are not y_2-trees. In the same way, s corresponds to twice marking y_2-trees that are y_1-trees too. We will show (at the end of the proof) that the generating function

$$\frac{w^m}{m} + \frac{y_2^m}{m} + \frac{1}{m}\sum_{k=1}^{m-1}\sum_{l=1}^{k} A_{mkl}s^l(\hat{l}_2 - y_2)^l w^{m-k-l} y_2^{k-l} \tag{23}$$

corresponds to a cycle of m trees where all nodes having a leaf at distance 2 are marked (A_{mkl}, see below, counts the number of cycles of length m containing k y_2-trees, $(k-l)$

of which are followed by a y_2-tree). Since

$$\sum_{m\ge 1}\frac{1}{m}\sum_{k=1}^{m-1}\sum_{l=1}^{k} A_{mkl}s^l(\hat{l}_2-y_2)^l w^{m-k-l}y_2^{k-l}$$
$$= \log\frac{1}{1-y_2-\frac{s(\hat{l}_2-y_2)}{1-w}} - \log\frac{1}{1-y_2},$$

we immediately get (22).

Therefore, it remains to interpret (23). The problem on a cycle is that a y_1-tree forces an additional mark at the next root on the cycle if and only if this next root is not marked, i.e., the corresponding tree is not a y_2-tree. For example, if a cycle of length m contains no y_2-tree, then it is immediately clear that $\frac{1}{m}w^m$ is the correct corresponding function, whereas the case of a cycle containing only y_2-trees the generating function of interest is $\frac{1}{m}y_2^m$. For the remaining cases consider a cycle containing exactly k $(0<k<m)$ y_2-trees such that l $(0<l\le k)$ of these trees are followed by a tree that is not a y_2-tree. Note that

$$A_{mkl} = \frac{m}{l}\binom{m-k-1}{l-1}\binom{k-1}{l-1} \tag{24}$$

is the number of such arrangements on a (labelled) cycle of length m. In any of these cases the corresponding generating function is $\frac{1}{m}y_2^{k-l}s^l w^{m-k-l}(\hat{l}_2-y_2)^l$. This proves (23). □

3. General theorems. Let $c(x)=\sum c_n x^n$ be the generating function of a combinatorial structure and $c(x,u)=\sum c_{nk}x^n u^k$ the bivariate generating function where a parameter of interest has been marked, i.e., $c(x,1)=c(x)$. Now we will be interested in the asymptotic distribution of this parameter in the system of combinatorial objects of size n when n tends to infinity. For this purpose we introduce a sequence of random variables X_n, $n\ge 1$, defined by

$$\mathbf{Pr}[X_n=k]=\frac{c_{nk}}{c_n}=\frac{[x^n u^k]c(x,u)}{[x^n]c(x,1)},$$

where $\mathbf{Pr}$ denotes probability. Now the above problem reduces to finding the limiting distribution of X_n.

An important analytic schema, related to combinatorial constructions "sequence" or "set of cycles," is

$$c(x,u)=\frac{1}{1-a(x,u)}\ .$$

The next three theorems study this schema when $a(x,u)$ has an algebraic singularity $\rho(u)$ of square-root type such that $a(\rho(1),1)=1$. According to further analytic properties of $a(x,u)$, the limiting distribution of X_n is shown to be either Gaussian, Rayleigh, or the convolution of Gaussian and Rayleigh, and in each case the global limit result (convergence of distribution functions) is accompanied by a local limit result (convergence of densities).

Let us first state precisely the general form of the analytic schemas under consideration.

Hypothesis [H]. Let $c(x,u) = \sum_{n,k} c_{nk} x^n u^k$ be a power series in two variables with nonnegative coefficients $c_{nk} \geq 0$ such that $c(x,1)$ has a radius of convergence of $\rho > 0$.

We suppose that $c(x,u)$ expresses as $c(x,u) = 1/d(x,u)$, where $d(x,u)$ has the local representation

$$d(x,u) = g(x,u) + h(x,u)\sqrt{1 - \frac{x}{\rho(u)}} \tag{25}$$

for $|u-1| < \varepsilon$ and $|x - \rho(u)| < \varepsilon$, $\arg(x - \rho(u)) \neq 0$, where $\varepsilon > 0$ is some fixed real number, and $g(x,u)$, $h(x,u)$, and $\rho(u)$ are analytic functions.

Furthermore, these functions satisfy $g(\rho,1) = 0$, $h(\rho,1) > 0$, and $\rho(1) = \rho$.

In addition, $x = \rho(u)$ is the only singularity on the circle of convergence $|x| = |\rho(u)|$ for $|u-1| < \varepsilon$ and $d(x,u)$, respectively $c(x,u)$, can be analytically continued to a region $|x| < \rho + \delta$, $|u| < 1 + \delta$, $|u-1| > \frac{\varepsilon}{2}$ for some $\delta > 0$.

Under this hypothesis, the limiting distribution of X_n in $c(x,u)$ depends on $\rho'(1)$ and $g_u(\rho(1),1)$, as stated in the following three theorems.

THEOREM 1. *Let $c(x,u)$ be a bivariate generating function satisfying* [H]. *If $\rho(u) = \rho = \text{const}$ for $|u-1| < \varepsilon$ and $g_u(\rho,1) < 0$, then the sequence of random variables X_n defined by*

$$\mathbf{Pr}[X_n = k] = \frac{[x^n u^k]c(x,u)}{[x^n]c(x,1)} \tag{26}$$

has a Rayleigh limiting distribution; *i.e.,*

$$\frac{X_n}{\sqrt{n}} \xrightarrow{d} \mathcal{R}(\lambda), \tag{27}$$

where $\lambda = \frac{h(\rho,1)^2}{2g_u(\rho,1)^2}$ and $\mathcal{R}(\lambda)$ has density $\lambda x \exp\left(-\frac{\lambda}{2}x^2\right)$ for $x \geq 0$. Expected value and variance are given by

$$\mathbf{E}X_n = \sqrt{\frac{\pi}{2\lambda}}\sqrt{n} + \mathcal{O}(1) \quad \textit{and} \quad \mathbf{V}X_n = \left(2 - \frac{\pi}{2}\right)\frac{n}{\lambda} + \mathcal{O}(\sqrt{n}). \tag{28}$$

Moreover, we have the local law

$$\mathbf{Pr}[X_n = k] = \frac{\lambda k}{n}\exp\left(-\frac{\lambda k^2}{2n}\right) + \mathcal{O}((k+1)n^{-\frac{3}{2}}) + \mathcal{O}(n^{-1}) \tag{29}$$

uniformly for all $k \geq 0$.

THEOREM 2. *Let $c(x,u)$ be a bivariate generating function satisfying* [H]. *If $\rho'(1) \neq 0$ and $\alpha = \frac{\partial}{\partial u} g(\rho(u),u)|_{u=1} = 0$, then X_n has a Gaussian limiting distribution*; *i.e.,*

$$\frac{X_n - \mu n}{\sqrt{\sigma^2 n}} \xrightarrow{d} \mathcal{N}(0,1), \tag{30}$$

where $\mu = -\rho'(1)/\rho$ and $\sigma^2 = \mu^2 + \mu - \rho''(1)/\rho$. Expected value and variance are given by

$$\mathbf{E}X_n = \mu n + \mathcal{O}(1) \quad \textit{and} \quad \mathbf{V}X_n = \sigma^2 n + \mathcal{O}(\sqrt{n}). \tag{31}$$

Furthermore, there is local law of the form

$$\mathbf{Pr}[X_n = k] = \frac{1}{\sqrt{2\pi\sigma^2 n}} \exp\left(-\frac{(k-\mu n)^2}{2\sigma^2 n}\right) + \mathcal{O}(n^{-\frac{3}{4}}) \tag{32}$$

uniformly for all $k \geq 0$.

Theorem 3. *Let $c(x,u)$ be a bivariate generating function satisfying* [H]*. If $\rho'(1) \neq 0$ and $\alpha = \frac{\partial}{\partial u} g(\rho(u),u)|_{u=1} < 0$, then the limiting distribution of X_n is the convolution of a Gaussian and a Rayleigh distribution; i.e.,*

$$\frac{X_n - \mu n}{\sqrt{\sigma^2 n}} \xrightarrow{d} \mathcal{N}(0,1) * \mathcal{R}(\lambda), \tag{33}$$

where $\lambda = \frac{h(\rho,1)^2\sigma^2}{2\alpha^2}$ and μ and σ^2 are defined as in Theorem 2. Expected value and variance are given by

$$\mathbf{E}X_n = \mu n - \frac{\sqrt{\pi}\alpha}{h(\rho,1)}\sqrt{n} + \mathcal{O}(1) \quad \text{and} \quad \mathbf{V}X_n = \left(\sigma^2 + \frac{(4-\pi)\alpha^2}{h(\rho,1)^2}\right) n + \mathcal{O}(\sqrt{n}) \tag{34}$$

and there is local law of the form

$$\begin{aligned}\mathbf{Pr}[X_n = k] = {}& \frac{\lambda}{1+\lambda}\frac{1}{\sqrt{2\pi\sigma^2 n}} \exp\left(-\frac{(k-\mu n)^2}{\sigma^2 n}\right) \\ & + \frac{\lambda}{(1+\lambda)^{\frac{3}{2}}}\frac{k-\mu n}{\sigma^2 n} \exp\left(-\frac{\lambda}{1+\lambda}\frac{(k-\mu n)^2}{\sigma^2 n}\right) \Phi\left(\frac{k-\mu n}{\sqrt{(1+\lambda)\sigma^2 n}}\right) \\ & + \mathcal{O}(n^{-\frac{3}{4}})\end{aligned} \tag{35}$$

uniformly for all $k \geq 0$, where

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{t^2}{2}\right) dt. \tag{36}$$

(If $\alpha > 0$ then the corresponding Rayleigh distribution is supported on the negative real axis and a similar local law holds.)

Remark. It should be noticed that condition $g(\rho,1) = 0$ in the theorems is not a real restriction. In fact, it turns out that the case $g(\rho,1) = 0$ is the most difficult one, and the limiting distribution for other cases can be found also.

If $g(\rho,1) > 0$ then $c(x,u)$ has a local representation of the form

$$c(x,u) = \frac{1}{g(x,u) + h(x,u)\sqrt{1 - x/\rho(u)}} = G(x,u) - H(x,u)\sqrt{1 - \frac{x}{\rho(u)}}. \tag{37}$$

On the other hand, if $g(\rho,1) < 0$, the algebraic singularity is not the dominating one. Here $d(\overline{\rho},1) = 0$ for $\overline{\rho} < \rho$ and (usually) $d_x(\overline{\rho},1) \neq 0$. Hence, by the Weierstrass preparation theorem, $d(x,u)$ has a local representation of the form $d(x,u) = D(x,u)(1 - x/\overline{\rho}(u))$, where $D(x,u)$ and $\overline{\rho}(u)$ are analytic functions satisfying $D(\overline{\rho},1) \neq 0$, $\overline{\rho}(1) = \overline{\rho}$, and $\overline{\rho}'(1) \neq 0$. Thus

$$c(x,u) = \frac{1/D(x,u)}{1 - \frac{x}{\overline{\rho}(u)}}. \tag{38}$$

In both cases, (37) and (38), we can apply Bender's theorem [5] (compare also with [6] and [7]) to get asymptotic normality if $\rho'(1) < 0$. (Evaluating the expected value shows that $\rho'(1)$ cannot be positive.)

When $\rho(u) = \text{const}$ (see also [14] for this case), the limiting distribution is Gaussian for $g(\rho, 1) > 0$ and discrete for $g(\rho, 1) < 0$ (for example, there is a derivated geometric law for the schema $c(x, u) = (1 - ua(x))^{-1}$).

Finally, we want to remark that the assumption $\rho(u) = \text{const}$ in Theorem 1 can be weakened to $\rho'(1) = 0$. However, the proof would be a little bit more complicated. Furthermore, no example is known where $\rho'(1) = 0$ but $\rho(u) \neq \text{const}$.

Before proving Theorems 1, 2, 3 (see section 5) we want to discuss why such theorems have some importance in relation to random mappings.

4. Applications to random mappings. In this section we apply our theorems to obtain the limiting distributions for various parameters of random mappings. It should be noted that some of the obtained results are known, but our intention is to provide all the results by applying only one general principle. The underlying point is that the combinatorial specification of random mappings out of Cayley trees, together with the analytic form of the Cayley trees series, imply that all bivariate generating functions $\hat{f}(x, u)$ constructed in section 2 satisfy Hypothesis [H].

4.1. Analytic frame. The basic property is that solutions of functional equations usually have algebraic singularities of square-root type.

PROPOSITION 1. *Let $F(a, x, u)$ be a power series on three variables with nonnegative coefficients and $F(0, 0, 0) = 0$. Suppose that the system of equations*

$$a_0 = F(a_0, x_0, 1), \tag{39}$$

$$1 = F_a(a_0, x_0, 1) \tag{40}$$

has positive solutions $a_0 > 0$, $x_0 > 0$ (which are supposed to be minimal) such that $(a_0, x_0, 1)$ is contained in the region of convergence of $F(a, x, u)$ and that

$$F_x(a_0, x_0, 1) \neq 0 \quad \text{and} \quad F_{aa}(a_0, x_0, 1) \neq 0. \tag{41}$$

Then there exists a unique analytic solution $a = a(x, u) = \sum_{nk} a_{nk} x^n u^k$ of

$$a = F(a, x, u) \tag{42}$$

with nonnegative coefficients $a_{nk} \geq 0$ and $a_{00} = 0$ such that $a(x, u)$ has the local representation

$$a(x, u) = g(x, u) - h(x, u) \sqrt{1 - \frac{x}{\rho(u)}} \tag{43}$$

for $|u - 1| < \varepsilon$ and $|x - \rho(u)| < \varepsilon$, $\arg(x - \rho(u)) \neq 0$, where $g(x, u)$, $h(x, u)$, and $\rho(u)$ are analytic functions that satisfy

$$g(x_0, 1) = a_0, \quad h(x_0, 1) = \sqrt{\frac{2x_0 F_x(a_0, x_0, 1)}{F_{aa}(a_0, x_0, 1)}}, \quad \text{and} \quad \rho(1) = x_0 \tag{44}$$

and $\varepsilon > 0$ is some fixed real number. Furthermore, if there are n_1, n_2, n_3 and $k_1 < k_2 < k_3$ such that $a_{n_1 k_1} a_{n_2 k_2} a_{n_3 k_3} > 0$ and $\gcd(k_3 - k_1, k_2 - k_1) = 1$ and if

$$\gcd\left\{ n - l \,:\, \sum_k a_{nk} > 0 \right\} = 1,$$

where

$$l = \min\left\{ m : \sum_k a_{mk} > 0 \right\},$$

then $x = \rho(u)$ is the only singularity on the circle of convergence $|x| = |\rho(u)|$ for $|u-1| < \varepsilon$, and there exists some $\delta > 0$ such that $a(x,u)$ can be analytically continued in the region $|x| < x_0 + \delta$, $|u| < 1+\delta$, $|u-1| > \frac{\varepsilon}{2}$.

The proof of Proposition 1 is a combination of the implicit function theorem and the Weierstrass preparation theorem (cf. [7, 8]).

Now it is easy to see the connection to random mappings. In any of the above combinatorial constructions the solution $a(x,u)$ (satisfying $a(\frac{1}{e}, 1) = 1$) of a functional equation of the type (42) is used to construct a final generating function that is more or less of the form

$$c(x,u) = \frac{1}{1 - a(x,u)}.$$

Hence, we can directly apply our theorems to obtain the kind of asymptotic distribution we are seeking.

4.2. Distribution of parameters. The examples mentioned in section 2 cover the three types of limiting distributions, Gaussian, Rayleigh, or a convolution of both. It should be noted that in Applications 2, 4, and 5, small structural modifications (neglecting cyclic edges) lead to different limit laws.

For the sake of brevity we will only mention the weak convergence law. However, in all the cases the local law and the asymptotic expansions for mean and variance hold, too.

Application 1 (see [16]). *Let X_n denote the number of noncyclic points at a fixed distance $d > 0$ to a cycle in random mappings of size n. Then*

$$\frac{X_n}{\sqrt{n}} \xrightarrow{d} \mathcal{R}(1). \tag{45}$$

Proof. From Proposition 1 it follows that $\hat{t}(x)$ has a local representation of the kind

$$\hat{t}(x) = a(x) - b(x)\sqrt{1 - ex},$$

where $a(x)$ and $b(x)$ are analytic functions around $x_0 = \frac{1}{e}$ with $a(\frac{1}{e}) = 1$ and $b(\frac{1}{e}) = \sqrt{2}$. Furthermore, we can use the Taylor series expansion of

$$A_d(x,u)^{-1} = \sum_{l,k \geq 0} c_{lk}(u-1)^l \left(x - \frac{1}{e}\right)^k,$$

where $c_{00} = 0$ and $c_{10} = -1$ to see that $A_d(x, u\hat{t}(x,u))$ has a representation of the kind

$$A_d(x, u\hat{t}(x,u))^{-1} = g(x,u) + h(x,u)\sqrt{1 - ex},$$

where $g(\frac{1}{e}, 1) = c_{00} = 0$ and $h(\frac{1}{e}, 1) = -c_{10} b(\frac{1}{e}) = \sqrt{2}$. Hence, we can apply Theorem 1. □

Application 2 (see [4]). *Let $r \geq 0$ be a fixed integer, and let X_n denote the number of points ν with $|\varphi^{-1}(\{\nu\})| = r$ in mappings $\varphi \in \mathcal{F}_n$. Then*

$$\frac{X_n - \mu n}{\sqrt{\sigma^2 n}} \xrightarrow{d} \mathcal{N}(0,1), \tag{46}$$

where $\mu = \frac{1}{er!}$ and $\sigma^2 = \mu + (1+(r-1)^2)\mu^2$.

Proof. Let

$$F(p,x,u) = xe^p + (u-1)x\frac{p^r}{r!}.$$

Then another application of Proposition 1 provides a local representation of

$$\hat{p}_r(x,u) = a(x,u) - b(x,u)\sqrt{1 - \frac{x}{\rho(u)}},$$

where $\hat{p}_r(x,1) = \hat{t}(x)$, and, consequently, $\rho(1) = \frac{1}{e}$, $a(\frac{1}{e},1) = 1$, and $b(\frac{1}{e},1) = \sqrt{2}$. Hence, we obtain

$$\begin{aligned}\hat{f}(x,u) &= \frac{1}{1 - \left(\hat{p}_r(x,u) - (u-1)x\frac{\hat{p}_r(x,u)^r}{r!} + (u-1)x\frac{\hat{p}_r(x,u)^{r-1}}{(r-1)!}\right)} \\ &= \frac{1}{g(x,u) + h(x,u)\sqrt{1 - x/\rho(u)}}\end{aligned}$$

in which $g(\frac{1}{e},1) = 0$, $h(\frac{1}{e},1) = \sqrt{2}$, and

$$\alpha = \frac{\partial}{\partial u} g(\rho(u),u)|_{u=1} = -\frac{\partial}{\partial u} a(\rho(u),u)|_{u=1} + \frac{1}{er!} - \frac{1}{e(r-1)!}.$$

Since $p(u) = a(\rho(u),u) = \hat{p}_r(\rho(u),u)$ satisfies the system of equations

$$\begin{aligned} p(u) &= F(p(u),\rho(u),u), \\ 1 &= F_p(p(u),\rho(u),u), \end{aligned}$$

implicit differentiation gives

$$\begin{aligned} \rho'(1) &= -\frac{F_u(1,\frac{1}{e},1)}{F_x(1,\frac{1}{e},1)} = -\frac{1}{er!}, \\ \rho''(1) &= \frac{1}{F_{pp}F_x^3}(F_x^2(F_{pu}^2 - F_{pp}F_{uu}) + F_u^2(F_{px}^2 - F_{pp}F_{xx}) \\ &\qquad -2F_xF_u(F_{px}F_{pu} - F_{pp}F_{xu})) \\ &= -\frac{(r-1)^2}{e^3(r!)^2}, \\ p'(1) &= \frac{F_{px}F_u - F_{pu}F_x}{F_{pp}F_x} = \frac{1}{er!} - \frac{1}{e(r-1)!}. \end{aligned}$$

Thus we can apply Theorem 2. □

Application 2′. *Let $r \geq 0$ be a fixed integer and let X_n denote the number of nodes with in-degree r in a sequence of Cayley trees of total size n. Then*

$$\frac{X_n - \mu n}{\sqrt{\sigma^2 n}} \xrightarrow{d} \mathcal{N}(0,1) * \mathcal{R}(\lambda),$$

where μ and σ^2 are the same as in Application 2, and $\lambda = \frac{\sigma^2(er!)^2}{(1-r)^2}$. In the special case $r = 1$, the limiting distribution is only Gaussian since $\lambda^{-1} = 0$.

Proof. In this case $\alpha = p'(1)$ is not equal to 0, except for $r = 1$. Hence, the convolution results by Theorem 3. □

Application 3. *Let $r \geq 0$ be a fixed integer, and let X_n denote the number of points ν with*

$$\left| \bigcup_{d \geq 0} \varphi^{-d}(\{\nu\}) \right| = r$$

in mappings $\varphi \in \mathcal{F}_n$. Then

$$\frac{X_n - \mu n}{\sqrt{\sigma^2 n}} \xrightarrow{d} \mathcal{N}(0,1), \tag{47}$$

where $\mu = \frac{r^{r-1}}{r!} e^{-r}$ and $\sigma^2 = \mu - 2r\mu^2$.

Proof. First notice that the analytic factor $\exp(\cdots)$ in (15) has no influence on the parameters of interest α, $\rho'(1)$, and $\rho''(1)$. Therefore, we can neglect it. Hence we can proceed as in the proof on Application 2. Here we have

$$\begin{aligned} \rho'(1) &= -t_r e^{-r-1}, \\ \rho''(1) &= (1+2r) t_r^2 e^{-2r-1}, \\ a'(1) &= t_r e^{-r}. \end{aligned}$$

Consequently, $\alpha = 0$ and we obtain a Gaussian limiting distribution. □

Application 4. *Let $d \geq 0$ be a fixed integer, and let X_n denote the number of points $\nu \in \varphi^d(\{1, \ldots, n\})$ mappings $\varphi \in \mathcal{F}_n$. Then*

$$\frac{X_n - \mu n}{\sqrt{\sigma^2 n}} \xrightarrow{d} \mathcal{N}(0,1), \tag{48}$$

where $\mu = h_d(\frac{1}{e})$ and $\sigma^2 = \frac{2}{e} h_d'(\frac{1}{e})(1 - \mu) - \mu$.

Proof. The proof is almost the same as the proof of Application 2. □

We want to mention that the mean value was already determined in [13].

Application 4$'$. *Let $d \geq 0$ be a fixed integer, and let X_n denote the number of points at distance $\geq d$ from a leaf of their own subtree in random mappings of size n. Then*

$$\frac{X_n - \mu n}{\sqrt{\sigma^2 n}} \xrightarrow{d} \mathcal{N}(0,1) * \mathcal{R}(\lambda), \tag{49}$$

where μ and σ^2 are the same as in Application 4, and $\lambda = \frac{\sigma^2}{h_d^2(\frac{1}{e})}$.

Proof. In this case, $\alpha = h_d(\frac{1}{e}) \neq 0$. Hence the convolution result. □

Application 5. *Let $d \geq 0$ be fixed and X_n denote the number of nodes that are connected to a leaf by a path of length d containing no cyclic edge in random mappings of size n. Then*

$$\frac{X_n - \mu n}{\sqrt{\sigma^2 n}} \xrightarrow{d} \mathcal{N}(0,1) * \mathcal{R}(\lambda), \tag{50}$$

where $\mu = c_d = c_d(\frac{1}{e}, 1)$, $\sigma^2 = c_d(1 - \frac{2}{e}c_{d,x}) + 2c_d c_{d,y} - c_{d,y}^2$, *and* $\lambda^{-1} = (c_d - c_{d,y})^2/\sigma^2$ $(c_{d,x} = \frac{\partial}{\partial x}c_d(\frac{1}{e}, 1)$, $c_{d,y} = \frac{\partial}{\partial y}c_d(\frac{1}{e}, 1))$. *Since* $\lambda^{-1} = 0$ *for* $d = 1$, *the limiting distribution is only Gaussian in this special case.*

Proof. The proof runs along the same lines as the preceding ones. You only have to apply Theorem 3 since $\alpha = c_d - c_{d,y} \neq 0$ for $d \neq 1$. □

APPLICATION 5′. *Let* $d \in \{0, 1, 2\}$ *be fixed and let* X_n *be the number of points* $x \in \varphi^d(\{1, \dots, n\} \setminus \varphi(\{1, \dots, n\}))$ *in mappings* $\varphi \in \mathcal{F}_n$. *Then*

$$\frac{X_n - \mu n}{\sqrt{\sigma^2 n}} \xrightarrow{d} \mathcal{N}(0, 1), \tag{51}$$

where μ *and* σ^2 *are as in Application* 5; *i.e.,*

$$c_d(e^{-1}, 1) = \begin{cases} e^{-1} & \textit{for } d = 0, \\ 1 - e^{-e^{-1}} & \textit{for } d = 1 \\ 1 - e^{-(1-e^{-1/e})} & \textit{for } d = 2, \end{cases}$$

and

$$\sigma^2 = \begin{cases} e^{-1} - 2e^{-2} & \textit{for } d = 0, \\ e^{-e^{-1}}\left(1 - 2e^{-1}\right)\left(1 - e^{-e^{-1}}\right) & \textit{for } d = 1, \\ 2e^{-1+e^{-1/e}-e^{-1}} - e^{-1+e^{-1/e}} & \textit{for } d = 2. \\ \quad -2e^{-2+e^{-1/e}-e^{-1}} - e^{-2+2e^{-1/e}-2e^{-1}} & \\ \quad +2e^{-3+2e^{-1/e}-e^{-1}} & \end{cases}$$

Proof. Especially in the case $d = 2$ you have to calculate α very carefully, but in all the cases $\alpha = 0$. □

Note. In the case $d > 2$, the combinatorial description is much more involved. Nevertheless, it may be conjectured that the limiting distribution is still Gaussian.

5. Proof of the theorems. The proofs of Theorems 1, 2, 3 proceed in the following way. First we derive asymptotic expansions for mean value and variance; then we prove a weak limit theorem using characteristic functions and, finally, we establish the corresponding local limit theorem. This procedure seems to be redundant, and in fact it is. But our aim is not only to prove special theorems but to provide an example for a general method to analyze the asymptotic distribution of a parameter in combinatorial constructions.

5.1. Preliminaries. We first list some useful formulae related to Gaussian and Rayleigh distributions.

LEMMA 6. *Let* γ *be a Hankel contour starting from* $+e^{2\pi i}\infty$, *passing around* 0, *and tending to* $+\infty$. *Then*

$$\frac{1}{2\pi i}\int_\gamma \frac{e^{-z}}{\sqrt{-z} - is}\, dz = \frac{1}{\sqrt{\pi}}\varphi_{\mathcal{R}}(\sqrt{2}s), \tag{52}$$

where

$$\begin{aligned} \varphi_{\mathcal{R}}(t) &= \int_0^\infty e^{itx} x e^{-x^2/2}\, dx \\ &= 1 + ite^{-t^2/2}\left(\sqrt{\frac{\pi}{2}} - i\int_0^t e^{u^2/2}\, du\right) \end{aligned}$$

denotes the characteristic function of the Rayleigh distribution.

Proof. It suffices to compare the Taylor expansion around $s=0$. By the Hankel integral representation of $\Gamma(s)^{-1}$ we get

$$\frac{1}{2\pi i}\int_\gamma \frac{e^{-z}}{\sqrt{-z}-is} = \frac{1}{2\pi i}\int_\gamma \sum_{n\ge 0}(is)^n(-z)^{-\frac{n+1}{2}}e^{-z}\,dz = \sum_{n\ge 0}\frac{(is)^n}{\Gamma\left(\frac{n+1}{2}\right)}.$$

On the other hand we have

$$\begin{aligned}\varphi_{\mathcal{R}}(t) &= \int_0^\infty \sum_{n\ge 0}(it)^n x^{n+1}e^{-x^2/2}\,dx\\ &= \sum_{n\ge 0}(it)^n 2^{\frac{n}{2}}\frac{\Gamma\left(\frac{n}{2}+1\right)}{\Gamma(n+1)}\\ &= \sqrt{\pi}\sum_{n\ge 0}\frac{1}{\Gamma\left(\frac{n+1}{2}\right)}\left(\frac{it}{\sqrt{2}}\right)^n,\end{aligned}$$

where we have used the duplication formula for the Γ-function. □

Lemma 7. *Let γ be as in Lemma* 6. *Then*

$$\frac{1}{2\pi i}\int_\gamma e^{-s\sqrt{-z}-z}\,dz = \frac{s}{2\sqrt{\pi}}e^{-s^2/4} \tag{53}$$

and

$$\frac{1}{2\pi i}\int_\gamma \frac{e^{-z}}{\sqrt{-z}}\,dz = \frac{1}{\sqrt{\pi}}. \tag{54}$$

Proof. (53) and (54) follow immediately from the substitution $z=w^2$. □

Lemma 8. *Let γ be as in Lemma* 6 *and α,β be real constants. Then*

$$\begin{aligned}&\frac{1}{2\pi i}\int_{-\infty}^{\infty}\int_\gamma \frac{e^{i\alpha w-w^2/2-z}}{\sqrt{-z}-i\beta w}\,dz\,dw\\ &\quad= \frac{e^{-\alpha^2/2}}{\sqrt{2}(\frac12+\beta^2)} - \frac{\sqrt{\pi}\alpha\beta}{(\frac12+\beta^2)^{\frac32}}\exp\left(-\frac{\alpha^2}{4(\frac12+\beta^2)}\right)\Phi\left(-\frac{\alpha\beta}{\sqrt{\frac12+\beta^2}}\right).\end{aligned} \tag{55}$$

Proof. Since both sides of (56) can be interpreted as analytic functions in α,β around the real axis, it suffices to prove (56) for the case $\alpha\beta>0$. In this case we can use the substitutions $z=\frac{u^2}{2}$ and $w=v+i\alpha$ to obtain

$$\frac{e^{-\alpha^2/2}}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{e^{-\frac12(u^2+v^2)}}{\frac{1}{\sqrt{2}}u+\beta v+i\alpha\beta}u\,du\,dv.$$

Then we can apply the polar substitution $u=r\cos\varphi$, $v=r\sin\varphi$ to get

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{e^{-\frac12(u^2+v^2)}}{\frac{1}{\sqrt{2}}u+\beta v+i\alpha\beta}u\,du\,dv$$

$$\begin{aligned}
&= \int_0^\infty \int_0^{2\pi} \frac{r^2 e^{\frac{1}{2}r^2}\cos\varphi}{r\left(\frac{1}{\sqrt{2}}\cos\varphi + \beta\sin\varphi\right) + i\alpha\beta}\, d\varphi\, dr \\
&= \int_0^\infty r^2 e^{-\frac{1}{2}r^2} \int_{|z|=1} \frac{\frac{1}{2}(z+z^{-1})}{r\left(\frac{1}{2\sqrt{2}}(z+z^{-1}) + \frac{\beta}{2i}(z-z^{-1})\right) + i\alpha\beta}\,\frac{dz}{iz}\, dr \\
&= \int_0^\infty r^2 e^{-\frac{1}{2}r^2} \frac{\sqrt{2}\pi}{r\left(\frac{1}{2}+\beta^2\right)} \left(1 - \frac{\alpha\beta}{\sqrt{\alpha^2\beta^2 + r^2\left(\frac{1}{2}+\beta^2\right)}}\right) dr \\
&= \frac{\sqrt{2}\pi}{\frac{1}{2}+\beta^2} - \frac{2\pi^{\frac{3}{2}}\alpha\beta}{\left(\frac{1}{2}+\beta^2\right)^{\frac{3}{2}}} \exp\left(\frac{\alpha^2\beta^2}{2\left(\frac{1}{2}+\beta^2\right)}\right) \Phi\left(-\frac{\alpha\beta}{\sqrt{\frac{1}{2}+\beta^2}}\right),
\end{aligned}$$

where the integral $\int_0^{2\pi} \dots d\varphi$ is solved by using the substitution $z = e^{i\varphi}$ and the residue theorem

$$\begin{aligned}
&\int_{|z|=1} \frac{z^2+1}{r(\frac{i}{\sqrt{2}}+\beta)z^2 - 2\alpha\beta z + r(\frac{i}{\sqrt{2}}-\beta)}\,\frac{dz}{z} \\
&\qquad = 2\pi i \left(\frac{-\frac{i}{\sqrt{2}} - \beta}{r(\frac{1}{2}+\beta^2)} + \frac{\frac{i}{\sqrt{2}}\alpha\beta + \beta\sqrt{\alpha^2\beta^2 + r^2(\frac{1}{2}+\beta^2)}}{r(\frac{1}{2}+\beta^2)\sqrt{\alpha^2\beta^2 + r^2(\frac{1}{2}+\beta^2)}}\right) \\
&\qquad = \frac{\sqrt{2}\pi}{r(\frac{1}{2}+\beta^2)} \left(1 - \frac{\alpha\beta}{\sqrt{\alpha^2\beta^2 + r^2(\frac{1}{2}+\beta^2)}}\right).
\end{aligned}$$

The residues have to be calculated for

$$z_1 = 0 \quad \text{and for} \quad z_2 = \frac{\alpha\beta - \sqrt{\alpha^2\beta^2 + r^2(\frac{1}{2}+\beta^2)}}{r(\frac{1}{2}+\beta^2)}.$$

This completes the proof of Lemma 8. □

5.2. Proof of Theorem 1. We first derive asymptotic expansions for mean value and variance. Since

$$c(x,u) = \frac{1}{g(x,u) + h(x,u)\sqrt{1 - x/\rho(u)}}, \tag{56}$$

we get

$$\begin{aligned}
[x^n]c(x,1) &= [x^n]\frac{1}{h(x,1)}\left(1 - \frac{x}{\rho}\right)^{-\frac{1}{2}} + \frac{\rho g_x(\rho,1)}{h(\rho,1)} + \mathcal{O}(\sqrt{1-x/\rho}) \\
&= \frac{\rho^{-n} n^{-\frac{1}{2}}}{h(\rho,1)\sqrt{\pi}}(1 + \mathcal{O}(n^{-1}))
\end{aligned}$$

and

$$\begin{aligned}
[x^n]c_u(x,1) &= [x^n]\left(\frac{-g_u(x,1)}{h(x,1)^2}\left(1 - \frac{x}{\rho}\right)^{-1} + \frac{-h_u(x,1)}{h(x,1)^2}\left(1 - \frac{x}{\rho}\right)^{-\frac{1}{2}}\right) \\
&= \frac{-g_u(\rho,1)\rho^{-n}}{h(\rho,1)^2}(1 + \mathcal{O}(n^{-\frac{1}{2}})).
\end{aligned}$$

Hence,

$$\mathbf{E}X_n = \frac{[x^n]c_u(x,1)}{[x^n]c(x,1)} = \frac{-g_u(\rho,1)}{h(\rho,1)}\sqrt{\pi n} + \mathcal{O}(1).$$

Similarly, we get

$$[x^n]c_{uu}(x,1) = \frac{4g_u(\rho,1)^2}{\sqrt{\pi}h(\rho,1)^3}\rho^{-n}n^{\frac{1}{2}}(1+\mathcal{O}(n^{-\frac{1}{2}}))$$

and

$$\mathbf{V}X_n = \frac{[x^n]c_{uu}(x,1)}{[x^n]c(x,1)} + \mathbf{E}X_n - (\mathbf{E}X_n)^2 = (4-\pi)\frac{g_u(\rho,1)^2}{h(\rho,1)^2}n + \mathcal{O}(n^{\frac{1}{2}}).$$

Next we will determine the characteristic function of $X_n/\sqrt{n}$. Since

$$\mathbf{E}e^{itX_n/\sqrt{n}} = \frac{[x^n]c(x,e^{\frac{it}{\sqrt{n}}})}{[x^n]c(x,1)}, \tag{57}$$

we have to expand $[x^n]c(x,u)$ for $u = e^{it/\sqrt{n}} = 1 + i\frac{t}{\sqrt{n}} + \mathcal{O}(n^{-1})$. For this purpose we will use Cauchy's formula

$$[x^n]c(x,u) = \frac{1}{2\pi i}\int_\Gamma c(z,u)\,\frac{dz}{z^{n+1}} \tag{58}$$

for the following path of integration $\Gamma = \Gamma_1 \cup \Gamma_2$:

$$\begin{aligned} \Gamma_1 &= \left\{z = \rho\left(1+\frac{s}{n}\right)\ :\ s \in \gamma'\right\}, \\ \Gamma_2 &= \left\{z = \mathrm{R}e^{i\vartheta}\ :\ R = \rho\left|1+\frac{\log^2 n + i}{n}\right|, \arg\left(1+\frac{\log^2 n+i}{n}\right) \le |\vartheta| \le \pi\right\}, \end{aligned} \tag{59}$$

where $\gamma' = \{s\ :\ |s| = 1, \Re s \le 0\} \cup \{s\ :\ 0 < \Re s < \log^2 n, \Im s = \pm 1\}$ is the major part of a Hankel contour γ.

First let us concentrate on the path Γ_1. By using the substitution $z = \rho\left(1+\frac{s}{n}\right)$ we get

$$\begin{aligned} \frac{1}{2\pi i}\int_{\Gamma_1} c(z,u)\,\frac{dz}{z^{n+1}} &= \frac{\rho^{-n}}{2\pi i}\int_{\gamma'} \frac{e^{-s}(1+\mathcal{O}(s^2n^{-1}))}{g_u(\rho,1)\frac{it}{\sqrt{n}} + h(\rho,1)\sqrt{\frac{-s}{n}} + \mathcal{O}\left(\frac{s}{n}\right)}\,\frac{ds}{n} \\ &= \frac{\rho^{-n}n^{-\frac{1}{2}}}{h(\rho,1)}\int_{\gamma'}\frac{e^{-s}}{\sqrt{-s}+i\frac{g_u(\rho,1)}{h(\rho,1)}t}\,ds + \mathcal{O}(\rho^{-n}n^{-1}). \end{aligned} \tag{60}$$

Since

$$\int_{\gamma\setminus\gamma'}\frac{e^{-s}}{\sqrt{-s}+iCt}\,ds = \mathcal{O}\left(e^{-\log^2 n}\right)$$

we immediately get by (60) and Lemma 6

$$\frac{1}{2\pi i}\int_{\Gamma_1} c(z,u)\,\frac{dz}{z^{n+1}} = \frac{\rho^{-n}n^{-\frac{1}{2}}}{\sqrt{\pi}h(\rho,1)}\varphi_{\mathcal{R}}\left(\frac{-\sqrt{2}g_u(\rho,1)}{h(\rho,1)}t\right) + \mathcal{O}(\rho^{-n}n^{-1}). \tag{61}$$

Now we make use of the fact that there are $\varepsilon_1 > 0$, $\delta_1 > 0$ such that

$$\max_{|x|=x_1,|\arg x|\ge\vartheta_1} |c(x,u)| = |c(x_1 e^{i\vartheta_1},u)|$$

for $1 \le x_1 \le 1+\delta_1$ and $|u-1| < \varepsilon_1$. You only have to observe that $c_{nk} \ge 0$, that $x = \rho$ is the only singularity on the circle of convergence, and that $c(x,u)$ has the local representation (56). Hence, it follows from

$$\left| c\left(\rho\left(1+\frac{\log^2 n + i}{n}\right), e^{\frac{it}{\sqrt{n}}}\right)\right| = \mathcal{O}\left(n^{\frac{1}{2}}\log n\right)$$

that

$$\int_{\Gamma_2} c(z,u)\,\frac{dz}{z^{n+1}} = \mathcal{O}\left(n^{\frac{1}{2}}\log n\, e^{-\log^2 n}\right). \tag{62}$$

Consequently, by (57), (58), (61), and (62),

$$\mathbf{E}e^{itX_n/\sqrt{n}} = \varphi_{\mathcal{R}}\left(\frac{-\sqrt{2}g_u(\rho,1)}{h(\rho,1)}t\right) + \mathcal{O}(n^{-\frac{1}{2}}).$$

Thus we have proved a weak limit theorem.

In order to prove the local limit theorem we again use Cauchy's formula

$$[x^n u^k]c(x,u) = \frac{1}{(2\pi i)^2}\int_\Gamma\int_\Delta c(z,u)\,\frac{du}{u^{k+1}}\,\frac{dz}{z^{n+1}},$$

where $\Gamma = \Gamma_1 \cup \Gamma_2$ is as above (see (59)), and Δ will be properly chosen.

If $z = \rho(1+\frac{s}{n}) \in \Gamma_1$ then the mapping $u \mapsto c(z,u)$ has a polar singularity at $u_0 = 1 + \frac{t_0}{\sqrt{n}}$, where

$$t_0 = -\frac{h(\rho,1)}{g_u(\rho,1)}\sqrt{-s} + \mathcal{O}\left(\frac{s}{n}\right)$$

with residue

$$\frac{1}{g_u(\rho,1)}\left(1+\mathcal{O}\left(\frac{s}{n}\right)\right).$$

Hence, we can transform Δ in a way that

$$\begin{aligned}
&\frac{1}{2\pi i}\int_\Delta c(z,u)\,\frac{du}{u^{k+1}}\\
&= -\frac{u_0^{-k-1}}{g_u(\rho,1)}\left(1+\mathcal{O}\left(\frac{s}{n}\right)\right) + \frac{1}{2\pi i}\int_{|u|=1+\varepsilon_2} c(z,u)\,\frac{du}{u^{k+1}}\\
&= \frac{-1}{g_u(\rho,1)}\exp\left(\frac{k}{\sqrt{n}}\frac{h(\rho,1)}{g_u(\rho,1)}\sqrt{-s}\right)\left(1+\mathcal{O}\left(\frac{(k+1)s}{n}\right)\right) + \mathcal{O}\left((1+\varepsilon_2)^{-k}\right).
\end{aligned}$$

Consequently,

$$\frac{1}{(2\pi i)^2}\int_{\Gamma_1}\int_\Delta c(z,u)\,\frac{du}{u^{k+1}}\,\frac{dz}{z^{n+1}}$$

$$
\begin{aligned}
&= \frac{-1}{g_u(\rho,1)} \frac{\rho^{-n}}{2\pi i} \int_{\gamma'} \exp\left(-s + \frac{k}{\sqrt{n}} \frac{h(\rho,1)}{g_u(\rho,1)} \sqrt{-s}\right) \left(1 + \mathcal{O}\left(\frac{(k+1)s^2}{n}\right)\right) \frac{ds}{n} \\
&\quad + \mathcal{O}\left(\rho^{-n}(1+\varepsilon_2)^{-k}\right) \\
&= \frac{k}{n^{\frac{3}{2}}} \frac{h(\rho,1)}{g_u(\rho,1)^2} \frac{\rho^{-n}}{2\sqrt{\pi}} \exp\left(-\frac{k^2}{4n} \frac{h(\rho,1)^2}{g_u(\rho,1)^2}\right) + \mathcal{O}\left(\rho^{-n} \frac{k+1}{n^2}\right) \\
&\quad + \mathcal{O}\left(\rho^{-n} \frac{(1+\varepsilon_2)^{-k}}{n}\right).
\end{aligned}
$$

By elementary considerations we obtain

$$
\max_{z\in\Gamma_2, |u|=1} c(z,u) = \mathcal{O}\left(n^{\frac{1}{2}} \log n\right).
$$

Hence, by choosing $\Delta = \{u : |u| = 1\}$ for $z \in \Gamma_2$ we can estimate the remaining integral by

$$
\frac{1}{(2\pi i)^2} \int_{\Gamma_2} \int_{\Delta} c(z,u) \frac{du}{u^{k+1}} \frac{dz}{z^{n+1}} = \mathcal{O}\left(n^{\frac{1}{2}} \log n \, e^{-\log^2 n}\right)
$$

and finally have proved the local limit theorem.

5.3. Proof of Theorem 2. As above we have

$$
[x^n]c(x,1) = \frac{\rho^{-n} n^{-\frac{1}{2}}}{h(\rho,1)\sqrt{\pi}} (1 + \mathcal{O}(n^{-1}))
$$

and from

$$
\begin{aligned}
c_u(x,1) = {} & \frac{-\rho'(1)}{2\rho h(\rho,1)} \left(1 - \frac{x}{\rho}\right)^{-\frac{3}{2}} - \frac{\alpha}{h(\rho,1)^2} \left(1 - \frac{x}{\rho}\right)^{-\frac{1}{2}} \\
& + \mathcal{O}\left(\left(1 - \frac{x}{\rho}\right)^{\frac{1}{2}}\right)
\end{aligned}
$$

we immediately get ($\alpha = 0$)

$$
\begin{aligned}
\mathbf{E}X_n &= \frac{[x^n]c_u(x,1)}{[x^n]c(x,1)} \\
&= -\frac{\rho'(1)}{\rho} n + \mathcal{O}(1) \\
&= \mu\, n + \mathcal{O}(1),
\end{aligned}
$$

and from a little bit more refined analysis we get

$$
\mathbf{V}X_n = \sigma^2 n + \mathcal{O}(\sqrt{n}).
$$

Since

$$
\varphi_{(X_n - \mu n)/\sqrt{\sigma^2 n}}(t) = e^{-it\sqrt{n}\mu/\sigma} \varphi_{X_n}\left(\frac{t}{\sqrt{\sigma^2 n}}\right) = e^{-it\sqrt{n}\mu/\sigma} \frac{[x^n]c(x, e^{it/\sqrt{\sigma^2 n}})}{[x^n]c(x,1)},
$$

we have to determine

$$
[x^n]c(x,u) = \frac{1}{2\pi i} \int_{\Gamma} c(z,u) \frac{dz}{z^{n+1}}
$$

for $u = e^{it/\sqrt{\sigma^2 n}} = 1 + i\frac{t}{\sqrt{\sigma^2 n}} - \frac{t^2}{2\sigma^2 n} + \mathcal{O}(n^{-2})$ in which we use the following path of integration $\Gamma = \Gamma_1 \cup \Gamma_2$:

$$\Gamma_1 = \left\{ z = \rho(u)\left(1 + \frac{s}{n}\right) \,:\, s \in \gamma' \right\},$$

$$\Gamma_2 = \left\{ z = \mathrm{R}e^{i(\vartheta - \arg(u))} \,:\, R = |\rho(u)| \left| 1 + \frac{\log^2 n + i}{n} \right|, \right. \tag{63}$$

$$\left. \arg\left(1 + \frac{\log^2 n + i}{n}\right) \le |\vartheta| \le \pi \right\}.$$

From

$$\rho(u)^{-n} = \rho^{-n} \exp\left(\sqrt{n}\frac{\mu}{\sigma} - \frac{t^2}{2}\right)\left(1 + \mathcal{O}(n^{-\frac12})\right)$$

and from

$$g(z,u) + h(z,u)\sqrt{1 - z/\rho(u)} = h(\rho,1)\sqrt{\frac{-s}{n}} + \mathcal{O}\left(\frac{s}{n}\right)$$

for $x \in \Gamma_1$, by applying Lemma 7 we directly get

$$\begin{aligned}
&\frac{1}{2\pi i}\int_{\Gamma_1} c(z,u)\,\frac{dz}{z^{n+1}} \\
&= \frac{\rho(u)^{-n}}{h(\rho,1)\sqrt{n}}\,\frac{1}{2\pi i}\int_{\gamma'} \frac{e^{-s}}{\sqrt{-s}}\left(1 + \mathcal{O}\left(\frac{s^2}{n} + \left|\frac{s}{n}\right|\right)\right) ds \\
&= \exp\left(\sqrt{n}\frac{\mu}{\sigma} - \frac{t^2}{2}\right)\frac{\rho^{-n}n^{-\frac12}}{h(\rho,1)\sqrt{\pi}}\left(1 + \mathcal{O}(n^{-\frac12})\right).
\end{aligned}$$

It remains to estimate the integral on Γ_2. But this can be done as in the proof of Theorem 1 since

$$\max_{z\in\Gamma_2} |c(z,u)| = \mathcal{O}(n^{\frac12}\log n).$$

In order to prove the local law we again use Cauchy's formula

$$[x^n u^k]c(x,u) = \frac{1}{(2\pi i)^2}\int_\Delta \int_\Gamma c(z,u)\,\frac{dz}{z^{n+1}}\,\frac{du}{u^{k+1}},$$

where $\Delta = \{u \,:\, |u| = 1\}$. For $u \in \Delta_1 = \{u = e^{it} \,:\, |t| \le n^{-5/12}\}$ let $\Gamma = \Gamma_1 \cup \Gamma_2$ as in the proof of the weak limit theorem; for $u \in \Delta_2 = \{u = e^{it} \,:\, n^{-5/12} < |t| < \varepsilon\}$ (for some sufficiently small $\varepsilon > 0$) let $\Gamma = \{z \,:\, |z| = \frac12(\rho + |\rho(e^{it})|)\}$; and for $u \in \Delta_3 = \{u = e^{it} \,:\, \varepsilon \le |t| \le \pi\}$ let $\Gamma = \{z \,:\, |z| = \rho(1+\delta)\}$ for some sufficiently small $\delta > 0$.

First, let $u = e^{it} \in \Gamma_1$, i.e., $|t| \le n^{-5/12}$, and $z \in \Gamma_1$. By direct approximation we have

$$g(z,u) + h(z,u)\sqrt{1 - z/\rho(u)} = n^{-\frac12}h(\rho,1)\sqrt{-s}(1 + \mathcal{O}(n^{-\frac13}))$$

(note that $\alpha = 0$ and that $|s| \ge 1$) and

$$z^{-n}u^{-k} = \rho^{-n}e^{-it(k-\mu n) - \frac12 t^2\sigma^2 n - s}(1 + \mathcal{O}(n^{-\frac14})).$$

Hence, we obtain by using the methods of [12] and a saddle-point-like integration (compare with [9])

$$\begin{aligned}&\frac{1}{(2\pi i)^2}\int_{\Delta_1}\int_{\Gamma_1} c(z,u)\,\frac{dz}{z^{n+1}}\,\frac{du}{u^{k+1}}\\&=\frac{\rho^{-n}}{\sqrt{n}h(\rho,1)}\frac{1}{(2\pi i)^2}\int_{-n^{-5/12}}^{n^{-5/12}}\int_{\gamma'}\frac{e^{-it(k-\mu n)-\frac{1}{2}t^2\sigma^2 n-s}}{\sqrt{-s}}(1+\mathcal{O}(n^{-\frac{1}{4}}))\,ds\,dt\\&=\frac{\rho^{-n}}{\sqrt{n}h(\rho,1)}\frac{1}{\sqrt{\pi}}\frac{1}{\sqrt{2\pi\sigma^2 n}}\left(\exp\left(-\frac{(k-\mu n)^2}{2\sigma^2 n}\right)+\mathcal{O}(n^{-\frac{1}{4}})\right).\end{aligned}$$

Therefore, the proof is finished if the remaining integrals are sufficiently small. As above we have

$$\frac{1}{(2\pi i)^2}\int_{\Delta_1}\int_{\Gamma_2} c(z,u)\,\frac{dz}{z^{n+1}}\,\frac{du}{u^{k+1}}=\mathcal{O}\left(\rho^{-n}n^{\frac{1}{12}}\log n\, e^{-\log^2 n}\right).$$

Next we get

$$\frac{1}{(2\pi i)^2}\int_{\Delta_2}\int_{\Gamma} c(z,u)\,\frac{dz}{z^{n+1}}\,\frac{du}{u^{k+1}}=\mathcal{O}\left(\rho^{-n}e^{-cn^{\frac{1}{6}}}\right)$$

since $|\rho(u)|\ge\rho(1+c_1 n^{-5/6})$ for $u\in\Delta_2$ (and some sufficiently small constants $c, c_1>0$). Finally, since $c(z,u)$ is bounded for $u\in\Delta_3$ and $z\in\Gamma$ we obtain

$$\frac{1}{(2\pi i)^2}\int_{\Delta_3}\int_{\Gamma} c(z,u)\,\frac{dz}{z^{n+1}}\,\frac{du}{u^{k+1}}=\mathcal{O}\left(\rho^{-n}(1+\delta)^{-n}\right),$$

which completes the proof of the local theorem.

5.4. Proof of Theorem 3. As in the proof of Theorem 2 we get

$$\mathbf{E}X_n=\mu\, n-\sqrt{\pi}\frac{\alpha}{h(\rho,1)}\sqrt{n}+\mathcal{O}(1)$$

and

$$\mathbf{V}X_n=\left(\sigma^2+(4-\pi)\frac{\alpha^2}{h(\rho,1)^2}\right)n+\mathcal{O}(\sqrt{n}).$$

Now, if we use the same normalization and the same path of integration (63) as in Theorem 2, by applying Lemma 6 we obtain for $u=e^{it/\sqrt{\sigma^2 n}}$

$$\begin{aligned}&\frac{1}{2\pi i}\int_{\Gamma_1} c(z,u)\,\frac{dz}{z^{n+1}}\\&=\frac{\rho(u)^{-n}}{h(\rho,1)\sqrt{n}}\frac{1}{2\pi i}\int_{\gamma'}\frac{e^{-s}}{\sqrt{-s}+i\alpha t/(h(\rho,1)\sigma)}\left(1+\mathcal{O}\left(\frac{s^2}{n}+\left|\frac{s}{n}\right|\right)\right)\,ds\\&=\frac{\rho^{-n}n^{-\frac{1}{2}}}{h(\rho,1)\sqrt{\pi}}\exp\left(\sqrt{n}\frac{\mu}{\sigma}-\frac{t^2}{2}\right)\varphi_{\mathcal{R}}\left(\frac{-\sqrt{2}\alpha}{h(\rho,1),\sigma}\right)\left(1+\mathcal{O}(n^{-\frac{1}{2}})\right).\end{aligned}$$

The remaining integral on Γ_2 can be estimated as above. Hence, we have proved the weak convergence property.

In order to prove the local law we will proceed as in the proof of Theorem 2. As above we can concentrate on the path of integration $\Delta_1 \times \Gamma_1$. The remaining integrals are negligible. Direct approximation yields

$$\begin{aligned}&\frac{1}{(2\pi i)^2}\int_{\Delta_1}\int_{\Gamma_1} c(z,u)\,\frac{dz}{z^{n+1}}\,\frac{du}{u^{k+1}}\\&= \frac{\rho^{-n}}{\sqrt{n}h(\rho,1)}\,\frac{1}{(2\pi i)^2}\int_{-n^{-5/12}}^{n^{-5/12}}\int_{\gamma'}\frac{e^{-it(k-\mu n)-\frac12 t^2\sigma^2 n-s}}{\sqrt{-s}+i\frac{\alpha}{h(\rho,1)}\sqrt{nt}}\left(1+\mathcal{O}(n^{-\frac14})\right)ds\,dt.\end{aligned}$$

Hence an application of Lemma 8 and easy tail estimates complete the proof of Theorem 3.

6. Conclusions. The main purpose of this paper is to provide general techniques to obtain the limiting distribution of parameters in combinatorial constructions. It is the second paper of a (planned) series of papers [10] devoted to this topic. Theorems 1–3 should be considered as examples of analytic theorems providing a link between combinatorial constructions and their asymptotic distributions. (They seem to be proper theorems to discuss random mappings.) The authors are convinced that the methods presented in the preceding proofs can be used in many other (different) problems. The basic ideas are singularity analysis (introduced by Flajolet and Odlyzko [12]) and saddle-point approximation.

Random mappings are widely and intensively discussed in literature; e.g., in Kolchin's book [15] a probabilistic approach via branching processes is presented, whereas Aldous and Pitman [3] use a completely different probabilistic concept related to Aldous's continuum random trees [1, 2]. Our concept of generating functions goes back to Arney and Bender [4] and to Flajolet and Odlyzko [13]. They could identify many limiting distributions and provided asymptotic expansions for mean and variance. (One gap could be filled by Application 3.) It should be mentioned, too, that Arney and Bender [4] discussed a slightly more general case, namely that the number of immediate predecessors of a point $|\varphi^{-1}(\{\nu\})|$ is not arbitrary but must be contained in a subset D of nonnegative integers; i.e., the corresponding tree function $\hat{t}_D(x)$ satisfies the functional equation

$$\hat{t}_D(x) = x\sum_{n\in D}\frac{\hat{t}_D(x)^n}{n!} = x\phi(\hat{t}_D(x))$$

and the corresponding generating function for those mappings

$$\hat{f}_D(x) = \frac{1}{1-x\phi'(\hat{t}_D(x))}.$$

From Proposition 1 it follows that if D contains a number ≥ 2 then it has a square-root singularity

$$\hat{t}_D(x) = g_D(x) - h_D(x)\sqrt{1-x/\rho_D},$$

around $x = \rho_D = t_0/\phi(t_0)$, where $t_0 > 0$ satisfies $t_0\phi'(t_0) = \phi(t_0)$. Hence, $\rho_D\phi'(t_0) = 1$ and we are in a similar situation as in the classical case. Especially we can adapt all the combinatorial constructions to this more general case and obtain analog results by applying our theorems.

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* III, Ann. Probab., 21 (1993), pp. 248–289.

[3] D. J. Aldous and J. Pitman, *Brownian bridge asymptotics for random mappings*, Random Structures Algorithms, 5 (1994), pp. 487–512.

[4] J. Arney and E. A. Bender, *Random mappings with constraints on coalescence and number of origins*, Pacific J. Math., 103 (1982), pp. 269–294.

[5] E. A. Bender, *Central and local limit theorems applied to asymptotic enumeration*, J. Combin. Theory Ser. A, 15 (1973), pp. 91–111.

[6] E. A. Bender and B. Richmond, *Central and local limit theorems applied to asymptotic enumeration.* II: *Multivariate generating functions*, J. Combin. Theory Ser. B, 34 (1983), pp. 255–265.

[7] M. Drmota, *Asymptotic distributions and a multivariate Darboux method in enumeration problems*, J. Combin. Theory Ser. A, 67 (1994), pp. 139–152.

[8] M. Drmota, *The height distribution of leaves in rooted trees*, Discrete Math. Appl., 4 (1994), pp. 45–58.

[9] M. Drmota, *A bivariate asymptotic expansion of coefficients of powers of generating functions*, European J. Combin., 15 (1994), pp. 139–152.

[10] M. Drmota and M. Soria, *Marking in combinatorial constructions: Generating functions and limiting distributions*, Theoret. Comput. Sci., 144 (1995), pp. 67–99.

[11] Ph. Flajolet, *Elements of a general theory of combinatorial structures*, in Fundamentals of Computation Theory, L. Budach, ed., Lecture Notes in Computer Science 199, Springer-Verlag, 1985, pp. 112–127.

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

[13] Ph. Flajolet and A. M. Odlyzko, *Random mapping statistics*, in Proceedings of Euroscript '89, J-J. Quisquater, ed., Lecture Notes in Computer Science 434, Springer-Verlag, 1990, pp. 329–354.

[14] Ph. Flajolet and M. Soria, *General combinatorial schemas: Special limit distributions*, manuscript.

[15] V. F. Kolchin, *Random Mappings*, Optimization Software, New York, 1986.

[16] L. Mutafchiev, *The limit distribution of the number of nodes in low strata of a random mapping*, Statist. Probab. Lett., 7 (1989), pp. 247–251.

[17] J. Vitter and Ph. Flajolet, *Analysis of algorithms and data structures*, in Handbook of Theoretical Computer Science, Vol. A: Algorithms and Complexity, Chap. 9, J. Van Leeuwen, ed., North–Holland, Amsterdam, 1990, pp. 432–524.

DE BRUIJN SEQUENCES AND PERFECT FACTORS[*]

CHRIS J. MITCHELL[†]

Abstract. In this paper we describe new constructions for de Bruijn sequences and Perfect Factors. These constructions are all based upon the idea of constructing one sequence (or set of sequences) from another. As a result of this fact, the sequences obtained from these construction methods possess simple decoding algorithms based on decoding the sequences used to construct them. Such decoding algorithms are of importance in position–location applications.

Key words. de Bruijn sequence, de Bruijn graph, window sequence, Perfect Factor

AMS subject classifications. 05C70, 05C38, 94A99, 68R10

PII. S0895480195290911

1. Introduction.

1.1. de Bruijn sequences, Perfect Factors, and the decoding problem.

In this paper we address two main issues relating to the existence and decoding of Perfect Factors and de Bruijn sequences.

• Perfect Factors, i.e., sets of uniformly long cycles whose elements are drawn from an alphabet of size c and in which every possible v-tuple of elements occurs exactly once, are of significance for two main reasons.

– They can be used to construct Perfect Maps (or two-dimensional de Bruijn arrays) (see, for example, [4, 9, 10]), which are of practical importance in certain position–location applications.

– They are special cases of Perfect Maps themselves, and hence their existence is of significance in deciding whether Perfect Maps exist for all parameter sets satisfying certain simple necessary conditions. (It has recently been established that these necessary conditions are sufficient for prime power size alphabets [12, 13].)

They are also of combinatorial interest in their own right [4].

It has been conjectured [6] that the simple necessary conditions for the existence of a Perfect Factor are sufficient for all finite alphabets and for all window sizes. This conjecture was established by Paterson for c, a prime power [11], and for $v < 5$ in [7]. In this paper we describe two new construction methods for Perfect Factors, yielding Perfect Factors with parameters not previously known to exist.

• The problem of *decoding* de Bruijn sequences and Perfect Maps, i.e., of finding the position within the sequence (or array) of any specified v-tuple (or subarray), is of fundamental importance in certain practical applications (see [2, 3, 14]). It has recently been shown that de Bruijn sequences which have simple decoding methods can be constructed [8]; in this paper we present another construction method for de Bruijn sequences which also yields sequences with a simple decoding technique.

In addition, it has been shown that Perfect Maps can be constructed using a combination of Perfect Factors and de Bruijn sequences, for which decoding the Perfect Map can be reduced to decoding its component sequences [9]. The methods for

[*] Received by the editors August 30, 1995; accepted for publication (in revised form) April 29, 1996.

http://www.siam.org/journals/sidma/10-2/29091.html

[†] Information Security Group, Royal Holloway, University of London, Egham, Surrey TW20 0EX, England (cjm@dcs.rhbnc.ac.uk).

constructing Perfect Factors presented here all allow simple decoding methods to be devised, and hence contribute to the simpler decoding of certain Perfect Maps.

1.2. Notation. We first set up some notation which we will use throughout the paper.

We are concerned here with c-ary periodic sequences, where by the term c-ary we mean sequences whose elements are drawn from the set $\{0, 1, \ldots, c-1\}$. We refer throughout to c-ary cycles of period n, by which we mean periodic sequences $[s_0, s_1, \ldots, s_{n-1}]$, where $s_i \in \{0, 1, \ldots, c-1\}$ for every i $(0 \leq i < n)$.

If $\boldsymbol{t} = (t_0, t_1, \ldots, t_{v-1})$ is a c-ary v-tuple (i.e., $t_i \in \{0, 1, \ldots, c-1\}$ for every i $(0 \leq i < v)$) and $\boldsymbol{s} = [s_0, s_1, \ldots, s_{n-1}]$ is a c-ary cycle of period n $(n \geq v)$, then we say that $\boldsymbol{t}$ *occurs in* $\boldsymbol{s}$ *at position* j if and only if

$$t_i = s_{i+j}$$

for every i $(0 \leq i < v)$, where $i + j$ is computed modulo n.

If $\boldsymbol{s}_0, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{t-1}$ are t cycles of the same length, n say, and if

$$\boldsymbol{s}_i = [s_{i0}, s_{i1}, \ldots, s_{i(n-1)}] \quad (0 \leq i < t),$$

then $\mathcal{I}(\boldsymbol{s}_0, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{t-1})$ denotes the t-fold interleaving of these cycles, i.e.,

$$\mathcal{I}(\boldsymbol{s}_0, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{t-1}) = [s_{00}, s_{10}, \ldots, s_{(t-1)0}, s_{01}, s_{11}, \ldots, s_{(t-1)(n-1)}],$$

a cycle of length nt.

Given a cycle $\boldsymbol{s} = [s_i]$ $(0 \leq i < n)$ and any integer k, we define $\boldsymbol{T}_k(\boldsymbol{s})$ to be the *cyclic shift* of $\boldsymbol{s}$ by k places *to the right.* That is, if we write $\boldsymbol{s}' = [s'_i] = \boldsymbol{T}_k(\boldsymbol{s})$ then

$$s'_{i+k} = s_i \quad (0 \leq i < n),$$

where $i + k$ is calculated modulo n.

Suppose $\boldsymbol{s} = [s_0, s_1, \ldots, s_{n-1}]$ and $\boldsymbol{s}' = [s'_0, s'_1, \ldots, s'_{n'-1}]$ are c-ary cycles of periods n and n', respectively. Then define the *concatenation* of $\boldsymbol{s}$ and $\boldsymbol{s}'$, written

$$\boldsymbol{s}||\boldsymbol{s}',$$

to be the c-ary cycle of period $n + n'$

$$\boldsymbol{t} = [t_0, t_1, \ldots, t_{n+n'-1}] = \boldsymbol{s}||\boldsymbol{s}',$$

where

$$t_i = \begin{cases} s_i & \text{if } 0 \leq i < n, \\ s'_{i-n} & \text{if } n \leq i < n + n'. \end{cases}$$

In addition, if $\boldsymbol{s}$ is a cycle of length n and $k > 0$, then $\boldsymbol{s}^k$ denotes the k-fold concatenation of $\boldsymbol{s}$ with itself, and hence $\boldsymbol{s}^k$ is a cycle of period nk.

Throughout we will write $\boldsymbol{0}^i$ for the i-tuple of all zeros and $\boldsymbol{1}^i$ for the i-tuple of all ones.

Finally note that throughout this paper the notation (m, n) represents the *greatest common divisor* of m and n (given that m, n are a pair of positive integers).

1.3. Fundamental definitions and results. We next define the objects of fundamental importance to this paper.

DEFINITION 1.1. *If* $\boldsymbol{s} = (s_0, s_1, \ldots, s_{n-1})$ *is a c-ary cycle of period n, then we say that* $\boldsymbol{s}$ *is a v*-window sequence *if no c-ary v-tuple occurs in two distinct positions within a period of* $\boldsymbol{s}$*. Equivalently, it contains n distinct v-tuples in a period of the cycle.*

Using this definition we also have the following one.

DEFINITION 1.2. *A c-ary* de Bruijn sequence of span v *is then simply a v-window sequence of period equal to* c^v*; equivalently, every possible c-ary v-tuple occurs precisely once in a period of a de Bruijn sequence.*

A c-ary punctured de Bruijn sequence of span v (*sometimes called a* pseudorandom sequence) *is a v-window sequence in which every c-ary v-tuple except for* $\boldsymbol{0}^v$ *occurs, and so a punctured de Bruijn sequence has period* $c^v - 1$*. A span v de Bruijn sequence can be "punctured" by deleting one of the zeros in* $\boldsymbol{0}^v$*, and a punctured de Bruijn sequence can be transformed into a de Bruijn sequence by adding a zero to any one of the* $c-1$ *occurrences of* $\boldsymbol{0}^{v-1}$.

We next have the following definition.

DEFINITION 1.3. *Suppose n, c, and v are positive integers, where* $c \geq 2$*. An* (n, c, v)*-Perfect Factor, or simply an* (n, c, v)*-PF, is a collection of* c^v/n *c-ary cycles of period n with the property that every c-ary v-tuple occurs in one of these cycles.*

Note that because we insist that a Perfect Factor contain exactly c^v/n cycles and because there are clearly c^v different c-ary v-tuples, each v-tuple will actually occur exactly once somewhere in the collection of cycles (and hence all the cycles are distinct). Also observe that a (c^v, c, v)-PF is simply a c-ary span v de Bruijn sequence.

The following necessary conditions for the existence of a Perfect Factor are trivial to establish.

LEMMA 1.4. *Suppose A is an* (n, c, v)*-PF. Then*

1. $n|c^v$*, and*
2. $v < n \leq c^v$ *(or* $n = v = 1$*).*

It was conjectured in [6] that these necessary conditions are sufficient for the existence of a Perfect Factor. Paterson [11] has shown that the conjecture holds if c is a prime power, and it has also been shown that the conjecture holds if $v < 5$ [7].

Finally, we define a related set of combinatorial objects, first introduced in [6].

DEFINITION 1.5. *Suppose n, k, c, and v are positive integers satisfying* $n|c^v$ *and* $c \geq 2$*. An* (n, k, c, v)*-Perfect Multifactor, or simply an* (n, k, c, v)*-PMF, is a collection of* c^v/n *c-ary cycles of period nk with the property that for every c-ary v-tuple* $\boldsymbol{t}$ *and for every integer j in the range* $0 \leq j < k$*,* $\boldsymbol{t}$ *occurs at a position* $p \equiv j \pmod{k}$ *in one of these cycles.*

Note that because a Perfect Multifactor contains exactly c^v/n cycles of length nk and because there are c^v different c-ary v-tuples, each v-tuple will actually occur exactly k times in the collection of cycles, once in each of the possible position congruency classes (mod k). This also implies that all the cycles are distinct.

The following necessary conditions for the existence of a Perfect Multifactor are simple to establish.

LEMMA 1.6. *Suppose A is an* (n, k, c, v)*-PMF. Then*

1. $n|c^v$ *and*
2. $v < nk$ *(or* $v = nk$ *and* $n = 1$*).*

It has been shown [6] that the above necessary conditions are sufficient if $k \geq v$.

2. A span-dividing construction for Perfect Factors. In this section we describe a novel method for constructing a Perfect Factor from a Perfect Multifactor. This method involves reducing the span and at the same time increasing the alphabet size. The method is of practical interest because a simple decoding algorithm for the Perfect Factor can be derived from a decoding algorithm for the Perfect Multifactor used to construct it.

2.1. The construction method.

CONSTRUCTION 2.1. *Suppose $c, k, n,$ and v are positive integers, where $c \geq 2$, $n | c^v$, and $k | v$, and let $A = \{\boldsymbol{a}_i : 0 \leq i < c^v/n\}$ be an (n, k, c, v)-PMF.*

Now define $D = \{\boldsymbol{d}_i : 0 \leq i < c^v/n\}$ to be the set of c^v/n c^k-ary cycles of period n defined so that $\boldsymbol{d}_i$ is obtained from $\boldsymbol{a}_i$ by dividing $\boldsymbol{a}_i$ into disjoint k-tuples and regarding each k-tuple as the c-ary representation of an element from an alphabet of size c^k.

THEOREM 2.2. *Suppose c, k, n, v, and A satisfy the conditions of Construction* 2.1. *If D is constructed from A using Construction* 2.1, *then D is an $(n, c^k, v/k)$-PF.*

Proof. Let $u = v/k$ and suppose $\boldsymbol{e}$ and $\boldsymbol{e}'$ are u-tuples from D occurring at positions p and p' in cycles $\boldsymbol{d}_i$ and $\boldsymbol{d}_{i'}$, respectively ($0 \leq p, p' < n$ and $0 \leq i, i' < c^v/n$). We need to show that these tuples are distinct unless $p = p'$ and $i = i'$.

Now if $\boldsymbol{e} = \boldsymbol{e}'$ then $\boldsymbol{f} = \boldsymbol{f}'$, where $\boldsymbol{f}$ and $\boldsymbol{f}'$ are c-ary v-tuples derived, respectively, from $\boldsymbol{e}$ and $\boldsymbol{e}'$ by substituting every c^k-ary element with a c-ary k-tuple (inverting the procedure used to derive D in Construction 2.1). Now $\boldsymbol{f}$ and $\boldsymbol{f}'$ occur at positions kp and kp' in cycles $\boldsymbol{a}_i$ and $\boldsymbol{a}_{i'}$, respectively. Hence, since A is a Perfect Multifactor and $kp \equiv kp' \pmod{k}$, we have

$$i = i' \text{ and } kp \equiv kp' \pmod{nk},$$

and the desired result follows. □

2.2. An example. We now give a simple example.

Example 2.3. Let $n = v = 4$ and $c = k = 2$. Also let

$$A = \left\{ \begin{array}{ll} \boldsymbol{a}_0 = [\,0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\,], & \boldsymbol{a}_1 = [\,1\ 0\ 0\ 0\ 0\ 1\ 1\ 1\,], \\ \boldsymbol{a}_2 = [\,0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\,], & \boldsymbol{a}_3 = [\,0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\,] \end{array} \right\},$$

a (4,2,2,4)-PMF.

Then, using the above construction, we obtain

$$D = \{\, \boldsymbol{d}_0 = [\,0\ 0\ 3\ 3\,],\ \boldsymbol{d}_1 = [\,2\ 0\ 1\ 3\,], \boldsymbol{d}_2 = [\,1\ 1\ 2\ 2\,],\ \boldsymbol{d}_3 = [\,0\ 2\ 3\ 1\,] \},$$

a (4,4,2)-PF.

2.3. A decoding algorithm. We now present a simple algorithm for decoding cycles which have been obtained using Construction 2.1; the algorithm is based on the use of a partial decoder for the Perfect Multifactor A.

ALGORITHM 2.4. *Suppose $c, k, n, v,$ and A satisfy the conditions of Construction* 2.1 *and D has been constructed from A using Construction* 2.1. *Suppose also that the pair of functions (E_1, E_2) acts as a partial decoder for A; i.e., if $\boldsymbol{x}$ is a c-ary v-tuple then $0 \leq E_1(\boldsymbol{x}) < c^v/n$, $0 \leq E_2(\boldsymbol{x}) < n$, and $\boldsymbol{x}$ occurs at position $kE_2(\boldsymbol{x})$ in cycle $\boldsymbol{a}_{E_1(\boldsymbol{x})}$ of A. That is, the partial decoder will find the unique location of the specified tuple in a position congruent to* 0 *modulo k.*

Then the pair (E_1, E_2) is a decoder for D; i.e., if $\boldsymbol{y}$ is a c^k-ary u-tuple, then $\boldsymbol{y}$ occurs at position $E_2(\boldsymbol{y})$ in cycle $\boldsymbol{d}_{E_1(\boldsymbol{y})}$.

Proof. This result follows immediately from the way in which D is constructed. □

2.4. Constructing suitable Perfect Multifactors. We now consider the problem of constructing Perfect Multifactors with parameters suitable for use in Construction 2.1. We first observe that, using Constructions 6.1 and 6.4 of [6], we have the following theorem.

THEOREM 2.5. *Suppose c, m, n, s, and v are positive integers, where $c \geq 2$, $m|n$, and $(s, m) = 1$, and suppose also that there exists an (n, c, v)-PF. Then an $(m, ns/m, c, v)$-PMF can be constructed.*

Remark 2.6. An examination of the construction methods in [6] reveals that a decoding algorithm for the Perfect Multifactors can very easily be derived from a decoding algorithm for the Perfect Factor used to construct it.

Note also that the (4,2,2,4)-PMF of Example 2.3 was obtained from an (8,2,4)-PF using exactly this method.

There are two simple ways in which we can combine Theorem 2.5 with our new construction method.

- First, suppose that $n = c^v$, $k|v$, and $(k, c^v) = 1$, and put $m = n$ and $s = k$ (and hence $(s, m) = 1$). Then, starting with a (c^v, c, v)-PF (a c-ary span v de Bruijn sequence), we can obtain a (c^v, k, c, v)-PMF. Now, since $k|v$, we can apply Construction 2.1 to obtain a $(c^v, c^k, v/k)$-PF, i.e., a c^k-ary span v/k d e Bruijn sequence. Most significantly, this new de Bruijn sequence can be trivially decoded using a decoder for the de Bruijn sequence used to construct it.
- Second, suppose $k|v$ and $n|c^v$, and put $m = n/(k, n)$ and $s = k/(k, n)$ (and hence $(s, m) = 1$). Then, starting with an (n, c, v)-PF, we can obtain an $(n/(k, n), k, c, v)$-PMF. Now, since $k|v$, we can apply Construction 2.1 to obtain an $(n/(k, n), c^k, v/k)$-PF. Again, this new Perfect Factor can be trivially decoded using a decoder for the Perfect Factor used to construct it.

Remark 2.7. Note that in the first case considered immediately above we could replace the initial de Bruijn sequence with any c-ary v-window sequence of period n, as long as $(n, k) = 1$ and $k|v$. We would then obtain a c^k-ary (v/k)-window sequence, also of period n. Thus if $(c^v - 1, k) = 1$ then we could start with a punctured c-ary span v de Bruijn sequence, in which case the final sequence would also be a punctured de Bruijn sequence.

2.5. Example.

Example 2.8. Let $v = 4$, $c = k = 2$, and $n = c^v - 1 = 15$ (and hence $(k, n) = (2, 15) = 1$). Also let

$$\boldsymbol{a}' = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix},$$

a 2-ary span 4 punctured de Bruijn sequence.

Then, using Constructions 6.1 and 6.4 of [6], we obtain

$$\boldsymbol{a} = \begin{bmatrix} 0\,0\,0\,1\,0\,0\,1\,1\,0\,1\,0\,1\,1\,1\,1\,0\,0\,0\,1\,0\,0\,1\,1\,0\,1\,0\,1\,1\,1\,1 \end{bmatrix}.$$

Using Construction 2.1, we obtain

$$\boldsymbol{d} = \begin{bmatrix} 0 & 1 & 0 & 3 & 1 & 1 & 3 & 2 & 0 & 2 & 1 & 2 & 2 & 3 & 3 \end{bmatrix},$$

a 4-ary span 2 punctured de Bruijn sequence.

3. Constructing Perfect Factors by interleaving. We now present another method for constructing Perfect Factors with a simple decoding algorithm. It also enables the construction of Perfect Factors for parameter sets for which the existence question was previously unanswered (examples of new parameter sets are given in section 3.4 below).

3.1. The construction method. We start by describing the method of construction.

CONSTRUCTION 3.1. *Suppose c, n, t, v are positive integers satisfying $c \geq 2$ and $t|n^{t-1}$. Moreover, suppose that*

$$A = \{\boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_{c^v/n-1}\}$$

is an (n, c, v)-PF.

Consider the set S of all n-ary t-tuples $(x_0, x_1, \ldots, x_{t-1})$ with the property that $\sum_{i=0}^{t-1} x_i \equiv n-1 \pmod{n}$. If $\boldsymbol{x}, \boldsymbol{y} \in S$ then write $\boldsymbol{x} \sim \boldsymbol{y}$ if and only if $\boldsymbol{x}$ can be obtained from $\boldsymbol{y}$ by a cyclic shift operation. It is straightforward to verify that $\sim$ is an equivalence relation on S which partitions S into n^{t-1}/t classes, each of size t. Now let

$$X = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n^{t-1}/t-1}\}$$

be a set of elements of S chosen so that X contains precisely one element of each equivalence class under $\sim$.

Next let

$$U = \{(\boldsymbol{a}_{i_0}, \boldsymbol{a}_{i_1}, \ldots, \boldsymbol{a}_{i_{t-1}}) \;:\; \boldsymbol{a}_{i_0}, \boldsymbol{a}_{i_1}, \ldots, \boldsymbol{a}_{i_{t-1}} \in A\}$$

be the set of all t-tuples of elements of A, and hence $|U| = c^{tv}/n^t$.

Finally, let B be the set of all interleaved cycles of the form

$$\mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_{i_0}), \boldsymbol{T}_{x_0}(\boldsymbol{a}_{i_1}), \boldsymbol{T}_{x_0+x_1}(\boldsymbol{a}_{i_2}), \ldots, \boldsymbol{T}_{x_0+x_1+\cdots+x_{t-2}}(\boldsymbol{a}_{i_{t-1}})),$$

where $(x_0, x_1, \ldots, x_{t-1}) \in X$ and $(\boldsymbol{a}_{i_0}, \boldsymbol{a}_{i_1}, \ldots, \boldsymbol{a}_{i_{t-1}}) \in U$. Hence $|B| = |X| \cdot |U| = (n^{t-1}/t)(c^{tv}/n^t) = c^{tv}/tn$.

We can now state and prove the following result.

THEOREM 3.2. *Suppose c, n, t, v, and A satisfy the conditions of Construction* 3.1. *If B is constructed from A using Construction* 3.1 *then B is a (tn, c, tv)-PF.*

Proof. Suppose $\boldsymbol{y}$ is any c-ary tv-tuple. We need to show that $\boldsymbol{y}$ occurs in one of the cycles of B. Suppose

$$\boldsymbol{y} = \mathcal{I}(\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{t-1}),$$

where $\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{t-1}$ are c-ary v-tuples. Now suppose that $\boldsymbol{z}_i$ occurs in cycle $\boldsymbol{a}_{\ell_i}$ at position k_i for every i satisfying $0 \leq i < t$. In addition, we define a further n-ary t-tuple $\boldsymbol{x} = (x_0, x_1, \ldots, x_{t-1})$, where $x_i \equiv k_i - k_{i+1} \pmod{n}$ for every i satisfying $0 \leq i < t-1$ and $x_{t-1} \equiv k_{t-1} - k_0 - 1 \pmod{n}$.

First observe that $\boldsymbol{x} \in S$ since

$$\sum_{i=0}^{t-1} x_i \equiv \sum_{i=0}^{t-2} (k_i - k_{i+1}) + (k_{t-1} - k_0 - 1) \equiv -1 \pmod{n}.$$

Hence there exists some cyclic shift of $\boldsymbol{x}$, say

$$\boldsymbol{T}_{t-u}(\boldsymbol{x}) = (x_u, x_{u+1}, \ldots, x_{t-1}, x_0, \ldots, x_{u-1}),$$

which is a member of X. Hence if we define the n-ary t-tuple $(v_0, v_1, \ldots, v_{t-1})$ by

$$v_i = \begin{cases} 0 & \text{if } i = 0, \\ \sum_{j=u}^{i+u-1} x_j \bmod n & \text{if } 0 < i \leq t-u, \\ \sum_{j=u}^{t-1} x_j + \sum_{j=0}^{i+u-t-1} x_j \bmod n & \text{if } t-u < i \leq t-1, \end{cases}$$

then the following cycle is a member of B:

$$\boldsymbol{w} = \mathcal{I}(\boldsymbol{T}_{v_0}(\boldsymbol{a}_{\ell_u}), \boldsymbol{T}_{v_1}(\boldsymbol{a}_{\ell_{u+1}}), \ldots, \boldsymbol{T}_{v_{t-u-1}}(\boldsymbol{a}_{\ell_{t-1}}), \boldsymbol{T}_{v_{t-u}}(\boldsymbol{a}_{\ell_0}), \ldots, \boldsymbol{T}_{v_{t-1}}(\boldsymbol{a}_{\ell_{u-1}})).$$

Now z_{u+i} occurs in $\boldsymbol{T}_{v_i}(\boldsymbol{a}_{\ell_{u+i}})$ at position $k_{u+i} + v_i$ $(0 \leq i < t-u)$ and z_i occurs in $\boldsymbol{T}_{v_{i+t-u}}(\boldsymbol{a}_{\ell_i})$ at position $k_i + v_{t-u+i}$ $(0 \leq i \leq u-1)$. In addition, by the definition of (x_i) we have

$$v_i = \begin{cases} 0 & \text{if } i = 0, \\ k_u - k_{u+i} \bmod n & \text{if } 0 < i < t-u, \\ k_u - k_{u-t+i} - 1 \bmod n & \text{if } t-u \leq i \leq t-1. \end{cases}$$

Thus z_{u+i} occurs in $\boldsymbol{T}_{v_i}(\boldsymbol{a}_{\ell_{u+i}})$ at position k_u $(0 \leq i < t-u)$ and z_i occurs in $\boldsymbol{T}_{v_{i+t-u}}(\boldsymbol{a}_{\ell_i})$ at position $k_u - 1$ $(0 \leq i \leq u-1)$. Hence $\boldsymbol{y}$ occurs in $\boldsymbol{w}$ at position $k_u t - u$ and the result follows. □

3.2. Examples. Before proceeding we give two simple examples of the construction method.

Example 3.3. Let $n = 4$ and $c = v = t = 2$. Then let A be the following $(4, 2, 2)$-PF (a de Bruijn sequence):

$$\boldsymbol{a}_0 = \left[\begin{array}{cccc} 0 & 0 & 1 & 1 \end{array}\right].$$

Then

$$S = \{\ (\ 0\quad 3\ ),\quad (\ 3\quad 0\ ),\quad (\ 2\quad 1\ ),\quad (\ 1\quad 2\ )\ \}.$$

Then we can define

$$X = \{\ (\ 0\quad 3\ ),\quad (\ 2\quad 1\ )\ \}.$$

In addition,

$$U = \{(\ \boldsymbol{a}_0,\quad \boldsymbol{a}_0\ )\}.$$

Hence

$$\begin{aligned} B &= \{\, \mathcal{I}(\ \boldsymbol{T}_0(\boldsymbol{a}_0),\ \boldsymbol{T}_0(\boldsymbol{a}_0)\ ),\ \mathcal{I}(\ \boldsymbol{T}_0(\boldsymbol{a}_0),\ \boldsymbol{T}_2(\boldsymbol{a}_0)\ )\,\} \\ &= \{\, \mathcal{I}(\ [\,0\ 0\ 1\ 1\,],\ [\,0\ 0\ 1\ 1\,]\ ),\ \mathcal{I}(\ [\,0\ 0\ 1\ 1\,],\ [\,1\ 1\ 0\ 0\,]\ )\,\} \\ &= \{\ [\,0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\,],\ [\,0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\,]\ \} \end{aligned}$$

is an $(8, 2, 4)$-PF.

Example 3.4. Let $n = c = t = 3$ and $v = 1$. Then let A be the following $(3, 3, 1)$-PF (a de Bruijn sequence):

$$\boldsymbol{a}_0 = [\ 0 \quad 1 \quad 2\ ].$$

Then

$$\begin{aligned} S = \{ &(\ 0 \quad 0 \quad 2\ ), \quad (\ 0 \quad 2 \quad 0\ ), \quad (\ 0 \quad 1 \quad 1\ ), \\ &(\ 2 \quad 0 \quad 0\ ), \quad (\ 2 \quad 2 \quad 1\ ), \quad (\ 2 \quad 1 \quad 2\ ), \\ &(\ 1 \quad 0 \quad 1\ ), \quad (\ 1 \quad 2 \quad 2\ ), \quad (\ 1 \quad 1 \quad 0\ )\ \}. \end{aligned}$$

Then we can define

$$X = \{\ (\ 0 \quad 0 \quad 2\ ), \quad (\ 0 \quad 1 \quad 1\ ), \quad (\ 2 \quad 2 \quad 1\ )\ \}.$$

In addition,

$$U = \{(\ \boldsymbol{a}_0, \quad \boldsymbol{a}_0, \quad \boldsymbol{a}_0\ )\}.$$

Hence

$$\begin{aligned} B &= \{\mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_0), \boldsymbol{T}_0(\boldsymbol{a}_0), \boldsymbol{T}_{0+0}(\boldsymbol{a}_0)), \mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_0), \boldsymbol{T}_0(\boldsymbol{a}_0), \boldsymbol{T}_{0+1}(\boldsymbol{a}_0)), \\ &\qquad \mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_0), \boldsymbol{T}_2(\boldsymbol{a}_0), \boldsymbol{T}_{2+2}(\boldsymbol{a}_0))\} \\ &= \{\mathcal{I}([0\,1\,2], [0\,1\,2], [0\,1\,2]), \mathcal{I}([0\,1\,2], [0\,1\,2], [2\,0\,1]), \mathcal{I}([0\,1\,2], [1\,2\,0], [2\,0\,1])\} \\ &= \{[0\,0\,0\,1\,1\,1\,2\,2\,2], [0\,0\,2\,1\,1\,0\,2\,2\,1], [0\,1\,2\,1\,2\,0\,2\,0\,1]\} \end{aligned}$$

is a $(9, 3, 3)$-PF.

3.3. A decoding algorithm. We next show how, given a Perfect Factor constructed using the above method, a simple decoding algorithm can be devised which reduces decoding the constructed Perfect Factors to decoding the Perfect Factor and the set of rotation vectors used as components in the construction.

Algorithm 3.5. *Suppose c, n, t, v, and A satisfy the conditions of Construction* 3.1 *and B has been constructed from A using Construction* 3.1*. Suppose also that the pair of functions (E_1, E_2) acts as a decoder for A; i.e., if $\boldsymbol{z}$ is a c-ary v-tuple then $0 \leq E_1(\boldsymbol{z}) < c^v/n$ and $0 \leq E_2(\boldsymbol{z}) < n$ and $\boldsymbol{z}$ occurs at position $E_2(\boldsymbol{z})$ in cycle $\boldsymbol{a}_{E_1(\boldsymbol{z})}$ of A.*

We also need to define labelings for the sets U and B (defined in Construction 3.1*). If $0 \leq i < c^{tv}/n^t$, then suppose $i_{t-1}i_{t-2}, \ldots, i_1 i_0$ is the (c^v/n)-ary representation of i (with least significant digit i_0), i.e., $0 \leq i_j < c^v/n$ $(0 \leq j < t)$ and*

$$i = \sum_{j=0}^{t-1} (c^v/n)^j i_j,$$

and let

$$\boldsymbol{u}_i = (\boldsymbol{a}_{i_0}, \boldsymbol{a}_{i_1}, \ldots, \boldsymbol{a}_{i_{t-1}}).$$

It should be clear that $U = \{\boldsymbol{u}_i \ :\ 0 \leq i < c^{tv}/n^t\}$.

Next, if $\boldsymbol{u}_i \in U$, say

$$\boldsymbol{u}_i = (\boldsymbol{a}_{i_0}, \boldsymbol{a}_{i_1}, \ldots, \boldsymbol{a}_{i_{t-1}}),$$

and $\boldsymbol{x}_j \in X$*, say*

$$\boldsymbol{x}_j = (x_0, x_1, \ldots, x_{n^{t-1}/t-1}),$$

then put

$$\boldsymbol{b}_{ij} = \mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_{i_0}), \boldsymbol{T}_{x_0}(\boldsymbol{a}_{i_1}), \boldsymbol{T}_{x_0+x_1}(\boldsymbol{a}_{i_2}), \ldots, \boldsymbol{T}_{x_0+x_1+\cdots+x_{t-2}}(\boldsymbol{a}_{i_{t-1}})),$$

and hence $B = \{\boldsymbol{b}_{ij} : 0 \leq i < c^{tv}/n^t, 0 \leq j < n^{t-1}/t\}$*.*

Define the triple of functions

$$\begin{aligned} F_{11} &: T \rightarrow \{0, 1, \ldots, c^{tv}/n^t - 1\}, \\ F_{12} &: T \rightarrow \{0, 1, \ldots, n^{t-1}/t - 1\}, \\ F_2 &: T \rightarrow \{0, 1, \ldots, nt - 1\} \end{aligned}$$

as follows, where T *is the set of all* c*-ary* tv*-tuples.*

First, suppose $\boldsymbol{y} \in T$ *and suppose*

$$\boldsymbol{y} = \mathcal{I}(\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{t-1}).$$

Next put

$$\boldsymbol{w} = (w_0, w_1, \ldots, w_{t-1}) = (E_2(\boldsymbol{z}_0), E_2(\boldsymbol{z}_1), \ldots, E_2(\boldsymbol{z}_{t-1}))$$

and let

$$\boldsymbol{x}' = (x'_0, x'_1, \ldots, x'_{t-1}) = (w_0 - w_1, w_1 - w_2, \ldots, w_{t-2} - w_{t-1}, w_{t-1} - w_0 - 1).$$

Now $\boldsymbol{x}' \in S$ *(as defined in Construction* 3.1*), and hence suppose*

$$\boldsymbol{x}' = \boldsymbol{T}_r(\boldsymbol{x}_q)$$

for some $\boldsymbol{x}_q \in X$ *(where* $0 \leq r < t$*). We now put* $F_{12}(\boldsymbol{y}) = q$*.*

Next put

$$\boldsymbol{g}' = (g'_0, g'_1, \ldots, g'_{t-1}) = (E_1(\boldsymbol{z}_0), E_1(\boldsymbol{z}_1), \ldots, E_1(\boldsymbol{z}_{t-1}))$$

and let

$$\boldsymbol{g} = (g_0, g_1, \ldots, g_{t-1}) = \boldsymbol{T}_r(\boldsymbol{g}').$$

Finally, put

$$F_{11}(\boldsymbol{y}) = \sum_{i=0}^{t-1} g_i (c^v/n)^i$$

and

$$F_2(\boldsymbol{y}) = tE_2(\boldsymbol{z}_r) - r.$$

THEOREM 3.6. *If* B *and* (F_{11}, F_{12}, F_2) *are defined as in Algorithm* 3.5*, then the pair* $((F_{11}, F_{12}), F_2)$ *is a decoder for* B*; i.e., if* $\boldsymbol{y}$ *is a* c*-ary* tv*-tuple, then* $\boldsymbol{y}$ *occurs at position* $F_2(\boldsymbol{y})$ *in cycle* $\boldsymbol{b}_{F_{11}(\boldsymbol{y}),F_{12}(\boldsymbol{y})}$*.*

Proof. Suppose $\boldsymbol{y}$ $(\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{t-1})$, F_{11}, F_{12}, and F_2 are as in the algorithm. We need to show that $\boldsymbol{y}$ occurs at position $F_2(\boldsymbol{y})$ in cycle $\boldsymbol{b}_{F_{11}(\boldsymbol{y}),F_{12}(\boldsymbol{y})}$.

First observe that

$$F_{11}(\boldsymbol{y}) = \sum_{i=0}^{t-1} g_i (c^v/n)^i$$

and

$$\boldsymbol{x}_{F_{12}(\boldsymbol{y})} = (x_0, x_1, \ldots, x_{t-1}) \in X.$$

Now, by definition,

$$\begin{aligned}
\boldsymbol{b}_{F_{11}(\boldsymbol{y}),F_{12}(\boldsymbol{y})} &= \mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_{g_0}), \boldsymbol{T}_{x_0}(\boldsymbol{a}_{g_1}), \ldots, \boldsymbol{T}_{x_0+x_1+\cdots+x_{t-2}}(\boldsymbol{a}_{g_{t-1}}))\\
&= \mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_{g'_r}), \boldsymbol{T}_{x_0}(\boldsymbol{a}_{g'_{r+1}}), \ldots, \boldsymbol{T}_{x_0+x_1+\cdots+x_{t-2}}(\boldsymbol{a}_{g'_{r-1}}))\\
&\qquad \text{(since } \boldsymbol{g} = \boldsymbol{T}_r(\boldsymbol{g}'))\\
&= \mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_{E_1(\boldsymbol{z}_r)}), \boldsymbol{T}_{x_0}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r+1})}), \ldots, \boldsymbol{T}_{x_0+x_1+\cdots+x_{t-2}}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r-1})}))\\
&\qquad \text{(by the definition of } \boldsymbol{g}')\\
&= \mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_{E_1(\boldsymbol{z}_r)}), \boldsymbol{T}_{x'_r}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r+1})}), \boldsymbol{T}_{x'_r+x'_{r+1}}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r+2})}), \ldots,\\
&\quad \boldsymbol{T}_{x'_r+x'_{r+1}+\cdots+x'_{r-2}}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r-1})}))\\
&\qquad \text{(since } \boldsymbol{x}' = \boldsymbol{T}_r(\boldsymbol{x}_q))\\
&= \mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_{E_1(\boldsymbol{z}_r)}), \boldsymbol{T}_{w_r-w_{r+1}}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r+1})}), \boldsymbol{T}_{w_r-w_{r+2}}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r+2})}), \ldots,\\
&\quad \boldsymbol{T}_{w_r-w_{t-1}}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{t-1})}), \boldsymbol{T}_{w_r-w_0-1}(\boldsymbol{a}_{E_1(\boldsymbol{z}_0)}), \ldots, \boldsymbol{T}_{w_r-w_{r-1}-1}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r-1})}))\\
&\qquad \text{(by the definition of } \boldsymbol{x}')\\
&= \mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_{E_1(\boldsymbol{z}_r)}), \boldsymbol{T}_{E_2(\boldsymbol{z}_r)-E_2(\boldsymbol{z}_{r+1})}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r+1})}), \boldsymbol{T}_{E_2(\boldsymbol{z}_r)-E_2(\boldsymbol{z}_{r+2})}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r+2})}), \ldots,\\
&\quad \boldsymbol{T}_{E_2(\boldsymbol{z}_r)-E_2(\boldsymbol{z}_{t-1})}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{t-1})}), \boldsymbol{T}_{E_2(\boldsymbol{z}_r)-E_2(\boldsymbol{z}_0)-1}(\boldsymbol{a}_{E_1(\boldsymbol{z}_0)}), \ldots,\\
&\quad \boldsymbol{T}_{E_2(\boldsymbol{z}_r)-E_2(\boldsymbol{z}_{r-1})-1}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r-1})}))\\
&\qquad \text{(by the definition of } \boldsymbol{w}).
\end{aligned}$$

Now since $\boldsymbol{z}_i$ occurs at position $E_2(\boldsymbol{z}_i)$ in $\boldsymbol{a}_{E_1(\boldsymbol{x}_i)}$ $(0 \leq i < t)$, we have the following:

- $\boldsymbol{z}_r$ occurs in $\boldsymbol{T}_0(\boldsymbol{a}_{E_1(\boldsymbol{z}_r)})$ at position $E_2(\boldsymbol{z}_r)$,
- $\boldsymbol{z}_{r+1}$ occurs in $\boldsymbol{T}_{E_2(\boldsymbol{z}_r)-E_2(\boldsymbol{z}_{r+1})}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r+1})})$ at position $E_2(\boldsymbol{z}_r)$,
- $\boldsymbol{z}_{r+2}$ occurs in $\boldsymbol{T}_{E_2(\boldsymbol{z}_r)-E_2(\boldsymbol{z}_{r+2})}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r+2})})$ at position $E_2(\boldsymbol{z}_r)$,
- $\boldsymbol{z}_{t-1}$ occurs in $\boldsymbol{T}_{E_2(\boldsymbol{z}_r)-E_2(\boldsymbol{z}_{t-1})}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{t-1})})$ at position $E_2(\boldsymbol{z}_r)$,
- $\boldsymbol{z}_0$ occurs in $\boldsymbol{T}_{E_2(\boldsymbol{z}_r)-E_2(\boldsymbol{z}_0)-1}(\boldsymbol{a}_{E_1(\boldsymbol{z}_0)})$ at position $E_2(\boldsymbol{z}_r) - 1$, and
- $\boldsymbol{z}_{r-1}$ occurs in $\boldsymbol{T}_{E_2(\boldsymbol{z}_r)-E_2(\boldsymbol{z}_{r-1})-1}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r-1})})$ at position $E_2(\boldsymbol{z}_r) - 1$.

Hence $\boldsymbol{y}$ occurs in $\boldsymbol{b}_{F_{11}(\boldsymbol{y}),F_{12}(\boldsymbol{y})}$ at position $tE_2(\boldsymbol{z}_r) - r = F_2(\boldsymbol{y})$, and the result follows. □

3.4. New parameter sets. We conclude our discussion of this method for constructing Perfect Factors by showing how it can be used to construct Perfect Factors with parameters for which the existence question was previously unresolved.

As has already been mentioned, in [7] the necessary conditions of Lemma 1.4 have been shown to be sufficient for the existence of a Perfect Factor when $v < 5$. Construction 3.1 does not help with any of the unresolved parameter sets for $v = 5$, and so we examine the case $v = 6$.

Now, by Theorem 7.1 of [6], Perfect Factors exist for all triples $(n, c, 6)$ satisfying the conditions of Lemma 1.4, with the possible exceptions of the following:

- $n = 10$, $c = 10d$ ($d \geq 1$),
- $n = 12$, $c = 6d$ ($d \geq 1$),
- $n = 15$, $c = 15d$ ($d \geq 1$),
- $n = 20$, $c = 10d$ ($d \geq 1$),
- $n = 30$, $c = 30d$ ($d \geq 1$), and
- $n = 60$, $c = 30d$ ($d \geq 1$).

Next observe that, by Theorem 26 of [7], the following Perfect Factors exist:

- $(6, 6d, 3)$-PFs, $d \geq 1$,
- $(10, 10d, 3)$-PFs, $d \geq 1$, and
- $(30, 30d, 3)$-PFs, $d \geq 1$.

Applying Construction 3.1 to all of these Perfect Factors (in each case with $t = 2$), we obtain Perfect Factors for precisely the parameter sets in the second, fourth, and sixth cases listed above.

This means that the only unresolved cases for $v = 6$ are as follows:

- $n = 10$, $c = 10d$ ($d \geq 1$),
- $n = 15$, $c = 15d$ ($d \geq 1$), and
- $n = 30$, $c = 30d$ ($d \geq 1$).

4. Summary and conclusions. Using recursive methods of construction, we have made further progress toward proving the conjecture of [6], namely, that Perfect Factors exist for all parameter sets satisfying the necessary conditions of Lemma 1.4. All the construction methods in this paper, both for de Bruijn sequences and for Perfect Factors, admit simple methods of decoding, making their use in practical applications advantageous.

Finally, it is interesting to observe that when put together with the de Bruijn sequence construction methods in [8] and [11] (special case of Lemma 5.1) there exists a series of construction methods for building one de Bruijn sequence out of another. If it turns out that some or all of these construction methods have "complexity-preserving properties" (cf. the Lempel construction [5]), then there may exist the means to make further progress with the long-standing problem of discovering for which linear complexities there exist de Bruijn sequences (see, for example, [1]).

Acknowledgment. The author would like to thank an anonymous referee for invaluable comments regarding Construction 2.1 which have both improved and shortened the paper.

REFERENCES

[1] S. Blackburn, T. Etzion, and K. Paterson, *Permutation polynomials, de Bruijn sequences and linear complexity*, J. Combin. Theory Ser. A, to appear.

[2] J. Bondy and U. Murty, *Graph Theory with Applications*, Elsevier, New York, 1976.

[3] J. Burns and C. Mitchell, *Coding schemes for two-dimensional position sensing*, in Cryptography and Coding III, M. Ganley, ed., Oxford University Press, London, 1993, pp. 31–66.

[4] T. Etzion, *Constructions for perfect maps and pseudo-random arrays*, IEEE Trans. Inform. Theory, 34 (1988), pp. 1308–1316.

[5] A. Lempel, *On a homomorphism of the de Bruijn graph and its application to the design of feedback shift registers*, IEEE Trans. Comput., C-19 (1970), pp. 1204–1209.

[6] C. Mitchell, *Constructing c-ary perfect factors*, Designs, Codes and Cryptography, 4 (1994), pp. 341–368.

[7] C. Mitchell, *New c-ary perfect factors in the de Bruijn graph*, in Codes and Cyphers, P. Farrell, ed., Formara Ltd., Southend, 1995, pp. 299–313; proc. of the fourth IMA Conference on Cryptography and Coding, Cirencester, 1993.

[8] C. Mitchell, T. Etzion, and K. Paterson, *A method for constructing decodable de Bruijn sequences*, IEEE Trans. Inform. Theory, 42 (1996), pp. 1472–1478.

[9] C. Mitchell and K. Paterson, *Decoding perfect maps*, Designs, Codes and Cryptography, 4 (1994), pp. 11–30.

[10] K. Paterson, *Perfect maps*, IEEE Trans. Inform. Theory, 40 (1994), pp. 743–753.

[11] K. Paterson, *Perfect factors in the de Bruijn graph*, Designs, Codes and Cryptography, 5 (1995), pp. 115–138.

[12] K. Paterson, *New classes of perfect maps* I, J. Combin. Theory Ser. A, 73 (1996), pp. 302–334.

[13] K. Paterson, *New classes of perfect maps* II, J. Combin. Theory Ser. A, 73 (1996), pp. 335–345.

[14] E. Petriu, *New pseudorandom/natural code conversion method*, Electronics Lett., 24 (1988), pp. 1358–1359.

SIAM J. Discrete Math.
Vol. 10, No. 2, pp. 282–293, May 1997

ON THE DIMENSION OF THE HULL*

NICOLAS SENDRIER†

Abstract. The hull [Assmus, Jr. and Key, *Discrete Math.*, 83 (1990), pp. 161–187], [Assmus, Jr. and Key, *Designs and Their Codes*, Cambridge University Press, 1992, p. 43] of a linear code is defined to be its intersection with its dual. We give here the number of distinct q-ary linear codes which have a hull of given dimension.

We will prove that, asymptotically, the proportion of q-ary codes whose hull has dimension l is a positive constant that depends only on l and q and consequently that the average dimension of the hull is asymptotically a positive constant depending only on q.

Key words. error correcting codes, self-dual codes, weakly self-dual codes, hull

AMS subject classifications. 11T71, 05A15

PII. S0895480195294027

1. Introduction. We will consider linear codes over a finite field $GF(q)$. A $[q;n,k]$ code will be a linear code of length n and dimension k over $GF(q)$.

We will first study in section 2 the properties of the Gaussian binomial coefficients. The coefficient $\begin{bmatrix} n \\ k \end{bmatrix}$ is the number of $[q;n,k]$ codes. The key result of this section is the inversion formula given in Corollary 2.8.

The hull [1] of a linear code is defined to be its intersection with its dual. In section 4 we express the number $A_{n,k,l}$ of $[q;n,k]$ codes whose hull has dimension l in terms of the number of weakly self-dual codes of given parameters, which is given in section 3 (Theorem 3.2, due to Pless [6]). Using the inversion formula of section 2, we obtain an explicit expression of $A_{n,k,l}$ (Theorem 4.5). We then obtain an asymptotic equivalent of $A_{n,k,l}$ for fixed l when n and k go to infinity (Theorem 4.18), and we prove that under the same conditions, the ratio $A_{n,k,l}/\begin{bmatrix} n \\ k \end{bmatrix}$ is equivalent to a constant, depending on q (Theorem 4.19). Finally, we give the average dimension of the hull of a linear code, which is asymptotically equal to $\sum_{i\geq 1} 1/(q^i+1)$.

Most of the results above will be restricted to the case $n \geq 2k$. However, since the hull of a code is equal to the hull of its dual, this assumption can be made without losing any generality.

2. Gaussian binomial coefficients. Most of the results presented here can be found in [4] and [8].

Definition 2.1. *Let n and k be two integers. The* q-ary Gaussian binomial coefficient $\begin{bmatrix} n \\ k \end{bmatrix}$ *is defined by*

$$\begin{bmatrix} n \\ k \end{bmatrix} = \frac{(q^n-1)(q^{n-1}-1)\dots(q^{n-k+1}-1)}{(q^k-1)(q^{k-1}-1)\dots(q-1)} \tag{1}$$

whenever $n \geq k > 0$ with $\begin{bmatrix} n \\ 0 \end{bmatrix} = 1$ and $\begin{bmatrix} n \\ k \end{bmatrix} = 0$ otherwise.

Note that the Gaussian coefficients are connected to the usual binomial coefficients by $\lim_{q\to 1} \begin{bmatrix} n \\ k \end{bmatrix} = \binom{n}{k}$. The coefficient $\begin{bmatrix} n \\ k \end{bmatrix}$ is the number of subspaces of dimension k

* Received by the editors October 3, 1995; accepted for publication (in revised form) April 29, 1996.

http://www.siam.org/journals/sidma/10-2/29402.html

† Projet CODES, INRIA, Domaine de Voluceau, BP 105, 78153 Le Chesnay Cedex, France (nicolas.sendrier@inria.fr).

of a vector space of dimension n over $GF(q)$. More generally, we have the following proposition.

PROPOSITION 2.2. *Let U be a vector space over $GF(q)$ of dimension n and let V be a subspace of U of dimension l. The number of subspaces C of U of dimension k containing V, that is, $V \subset C \subset U$, is equal to $\begin{bmatrix} n-l \\ k-l \end{bmatrix}$.*

Proof. See, for instance, [5, Thm. 4, p. 698]. □

We have the following identities.

PROPOSITION 2.3. *Let $n \geq k \geq i$ be positive integers.*

$$\begin{bmatrix} n \\ k \end{bmatrix} = \begin{bmatrix} n \\ n-k \end{bmatrix}, \tag{2a}$$

$$\begin{bmatrix} n \\ k \end{bmatrix} = \begin{bmatrix} n \\ i \end{bmatrix}\begin{bmatrix} n-i \\ n-k \end{bmatrix} \Big/ \begin{bmatrix} k \\ i \end{bmatrix}, \tag{2b}$$

$$\begin{bmatrix} n \\ k \end{bmatrix} = \frac{q^n-1}{q^{n-k}-1}\begin{bmatrix} n-1 \\ k \end{bmatrix}, \tag{2c}$$

$$\begin{bmatrix} n \\ k \end{bmatrix} = \frac{q^{n-k+1}-1}{q^k-1}\begin{bmatrix} n \\ k-1 \end{bmatrix}, \tag{2d}$$

$$\begin{bmatrix} n \\ k \end{bmatrix} = \begin{bmatrix} n-1 \\ k \end{bmatrix} + q^{n-k}\begin{bmatrix} n-1 \\ k-1 \end{bmatrix}. \tag{2e}$$

2.1. Asymptotic behavior. For all $i > 0$, let $[i] = (q-1)(q^2-1)\dots(q^i-1)$ and $[0] = 1$. Using the fact that $n(n+1)/2 - k(k+1)/2 - (n-k)(n-k+1)/2 = k(n-k)$, we can rewrite (1) as

$$\begin{bmatrix} n \\ k \end{bmatrix} = \frac{[n]}{[k][n-k]} = q^{k(n-k)}\frac{g_{q,n}}{g_{q,k}g_{q,n-k}}, \tag{3}$$

where the sequence $(g_{q,n})_{n\geq 0}$ is defined for all $q > 1$ by

$$g_{q,n} = \prod_{i=1}^{n}\left(1 - \frac{1}{q^i}\right). \tag{4}$$

This sequence is obviously decreasing and positive; we will see that it goes exponentially quickly to its limit. We will first need the following result.

PROPOSITION 2.4 (see [3, Chap. II, p. 106]).

$$\prod_{i\geq 0}(1 + t^i u) = \sum_{n\geq 0}\frac{t^{\binom{n}{2}}u^n}{(1-t)(1-t^2)\dots(1-t^n)}. \tag{5}$$

PROPOSITION 2.5. *The sequence $(g_{q,n})_{n\geq 0}$ is strictly decreasing for $q > 1$. We will denote by $g_{q,\infty}$ its limit when n goes to infinity. We have*

$$\frac{g_{q,\infty}}{g_{q,n}} = \sum_{i\geq 0}\frac{1}{q^{ni}}\frac{(-1)^i}{(q-1)(q^2-1)\dots(q^i-1)}. \tag{6}$$

Proof. By definition (4) of $g_{q,n}$,

$$\frac{g_{q,\infty}}{g_{q,n}} = \prod_{i\geq n+1}\left(1 - \frac{1}{q^i}\right) = \prod_{i\geq 0}\left(1 - \frac{1}{q^{n+1}q^i}\right).$$

We then write (5) with $t = 1/q$ and $u = -1/q^{n+1}$, and we get (6). □

COROLLARY 2.6. *For all integers $n \geq 0$,*

$$1 - \frac{1}{(q-1)q^n} \leq \frac{g_{q,\infty}}{g_{q,n}} \leq 1. \tag{7}$$

Proof. We can rewrite (6) as $\sum_{i\geq 0}(-1)^i G_i$, where $G_i^{-1} = (q-1)\dots(q^i-1)q^{ni}$. The sequence G_i is strictly positive and decreasing for $q > 1$, and thus, from a classical property of alternate series, we have inequalities (7). □

2.2. An inversion formula. A classical inversion formula given, for instance, in [3, p. 143] says that in any commutative ring with identity if for all $n \geq 0$, $u_n = \sum_{k=0}^{n}\binom{n}{k}v_k$, then for all $n \geq 0$ we have $v_n = \sum_{k=0}^{n}\binom{n}{k}(-1)^{n-k}u_k$. A similar identity holds for Gaussian binomial coefficients. To obtain this formula we will first examine how the two bases $(x^n)_{n\geq 0}$ and $(p_n(x))_{n\geq 0}$, where $p_n(x) = (x-1)(x-q)\dots(x-q^{n-1})$, of the vector space of univariate polynomials are related.

PROPOSITION 2.7. *For all integers $n \geq 0$, let $p_n(x) = (x-1)(x-q)\dots(x-q^{n-1})$. We have*

1. $x^n = \sum_{k=0}^{n}\begin{bmatrix}n\\k\end{bmatrix}p_k(x)$,
2. $p_n(x) = \sum_{k=0}^{n}\begin{bmatrix}n\\k\end{bmatrix}(-1)^{n-k}q^{\binom{n-k}{2}}x^k$.

Proof. See [4] and [8]. □

COROLLARY 2.8 (inversion formula). *Let $(u_i)_{i\geq 0}$ and $(v_i)_{i\geq 0}$ be two sequences. For all $k \geq 0$,*

$$\left(\forall l, 0 \leq l \leq k, v_l = \sum_{i=0}^{l}\begin{bmatrix}l\\i\end{bmatrix}u_i\right) \Leftrightarrow \left(\forall l, 0 \leq l \leq k, u_l = \sum_{i=0}^{l}\begin{bmatrix}l\\i\end{bmatrix}(-1)^{l-i}q^{\binom{l-i}{2}}v_i\right). \tag{8}$$

Proof (see [3, pp. 118–119, 143]). We can express $v_l = \sum_{i=0}^{l}\begin{bmatrix}l\\i\end{bmatrix}u_i$ as $U = PV$, where $U = (u_0, u_1, \dots)$, $V = (v_0, v_1, \dots)$, and P is an infinite triangular matrix of general term $\begin{bmatrix}l\\i\end{bmatrix}$. The inverse P^{-1} of P is given for $u_i = p_i(x)$ and $v_i = x^i$ by Proposition 2.7, and its general term is $\begin{bmatrix}l\\i\end{bmatrix}(-1)^{l-i}q^{\binom{l-i}{2}}$. And thus from $V = P^{-1}U$ we obtain $u_l = \sum_{i=0}^{l}\begin{bmatrix}l\\i\end{bmatrix}(-1)^{l-i}q^{\binom{l-i}{2}}v_i$. □

3. Weakly self-dual codes.

DEFINITION 3.1. *A code C is said to be* weakly self dual *(w.s.d.) if $C \subset C^{\perp}$.*

We will denote by $\sigma_{n,k}$ the number of w.s.d. $[q; n, k]$ codes. We have [9, 6, 7].

THEOREM 3.2. *Let n be a positive integer. The number of w.s.d. $[q; n, k]$ codes is equal to*

1. *if n is odd and $k \leq (n-1)/2$*

$$\sigma_{n,k} = \prod_{i=1}^{k}\frac{q^{n-2i+1}-1}{q^i-1},$$

2. *if n and q are even and $k \leq n/2$*

$$\sigma_{n,k} = \frac{q^{n-k}-1}{q^n-1}\prod_{i=1}^{k}\frac{q^{n-2i+2}-1}{q^i-1},$$

3. *if* $((n \equiv 0 \bmod 4$ *and* q *odd*$)$ *or* $(n \equiv 2 \bmod 4$ *and* $q \equiv 1 \bmod 4))$ *and* $k \leq n/2$

$$\sigma_{n,k} = \frac{q^{n/2-k}+1}{q^{n/2}+1} \prod_{i=1}^{k} \frac{q^{n-2i+2}-1}{q^i-1},$$

4. *if* $n \equiv 2 \bmod 4$*,* $q \equiv 3 \bmod 4$*, and* $k \leq n/2-1$

$$\sigma_{n,k} = \frac{q^{n/2-k}-1}{q^{n/2}-1} \prod_{i=1}^{k} \frac{q^{n-2i+2}-1}{q^i-1},$$

5. *else (*k *too large)* $\sigma_{n,k} = 0$.

Proposition 3.3. *Let* $m = \lfloor n/2 \rfloor$*. For all* $k \leq m$*, we have*

$$\sigma_{n,k} = s_{n,k} \frac{q^{k(n-k)}}{q^{k(k+1)/2}} \frac{g_{q^2,m}}{g_{q^2,m-k}\, g_{q,k}},$$

where

$$s_{n,k} = \begin{cases} 1 & \textit{if } n \textit{ is odd,} \\ \dfrac{q^n - q^k}{q^n - 1} & \textit{if } n \textit{ and } q \textit{ are even,} \\ \dfrac{q^{n/2}+\varepsilon q^k}{q^{n/2}+\varepsilon} & \textit{if } n \textit{ is even and } q \textit{ is odd} \end{cases}$$

with $\varepsilon = -1$ *if* $n \equiv 2 \bmod 4$ *and* $q \equiv 3 \bmod 4$ *and* $\varepsilon = 1$ *otherwise.*

Proof. If n is odd, we have $n = 2m+1$, and from Theorem 3.2

$$\sigma_{n,k} = \prod_{i=1}^{k} \frac{q^{n+1-2i}-1}{q^i-1} = \frac{q^{nk+k-2\sum_{i=1}^{k} i}}{q^{k(k+1)/2}} \prod_{i=1}^{k} \frac{1-1/q^{2m+2-2i}}{1-1/q^i}$$
$$= \frac{q^{k(n-k)}}{q^{k(k+1)/2}} \frac{g_{q^2,m}}{g_{q^2,m-k}\, g_{q,k}}. \tag{9}$$

If n is even, we have $n = 2m$. From Theorem 3.2, we find that

$$\sigma_{n,k} = \frac{s_{n,k}}{q^k} \prod_{i=1}^{k} \frac{q^{n-2i+2}-1}{q^i-1} = \frac{s_{n,k}}{q^k} \sigma_{n+1,k},$$

and writing (9) for $\sigma_{n+1,k}$, we get

$$\sigma_{n,k} = \frac{s_{n,k}}{q^k} \frac{q^{k(n+1-k)}}{q^{k(k+1)/2}} \frac{g_{q^2,m}}{g_{q^2,m-k}\, g_{q,k}} = s_{n,k} \frac{q^{k(n-k)}}{q^{k(k+1)/2}} \frac{g_{q^2,m}}{g_{q^2,m-k}\, g_{q,k}}. \qquad \square$$

Proposition 3.4. *For all* $k \leq n/2$*,*

$$1 - \frac{1}{q^{n/2-k}} \leq s_{n,k} \leq 1 + \frac{1}{q^{n/2-k}}.$$

Proof. If $k \leq n/2$, we have the following inequalities:

$$1 - \frac{1}{q^{n/2-k}} \leq \frac{q^{n/2}-q^k}{q^{n/2}-1} \leq \frac{q^n-q^k}{q^n-1} \leq 1 \leq \frac{q^{n/2}+q^k}{q^{n/2}+1} \leq 1 + \frac{1}{q^{n/2-k}},$$

which proves the result. $\square$

4. Hull of a linear code.

Definition 4.1. *The hull of a linear code is defined to be the intersection of the code with its dual.*

We will denote by $\mathcal{H}(C) = C \cap C^\perp$ the hull of a code C.

Lemma 4.2. *Let V be a w.s.d. $[q;n,l]$ code. The number of $[q;n,k]$ codes C such that $V \subset \mathcal{H}(C)$ is equal to $\left[\begin{smallmatrix} n-2l \\ k-l \end{smallmatrix}\right]$.*

Proof. Let C be a $[q;n,k]$ code. We have $V \subset \mathcal{H}(C) = C \cap C^\perp$ if and only if $V \subset C \subset V^\perp$, and from Proposition 2.2 the number of such codes is equal to $\left[\begin{smallmatrix} n-2l \\ k-l \end{smallmatrix}\right]$. □

Lemma 4.3. *Let C be a $[q;n,k]$ code and let $\mathcal{H}(C)$ be its hull, the number of w.s.d. $[q;n,l]$ codes V such that $V \subset \mathcal{H}(C)$ is equal to $\left[\begin{smallmatrix} \dim \mathcal{H}(C) \\ l \end{smallmatrix}\right]$.*

Proof. The hull $\mathcal{H}(C)$ of C is w.s.d.; so are any of its subspaces. The number of subspaces of dimension l of $\mathcal{H}(C)$ is $\left[\begin{smallmatrix} \dim \mathcal{H}(C) \\ l \end{smallmatrix}\right]$, and thus we get the result. □

Proposition 4.4. *For all i, $0 \le i \le k$, let $A_{n,k,i}$ denote the number of $[q;n,k]$ codes whose hull has dimension i. We have, for all l, $0 \le l \le k$,*

$$\begin{bmatrix} n-2l \\ k-l \end{bmatrix} \sigma_{n,l} = \sum_{i=l}^{k} \begin{bmatrix} i \\ l \end{bmatrix} A_{n,k,i}. \tag{10}$$

Proof. Let C be a $[q;n,k]$ code. From Lemma 4.3, $\mathcal{H}(C)$ contains $\left[\begin{smallmatrix} \dim \mathcal{H}(C) \\ l \end{smallmatrix}\right]$ different w.s.d. $[q;n,l]$ codes. From Lemma 4.2 any $[q;n,l]$ w.s.d. code is contained in the hull of $\left[\begin{smallmatrix} n-2l \\ k-l \end{smallmatrix}\right]$ different $[q;n,k]$ codes. Finally, the number of w.s.d. $[q;n,l]$ codes is $\sigma_{n,l}$, and we get

$$\sigma_{n,l} = \left(\sum_{\substack{C \subset GF(q)^n \\ \dim C = k}} \begin{bmatrix} \dim \mathcal{H}(C) \\ l \end{bmatrix} \right) \begin{bmatrix} n-2l \\ k-l \end{bmatrix}^{-1},$$

which leads to the result since $A_{n,k,i}$ is the number of $[q;n,k]$ codes whose hull has dimension i and $\left[\begin{smallmatrix} i \\ l \end{smallmatrix}\right] = 0$ when $i < l$. □

Theorem 4.5. *Let n be a positive integer and let $\sigma_{n,i}$ denote for all i the number of w.s.d. $[q;n,i]$ codes. For all $k \le n/2$ and all $l \le k$, the number of $[q;n,k]$ codes whose hull has dimension l is equal to*

$$A_{n,k,l} = \sum_{i=l}^{k} \begin{bmatrix} n-2i \\ k-i \end{bmatrix} \begin{bmatrix} i \\ l \end{bmatrix} (-1)^{i-l} q^{\binom{i-l}{2}} \sigma_{n,i}. \tag{11}$$

Proof. For all l, $0 \le l \le k$, let

$$\begin{bmatrix} k \\ l \end{bmatrix} V_{n,k,l} = \begin{bmatrix} n-2k+2l \\ l \end{bmatrix} \sigma_{n,k-l} \quad \text{and} \quad \begin{bmatrix} k \\ l \end{bmatrix} U_{n,k,l} = A_{n,k,k-l}. \tag{12}$$

We write (10) with $k-l$ instead of l, and we get for all $l \le k$

$$\begin{bmatrix} k \\ l \end{bmatrix} V_{n,k,l} = \begin{bmatrix} n-2k+2l \\ l \end{bmatrix} \sigma_{n,k-l} = \sum_{i=k-l}^{k} \begin{bmatrix} i \\ k-l \end{bmatrix} A_{n,k,i} = \sum_{j=0}^{l} \begin{bmatrix} k-j \\ k-l \end{bmatrix} \begin{bmatrix} k \\ j \end{bmatrix} U_{n,k,j},$$

where $j = k-i$ in the last summation, and thus for all $l \le k$, using (2b),

$$V_{n,k,l} = \sum_{j=0}^{l} \frac{\left[\begin{smallmatrix} k-j \\ k-l \end{smallmatrix}\right] \left[\begin{smallmatrix} k \\ j \end{smallmatrix}\right]}{\left[\begin{smallmatrix} k \\ l \end{smallmatrix}\right]} U_{n,k,j} = \sum_{j=0}^{l} \begin{bmatrix} l \\ j \end{bmatrix} U_{n,k,j}. \tag{13}$$

We now apply the inversion formula of Corollary 2.8 to (13), and we have for all $l \leq k$

$$\begin{aligned} \frac{A_{n,k,k-l}}{\begin{bmatrix}k\\l\end{bmatrix}} &= U_{n,k,l} = \sum_{j=0}^{l} \begin{bmatrix}l\\j\end{bmatrix} (-1)^{l-j} q^{\binom{l-j}{2}} V_{n,k,j} \\ &= \sum_{i=k-l}^{k} \begin{bmatrix}l\\k-i\end{bmatrix} (-1)^{l-k+i} q^{\binom{l-k+i}{2}} \frac{\begin{bmatrix}n-2i\\k-i\end{bmatrix}\sigma_{n,i}}{\begin{bmatrix}k\\i\end{bmatrix}}, \end{aligned} \tag{14}$$

where $i = k - j$ in the last summation. If we then replace $k - l$ by l in (14) we obtain for all $l \leq k$

$$A_{n,k,l} = \sum_{i=l}^{k} \begin{bmatrix}n-2i\\k-i\end{bmatrix} \frac{\begin{bmatrix}k-l\\k-i\end{bmatrix}\begin{bmatrix}k\\l\end{bmatrix}}{\begin{bmatrix}k\\i\end{bmatrix}} (-1)^{i-l} q^{\binom{i-l}{2}} \sigma_{n,i} = \sum_{i=l}^{k} \begin{bmatrix}n-2i\\k-i\end{bmatrix} \begin{bmatrix}i\\l\end{bmatrix} (-1)^{i-l} q^{\binom{i-l}{2}} \sigma_{n,i}. \qquad \square$$

The result above will be practically useful only when $k - l$ is small. When the number of terms in the summation (11) gets large, the formula becomes intractable. Furthermore, it gives no precise idea of the asymptotic behavior of $A_{n,k,l}$ when n and k get large.

4.1. Asymptotic behavior. For all l, $0 \leq l \leq k$, let

$$b_{n,k,l} = \frac{\begin{bmatrix}n-2k+2l\\l\end{bmatrix}\sigma_{n,k-l} q^{k(k+1)/2}}{\begin{bmatrix}n\\k\end{bmatrix}\begin{bmatrix}k\\l\end{bmatrix}} \quad \text{and} \quad a_{n,k,l} = \frac{A_{n,k,k-l} q^{k(k+1)/2}}{\begin{bmatrix}n\\k\end{bmatrix}\begin{bmatrix}k\\l\end{bmatrix}}.$$

PROPOSITION 4.6. *Let $m = \lfloor n/2 \rfloor$. For all $k \leq m$ and for all l, $0 \leq l \leq k$, we have*

$$b_{n,k,l} = q^{l(l+1)/2} \frac{g_{q^2,m}\, g_{q,n-2k+2l}\, g_{q,n-k}}{g_{q^2,m-k+l}\, g_{q,n-2k+l}\, g_{q,n}} s_{n,k-l}.$$

Proof. From Proposition 3.3, we have

$$\sigma_{n,k-l} = \frac{q^{(k-l)(n-k+l)}}{q^{(k-l)(k-l+1)/2}} \frac{g_{q^2,m}}{g_{q^2,m-k+l}\, g_{q,k-l}} s_{n,k-l}.$$

From (3) we have

$$\begin{bmatrix}n-2k+2l\\l\end{bmatrix} = q^{l(n-2k+l)} \frac{g_{q,n-2k+2l}}{g_{q,l}\, g_{q,n-2k+l}}$$

and

$$\begin{bmatrix}n\\k\end{bmatrix}\begin{bmatrix}k\\l\end{bmatrix} = q^{k(n-k)+l(k-l)} \frac{g_{q,n}\, g_{q,k}}{g_{q,k}\, g_{q,n-k}\, g_{q,l}\, g_{q,k-l}},$$

and thus, using $k(k+1)/2 - l(l+1)/2 = (k-l)(k-l+1)/2 + l(k-l)$,

$$\frac{\begin{bmatrix}n-2k+2l\\l\end{bmatrix}\sigma_{n,k-l}}{\begin{bmatrix}n\\k\end{bmatrix}\begin{bmatrix}k\\l\end{bmatrix}} = \frac{q^{l(l+1)/2}}{q^{k(k+1)/2}} \frac{g_{q^2,m}\, g_{q,n-2k+2l}\, g_{q,n-k}}{g_{q^2,m-k+l}\, g_{q,n-2k+l}\, g_{q,n}} s_{n,k-l}. \qquad \square$$

PROPOSITION 4.7. *For all l, $0 \leq l \leq k$, we have*

$$\sum_{i=0}^{l} \begin{bmatrix}l\\i\end{bmatrix} a_{n,k,i} = b_{n,k,l}. \tag{15}$$

Proof. We have $a_{n,k,l} = U_{n,k,l} q^{k(k+1)/2} / \left[\begin{smallmatrix} n \\ k \end{smallmatrix}\right]$ and $b_{n,k,l} = V_{n,k,l} q^{k(k+1)/2} / \left[\begin{smallmatrix} n \\ k \end{smallmatrix}\right]$, where $U_{n,k,l}$ and $V_{n,k,l}$ are defined by (12). Then (13) will give

$$\frac{\left[\begin{smallmatrix} n \\ k \end{smallmatrix}\right] b_{n,k,l}}{q^{k(k+1)/2}} = \sum_{i=0}^{l} \begin{bmatrix} l \\ i \end{bmatrix} \frac{\left[\begin{smallmatrix} n \\ k \end{smallmatrix}\right] a_{n,k,i}}{q^{k(k+1)/2}}. \qquad \square$$

Lemma 4.8. *Let $m = \lfloor n/2 \rfloor$. For all $i \leq m$,*

$$\frac{g_{q^2,m} g_{q,n-i}}{g_{q^2,m-i} g_{q,n}} \leq 1.$$

Proof. By the definition of the sequences $g_{q,n}$ and $g_{q^2,n}$, we have

$$\frac{g_{q^2,m} g_{q,n-i}}{g_{q^2,m-i} g_{q,n}} = \frac{\prod_{j=m-i+1}^{m} 1 - 1/q^{2j}}{\prod_{j=n-i+1}^{n} 1 - 1/q^{j}} = \prod_{j=1}^{i} \frac{1 - 1/q^{2m-2j+2}}{1 - 1/q^{n-j+1}}.$$

For all j, $1 \leq j \leq i$, and whatever is the parity of n, we have $1 - 1/q^{2m-2j+2} \leq 1 - 1/q^{n-j+1}$, which gives us the result. □

Lemma 4.9. *For all integers $0 \leq u \leq v$ and $i \geq 0$, we have*

$$\frac{g_{q,u+i}}{g_{q,u}} \leq \frac{g_{q,v+i}}{g_{q,v}}.$$

Proof. Since $u \leq v$, we have $1 - 1/q^{u+j} \leq 1 - 1/q^{v+j}$ for all j, $1 \leq j \leq i$, and thus

$$\frac{g_{q,u+i}}{g_{q,u}} = \prod_{j=1}^{i} \left(1 - \frac{1}{q^{u+j}}\right) \leq \prod_{j=1}^{i} \left(1 - \frac{1}{q^{v+j}}\right) = \frac{g_{q,v+i}}{g_{q,v}}. \qquad \square$$

Lemma 4.10. *Let $m = \lfloor n/2 \rfloor$. For all $i \leq m$,*

$$\frac{g_{q^2,\infty}\, g_{q,n-2i}}{g_{q^2,m-i}\, g_{q,\infty}} \geq 1.$$

Proof. We have, by definition,

$$\frac{g_{q^2,\infty}\, g_{q,n-2i}}{g_{q^2,m-i}\, g_{q,\infty}} = \frac{\prod_{j>m-i} (1 - 1/q^{2j})}{\prod_{j>n-2i} (1 - 1/q^{j})} = \prod_{j>0} \frac{1 - 1/q^{2m-2i+2j}}{1 - 1/q^{n-2i+j}}.$$

For all $j > 0$ and whatever is the parity of n, we have $1 - 1/q^{2m-2i+2j} \geq 1 - 1/q^{n-2i+j}$. □

Proposition 4.11. *Let $\delta_{n,k,l} = q^{l(l+1)/2} - b_{n,k,l}$. For all l, $0 \leq l \leq k$, we have*

$$-\frac{q^{l(l-1)/2}}{q^{n/2-k}} \leq \delta_{n,k,l} \leq \frac{q}{q-1} \frac{q^{l(l-1)/2}}{q^{n/2-k}}.$$

Proof. Let $m = \lfloor n/2 \rfloor$. We have

$$b_{n,k,l} = q^{l(l+1)/2} \frac{g_{q^2,m}\, g_{q,n-2k+2l}\, g_{q,n-k}}{g_{q^2,m-k+l}\, g_{q,n-2k+l}\, g_{q,n}} s_{n,k-l}$$

using Lemma 4.8 (with $i = k - l$), Lemma 4.9 (with $u = n - 2k + l$, $v = n - k$, and $i = l$), Lemma 4.10 (with $i = k - l$), and the fact that $g_{q,n}$ and $g_{q^2,n}$ are decreasing. We then obtain

$$\frac{g_{q,\infty}}{g_{q,n-2k+l}} s_{n,k-l} \leq \frac{b_{n,k,l}}{q^{l(l+1)/2}} \leq s_{n,k-l}.$$

From Corollary 2.6 and Proposition 3.4, we get

$$L = \left(1 - \frac{1}{(q-1)q^{n-2k+l}}\right)\left(1 - \frac{1}{q^{n/2-k+l}}\right) \leq \frac{b_{n,k,l}}{q^{l(l+1)/2}} \leq 1 + \frac{1}{q^{n/2-k+l}}.$$

Let's consider the left-hand term L of this inequality

$$\begin{aligned} L &\geq 1 - \frac{1}{(q-1)q^{n-2k+l}} - \frac{1}{q^{n/2-k+l}} \\ &\geq 1 - \frac{1}{(q-1)q^{n/2-k+l}} - \frac{1}{q^{n/2-k+l}} = 1 - \frac{q}{(q-1)q^{n/2-k+l}}, \end{aligned}$$

and finally, we have

$$1 - \frac{q}{(q-1)q^{n/2-k+l}} \leq \frac{b_{n,k,l}}{q^{l(l+1)/2}} \leq 1 + \frac{1}{q^{n/2-k+l}},$$

which concludes the proof. □

Proposition 4.12. *Let $(u_l)_{l\geq 0}$ be the sequence solution of $\sum_{i=0}^{l} \begin{bmatrix} l \\ i \end{bmatrix} u_i = q^{l(l+1)/2}$ and let $\gamma_{n,k,l} = u_l - a_{n,k,l}$. For all l, $0 \leq l \leq k$, we have $\sum_{i=0}^{l} \begin{bmatrix} l \\ i \end{bmatrix} \gamma_{n,k,i} = \delta_{n,k,l}$.*

This proposition states that equation (15) can be cut into two pieces. We have for all l, $0 \leq l \leq k$,

$$\left\{ \begin{array}{lclcl} a_{n,k,l} & = & u_l & - & \gamma_{n,k,l} \\ b_{n,k,l} & = & q^{l(l+1)/2} & - & \delta_{n,k,l} \end{array} \right. \text{ and } \left\{ \begin{array}{lcl} \sum_{i=0}^{l} \begin{bmatrix} l \\ i \end{bmatrix} u_i & = & q^{l(l+1)/2} \\ \sum_{i=0}^{l} \begin{bmatrix} l \\ i \end{bmatrix} \gamma_{n,k,i} & = & \delta_{n,k,l} \end{array} \right. .$$

Using these equations, we will first find a closed expression for u_l (Corollary 4.15) and second see how $\gamma_{n,k,l}$ and u_l compare (Corollary 4.17).

4.1.1. First-order term. We now wish to solve the equation $\sum_{i=0}^{l} \begin{bmatrix} l \\ i \end{bmatrix} u_i = q^{l(l+1)/2}$. By use of the inversion formula (8) we get

$$u_l = \sum_{i=0}^{l} \begin{bmatrix} l \\ i \end{bmatrix} (-1)^{l-i} q^{\binom{i+1}{2} + \binom{l-i}{2}}. \tag{16}$$

Lemma 4.13. *For all $l \geq 0$, we have*

$$w_l = \sum_{i=0}^{l} \begin{bmatrix} l \\ i \end{bmatrix} (-1)^{l-i} q^{\binom{i+1}{2} + \binom{l-i+1}{2}} = \left\{ \begin{array}{ll} 0 & \textit{if } l \textit{ is odd,} \\ u_l & \textit{if } l \textit{ is even.} \end{array} \right. \tag{17}$$

Proof. From $\binom{l-i+1}{2} = \binom{l-i}{2} + l - i$ and (16) we get

$$w_l = \sum_{i=0}^{l} \begin{bmatrix} l \\ i \end{bmatrix} (-1)^{l-i} q^{\binom{i+1}{2} + \binom{l-i+1}{2}} = u_l + \sum_{i=0}^{l-1} \begin{bmatrix} l \\ i \end{bmatrix} (-1)^{l-i} q^{\binom{i+1}{2} + \binom{l-i}{2}} (q^{l-i} - 1),$$

and from (2c) we have $\left[{l \atop i}\right](q^{l-i}-1) = \left[{l-1 \atop i}\right](q^l-1)$. Thus

$$w_l = u_l + (q^l-1)\sum_{i=0}^{l-1}\begin{bmatrix} l-1 \\ i \end{bmatrix}(-1)^{l-i}q^{\binom{i+1}{2}+\binom{l-i}{2}} = u_l-(q^l-1)w_{l-1}.$$

Now $w_l = 0$ when l is odd because the terms for i and $l-i$ in the sum are the opposite of each other. And when l is even, $l-1$ is odd and $w_l = u_l-(q^l-1)w_{l-1} = u_l$. □

Proposition 4.14. *Let u_l be the sequence defined by*

$$\sum_{i=0}^{l}\begin{bmatrix} l \\ i \end{bmatrix}u_i = q^{l(l+1)/2}.$$

We have for all $l \geq 0$

$$u_l = \prod_{\substack{0\leq i\leq l\\ i\ even}} q^i \prod_{\substack{0\leq i\leq l\\ i\ odd}}(q^i-1)$$

or, equivalently, $u_0 = 1$, and for all $l > 0$

$$u_l = \begin{cases} u_{l-1}q^l & \text{if } l \text{ is even,}\\ u_{l-1}(q^l-1) & \text{if } l \text{ is odd.}\end{cases}$$

Proof. We will prove the result by induction. Clearly $u_0 = 1$. From (16) and (2e), we have

$$\begin{aligned} u_l &= \sum_{i=0}^{l}\left(\begin{bmatrix} l-1 \\ i-1 \end{bmatrix}q^{l-i} + \begin{bmatrix} l-1 \\ i \end{bmatrix}\right)(-1)^{l-i}q^{\binom{i+1}{2}+\binom{l-i}{2}} \\ &= \sum_{i=1}^{l}\begin{bmatrix} l-1 \\ i-1 \end{bmatrix}(-1)^{l-i}q^{l-i}q^{\binom{i+1}{2}+\binom{l-i}{2}} + \sum_{i=0}^{l-1}\begin{bmatrix} l-1 \\ i \end{bmatrix}(-1)^{l-i}q^{\binom{i+1}{2}+\binom{l-i}{2}} \\ &= q^l\underbrace{\sum_{i=0}^{l-1}\begin{bmatrix} l-1 \\ i \end{bmatrix}(-1)^{l-i-1}q^{\binom{i+2}{2}-(i+1)+\binom{l-i-1}{2}}}_{=u_{l-1}} - \underbrace{\sum_{i=0}^{l-1}\begin{bmatrix} l-1 \\ i \end{bmatrix}(-1)^{l-i-1}q^{\binom{i+1}{2}+\binom{l-i}{2}}}_{=w_{l-1}}. \end{aligned}$$

Thus we have $u_l = q^l u_{l-1} - w_{l-1}$, where w_l is defined by (17). Lemma 4.13 then gives $u_l = q^l u_{l-1}$ if l is even and $u_l = (q^l-1)u_{l-1}$ if l is odd. □

Corollary 4.15. *For all $l \geq 0$, we have*

$$u_l = q^{l(l+1)/2}\frac{g_{q,l}}{g_{q^2,\lfloor l/2\rfloor}}.$$

Proof. From Proposition 4.14, we have

$$u_l = q^{l(l+1)/2}\prod_{\substack{0\leq i\leq l\\ i\ odd}}\left(1-\frac{1}{q^i}\right) = q^{l(l+1)/2}\prod_{0\leq i\leq l}\left(1-\frac{1}{q^i}\right)\prod_{0\leq i\leq\lfloor l/2\rfloor}\left(1-\frac{1}{q^{2i}}\right)^{-1},$$

which means exactly $u_l = q^{l(l+1)/2}g_{q,l}/g_{q^2,\lfloor l/2\rfloor}$. □

4.1.2. Second-order term.

PROPOSITION 4.16. *For all l, $0 \le l \le k$, we have*

$$|\gamma_{n,k,l}| \le (l+1)\frac{q}{q-1}\frac{q^{l(l-1)/2}}{g_{q,\lfloor l/2\rfloor}q^{n/2-k}}.$$

Proof. The inversion formula gives

$$\gamma_{n,k,l} = \sum_{i=0}^{l}\begin{bmatrix} l \\ i \end{bmatrix}(-1)^{l-i}q^{\binom{l-i}{2}}\delta_{n,k,i},$$

$$|\gamma_{n,k,l}| \le \sum_{i=0}^{l}\begin{bmatrix} l \\ i \end{bmatrix}q^{\binom{l-i}{2}}|\delta_{n,k,i}| \le \frac{q}{(q-1)q^{n/2-k}}\sum_{i=0}^{l}\begin{bmatrix} l \\ i \end{bmatrix}q^{\binom{l-i}{2}+\binom{i}{2}},$$

and since $\binom{l-i}{2}+\binom{i}{2} = \binom{l}{2} - i(l-i)$, we have

$$|\gamma_{n,k,l}| \le \frac{q}{q-1}\frac{q^{\binom{l}{2}}}{q^{n/2-k}}\sum_{i=0}^{l}\begin{bmatrix} l \\ i \end{bmatrix}\frac{1}{q^{i(l-i)}} = \frac{q}{q-1}\frac{q^{\binom{l}{2}}}{q^{n/2-k}}\sum_{i=0}^{l}\frac{g_{q,l}}{g_{q,i}\,g_{q,l-i}}.$$

Finally, we have $g_{q,l} \le g_{q,i}$ and $g_{q,l} \le g_{q,l-i}$. Thus $g_{q,l}/(g_{q,i}g_{q,l-i}) \le \min(1/g_{q,i}, 1/g_{q,l-i}) \le 1/g_{q,\lfloor l/2\rfloor}$ and

$$|\gamma_{n,k,l}| \le \frac{q}{q-1}\frac{q^{\binom{l}{2}}}{q^{n/2-k}}\frac{l+1}{g_{q,\lfloor l/2\rfloor}}. \qquad \square$$

COROLLARY 4.17. *There exists a constant K depending only on q such that for all l, $0 \le l \le k$,*

$$\frac{|\gamma_{n,k,l}|}{u_l} \le K\frac{k}{q^{n/2-k+l}}.$$

Proof. From Proposition 4.16 and Corollary 4.15, we can easily find such a constant. $\square$

4.2. Dimension of the hull.

THEOREM 4.18. *Let n be a positive integer. For all $k \le n/2$ and all $l \le k$, the number of $[q; n, k]$ codes whose hull has dimension l is equal to*

$$A_{n,k,l} = \begin{bmatrix} n \\ k \end{bmatrix}\frac{1}{q^{l(l+1)/2}}\frac{g_{q,k}}{g_{q^2,\lfloor (k-l)/2\rfloor}g_{q,l}}\left(1 + O\left(\frac{k}{q^{n/2-l}}\right)\right).$$

Proof. By definition, we have $A_{n,k,k-l}q^{k(k+1)/2} = \begin{bmatrix} n \\ k \end{bmatrix}\begin{bmatrix} k \\ l \end{bmatrix}a_{n,k,l}$. We have $a_{n,k,k-l} = u_{k-l} + \gamma_{n,k,k-l}$, and thus from Corollary 4.17 we get

$$A_{n,k,l}q^{k(k+1)/2} = \begin{bmatrix} n \\ k \end{bmatrix}\begin{bmatrix} k \\ l \end{bmatrix}u_{k-l}\left(1 + O\left(\frac{k}{q^{n/2-l}}\right)\right).$$

Finally, from Corollary 4.15 and (3) we obtain the result. $\square$

This result gives an accurate estimate as long as l is not close to $n/2$. This will always be the case if $n-2k$ is large. If $n-2k$ is small and l is close to k, then formula (11) of Theorem 4.5 will apply.

The fraction $A_{n,k,l}/\left[{n \atop k}\right]$ represents the proportion of $[q;n,k]$ codes whose hull has a given dimension l. The next theorem states that this ratio is independent of n and k when these numbers grow.

THEOREM 4.19. *Let $A_{n,k,l}$ denote the number of $[q;n,k]$ codes whose hull has dimension l. For all l, the proportion $A_{n,k,l}/\left[{n \atop k}\right]$ of such codes is convergent when n and k go to infinity. We will denote by R_l this limit. We have for all $l \geq 0$,*

$$R_l = \frac{R_0}{g_{q,l}q^{l(l+1)/2}} = \frac{R_0}{(q-1)(q^2-1)\dots(q^l-1)}, \text{ where } R_0 = \frac{g_{q,\infty}}{g_{q^2,\infty}}.$$

Proof. It is immediate from the application of Theorem 4.18. □

COROLLARY 4.20. *The average dimension of the hull of a q-ary linear code is asymptotically equal to*

$$\sum_{l\geq 1} lR_l = \sum_{i\geq 1} \frac{1}{q^i+1}.$$

Proof. Applying (5) with $t = 1/q$ and $u = tz$, we obtain

$$\prod_{i\geq 0}\left(1+\frac{z}{q^{i+1}}\right) = \sum_{l\geq 0}\frac{z^l}{(q-1)(q^2-1)\dots(q^l-1)},$$

from which we obtain the series

$$\mathcal{R}(z) = \sum_{l\geq 0} R_l z^l = R_0 \prod_{i\geq 1}\left(1+\frac{z}{q^i}\right).$$

(Note that when $z = 1$ we have

$$\mathcal{R}(1) = \sum_{l\geq 0} R_l = R_0\prod_{i\geq 1}\left(1+\frac{1}{q^i}\right) = R_0\frac{\prod_{i\geq 1}(1-1/q^{2i})}{\prod_{i\geq 1}(1-1/q^i)} = R_0\frac{g_{q^2,\infty}}{g_{q,\infty}} = 1,$$

which was predictable since the sum of the ratios R_l for all l must be one.) The average dimension of the hull can be obtained by differentiation of the series $\mathcal{R}(z)$,

$$\frac{d\mathcal{R}(z)}{dz} = \sum_{l\geq 1} lR_l z^{l-1} = \sum_{i\geq 1}\frac{1}{q^i}\frac{\mathcal{R}(z)}{1+z/q^i} = \mathcal{R}(z)\sum_{i\geq 1}\frac{1}{q^i+z},$$

and thus, for $z = 1$,

$$\sum_{l\geq 1} lR_l = \sum_{i\geq 1}\frac{1}{q^i+1}. \qquad \square$$

5. Conclusion. We proved that the expected dimension of the hull of a random $[q;n,k]$ code is a constant, given by Corollary 4.20, when n and k go to infinity. Furthermore, this constant is accurate even for relatively small values of n and k. For instance, in the binary case with $n = 40$ and $k = 20$ the average dimension of the hull computed by the asymptotic formula has a relative difference of 10^{-6} with the exact value computed by (11). This figure drops to 10^{-15} when $n = 2k = 100$.

A basis of $\mathcal{H}(C) = C \cap C^{\perp}$ can be obtained by computing the null space of the matrix whose columns are a basis of C followed by a basis of $C^{\perp}$ (see [11, p. 199]).

Thus, computing the hull of an $[n,k]$ linear code is equivalent to a Gaussian elimination on an $n \times n$ matrix.

The weight distribution of the hull is an invariant in the sense that it will not vary when the support of the code is permuted. This invariant will be, in general, easy to compute, first because the hull will have a small dimension and second because a basis of the hull is easy to obtain. Furthermore, this invariant has a good chance, at least for small values of q, to be different for two nonequivalent codes. This fact is used in [10] to obtain an algorithm for finding the permutation between two equivalent random binary linear codes, which is efficient even for codes of length larger than 1000. This algorithm needs an invariant that must be computed many times for many different codes and would not have been tractable with the usual invariants (minimum distance, weight distribution ...).

Acknowledgments. The author wishes to thank Ph. Flajolet for his indications of great import. He is also particularly thankful to E. F. Assmus, Jr. for his valuable advice and for his interest in this work.

REFERENCES

[1] E. Assmus, Jr. and J. Key, *Affine and projective planes*, Discrete Math., 83 (1990), pp. 161–187.

[2] E. Assmus, Jr. and J. Key, *Designs and their Codes*, Cambridge University Press, London, 1992.

[3] L. Comtet, *Advanced Combinatorics*, D. Reidel, Dordrecht, the Netherlands, 1974.

[4] J. Goldman and G.-C. Rota, *The number of subspaces of a vector space*, in Recent Progress in Combinatorics, W. Tutte, ed., Academic Press, New York, 1969, pp. 75–83.

[5] F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*, North–Holland, Amsterdam, 1977.

[6] V. Pless, *The number of isotropic subspaces in a finite geometry*, Rend. Sc. Fis. Mat. e Nat., Accad. Naz. Lincie, Ser. VIII, 39 (1965), pp. 418–421.

[7] V. Pless, *On the uniqueness of the Golay codes*, J. Combin. Theory, 5 (1968), pp. 215–228.

[8] G. Pólya and G. Alexanderson, *Gaussian binomial coefficients*, Elem. Math., 26 (1971), pp. 102–109.

[9] B. Segre, *Le geometrie di Galois*, Annali di Mat. Pura Appl., Ser. 4a, 49 (1959), pp. 1–96.

[10] N. Sendrier, *Un algorithme pour trouver la permutation entre deux codes binaires équivalents*, Rapport de Recherche 2853, INRIA, France, 1996.

[11] G. Strang, *Linear Algebra and its Applications*, 3rd ed., Saunders HBJ, 1988.

SIAM J. DISCRETE MATH.
Vol. 10, No. 2, pp. 294–308, May 1997

A MIN–MAX THEOREM FOR BISUBMODULAR POLYHEDRA*

SATORU FUJISHIGE†

Abstract. For a family $\mathcal{F} \subseteq 3^E$ closed with respect to the reduced union and intersection and for a bisubmodular function $f : \mathcal{F} \to \mathbf{R}$ with $(\emptyset, \emptyset) \in \mathcal{F}$ and $f(\emptyset, \emptyset) = 0$, the bisubmodular polyhedron associated with $(\mathcal{F}, f)$ is given by

$$\mathrm{P}_*(f) = \{x \mid x \in \mathbf{R}^E \quad \forall (X,Y) \in \mathcal{F} : x(X,Y) \leq f(X,Y)\},$$

where $x(X,Y) = \sum_{e \in X} x(e) - \sum_{e \in Y} x(e)$. We show a min–max relation that characterizes the distance between $\mathrm{P}_*(f)$ and a given point x^0 with respect to the l_1 norm: for any vector $x^0 \in \mathbf{R}^E$,

$$\min\left\{\sum_{e \in E} |x(e) - x^0(e)| \;\middle|\; x \in \mathrm{P}_*(f)\right\} = \max\{x^0(X,Y) - f(X,Y) \mid (X,Y) \in \mathcal{F}\},$$

where if f is integer valued and x^0 is integral, then the minimum is attained by an integral $x \in \mathrm{P}_*(f)$. This is in a sense equivalent to but is in a nicer symmetric form than a min–max theorem of Cunningham and Green-Krótki [*Combinatorica*, 11 (1991), pp. 219–230] shown to be associated with b-matching degree-sequence polyhedra and generalizes the well-known min–max theorem concerning a vector reduction of polymatroids and submodular systems. We also give an application of the theorem to a separable convex optimization problem on bisubmodular polyhedra.

Key words. bisubmodular functions, bisubmodular polyhedra, min–max theorem, submodular functions

AMS subject classifications. 52B40, 90C27, 52A41

PII. S0895480194264344

1. Introduction.

Bisubmodular functions have recently been investigated as a generalization of ordinary submodular functions (see [2], [7], [12], [14], [15], and [16]). A characterization of b-matching degree-sequence polyhedra is nicely given by means of bisubmodular functions in [8].

A bisubmodular function is a generalization of an ordinary submodular function, and some of the results on submodular functions can naturally be generalized to those on bisubmodular functions. In this paper we show a fundamental min–max theorem for bisubmodular polyhedra associated with bisubmodular functions with respect to the l_1 norm. The form of this min–max theorem is not straightforwardly anticipated from those of ordinary submodular functions. The theorem is in a sense equivalent to but is in a nicer symmetric form than a min–max theorem given by Cunningham and Green-Krótki [8] associated with b-matching degree-sequence polyhedra and generalizes the well-known min–max theorem concerning a vector reduction of polymatroids and submodular systems [11], [12].

In section 2 we give some definitions and preliminaries, which generalize those for polymatroids and submodular systems. We show a min–max theorem for bisubmodular polyhedra in section 3. Section 4 furnishes some greedy-type monotone algorithms for solving the minimization problem associated with the min–max theorem. In section 5 we also show an application of the min–max theorem to a separable convex optimization problem over a bisubmodular polyhedron (cf. [3], [4]).

* Received by the editors March 9, 1994; accepted for publication (in revised form) May 15, 1996. This research was supported by the Alexander-von-Humboldt Foundation and Sonderforschunsbereich 303 (DFG), Germany.

http://www.siam.org/journals/sidma/10-2/26434.html

† Institute of Socio-Economic Planning, University of Tsukuba, Tsukuba, Ibaraki 305, Japan (fujishig@shako.sk.tsukuba.ac.jp).

2. Definitions and preliminaries. For a finite nonempty set E define

$$3^E = \{(X,Y) \,|\, X,Y \subseteq E, X \cap Y = \emptyset\}. \tag{1}$$

Note that each element $(X,Y) \in 3^E$ can be made to correspond one-to-one to its characteristic vector $\chi_{(X,Y)} \in \{0,\pm 1\}^E$, where

$$\chi_{(X,Y)}(e) = \begin{cases} 1 & \text{if } e \in X, \\ -1 & \text{if } e \in Y, \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

for each $e \in E$. We call an element of 3^E a *signed subset* of E. For any $(X_i,Y_i) \in 3^E$ $(i=1,2)$ we write $(X_1,Y_1) \sqsubseteq (X_2,Y_2)$ if $X_1 \subseteq X_2$ and $Y_1 \subseteq Y_2$. Also, we write $(X_1,Y_1) \sqsubset (X_2,Y_2)$ if $(X_1,Y_1) \sqsubseteq (X_2,Y_2)$ and $(X_1,Y_1) \neq (X_2,Y_2)$. The binary relation $\sqsubseteq$ is a partial order on 3^E.

We consider two binary operations—$\sqcup$ (*reduced union*) and $\sqcap$ (*intersection*)—on 3^E, defined as follows. For any $(X_i,Y_i) \in 3^E$ $(i=1,2)$,

$$(X_1,Y_1) \sqcup (X_2,Y_2) = ((X_1 \cup X_2) - (Y_1 \cup Y_2), (Y_1 \cup Y_2) - (X_1 \cup X_2)), \tag{3}$$

$$(X_1,Y_1) \sqcap (X_2,Y_2) = (X_1 \cap X_2, Y_1 \cap Y_2). \tag{4}$$

Let $\mathcal{F}$ be a family of signed subsets of E that is closed with respect to the reduced union $\sqcup$ and the intersection $\sqcap$. We call such a family $\mathcal{F}$ a $\{\sqcup,\sqcap\}$-*closed family* (or a *signed ring family*). A function $f : \mathcal{F} \to \mathbf{R}$ is a *bisubmodular function* if for each $(X_i,Y_i) \in \mathcal{F}$ $(i=1,2)$ we have

$$f(X_1,Y_1) + f(X_2,Y_2) \geq f((X_1,Y_1) \sqcup (X_2,Y_2)) + f((X_1,Y_1) \sqcap (X_2,Y_2)). \tag{5}$$

In the following we assume that $(\emptyset,\emptyset) \in \mathcal{F}$ and $f(\emptyset,\emptyset) = 0$. Then the pair $(\mathcal{F},f)$ is called a *bisubmodular system* on E (see [2]). When $\mathcal{F} = 3^E$, a bisubmodular system is called a polypseudomatroid [7], [12] (also see [5], [6], [9], [10], [14], [15], and [16] for related concepts).

It should be noted that the argument throughout this paper is valid when $\mathbf{R}$ is any totally ordered additive group such as the sets of reals, rationals, and integers.

The *bisubmodular polyhedron* $\mathrm{P}_*(f)$ associated with the bisubmodular system $(\mathcal{F},f)$ on E is given by

$$\mathrm{P}_*(f) = \{x \,|\, x \in \mathbf{R}^E \quad \forall (X,Y) \in \mathcal{F} : x(X,Y) \leq f(X,Y)\} \tag{6}$$

(see [6]), where for any $X \subseteq E$ $x(X) = \sum_{e \in X} x(e)$, $x(\emptyset) = 0$, and for any $(X,Y) \in 3^E$,

$$x(X,Y) = x(X) - x(Y). \tag{7}$$

It should be noted that we always have $\mathrm{P}_*(f) \neq \emptyset$ (cf. [12]).

For any $x \in \mathrm{P}_*(f)$ and any $e \in E$, if we have

$$\forall \alpha > 0 : x + \alpha\chi_e \notin \mathrm{P}_*(f), \tag{8}$$

we say that x is *positively saturated at* e, where χ_e is a unit vector in $\{0,1\}^E$ defined as $\chi_e(e) = 1$ and $\chi_e(e') = 0$ for $e' \in E - \{e\}$. Similarly, we say that x is *negatively saturated at* e if

$$\forall \alpha > 0 : x - \alpha\chi_e \notin \mathrm{P}_*(f). \tag{9}$$

Denote by $\mathrm{sat}^{(+)}(x)$ (or $\mathrm{sat}^{(-)}(x)$) the set of elements of E at which x is positively (or negatively) saturated. Note that we may have $\mathrm{sat}^{(+)}(x) \cap \mathrm{sat}^{(-)}(x) \neq \emptyset$. We call $\mathrm{sat}^{(+)}$ and $\mathrm{sat}^{(-)}$ the *signed saturation functions*, which generalize the saturation function for polymatroids and submodular systems (see [12]).

For any $e \in E - \mathrm{sat}^{(+)}(x)$ define

$$\hat{\mathrm{c}}(x, +e) = \max\{\alpha \mid \alpha \in \mathbf{R}, x + \alpha\chi_e \in \mathrm{P}_*(f)\}. \tag{10}$$

Also, for any $e \in E - \mathrm{sat}^{(-)}(x)$ define

$$\hat{\mathrm{c}}(x, -e) = \max\{\alpha \mid \alpha \in \mathbf{R}, x - \alpha\chi_e \in \mathrm{P}_*(f)\}. \tag{11}$$

Note that the maximum in (10) or (11) may be $+\infty$. We call $\hat{\mathrm{c}}(x, \pm e)$ the *signed saturation capacities*.

LEMMA 2.1. *Suppose $x \in \mathrm{P}_*(f)$. For any $e \in E - \mathrm{sat}^{(+)}(x)$,*

$$\hat{\mathrm{c}}(x, +e) = \min\{f(X, Y) - x(X, Y) \mid e \in X, (X, Y) \in \mathcal{F}\}. \tag{12}$$

Also, for any $e \in E - \mathrm{sat}^{(-)}(x)$,

$$\hat{\mathrm{c}}(x, -e) = \min\{f(X, Y) - x(X, Y) \mid e \in Y, (X, Y) \in \mathcal{F}\}. \tag{13}$$

Here the minimum over the empty set should be regarded as $+\infty$.

Proof. The proof is easy. □

It may be noted that if we define $\hat{\mathrm{c}}(x, \pm e) = 0$ for $e \in \mathrm{sat}^{(\pm)}(x)$, then (12) and (13) hold for any $e \in E$ (similar comments may apply to signed exchange capacities (22)∼(25) given later).

For any $x \in \mathrm{P}_*(f)$ let $\mathcal{F}(x)$ be the collection of tight signed sets for x in $\mathrm{P}_*(f)$; i.e.,

$$\mathcal{F}(x) = \{(X, Y) \mid x(X, Y) = f(X, Y)\}. \tag{14}$$

We can easily show that $\mathcal{F}(x)$ is closed with respect to $\sqcup$ and $\sqcap$ (see [13], [6] for the case when $\mathcal{F} = 3^E$). Note that we have $e \in \mathrm{sat}^{(+)}(x)$ (or $e \in \mathrm{sat}^{(-)}(x)$) if and only if there exists some $(X, Y) \in \mathcal{F}(x)$ such that $e \in X$ (or $e \in Y$). Therefore, for any $e \in \mathrm{sat}^{(+)}(x)$ define

$$\mathrm{dep}(x, +e) = \sqcap\{(X, Y) \mid e \in X, (X, Y) \in \mathcal{F}(x)\}, \tag{15}$$

and for any $e \in \mathrm{sat}^{(-)}(x)$ define

$$\mathrm{dep}(x, -e) = \sqcap\{(X, Y) \mid e \in Y, (X, Y) \in \mathcal{F}(x)\}. \tag{16}$$

We call dep the *signed dependence function*, which generalizes the dependence function for polymatroids and submodular systems (see [12]).

For any signed subset $W = (X, Y)$ of E we define

$$W^+ = X, \qquad W^- = Y. \tag{17}$$

We can easily see that for any $e \in \mathrm{sat}^{(+)}(x)$,

$$\mathrm{dep}(x, +e)^+ = \{e' \mid e' \in E, \exists \alpha > 0 : x + \alpha(\chi_e - \chi_{e'}) \in \mathrm{P}_*(f)\}, \tag{18}$$

$$\mathrm{dep}(x,+e)^{-}=\{e'\mid e'\in E,\exists\alpha>0: x+\alpha(\chi_e+\chi_{e'})\in \mathrm{P}_*(f)\}. \tag{19}$$

Similarly, for any $e\in \mathrm{sat}^{(-)}(x)$,

$$\mathrm{dep}(x,-e)^{+}=\{e'\mid e'\in E,\exists\alpha>0: x+\alpha(-\chi_e-\chi_{e'})\in \mathrm{P}_*(f)\}, \tag{20}$$

$$\mathrm{dep}(x,-e)^{-}=\{e'\mid e'\in E,\exists\alpha>0: x+\alpha(-\chi_e+\chi_{e'})\in \mathrm{P}_*(f)\}. \tag{21}$$

For any $(X_i,Y_i)\in 3^E$ $(i=1,2)$ we say that (X_1,Y_1) is *compliant with* (X_2,Y_2) if $X_1\cap Y_2=\emptyset$ and $Y_1\cap X_2=\emptyset$.

Suppose $e\in \mathrm{sat}^{(+)}(x)$. Then, for any $(X,Y)\in\mathcal{F}(x)$ with $e\notin Y$, (X,Y) must be compliant with $\mathrm{dep}(x,+e)$ due to the minimality of $\mathrm{dep}(x,+e)$ since $(\mathrm{dep}(x,+e)\sqcup(X,Y))\sqcap\mathrm{dep}(x,+e)\in\mathcal{F}(x)$. Similarly, for any $(X,Y)\in\mathcal{F}(x)$ with $e\notin X$, (X,Y) is compliant with $\mathrm{dep}(x,-e)$ if it is defined.

Also define the following:

(i) for any $e'\in\mathrm{dep}(x,+e)^{+}$ with $e'\neq e$,

$$\tilde{\mathrm{c}}(x,+e,-e')=\max\{\alpha\mid\alpha\in\mathbf{R},x+\alpha(\chi_e-\chi_{e'})\in \mathrm{P}_*(f)\}, \tag{22}$$

(ii) for any $e'\in\mathrm{dep}(x,+e)^{-}$,

$$\tilde{\mathrm{c}}(x,+e,+e')=\max\{\alpha\mid\alpha\in\mathbf{R},x+\alpha(\chi_e+\chi_{e'})\in \mathrm{P}_*(f)\}, \tag{23}$$

(iii) for any $e'\in\mathrm{dep}(x,-e)^{-}$ with $e'\neq e$,

$$\tilde{\mathrm{c}}(x,-e,+e')=\max\{\alpha\mid\alpha\in\mathbf{R},x+\alpha(-\chi_e+\chi_{e'})\in \mathrm{P}_*(f)\}, \tag{24}$$

(iv) for any $e'\in\mathrm{dep}(x,-e)^{+}$,

$$\tilde{\mathrm{c}}(x,-e,-e')=\max\{\alpha\mid\alpha\in\mathbf{R},x+\alpha(-\chi_e-\chi_{e'})\in \mathrm{P}_*(f)\}, \tag{25}$$

where the values of the right-hand sides are positive and may be $+\infty$. We call $\tilde{\mathrm{c}}(x,\pm e,\pm e')$ the *signed exchange capacities.*

LEMMA 2.2. *For any $x\in\mathrm{P}_*(f)$ we have the following:*

(i) *for any $e'\in\mathrm{dep}(x,+e)^{+}$ with $e'\neq e$,*

$$\begin{aligned}\tilde{\mathrm{c}}(x,+e,-e')=\min\{&f(X,Y)-x(X,Y)\mid(X,Y)\in\mathcal{F},\\&(e\in X,e'\notin X\cup Y)\text{ or }(e\notin X\cup Y,e'\in Y)\},\end{aligned}\tag{26}$$

(ii) *for any $e'\in\mathrm{dep}(x,+e)^{-}$,*

$$\begin{aligned}\tilde{\mathrm{c}}(x,+e,+e')=\min\{&f(X,Y)-x(X,Y)\mid(X,Y)\in\mathcal{F},\\&(e\in X,e'\notin X\cup Y)\text{ or }(e\notin X\cup Y,e'\in Y)\},\end{aligned}\tag{27}$$

(iii) *for any $e'\in\mathrm{dep}(x,-e)^{-}$ with $e'\neq e$,*

$$\begin{aligned}\tilde{\mathrm{c}}(x,-e,+e')=\min\{&f(X,Y)-x(X,Y)\mid(X,Y)\in\mathcal{F},\\&(e\in Y,e'\notin X\cup Y)\text{ or }(e\notin X\cup Y,e'\in X)\},\end{aligned}\tag{28}$$

(iv) *for any $e'\in\mathrm{dep}(x,-e)^{+}$,*

$$\begin{aligned}\tilde{\mathrm{c}}(x,-e,-e')=\min\{&f(X,Y)-x(X,Y)\mid(X,Y)\in\mathcal{F},\\&(e\in Y,e'\notin X\cup Y)\text{ or }(e\notin X\cup Y,e'\in X)\},\end{aligned}\tag{29}$$

where the minimum over the empty set should be regarded as $+\infty$.

Proof. We show (i) only (the proofs of (ii)$\sim$(iv) are similar).

Note that for $y_\alpha \equiv x + \alpha(\chi_e - \chi_{e'})$ and $(X,Y) \in \mathcal{F}$ the value $f(X,Y) - y_\alpha(X,Y)$ decreases as α increases if and only if (1) $e \in X$ and $e' \notin X$ or (2) $e \notin Y$ and $e' \in Y$. Moreover, for any $(X,Y) \in \mathcal{F}$ such that $e \in X$ and $e' \in Y$ we have

$$\begin{aligned}(30)\quad f(X,Y) - x(X,Y) &= f(X,Y) - x(X,Y) + f(\mathrm{dep}(x,+e)) - x(\mathrm{dep}(x,+e)) \\ &\geq f((X,Y) \sqcup \mathrm{dep}(x,+e)) - x((X,Y) \sqcup \mathrm{dep}(x,+e)) \\ &\quad + f((X,Y) \sqcap \mathrm{dep}(x,+e)) - x((X,Y) \sqcap \mathrm{dep}(x,+e)),\end{aligned}$$

where $e \in ((X,Y)\sqcup\mathrm{dep}(x,+e))^+$, $e' \notin ((X,Y)\sqcup\mathrm{dep}(x,+e))^+ \cup ((X,Y)\sqcup\mathrm{dep}(x,+e))^-$, $e \in ((X,Y)\sqcap\mathrm{dep}(x,+e))^+$, and $e' \notin ((X,Y)\sqcap\mathrm{dep}(x,+e))^+ \cup ((X,Y)\sqcap\mathrm{dep}(x,+e))^-$. We see from (30) that the value of its left-hand side is at least twice the minimum value of (26). Hence, the maximum value of α in (22) is equal to the minimum value of $f(X,Y) - x(X,Y)$ for $(X,Y) \in \mathcal{F}$ such that (1) $e \in X$, $e' \notin X \cup Y$ or (2) $e \notin X \cup Y$, $e' \in Y$. $\square$

Remark 2.1. Consider (i) in Lemma 2.2. For any $e \in \mathrm{sat}^{(+)}(x)$ and $e' \in \mathrm{dep}(x,+e)^+$ with $e \neq e'$ put

$$(31)\qquad \alpha = \tilde{\mathrm{c}}(x,+e,-e'),$$

$$(32)\qquad y = x + \alpha(\chi_e - \chi_{e'}),$$

where we assume $\alpha < +\infty$. We have $e \in \mathrm{sat}^{(+)}(y)$ since $\mathrm{dep}(x,+e) \in \mathcal{F}(y)$. Also, there exists a signed set $(X,Y) \in \mathcal{F}(y)$ such that (1) $e \in X$, $e' \notin X \cup Y$ or (2) $e \notin X \cup Y$, $e' \in Y$. In case (1),

$$(33)\qquad \mathrm{dep}(x,+e) \sqcap (X,Y) \sqsubset \mathrm{dep}(x,+e)$$

and in case (2),

$$(34)\qquad \mathrm{dep}(x,+e) \sqcap (\mathrm{dep}(x,+e) \sqcup (X,Y)) \sqsubset \mathrm{dep}(x,+e).$$

Therefore, by the minimality of $\mathrm{dep}(y,+e)$ we have

$$(35)\qquad \mathrm{dep}(y,+e) \sqsubset \mathrm{dep}(x,+e),$$

$$(36)\qquad e' \notin \mathrm{dep}(y,+e)^+;$$

i.e., after the full exchanging of (32) $\mathrm{dep}(x,+e)$ strictly decreases with respect to the partial order $\sqsubseteq$ and e' is removed from $\mathrm{dep}(x,+e)$. We can show similar facts concerning the full exchangings in (ii)$\sim$(iv) in Lemma 2.2. $\square$

For a bisubmodular system $(\mathcal{F}, f)$ on E and a signed set $(\hat{X},\hat{Y}) \in \mathcal{F}$ define $\mathcal{F}^{(\hat{X},\hat{Y})} \subseteq \mathcal{F}$ and $f^{(\hat{X},\hat{Y})} : \mathcal{F}^{(\hat{X},\hat{Y})} \to \mathbf{R}$ by

$$(37)\qquad \mathcal{F}^{(\hat{X},\hat{Y})} = \{(X,Y) \mid (X,Y) \in \mathcal{F}, (X,Y) \sqsubseteq (\hat{X},\hat{Y})\},$$

$$(38)\qquad f^{(\hat{X},\hat{Y})}(X,Y) = f(X,Y) \qquad ((X,Y) \in \mathcal{F}^{(\hat{X},\hat{Y})}).$$

We call $(\mathcal{F}^{(\hat{X},\hat{Y})}, f^{(\hat{X},\hat{Y})})$, $\mathcal{F}^{(\hat{X},\hat{Y})}$, and $f^{(\hat{X},\hat{Y})}$ the *restrictions* of $(\mathcal{F}, f)$, $\mathcal{F}$, and f to $(\hat{X}, \hat{Y})$, respectively.

Any $(S,T) \in 3^E$ with $S \cup T = E$ is called an *orthant.* For a bisubmodular system $(\mathcal{F}, f)$ on E, if we have an orthant (S,T) in $\mathcal{F}$, the *base polyhedron in the orthant* (S,T) is defined by

$$\mathrm{B}_{(S,T)}(f) = \{x \mid x \in \mathbf{R}^E, x \in \mathrm{P}_*(f), x(S,T) = f(S,T)\}. \tag{39}$$

The polyhedron $\mathrm{B}_{(S,T)}(f)$ can also be expressed as

(40)
$$\mathrm{B}_{(S,T)}(f) = \{x \mid x \in \mathbf{R}^E \quad \forall (X,Y) \in \mathcal{F}^{(S,T)} : x(X,Y) \le f(X,Y), x(S,T) = f(S,T)\}$$

(see [12]).

3. A min–max theorem. Consider a bisubmodular system $(\mathcal{F}, f)$ on E. We show the following min–max theorem for bisubmodular polyhedra with respect to the l_1 norm.

THEOREM 3.1. *For any vector $x^0 \in \mathbf{R}^E$,*

$$\min\left\{\sum_{e\in E} |x(e) - x^0(e)| \,\middle|\, x \in \mathrm{P}_*(f)\right\} = \max\{x^0(X,Y) - f(X,Y) \mid (X,Y) \in \mathcal{F}\}.$$
(41)

Moreover, if f is integer valued and x^0 is integral, then there exists an integral $x \in \mathrm{P}_*(f)$ that attains the minimum in the left-hand side of (41).

Proof. For any $x \in \mathrm{P}_*(f)$ and any $(X,Y) \in \mathcal{F}$ we have

$$\begin{aligned}\sum_{e\in E} |x(e) - x^0(e)| &\ge \sum_{e\in X} |x(e) - x^0(e)| + \sum_{e\in Y} |x(e) - x^0(e)| \\ &\ge x^0(X) - x(X) + x(Y) - x^0(Y) \\ &\ge x^0(X,Y) - f(X,Y).\end{aligned} \tag{42}$$

We show that (42) holds with equalities for some $x \in \mathrm{P}_*(f)$ and $(X,Y) \in \mathcal{F}$, which will complete the proof of the former part of the theorem.

Let $\hat{x}$ be a vector in $\mathrm{P}_*(f)$ that attains the minimum in the left-hand side of (41). Define

$$A_+ = \{e \mid e \in E, \hat{x}(e) < x^0(e)\}, \tag{43}$$

$$A_- = \{e \mid e \in E, \hat{x}(e) > x^0(e)\}, \tag{44}$$

$$A_0 = \{e \mid e \in E, \hat{x}(e) = x^0(e)\}. \tag{45}$$

Then it follows from the optimality of $\hat{x}$ that

$$A_+ \subseteq \mathrm{sat}^{(+)}(\hat{x}), \qquad A_- \subseteq \mathrm{sat}^{(-)}(\hat{x}), \tag{46}$$

and we have

$$\mathrm{dep}(\hat{x}, +e)^+ \subseteq A_+ \cup A_0 \qquad (e \in A_+), \tag{47}$$

$$\mathrm{dep}(\hat{x},+e)^- \subseteq A_- \cup A_0 \qquad (e \in A_+), \tag{48}$$

$$\mathrm{dep}(\hat{x},-e)^- \subseteq A_- \cup A_0 \qquad (e \in A_-), \tag{49}$$

$$\mathrm{dep}(\hat{x},-e)^+ \subseteq A_+ \cup A_0 \qquad (e \in A_-). \tag{50}$$

From (43)∼(50), defining

$$(\hat{X},\hat{Y}) = (\sqcup\{\mathrm{dep}(\hat{x},+e) \mid e \in A_+\}) \sqcup (\sqcup\{\mathrm{dep}(\hat{x},-e) \mid e \in A_-\}), \tag{51}$$

we have

$$(\hat{X},\hat{Y}) \sqsupseteq (A_+, A_-), \tag{52}$$

$$(\hat{X},\hat{Y}) \in \mathcal{F}(\hat{x}) \tag{53}$$

since $\mathcal{F}(\hat{x})$ is $\{\sqcup,\sqcap\}$-closed. Consequently,

$$\begin{aligned}\sum_{e\in E}|\hat{x}(e) - x^0(e)| &= x^0(\hat{X}) - \hat{x}(\hat{X}) + \hat{x}(\hat{Y}) - x^0(\hat{Y}) \\ &= x^0(\hat{X},\hat{Y}) - f(\hat{X},\hat{Y}).\end{aligned} \tag{54}$$

This completes the proof of the former part of the theorem.

For the latter part of the theorem concerning the integrality property, note that if f is integer valued and x^0 is integral, then the bisubmodular polyhedron $\mathrm{P}_*(f)$ is integral (cf. [7], [13], [6], [12]), so that the above argument is also valid if we consider $\mathbf{R}$ as the set $\mathbf{Z}$ of integers. □

It should be noted that the left-hand side of (41) is equal to the distance between $\mathrm{P}_*(f)$ and x^0 with respect to the l_1 norm.

From the proof of Theorem 3.1 we also have the following theorem.

THEOREM 3.2. *A vector $\hat{x} \in \mathrm{P}_*(f)$ attains the minimum of the left-hand side of* (41) *if and only if $\hat{x}$ satisfies* (46)∼(50)*, where A_+, A_-, and A_0 are, respectively, defined by* (43)∼(45)*.*

Proof. A proof of the "only if" part is included in that of Theorem 3.1. Moreover, following the proof of Theorem 3.1 from (51) until (54), we actually have shown the "if" part of the present theorem. □

Remark 3.1. We can easily see that the condition of (46)∼(50) is equivalent to the existence of a signed set $(X,Y) \in \mathcal{F}(\hat{x})$ such that

$$\hat{x}(e) \le x^0(e) \qquad (e \in X), \tag{55}$$

$$\hat{x}(e) \ge x^0(e) \qquad (e \in Y), \tag{56}$$

$$\hat{x}(e) = x^0(e) \qquad (e \in E - (X \cup Y)). \tag{57}$$

□

We also have the following theorem.

THEOREM 3.3. *Let $\hat{x}$ be any minimizer of the left-hand side of* (41) *and $(\hat{X},\hat{Y})$ be any maximizer of the right-hand side of* (41)*. Then for A_+ and A_- defined by* (43) *and* (44)*, respectively, we have* (52) *and* (53)*.*

Proof. For $x = \hat{x}$ and $(X,Y) = (\hat{X},\hat{Y})$ (42) holds with equalities, which implies (52) and (53). □

COROLLARY 3.4. *Let $\hat{x}$ and $(\hat{X}, \hat{Y})$ be those appearing in Theorem* 3.3. *Suppose $(\hat{X}, \hat{Y}) \neq (\emptyset, \emptyset)$. Then we have*

$$\hat{x}^{(\hat{X},\hat{Y})} \in \mathrm{B}_{(\hat{X},\hat{Y})}(f^{(\hat{X},\hat{Y})}), \tag{58}$$

where $\hat{x}^{(\hat{X},\hat{Y})}$ is the restriction of vector $\hat{x}$ to $\hat{X} \cup \hat{Y}$, $f^{(\hat{X},\hat{Y})}$ is the restriction of f to $(\hat{X}, \hat{Y})$, and $\mathrm{B}_{(\hat{X},\hat{Y})}(f^{(\hat{X},\hat{Y})})$ is the base polyhedron of $(\mathcal{F}^{(\hat{X},\hat{Y})}, f^{(\hat{X},\hat{Y})})$ in the orthant $(\hat{X}, \hat{Y})$.

Proof. The present corollary easily follows from Theorem 3.3 (recall expressions (39) and (40) for the base polyhedron in an orthant). □

Theorem 3.3 and Corollary 3.4 will be used in the following sections.

Using Theorem 3.1, we can show a theorem of Cunningham and Green-Krótki [8] given below. (Here we generalize it to the case when possibly $\mathcal{F} \neq 3^E$. Also note that the integrality property is not explicitly stated in [8].)

THEOREM 3.5 (Cunningham and Green-Krótki [8]). *Let x^0 be a vector in $\mathbf{R}^E$. If there is a vector $x \in \mathrm{P}_*(f)$ such that $x \leq x^0$, then*

$$\max\{x(E) \mid x \leq x^0, x \in \mathrm{P}_*(f)\} = \min\{f(X,Y) + x^0(Y) + x^0(E - X) \mid (X,Y) \in \mathcal{F}\}. \tag{59}$$

Moreover, if f is integer valued and x^0 is integral, then the maximum in the left-hand side of (59) *is attained by an integral x.* □

The following lemma is essential in regard to the relationship between Theorem 3.1 and the theorem of Cunningham and Green-Krótki.

LEMMA 3.6. *Let x^0 be a vector in $\mathbf{R}^E$ and suppose that there is a vector $x \in \mathrm{P}_*(f)$ such that $x \leq x^0$. Then we have*

$$\min\{x^0(E) - x(E) \mid x \leq x^0, x \in \mathrm{P}_*(f)\} = \min\left\{\sum_{e \in E} |x^0(e) - x(e)| \,\middle|\, x \in \mathrm{P}_*(f)\right\}, \tag{60}$$

and any vector $\hat{x}$ that attains the minimum of the left-hand side of (60) *also attains the minimum of the right-hand side of* (60). *Moreover, if f is integer valued and x^0 is integral, then there exists an integral such $\hat{x}$.*

Proof. Let $\hat{x}$ be a minimizer of the left-hand side of (60). Then we can easily see that (46)∼(50) hold. It follows from Theorem 3.2 that $\hat{x}$ also attains the minimum of the right-hand side of (60) and that both sides of (60) have the same value. Since the above argument is also valid if we consider it within the set $\mathbf{Z}$ of integers, the latter integrality part of the present lemma follows. □

It should be noted that in the above proof of the integrality property in Lemma 3.6 we have implicitly used the fact that if f is integer valued, x^0 is integral, and there exists a vector $x \in \mathrm{P}_*(f)$ such that $x \leq x^0$, then there exists an integral such vector x. This fact can be shown by the integrality property of $\mathrm{P}_*(f)$ and an algorithm proposed in section 4 (unfortunately, this integrality property was not proved in [8]).

Now Theorem 3.5 can be shown as follows. Note that (59) is equivalent to the following:

$$\min\{x^0(E) - x(E) \mid x \leq x^0, x \in \mathrm{P}_*(f)\} = \max\{x^0(X,Y) - f(X,Y) \mid (X,Y) \in \mathcal{F}\}. \tag{61}$$

We see from Lemma 3.6 and Theorem 3.1 that (61) together with its integrality property holds.

Consequently, Theorem 3.1 generalizes the well-known min–max relation arising from a vector reduction of polymatroids and submodular systems [11], [12].

Conversely, Theorem 3.1 can also be shown by using Theorem 3.5. For a subset T of E define the partial order $\leq_T$ among vectors in $\mathbf{R}^E$ by $x \leq_T y$ if and only if $x(e) \leq y(e)$ $(e \in E - T)$ and $x(e) \geq y(e)$ $(e \in T)$. Then reflected versions of Theorem 3.5 and Lemma 3.6 are, respectively, given as follows. Let T be a subset of E and x^0 be a vector in $\mathbf{R}^E$.

Theorem 3.7. *If there is a vector $x \in \mathrm{P}_*(f)$ such that $x \leq_T x^0$, then*

$$\begin{aligned}(62)\qquad &\max\{x(E-T,T) \mid x \leq_T x^0, x \in \mathrm{P}_*(f)\}\\ &= \min\{f(X,Y) - x^0(X,Y) + x^0(E-T,T) \mid (X,Y) \in \mathcal{F}\}.\end{aligned}$$

Moreover, if f and x^0 are integral, then the maximum in the left-hand side of (62) *is attained by an integral x.* □

Lemma 3.8. *If there is a vector $x \in \mathrm{P}_*(f)$ such that $x \leq_T x^0$, then*

$$\begin{aligned}(63)\qquad &\min\{x^0(E-T,T) - x(E-T,T) \mid x \leq_T x^0, x \in \mathrm{P}_*(f)\}\\ &= \min\left\{\sum_{e\in E} |x^0(e) - x(e)| \;\middle|\; x \in \mathrm{P}_*(f)\right\},\end{aligned}$$

and any vector $\hat{x}$ that attains the minimum of the left-hand side of (63) *also attains the minimum of the right-hand side of* (63)*. Moreover, if f and x^0 are integral, then there exists an integral such $\hat{x}$.* □

We prove Theorem 3.1 by using Theorem 3.7 and Lemma 3.8 as follows. Let x^1 be a vector that attains the minimum of the left-hand side of (41). Put $T = \{e \mid e \in E, x^1(e) > x^0(e)\}$. Then the assumptions of Lemma 3.8 and Theorem 3.7 hold for this T. Note that relation (62) can be rewritten as

$$\begin{aligned}(64)\qquad &\min\{x^0(E-T,T) - x(E-T,T) \mid x \leq_T x^0, x \in \mathrm{P}_*(f)\}\\ &= \max\{x^0(X,Y) - f(X,Y) \mid (X,Y) \in \mathcal{F}\}.\end{aligned}$$

Therefore, from Lemma 3.8 we have Theorem 3.1.

Furthermore, we show the following min–max relation, which is closely related to Theorems 3.1 and 3.5.

Theorem 3.9. *For any vector $x^0 \in \mathbf{R}^E$,*

$$(65)\qquad \min\left\{\sum_{e\in E} \max\{0, x(e) - x^0(e)\} \;\middle|\; x \in \mathrm{P}_*(f)\right\} = \max\{x^0(\emptyset,Y) - f(\emptyset,Y) \mid (\emptyset,Y) \in \mathcal{F}\}.$$

Moreover, if f and x^0 are integral, then there exists an integral x in $\mathrm{P}_*(f)$ that attains the minimum of the left-hand side of (65).

Proof. For any $x \in \mathrm{P}_*(f)$ and $(\emptyset, Y) \in \mathcal{F}$,

$$\begin{aligned}(66)\qquad \sum_{e\in E} \max\{0, x(e) - x^0(e)\} &\geq x(Y) - x^0(Y) = x^0(\emptyset,Y) - x(\emptyset,Y)\\ &\geq x^0(\emptyset,Y) - f(\emptyset,Y).\end{aligned}$$

On the other hand, let $\hat{x}$ be a vector that attains the minimum of the left-hand side of (65). Then for A_+, A_-, and A_0 defined by (43)∼(45) we have

$$(67)\qquad A_- \subseteq \mathrm{sat}^{(-)}(\hat{x})$$

and

$$\text{dep}(\hat{x}, -e)^- \subseteq A_- \cup A_0 \qquad (e \in A_-), \tag{68}$$

$$\text{dep}(\hat{x}, -e)^+ = \emptyset \qquad (e \in A_-) \tag{69}$$

due to the optimality of $\hat{x}$. Since $\text{dep}(\hat{x}, -e)$ $(e \in A_-)$ are compliant with each other, we have from (68) and (69) that

$$(\emptyset, A_-) \sqsubseteq (\emptyset, \hat{Y}) \equiv \sqcup\{\text{dep}(\hat{x}, -e) \mid e \in A_-\} \sqsubseteq (\emptyset, A_0 \cup A_-). \tag{70}$$

It follows from (70) that

$$\hat{x}(\emptyset, \hat{Y}) = f(\emptyset, \hat{Y}), \tag{71}$$

$$\hat{x}(e) \leq x^0(e) \qquad (e \in E - \hat{Y}). \tag{72}$$

Hence, (66) with $x = \hat{x}$ and $Y = \hat{Y}$ holds with equality and we thus have (65). Moreover, the integrality property holds since when f and x^0 are integral, the above argument is valid if we restrict $\mathbf{R}$ to the set $\mathbf{Z}$ of integers. □

We can also have a reflected version of Theorem 3.9.

4. Algorithms. Given any vector $x^0 \in \mathbf{R}^E$, consider the following problem that appeared in the left-hand side of (41):

$$\begin{aligned} (P) \quad & \text{Minimize} \sum_{e \in E} |x(e) - x^0(e)| \\ & \text{subject to } x \in \mathrm{P}_*(f). \end{aligned} \tag{73}$$

We propose an algorithm for solving Problem (P) under the assumption that we can easily calculate signed saturation capacities $\hat{\text{c}}(x, \pm e)$ and signed exchange capacities $\tilde{\text{c}}(x, \pm e, \pm e')$. When $\mathcal{F} = 3^E$, an extreme point of $\mathrm{P}_*(f)$ as an initial vector x in Step 0 is obtained by the greedy algorithm (see [6], [7], [10], [12], [13], [14]). We can also easily find an initial vector x in $\mathrm{P}_*(f)$ for a general $\{\sqcup, \sqcap\}$-closed family $\mathcal{F}$ (see [1]).

An Algorithm for Problem (P).

Step 0: Find an initial vector x in $\mathrm{P}_*(f)$.

Step 1: For each $e \in E$ such that $x(e) < x^0(e)$ do the following (1-1)∼(1-3):

(1-1) If $x(e) < x^0(e)$ and $e \notin \text{sat}^{(+)}(x)$, then put

$$\hat{\alpha} \leftarrow \min\{x^0(e) - x(e), \hat{\text{c}}(x, +e)\}, \tag{74}$$

$$x \leftarrow x + \hat{\alpha}\chi_e. \tag{75}$$

(1-2) While $x(e) < x^0(e)$ and there exists an element $e' \in \text{dep}(x, +e)^- - \{e\}$ with $x(e') < x^0(e')$, choose one such e' and put

$$\hat{\beta} \leftarrow \min\{x^0(e) - x(e), x^0(e') - x(e'), \tilde{\text{c}}(x, +e, +e')\}, \tag{76}$$

$$x \leftarrow x + \hat{\beta}(\chi_e + \chi_{e'}). \tag{77}$$

(1-3) While $x(e) < x^0(e)$ and there exists an element $e' \in \text{dep}(x, +e)^+$ with $x(e') > x^0(e')$, choose one such e' and put

$$\hat{\beta} \leftarrow \min\{x^0(e) - x(e), x(e') - x^0(e'), \tilde{\text{c}}(x, +e, -e')\}, \tag{78}$$

$$x \leftarrow x + \hat{\beta}(\chi_e - \chi_{e'}). \tag{79}$$

Step 2: For each $e \in E$ such that $x(e) > x^0(e)$ do the following (2-1) and (2-2):
(2-1) If $x(e) > x^0(e)$ and $e \notin \mathrm{sat}^{(-)}(x)$, then put

$$\hat{\alpha} \leftarrow \min\{x(e) - x^0(e), \hat{\mathrm{c}}(x, -e)\}, \tag{80}$$

$$x \leftarrow x - \hat{\alpha}\chi_e. \tag{81}$$

(2-2) While $x(e) > x^0(e)$ and there exists an element $e' \in \mathrm{dep}(x, -e)^+ - \{e\}$ with $x(e') > x^0(e')$, choose one such e' and put

$$\hat{\beta} \leftarrow \min\{x(e) - x^0(e), x(e') - x^0(e'), \tilde{\mathrm{c}}(x, -e, -e')\}, \tag{82}$$

$$x \leftarrow x + \hat{\beta}(-\chi_e - \chi_{e'}). \tag{83}$$

(End)

The above algorithm terminates after calculating signed saturation capacities $\mathrm{O}(|E|)$ times and signed exchange capacities $\mathrm{O}(|E|^2)$ times because of Remark 2.1.

Let us examine the above algorithm to see its validity.

Suppose $\{e \mid e \in E, x(e) < x^0(e)\} = \{e_1, e_2, \ldots, e_l\}$ and that Steps (1-1)∼(1-3) are carried out in the order of $e_1, e_2, \ldots, e_l$. For element e_1, after finishing Step (1-1) we have $x(e_1) = x^0(e_1)$ or $x(e_1) < x^0(e_1)$ and $e_1 \in \mathrm{sat}^{(+)}(x)$. In the latter case, $\mathrm{dep}(x, +e_1)$ is defined and we move on to Step (1-2). After finishing Step (1-2) we have $x(e_1) = x^0(e_1)$ or

$$x(e') \geq x^0(e') \qquad (e' \in \mathrm{dep}(x, +e_1)^-), \tag{84}$$

since performing (76) and (77) for e' yields (1) $x(e_1) = x^0(e_1)$, (2) $x(e') = x^0(e')$, or (3) $\mathrm{dep}(x, +e_1)$ becomes strictly smaller than the previous one with respect to $\sqsubseteq$ and e' is removed from the previous $\mathrm{dep}(x, +e_1)$ due to Remark 2.1. Similarly, after finishing Step (1-3) we have $x(e_1) = x^0(e_1)$ or

$$x(e') \leq x^0(e') \qquad (e' \in \mathrm{dep}(x, +e_1)^+). \tag{85}$$

For each $k \in \{1, 2, \ldots, l\}$ denote by x_k the x obtained after Step (1-3) for element e_k. We show that for each $k \in \{1, 2, \ldots, l\}$, if $x_k(e_k) < x^0(e_k)$, then $\mathrm{dep}(x_k, +e_k) = \mathrm{dep}(x_l, +e_k)$. Suppose that we are to carry out Step (1-1) for element e_k for some $k \in \{2, \ldots, l\}$. We assume that $\mathrm{dep}(x_i, +e_i) = \mathrm{dep}(x_{k-1}, +e_i)$ for each $i \in \{1, 2, \ldots, k-1\}$ with $x_i(e_i) < x^0(e_i)$ and that, defining

$$(X_{k-1}, Y_{k-1}) = \sqcup\{\mathrm{dep}(x_i, +e_i) \mid i \in \{1, 2, \ldots, k-1\}, x_i(e_i) < x^0(e_i)\}, \tag{86}$$

where the reduced union over the empty set should be regarded as $(\emptyset, \emptyset)$, we have

$$x_{k-1}(e) \leq x^0(e) \qquad (e \in X_{k-1}), \tag{87}$$

$$x_{k-1}(e) \geq x^0(e) \qquad (e \in Y_{k-1}). \tag{88}$$

Note that this assumption is valid for $k = 2$ (see (84) and (85)). In Step (1-1) for element e_k, if $e_k \in X_{k-1}$, then for the current x we have $e_k \in \mathrm{sat}^{(+)}(x)$ and we do nothing in Step (1-1). In Step (1-2), suppose $x(e_k) < x^0(e_k)$. Note that $e_k \notin Y_{k-1}$ by the assumption. If $e_k \in X_{k-1}$, then $\mathrm{dep}(x, +e_k) \sqsubseteq (X_{k-1}, Y_{k-1})$, so that x is not changed in Step (1-2) due to (88). Also, if $e_k \notin X_{k-1}$, i.e., $e_k \notin X_{k-1} \cup Y_{k-1}$, then it follows from the minimality of the signed dependence function that $\mathrm{dep}(x, +e_k)$ is compliant with (X_{k-1}, Y_{k-1}); i.e.,

$$\mathrm{dep}(x, +e_k)^- \cap X_{k-1} = \emptyset, \qquad \mathrm{dep}(x, +e_k)^+ \cap Y_{k-1} = \emptyset. \tag{89}$$

Therefore, for each $e'' \in X_{k-1} \cup Y_{k-1}$ the value $x(e'')$ is not changed by Steps (1-2) and (1-3) for e_k. Hence, we have $\mathrm{dep}(x_i, +e_i) = \mathrm{dep}(x_k, +e_i)$ for $i \in \{1, 2, \ldots, k\}$ with $x_i(e_i) < x^0(e_i)$. After finishing Steps (1-1)$\sim$(1-3) for e_k, if $x_k(e_k) < x^0(e_k)$, then we have

$$x_k(e') \geq x^0(e') \qquad (e' \in \mathrm{dep}(x_k, +e_k)^-), \tag{90}$$

$$x_k(e') \leq x^0(e') \qquad (e' \in \mathrm{dep}(x_k, +e_k)^+). \tag{91}$$

Hence, (87) and (88) hold with $k-1$ replaced by k.

After finishing Step 1 we have $(X_l, Y_l) \in \mathcal{F}(x)$ for the current x and

$$x(e) \leq x^0(e) \qquad (e \in X_l), \tag{92}$$

$$x(e) \geq x^0(e) \qquad (e \in E - X_l). \tag{93}$$

Now suppose we have moved on to Step 2. Suppose $\{e \,|\, e \in E,\ x(e) > x^0(e)\} = \{e_{l+1}, e_{l+2}, \ldots, e_m\}$ and that we treat these elements in the order of $e_{l+1}, e_{l+2}, \ldots, e_m$. Denote by x_k the x obtained after Step (2-2) for element e_k for $k \in \{l{+}1, l{+}2, \ldots, m\}$. In Step (2-1) for element e_{l+1}, if $x(e_{l+1}) > x^0(e_{l+1})$ and $e_{l+1} \notin \mathrm{sat}^{(-)}(x)$, then $e_{l+1} \notin X_l \cup Y_l$ because of (92) and (93) and since $\mathrm{sat}^{(-)}(x) \supseteq Y_l$. After finishing Step (2-1) for e_{l+1}, if $x(e_{l+1}) > x^0(e_{l+1})$, then we have $\mathrm{dep}(x, -e_{l+1})^- \cap X_l = \emptyset$ due to the minimality of dep. (Due to this fact we do not carry out "Step (2-3)" similar to Step (1-3).) Therefore, if we carry out (82) and (83), we also have $e' \notin X_l \cup Y_l$. Consequently, the values of $x(e)$ $(e \in X_l \cup Y_l)$ are not changed and if $x(e_{l+1}) > x^0(e_{l+1})$ after Step (2-2), we have $\mathrm{dep}(x_{l+1}, -e_{l+1})$ compliant with (X_l, Y_l) such that

$$x(e') \leq x^0(e') \qquad (e' \in \mathrm{dep}(x_{l+1}, -e_{l+1})^+), \tag{94}$$

$$x(e') \geq x^0(e') \qquad (e' \in \mathrm{dep}(x_{l+1}, -e_{l+1})^-). \tag{95}$$

Put

$$X_{l+1} \cup Y_{l+1} = (X_l, Y_l) \sqcup \mathrm{dep}(x_{l+1}, -e_{l+1}). \tag{96}$$

By repeating almost the same argument for Step 1 we can show that after Step 2, putting

$$(X_m, Y_m) = (X_l, Y_l) \sqcup (\sqcup\{\mathrm{dep}(x, -e_i) \,|\, i \in \{l+1, \ldots, m\}, x(e_i) > x^0(e_i)\}) \tag{97}$$

for the finally obtained x, we have

$$(X_m, X_m) \in \mathcal{F}(x), \tag{98}$$

$$x(e) \leq x^0(e) \qquad (e \in X_m), \tag{99}$$

$$x(e) \geq x^0(e) \qquad (e \in Y_m), \tag{100}$$

$$x(e) = x^0(e) \qquad (e \in E - (X_m \cup Y_m)). \tag{101}$$

It follows from (98)∼(101) that the finally obtained x satisfies the optimality condition given in Theorem 3.2 (see Remark 3.1 below Theorem 3.2).

Hence, we have the following theorem.

THEOREM 4.1. *The vector x obtained when the algorithm terminates is an optimal solution of Problem (P).* □

It should be noted that by performing the algorithm each component $x(e)$ with $x(e) < x^0(e)$ is monotonically increased and each component $x(e)$ with $x(e) > x^0(e)$ is monotonically decreased. Therefore, if there is any $x \in \mathrm{P}_*(f)$ such that $x \leq x^0$, then, starting from such an x, the algorithm finds an optimal $x \in \mathrm{P}_*(f)$ such that $x \leq x^0$. Also, note that when f is integer valued and x^0 is integral, starting from an integral $x \in \mathrm{P}_*(f)$, we reach an integral optimal solution by the algorithm.

Let us also consider the following problem associated with the min–max relation in Theorem 3.9:

$$(102)\qquad \begin{array}{lll} (P') & \text{Minimize} & \displaystyle\sum_{e\in E} \max\{0, x(e) - x^0(e)\} \\ & \text{subject to} & x \in \mathrm{P}_*(f). \end{array}$$

An algorithm for Problem (P') is now given similarly as the above algorithm for Problem (P).

AN ALGORITHM FOR PROBLEM (P').

Step 0′: Find an initial vector x in $\mathrm{P}_*(f)$.

Step 1′: For each $e \in E$ such that $x(e) > x^0(x)$ do the following (1-1)′∼(1-3)′:

(1-1)′ If $x(e) > x^0(e)$ and $e \notin \mathrm{sat}^{(-)}(x)$, then put

$$(103)\qquad \hat{\alpha} \leftarrow \min\{x(e) - x^0(e), \hat{\mathrm{c}}(x, -e)\},$$

$$(104)\qquad x \leftarrow x - \hat{\alpha}\chi_e.$$

(1-2)′ While $x(e) > x^0(e)$ and there exists an element $e' \in \mathrm{dep}(x, -e)^+ - \{e\}$, choose one such e' and put

$$(105)\qquad \hat{\beta} \leftarrow \min\{x(e) - x^0(e), \tilde{\mathrm{c}}(x, -e, -e')\},$$

$$(106)\qquad x \leftarrow x + \hat{\beta}(-\chi_e - \chi_{e'}).$$

(1-3)′ While $x(e) > x^0(e)$ and there exists an element $e' \in \mathrm{dep}(x, -e)^- - \{e\}$ with $x(e') < x^0(e)$, choose one such e' and put

$$(107)\qquad \hat{\beta} \leftarrow \min\{x(e) - x^0(e), x^0(e') - x(e'), \tilde{\mathrm{c}}(-e, +e')\},$$

$$(108)\qquad x \leftarrow x + \hat{\beta}(-\chi_e + \chi_{e'}).$$

(End)

We omit the proof of the validity of this algorithm. (The proof is similar to that for the algorithm for Problem (P).)

When f and x^0 are integral, starting from an integral initial vector x in $\mathrm{P}_*(f)$, we reach an integral optimal solution of Problem (P'). If there exists a vector $x \in \mathrm{P}_*(f)$ such that $x \leq x^0$, then the above algorithm finds such a vector (an integral such vector when f and x^0 are integral, starting from an integral initial vector in $\mathrm{P}_*(f)$).

5. A separable convex optimization problem. We show an application of the results obtained in sections 3 and 4 to a separable convex optimization problem over a bisubmodular polyhedron.

Given a bisubmodular system $(\mathcal{F}, f)$ on E and a convex function $w_e : \mathbf{R} \to \mathbf{R}$ for each $e \in E$, consider the following optimization problem:

$$\begin{aligned} (\hat{P}) \quad & \text{Minimize} \sum_{e \in E} w_e(x(e)) \\ & \text{subject to } x \in \mathrm{P}_*(f). \end{aligned} \tag{109}$$

In [3] the problem of (109) where $\mathbf{R}$ is the set $\mathbf{Z}$ of integers is considered and an incrementally greedy algorithm is given (also see [4]). Recall that the argument in this paper is valid for any totally ordered additive group $\mathbf{R}$.

The following theorem relates optimal solutions of $(\hat{P})$ to optimal solutions of (P) in (73).

THEOREM 5.1. *Suppose that there is a global minimizer x^0 of $\sum_{e \in E} w_e(x(e))$ over $\mathbf{R}^E$. Then there exists an optimal solution of Problem $(\hat{P})$ that is also an optimal solution of Problem (P) with this x^0.*

Proof. Because of the existence of x^0 and the convexity of w_e $(e \in E)$ there exists an optimal solution x^* of Problem $(\hat{P})$. Then by starting from $x = x^*$ the algorithm presented in section 4 with the present x^0 gives us an optimal solution of Problem (P) that is also optimal for Problem $(\hat{P})$, since the augmentations and the exchangings performed in the algorithm do not increase the value of the objective function of Problem $(\hat{P})$. □

It follows from Theorem 5.1 and Corollary 3.4 that if a global minimizer x^0 of the objective function of $(\hat{P})$ is given and $(\hat{X}, \hat{Y})$ given by (51) using this x^0 is not equal to $(\emptyset, \emptyset)$, then the restriction of an optimal solution of $(\hat{P})$ to $(\hat{X}, \hat{Y})$ lies in the base polyhedron in the orthant given in Corollary 3.4. Note that if $(\hat{X}, \hat{Y}) = (\emptyset, \emptyset)$, i.e., $\hat{x} = x^0$, then we are finished. Therefore, an optimal solution of Problem $(\hat{P})$ can be found by the decomposition algorithm developed in [12, section 8.2].

We can remove the assumption that there is a global minimizer of the objective function of $(\hat{P})$ over $\mathbf{R}^E$ as follows. Define $x^0 \in (\mathbf{R} \cup \{+\infty\})^E$ by

$$x^0(e) = \begin{cases} \text{a minimizer of } w_e(\cdot) \text{ in } \mathbf{R} \text{ if any exists,} \\ +\infty \text{ if } w_e(\cdot) \text{ is strictly monotone decreasing,} \\ -\infty \text{ if } w_e(\cdot) \text{ is strictly monotone increasing} \end{cases} \tag{110}$$

for each $e \in E$. With this x^0 carry out the algorithm in section 4. We can see that during the execution of the algorithm if any parameter $\hat{\alpha}$ or $\hat{\beta}$ becomes $+\infty$, Problem $(\hat{P})$ does not have an optimal solution and that if the algorithm terminates with a finite $x = \hat{x}$, then Problem $(\hat{P})$ has an optimal solution. In the latter case, the proof of Theorem 5.1 is valid for x^0 defined by (110) and we can show that if $(\hat{X}, \hat{Y}) \neq (\emptyset, \emptyset)$, the restriction of an optimal solution of Problem $(\hat{P})$ to $(\hat{X}, \hat{Y})$ lies in $\mathrm{B}_{(\hat{X},\hat{Y})}(f^{(\hat{X},\hat{Y})})$ of Corollary 3.4, where $(\hat{X}, \hat{Y})$ is defined by (51) using $\hat{x}$ obtained as above.

Acknowledgments. The present research was carried out while the author was at Forschungsinstitut für Diskrete Mathematik, Universität Bonn, Germany. The author is very grateful to Kazutoshi Ando, Bill Cunningham, and Jim Geelan for their useful comments on an earlier version of this paper.

REFERENCES

[1] K. Ando and S. Fujishige, *On structures of bisubmodular polyhedra*, Math. Programming, 74 (1996), pp. 293–317.

[2] K. Ando, S. Fujishige, and T. Naitoh, *Proper Bisubmodular Systems and Bidirected Flows*, Discussion Paper Series No. 532, Institute of Socio-Economic Planning, University of Tsukuba, Japan, 1993.

[3] K. Ando, S. Fujishige, and T. Naitoh, *A greedy algorithm for minimizing a separable convex function over an integral bisubmodular polyhedron*, J. Oper. Res. Soc. Japan, 37 (1994), pp. 188–196.

[4] K. Ando, S. Fujishige, and T. Naitoh, *A greedy algorithm for minimizing a separable convex function over a finite jump system*, J. Oper. Res. Soc. Japan, 38 (1995), pp. 362–375.

[5] A. Bouchet, *Greedy algorithm and symmetric matroids*, Math. Programming, 38 (1987), pp. 147–159.

[6] A. Bouchet and W. H. Cunningham, *Delta-matroids, jump systems and bisubmodular polyhedra*, SIAM J. Discrete Math., 8 (1995), pp. 17–32.

[7] R. Chandrasekaran and S. N. Kabadi, *Pseudomatroids*, Discrete Math., 71 (1988), pp. 205–217.

[8] W. H. Cunningham and J. Green-Krótki, *b-matching degree-sequence polyhedra*, Combinatorica, 11 (1991), pp. 219–230.

[9] A. Dress and T. F. Havel, *Some combinatorial properties of discriminants in metric vector spaces*, Adv. Math., 62 (1986), pp. 285–312.

[10] F. D. J. Dunstan and D. J. A. Welsh, *A greedy algorithm for solving a certain class of linear programmes*, Math. Programming, 62 (1973), pp. 338–353.

[11] J. Edmonds, *Submodular functions, matroids, and certain polyhedra*, in Combinatorial Structures and Their Applications, R. Guy, H. Hanani, N. Sauer, and J. Schönheim, eds., Gordon and Breach, New York, 1970, pp. 69–87.

[12] S. Fujishige, *Submodular Functions and Optimization*, North–Holland, Amsterdam, 1991.

[13] S. N. Kabadi and R. Chandrasekaran, *On totally dual integral systems*, Discrete Appl. Math., 26 (1990), pp. 87–104.

[14] M. Nakamura, *A characterization of greedy sets: Universal polymatroids* (I), Scientific Papers of College of Arts and Science, University of Tokyo, 38 (1988), pp. 155–167.

[15] M. Nakamura, *Δ-polymatroids and an extension of Edmonds–Giles' TDI scheme*, Proc. of the Third IPCO Conference, 1993, pp. 401–412.

[16] L. Qi, *Directed submodularity, ditroids and directed submodular flows*, Math. Programming, 42 (1988), pp. 579–599.

SIAM J. DISCRETE MATH.
Vol. 10, No. 2, pp. 309–317, May 1997

011

DECOMPOSING 4-REGULAR GRAPHS INTO TRIANGLE-FREE 2-FACTORS[*]

EDWARD BERTRAM[†] AND PETER HORÁK[‡]

Abstract. There is a polynomial algorithm which finds a decomposition of any given 4-regular graph into two triangle-free 2-factors or shows that such a decomposition does not exist.

Key words. decomposition, triangle-free, 2-factor, polynomial algorithm

AMS subject classifications. 05C70, 05C85, 68R10

PII. S089548019427144X

Introduction. A 2-factor of a graph G is a subgraph F of G such that any vertex of G is of degree 2 in F. Hell et al. [5] proved that given a set L of natural numbers, recognizing whether a graph G admits a 2-factor F such that no cycle of F is of length from L is NP-hard unless $L \subseteq \{3, 4\}$. On the other hand, an elegant criterion for deciding if a graph possesses an (unrestricted) 2-factor (i.e., $L = \{\emptyset\}$) was given by Tutte [8], and there is a polynomial algorithm to find such a 2-factor (or to determine that none exist) [3]. Hartvigsen [4] proved that the problem of whether a graph G admits a triangle-free 2-factor ($L = \{3\}$) can also be solved in polynomial time.

In this paper we study a modification of this problem. How difficult is it to recognize whether a 4-regular graph can be decomposed into two triangle-free 2-factors? For a long time it had been thought that the general graph decomposition problem of whether a graph H can be written as an edge-disjoint union of copies of a graph G was difficult. This was confirmed when Dolinski and Tarsi [2] proved that unless G is of the form $tK_2 \cup nP_3$, the decomposition problem is NP-complete. In view of their result it is not surprising that there is interest in restricted decomposition problems. The main result of this paper says that there is a polynomial algorithm for finding a decomposition of any given 4-regular graph into two triangle-free 2-factors (or showing that none exists). In fact, to be able to proceed with the induction, we prove a slightly stronger result.

It would be nice to know the complexity of recognizing $2n$-regular graphs which admit a decomposition into two triangle-free n-factors and the complexity of recognizing $2n$-regular graphs which admit a decomposition into n triangle-free 2-factors. We believe that the following is true.

Conjecture. The two decision problems are NP-complete for all $n \geq 3$.

We point out that using a different approach Koudier and Sabidussi recently published [6] an elegant sufficient condition for a 4-regular graph G to have a decomposition into two triangle-free 2-factors. They showed that G possesses such a decomposition if G has at most two essential cut vertices (a cut vertex is essential if it lies on a triangle). We do not see how to prove the result of Koudier and Sabidussi using methods of this paper. On the other hand, it seems to us that a decision pro-

[*] Received by the editors July 20, 1994; accepted for publication (in revised form) May 31, 1996. http://www.siam.org/journals/sidma/10-2/27144.html

[†] Department of Mathematics, University of Hawaii, Honolulu, HI 96822 (ed@math.hawaii.edu).

[‡] Department of Mathematics, Kuwait University, P.O. Box 5969 Code No. 13060, Kuwait (horak@math-1.sci.kuniv.edu.kw).

cedure, which would determine for all 4-regular graphs whether they have a required decomposition, is not within the methods of [6].

Preliminaries. Except for concepts and notation introduced here we use standard graph-theoretical terminology. A graph G is said to be even (odd) if G has an even (odd) number of edges. We will say that a graph G belongs to class $\mathcal{G}(4,2)$ if all vertices of G are either of degree 4 or of degree 2. Let v be a cut vertex of an even graph $G \in \mathcal{G}(4,2)$. Then $d(v) = 4$ and the graph $G - v$ has two components. We will say that v is an even (odd) cut vertex if the parity of the number of edges of both components is even (odd). It turns out that instead of the language of decompositions it is more convenient to use that of coloring edges. A coloring C of the edges of $G \in \mathcal{G}(4,2)$ with two colors will be called *proper* if

(i) each vertex of G is adjacent to the same number of edges in each color,

(ii) both monochromatic components of C are triangle free.

If G admits a proper coloring C we will also say that G admits a *triangle-free splitting*. Clearly, a triangle-free splitting of a 4-regular graph is a decomposition of G into two triangle-free 2-factors. Further, it makes sense to ask whether a graph $G \in \mathcal{G}(4,2)$ has a triangle-free splitting only if G is even. For an edge e of G we denote by e_t the number of triangles of G containing e. Since $G \in \mathcal{G}(4,2)$, $e_t \leq 3$ for any edge e. Let edges x, y, and z form a triangle T. Then we say that T is of type (x_t, y_t, z_t).

Now we state several auxiliary results. We start with a parity lemma.

Lemma 1. *Let $G \in \mathcal{G}(4,2)$ and C be a coloring of the edges of G by two colors such that each vertex of G of degree 4 is incident with two edges of each color. Let N be the number of vertices v of G of degree 2 such that both edges incident with v get the same color in C. Then the parity of N is the same as the parity of the size of G.*

Proof. Since each vertex of G of degree 4 is incident with two edges in each color, the maximum degree of both monochromatic subgraphs in C is 2. Therefore, each component in both monochromatic subgraphs is either a cycle or a path. Further, a vertex v is a terminal vertex of such a monochromatic path if and only if v is of degree 2 in G and the two edges of G incident with v are of distinct colors in C. Since any path has two terminal vertices there is in G an even number of vertices v of degree 2 with edges incident to v being of distinct colors. Clearly, the parity of the size of G equals the parity of the number of vertices of G of degree 2, which yields that the parity of N equals the parity of the size of G. □

As an immediate consequence of Lemma 1 we get the following lemma.

Lemma 2. *Let a graph $G \in \mathcal{G}(4,2)$ admit a triangle-free splitting and let a vertex v of G be a cut vertex of G. If S is an odd component of $G - v$ then in any proper coloring C of G both edges incident with v and having the other endpoint in S must be colored with the same color.*

Proof. Consider the odd subgraph H of G formed by the edges of S and two edges incident with v having the other end vertex in S. Let C be a proper coloring of G. Then all the vertices of H of degree 2, except v, have the two edges incident with them colored by different colors. The rest of the proof follows from Lemma 1. □

By a *three-triangle* graph, or simply a TT graph, we mean a graph consisting of three triangles with a common edge; see Fig. 1.

Our final lemma will play a crucial role in proving the main result of the paper. First, we introduce one more notion. Let $T = \{v, w, z\}$ be a triangle of G, $d(v) = 4$, and let x, y be the other vertices of G adjacent to v. Then by the *splitting of v with respect to T* we understand the graph $G' = (G - v) \cup \{v'w, v'z, v''x, v''y\}$, where v'

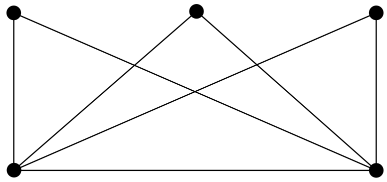


FIG. 1.

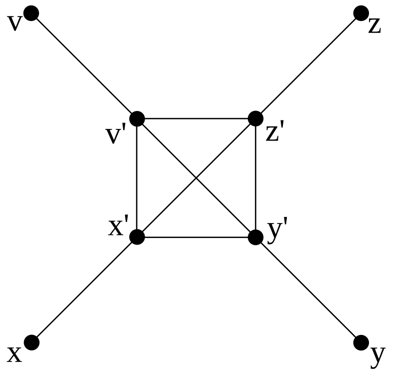


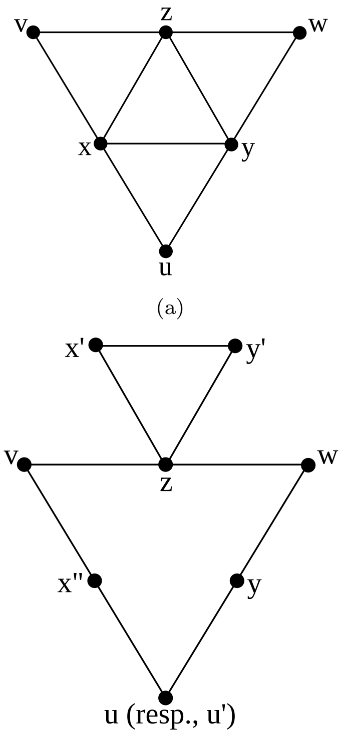
FIG. 2.

and v'' are two new vertices. For the sake of simplicity we view G' as a graph having the same edge set as G.

LEMMA 3. *Let $G \in \mathcal{G}(4,2)$ be an even, connected graph with the following property:* (A) *if T is either a triangle of G of type* $(1,1,1)$ *or an induced TT subgraph of G then one of the vertices of T is a cut vertex of G and the other vertices of T are incident in G only with edges of T. Then G admits a triangle-free splitting.*

Proof. We prove the statement by induction with respect to the number of triangles in G. If there is no triangle in G, then one can get the desired coloring by taking an Eulerian trail of G and alternately coloring its edges. So suppose that there are triangles in G. We will distinguish among five cases. In each of them we construct a graph $G' \in \mathcal{G}(4,2)$ with fewer triangles than G so that each component of G' satisfies the assumptions of the statement. By the induction hypothesis there is a proper coloring C' of G' and we extend (modify) C' to a proper coloring C of G.

Case 1. There are vertices x', y', z', w' in G so that the subgraph induced by them is K_4. The other neighbors of these vertices are x, y, z, w, respectively; see Fig. 2. To get the graph G' we remove a cycle K of length 4 on the vertices x', y', z', and v'. Clearly, for any choice of such a cycle K one cannot create a new induced TT subgraph, and all possible new triangles of type (1,1,1) (this can happen when some of the vertices x, y, z, v are identical) satisfy (A). A choice of K could lead to a disconnected graph G' with possibly odd components to which we cannot apply the induction hypothesis. This could happen only when $H = G$–$\{x', y', z', v'\}$ is a disconnected graph. However, then H has exactly two components. We choose K in such a way that the remaining two edges of K_4 make G' a connected graph. Clearly, G' has a smaller number of triangles than G and by the induction hypothesis there is a proper coloring C' of G'. There are now two possibilities for coloring the edges of the cycle K alternatively by two colors and one of them, in some cases either of them,



(a)

(b)

FIG. 3.

gives the extension of C' to a proper coloring of G. Indeed, one must be careful only in the case when some of vertices x, y, z, v coincide and some additional conditions are met by C'. First suppose that, say $x \equiv y$, the edge $x'y'$ is in K, and the edges xx', yy' get the same color in C'. In order not to get a monochromatic triangle start coloring K from the edge $x'y'$ and assign to $x'y'$ the color not assigned to xx'. We note that if also $z \equiv v$ then the edges zz', vv' must have the same color as xx' and yy'. Alternating the coloring of the edges of K guarantees that $z'v'$ gets the same color as $x'y'$ and no monochromatic triangle is created. By the same token one can deal with the cases when three or all four vertices of x, y, z, v coincide.

Case 2. There is in G a triangle $T = \{x, y, z\}$ of type (2,2,2) and G has no K_4. Suppose first that one of the vertices u, v, w (see Fig. 3(a)), say u, is not an odd cut vertex. Then to construct G' we first, if u is of degree 4, split at u with respect to the triangle $\{u, x, y\}$ and then split at x and y with respect to the triangle T; see Fig. 3(b). Clearly, G' satisfies condition (A) and by the induction hypothesis there is a proper coloring of G'. The same coloring (we view G' as having the same edge set as G) provides a proper coloring of G. Indeed, the vertices x, y, u are incident with an equal number of edges in both colors as the vertices x', x'', y', y'',u', u'' have that property. Further, by Lemma 2, the edges zx', zy' are of the same color. This means the edges zv, zw are of the other color and hence the triangles $\{v, x, z\}$ and $\{w, y, z\}$ are not monochromatic. The triangle $\{u, x, y\}$ cannot be monochromatic because the edges ux, uy are of different colors. To finish the proof of this case we suppose that all vertices u, v, w are odd cut vertices. To get the graph G' we first split at w with

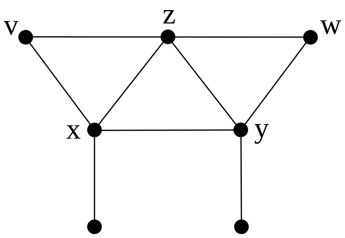


Fig. 4.

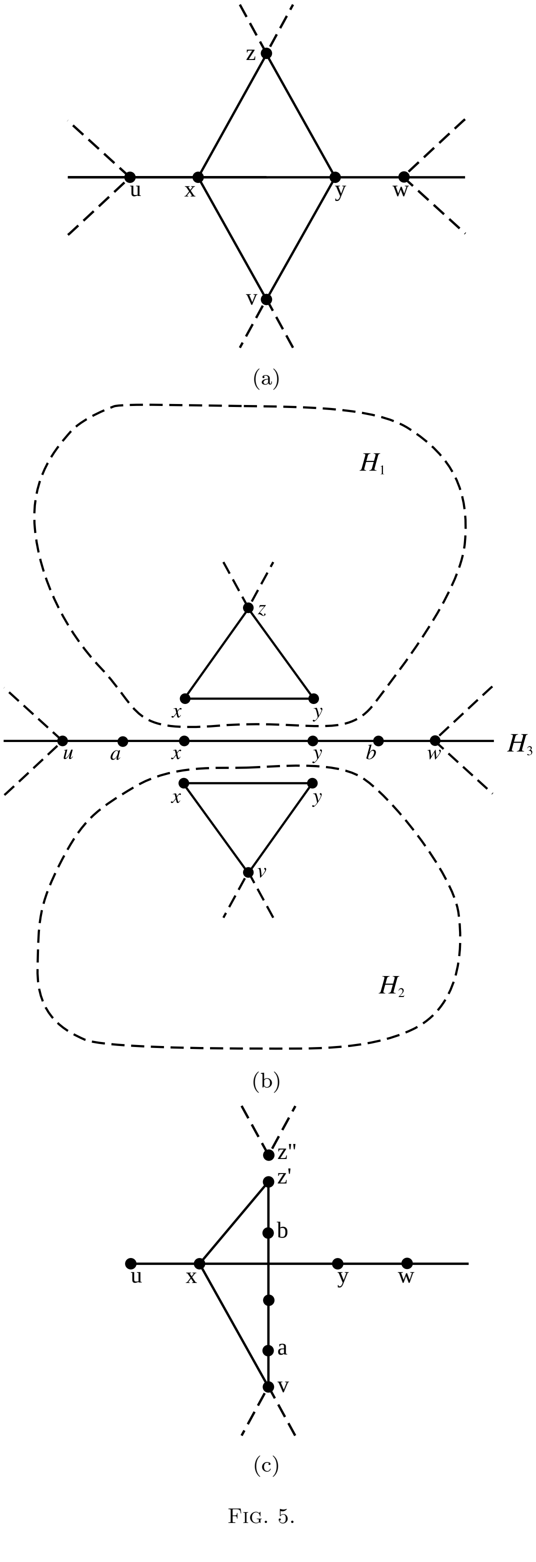
respect to triangle $\{w, y, z\}$, obtaining a graph with two components. Then as before we split at x and y with respect to the triangle T. Finally, we add to vertices w', w'' a loop and then subdivide both loops by four new vertices. Clearly, both components are even and satisfy the assumptions of the statement. We can choose proper colorings of C', C'' of the two components of G' in such a way that the edges of G incident to w' have different colors from the edges incident to w'' (by Lemma 2 the edges incident to w' (to w'') have the same color). Now it is a routine matter to check that a coloring of G given by the restriction of C' and C'' to the edges of G is a proper coloring, since the edges xz, yz, and uy are of the same color and the edges vz, wz, and xy are of the other color. This implies that no triangle on the vertex set from $\{x, y, z, u, v, w\}$ is monochromatic.

Case 3. There is in G a triangle $T = \{x, y, z\}$ of type (1,2,2); see Fig. 4. This case is simpler then the previous one, and we will use an argument very similar to that of the first part of Case 3. To construct G' we split at the vertices x and y with respect to T. By the induction hypothesis there is a proper coloring C' of G'. As before, C' also provides a proper coloring of G (again we view G' as having the same edge set as G).

Case 4. There is in G a triangle T=$\{x, y, z\}$ of type (1,1,2) as in Fig. 5(a). The edges depicted by broken lines may or may not be in G.

We assume that edges uv, vw, uz, zw are not in G, for otherwise there would be a triangle of type (1,2,2). Suppose first that both vertices v and z are odd cut vertices. In this case the broken edges incident with v and z are in G. Then we construct three even connected graphs H_1, H_2, H_3 as in Fig. 5(b), a, b being new vertices. Any of them satisfies the assumptions of the statement and has fewer triangles than G. By induction we may take such proper colorings of these graphs where the edge xy has in all three of them the same color. The union of the three colorings provides a proper coloring of G, where the edges ux, yw get the color of the edge xy. Thus we may now assume that z is not an odd cut vertex of G. To get G' first, if z is of degree 4, we split at z with respect to T and then modify the obtained graph (possibly having two even components) as in Fig. 5(c). A proper coloring of G can be obtained from any proper coloring of G' by giving the edge vy the color of the edge va and giving the edge zy the color of zb.

Case 5. There is in G a subgraph T which is either a triangle of type (1,1,1) or an induced TT subgraph of G, satisfying (A), and a vertex x of T is a cut vertex of G. To obtain G' we subdivide the two edges of T incident with x by two new vertices y, z. Since G' has fewer triangles than G and G' satisfies the assumptions of the statement there is a proper coloring C' of G'. To obtain a proper coloring C of G we color the edges of G which do not belong to T by the same color as in C', the edges of T incident with x get the color used in C' for edges xy, zy (by Lemma 2 the



(a)

(b)

(c)

FIG. 5.

two edges have in C' the same color), and the other edges (edge) of T are colored in an obvious way to get a proper coloring. $\square$

The main result. The following statement constitutes the main result of this paper.

THEOREM 1. *The decision problem "Given $G \in \mathcal{G}(4,2)$, does G have a triangle-free splitting?" can be solved in polynomial time.*

Proof. To prove the statement we show that the decision problem can be reduced to a special case of the general f-factor problem; for a detailed discussion of the matter see [7]. For the sake of completeness we recall that the general f-factor problem asks whether there exists a subgraph of a graph $G = (V, E)$, say $F = (V, E')$, $E' \subset E$, such that $d_F(v) \in B_v$, where B_v is a subset of the set $\{1, \dots, d_G(v)\}$ for all $v \in V$. This problem is NP-complete. However, Cornuéjols [1] showed that if in each B_v all the gaps (if any) have length 1 then the problem can be solved in polynomial time (a set B_v is said to have a gap of length p if there is an integer $k \in B_v$ so that $k+1+p \in B_v$ but no number between these two is in B_v).

If there is no triangle of type (1,1,1) and no induced TT subgraph in G we are done using Lemma 3. So suppose that $T = \{T_1, \dots, T_n\}, n > 0$, is the set of all triangles of G of type (1,1,1) and of all induced TT subgraphs of G. Denote by $G(T)$ the graph obtained from G by removing all the edges of the subgraphs from T; if $T_i \in T$ is an induced TT subgraph then we also remove from G the two vertices of T_i which are of degree 4 in T_i. Let $O_1, \dots, O_s$ and $E_1, \dots, E_r$ be odd and even components of $G(T)$, respectively. A component comprising a single vertex is considered an even component. We construct a bipartite graph B with bipartition $(T', E \cup O)$, where the vertices of $T' = \{t_1, \dots, t_n\}$ represent the subgraphs from T and the vertices of $E \cup O = \{o_1, \dots, o_s\} \cup \{e_1, \dots, e_r\}$ represent the components of $G(T)$. Further, $t_io_j(t_ie_j)$ is an edge of B if an edge of the subgraph T_i is incident with a vertex of O_j (a vertex of E_j). Clearly, $d(t_i) \leq 3$ for $i = 1, \dots, n$. Now we prove the following.

(*) The graph G has a triangle-free splitting if and only if there is a subgraph F of B such that $d_F(t_i) = 1$ for $i = 1, \dots, n$, $d_F(o_j)$ is odd for $j = 1, \dots, s$, and $d_F(e_j)$ is even for $j = 1, \dots, r$.

First we prove the necessity of the condition. Let C be a proper coloring of G. Each subgraph $T_i \in T$ has three vertices of degree 2 in T_i. Exactly one of them has both edges incident with the vertex colored with the same color. We call this vertex the monochromatic vertex of T_i. We define a subgraph F of B by letting an edge t_io_j (t_ie_j) belong to F if the monochromatic vertex of T_i is in the component O_j (E_j). Since each subgraph T_i has exactly one monochromatic vertex, $d_F(t_i) = 1$. Further, if $v \in O_j$ $(v \in E_j)$ is a monochromatic vertex of T_i then v must be of degree 4 in G and the two edges incident to v which are not in T_i must be of the same color. Thus the coloring C restricted to a component K of $G(T)$ provides a coloring of K such that each vertex v of degree 2 in K has both edges incident with it of the same color if and only if v is a monochromatic vertex of a subgraph from T. By Lemma 1 the parity of the number of such vertices coincides with the parity of the size of the component. Hence the parity of the degree of the vertex k in F, where k is the vertex representing the component K, is the same as the parity of the size of K. This finishes the proof of this part of the statement.

Suppose now that F is a subgraph of B as in (*). We show how to construct a proper coloring of G. If t_io_j (t_ie_j) is an edge of F then we choose a vertex of T_i which is in O_j (E_j) to be a monochromatic vertex of the subgraph T_i. Now we take G and split at each vertex of T_i which is of degree 2 in T_i and is not its monochromatic vertex.

Denote the obtained graph as G^*. Components of G^* can be matched in a natural way with the components of $G(T)$. In fact, if D is a component of $G(T)$ then the match of D in G^* is a component D^*, where D^* comprises all the edges of D and the edges of those subgraphs from T which have their monochromatic vertex in D. The size of D^* is even since the parity of the number of subgraphs from T which are "attached" to D to form D^* equals the parity of the size of D. Clearly, each component of G^* satisfies the assumptions of Lemma 3, and therefore each component of G^* has a proper coloring. A coloring C of the edges of G which is the union of proper colorings of components of G^* is a proper coloring. Indeed, if a vertex v was split during the procedure of constructing G^* then both new vertices v_1 and v_2 are of degree 2 and the edges incident with $v_i, i = 1, 2$, are of different colors. Thus, in G, v is incident with two edges of each color. We note that a vertex v which is the monochromatic vertex of T_i is really incident with the edges of T_i of the same color since v is an odd cut vertex in the component of G^* containing the edges of T_i; cf. Lemma 2.

It is obvious that the reductions from the graph G to the graph $G(T)$ and from $G(T)$ to the graph B are polynomial. From the mentioned result of Cornuéjols it also follows that the decision problem of whether B possesses a required subgraph F described in the condition (*) can be solved in polynomial time. Thus our decomposition problem is polynomial. $\square$

Remark. Clearly, the proof of Lemma 3 provides a polynomial algorithm for finding a proper coloring of the components of the graph G^*. Thus, following the proof of Theorem 1, together with the polynomial algorithm for the special case of the general f-factor problem, one can easily obtain a polynomial algorithm for the decomposition problem.

Finally, we show how to construct even graphs from $\mathcal{G}(4,2)$ which do not have a triangle-free splitting. We will make use of the following theorem. Here, the bipartite graph B and the set T of subgraphs of G are the same as in the proof of Theorem 1.

THEOREM 2. *Let $G \in \mathcal{G}(4,2)$ admit a triangle-free splitting. Then the number of odd components of B is at most the cardinality of T.*

Proof. By the condition (*), B has a subgraph F such that each vertex of B representing a subgraph of T is of degree 1 in F and each vertex of B representing an odd connectivity component of $B(T)$ is of degree at least 1. Thus the number of odd components of $B(T)$ is at most the cardinality of T. $\square$

Theorem 2 provides a hint toward constructing some even graphs $G \in \mathcal{G}(4,2)$ which do not have a triangle-free splitting. Let H be a graph having a triangle T of type (1,1,1) such that all vertices of T are odd cut vertices. By Lemma 2, in any proper coloring of H all edges of T must have the same color, which is a contradiction. Thus, H does not admit a triangle-free splitting. Suppose now that u, v are vertices of H of degree 2. Consider an even graph H' consisting of a triangle $T' = \{x, y, z\}$, where x, y are of degree 2 and z is an odd cut vertex. Construct a new graph H'' by identifying vertices x and u, obtaining a new vertex of degree 4, and possibly also identifying y and v. H'' does not admit a proper coloring since this coloring restricted to the edges of H would have to be a proper coloring of H. Thus, in this way, we can construct an infinite class of graphs not admitting a triangle-free splitting.

Acknowledgments. This research was carried out while the second author was visiting the Department of Mathematics of the University of Hawaii at Manoa. He would like to thank the department for its hospitality. He is also grateful to K. Heinrich for drawing his attention to the problem studied in this paper and to Zs. Tuza for early discussions.

REFERENCES

[1] G. Cornuéjols, *General factors of graphs*, J. Combin. Theory Ser. B, 45 (1988), pp. 185–198.

[2] A. Dolinski and M. Tarsi, *Graph decomposition is NPC, a complete proof of Holyer's conjecture*, in Proc. 24th Annual ACM symposium on Theory of Computing, Victoria, BC, 1992, pp. 252–263.

[3] J. Edmonds and E. L. Johnson, *Matching: A well solved class of integer linear programs*, in Combinatorial Structures and Their Applications, R. Guy, H. Hanani, N. Sauer, and J. Schönheim, eds., Gordon and Breach, 1970, pp. 89–92.

[4] D. Hartvigsen, *Extensions of Matching Theory*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, 1984.

[5] P. Hell, D. Kirkpatrick, J. Kratochvíl, and I. Kříž, *On restricted two-factors*, SIAM J. Discrete Math., 1 (1988), pp. 472–484.

[6] M. Koudier and G. Sabidussi, *Factorization of 4-regular graphs and Petersen's theorem*, J. Combin. Theory Ser. B, 63 (1995), pp. 170–184.

[7] L. Lovász and M. D. Plummer, *Matching Theory*, Ann. Discrete Math. 29, North–Holland, 1986.

[8] W. Tutte, *The factors of graphs*, Canad. J. Math., 4 (1952), pp. 314–328.

SIAM J. DISCRETE MATH.
Vol. 10, No. 2, pp. 318–335, May 1997

THE STRUCTURE OF RANDOM GRAPH ORDERS[*]

BÉLA BOLLOBÁS[†] AND GRAHAM BRIGHTWELL[‡]

Abstract. The random graph order $P_{n,p}$ is defined by taking a random graph $G_{n,p}$ on vertex set $[n]$, treating an edge ij with $i \prec j$ in $[n]$ as a relation $i < j$, and taking the transitive closure. A *post* in a partial order is an element comparable with all others. We investigate the occurrence of posts in random graph orders, showing in particular that $P_{n,p}$ almost surely has posts if $np^{-1}e^{-\pi^2/3p} \to \infty$, but almost surely does not if this quantity tends to 0. If there are many posts, the partial order decomposes as a linear sum of smaller orders, and we use this decomposition to show that many parameters of a random graph order—for instance, the height, the logarithm of the number of linear extensions, and the number of incomparable pairs—behave as normal random variables. For instance, for the height $H_{n,p}$, we prove that, for p in an appropriate range, there are functions $\alpha_H(p) = e(1+o(1))p$ and $\beta_H(p)$ such that $(H_{n,p} - \alpha_H(p)n)/\sqrt{n}\beta_H(p) \xrightarrow{d} N(0,1)$.

Key words. partial order, random graph, random partial order

AMS subject classifications. 06A06, 05C80

PII. S0895480194281215

1. Introduction.

The *random graph order* $P_{n,p}$ is defined by taking a random graph $G_{n,p} \in \mathcal{G}(n,p)$ with vertex set $[n] = \{1, 2, \ldots, n\}$, interpreting an edge between vertices i and j with $i \prec j$ as a relation $i < j$, and taking the transitive closure of the relation $<$. Thus $i < j$ in $P_{n,p}$ if there is an increasing sequence $i = i_1 \prec i_2 \prec \cdots \prec i_k = j$ of vertices such that $i_l i_{l+1}$ is an edge of $G_{n,p}$ for each l. (Note that here, as throughout, we reserve the symbol $\prec$ to denote the underlying linear order on the integers and use $<$ for the random graph order.) In general, p will be a function of n, and we set $q = q(n) = 1 - p$ throughout.

The infinite random graph orders $P_{\mathbf{N},p}$ and $P_{\mathbf{Z},p}$ are defined in exactly the same way, starting with the vertex set $\mathbf{N}$ or $\mathbf{Z}$ in the standard order $\prec$. Observe that $P_{n,p}$ is distributed as the restriction of $P_{\mathbf{N},p}$ or $P_{\mathbf{Z},p}$ to the subset $[n]$: we shall identify $P_{n,p}$ with such a restriction whenever convenient. These infinite random orders will be of use later, but all our results are concerned with finite random graph orders.

Random graph orders have been investigated by Barak and Erdős [5], Albert and Frieze [1], Alon et al. [2], Newman [21], Simon, Crippa, and Collenberg [23], and in two earlier papers [11, 12] of the authors. See also the survey article by Brightwell [14].

One feature of the random graph order that has become apparent is that there is some type of "phase transition" as $p = p(n)$ increases from $o(1/\log n)$ to $\omega(n)/\log n$. For instance, the width $W_{n,p}$ of $P_{n,p}$, that is, the size of a largest antichain in the partial order, behaves very differently on either side of this transition. The following result is taken from Bollobás and Brightwell [11].

THEOREM 1.1. (i) *If* $p \log n \to 0$ *and* $pn \to \infty$*, then* $W_{n,p}$ *almost surely lies between* $1.455p^{-1}$ *and* $2.428p^{-1}$.

[*] Received by the editors December 22, 1994; accepted for publication (in revised form) May 31, 1996.

http://www.siam.org/journals/sidma/10-2/28121.html

[†] Department of Mathematical Sciences, University of Memphis, Memphis, TN 38152 and Department of Pure Mathematics and Mathematical Statistics, University of Cambridge, 16 Mill Lane, Cambridge CB2 1SB, UK (b.bollobas@pmms.cam.ac.uk).

[‡] Department of Mathematics, London School of Economics, Houghton St., London WC2A 2AE, UK (g.r.brightwell@lse.ac.uk).

(ii) *If* $p \log n \to \infty$*, then* $W_{n,p}$ *is almost surely*

$$(1+o(1))\sqrt{\frac{2\log n}{\log(1/q)}}. \qquad \square$$

The proof of Theorem 1.1 in [11] goes some way toward explaining this phenomenon. If $p \log n \to 0$, then there are antichains of size cp^{-1} at all levels of the partial order, but there are none that are substantially larger. However, if $p \log n \to \infty$, then n is larger compared to $1/p$, so we become overwhelmingly likely to find the occasional exceptionally large antichain.

Our aim in this paper is to explore the structure of random graph orders in the upper range of p and hopefully to shed more light on the phase transition.

Let $P_1, \dots, P_m$ be partial orders on disjoint vertex sets $X_1, \dots, X_m$, respectively. The *linear sum* $P_1 \oplus \cdots \oplus P_m$ is the partial order defined on $\bigcup_{i=1}^m X_i$ by setting $x < y$ if either (a) $x \in X_i$, $y \in X_j$, and $i \prec j$ or (b) $x, y \in X_i$ and $x < y$ in P_i. Less formally, the linear sum is the partial order obtained by stacking the partial orders on top of one another in sequence.

We define a *post* in a partial order to be an element comparable with all others. Posts in random graph orders were introduced in Alon et al. [2], who showed that, in the case where p is constant, there are posts, indeed very many of them, in $P_{n,p}$.

Suppose that the random graph order $P_{n,p}$ has posts $x_1 \prec x_2 \prec \cdots \prec x_m$. Then $P_{n,p}$ breaks up as the linear sum of $m+1$ (or possibly just m) smaller partial orders. To be precise, for $1 \le j \le m-1$ let Q_j be the partial order induced by $P_{n,p}$ on the set $(x_j, x_{j+1}]$, let Q_0 be the partial order induced on $[1, x_1]$, and Q_m the order induced on $(x_m, n]$ (which could be empty). Then $P_{n,p} = Q_0 \oplus \cdots \oplus Q_m$. We call the unlabeled partial orders Q_i the *factors* of $P_{n,p}$.

It seems to us to be fundamental to the study of random graph orders to see whether or not the partial order is the linear sum of smaller factors. Dealing solely with whether there are posts turns out to be rather simpler, and it seems clear that the cut-off points for the existence of posts and the break-up into factors will be very close, possibly even almost surely identical. We don't address these issues here, but rather restrict our attention to posts. We prove the following result, extending a result of Alon et al. [2] as far as possible.

THEOREM 1.2. *Set* $Y(n,p) = np^{-1}e^{-\pi^2/3p}$*. If* $Y(n,p) \to \infty$ *as* $n \to \infty$*, then there are almost surely posts in* $P_{n,p}$*. If* $Y(n,p) \to 0$*, then there are almost surely no posts in* $P_{n,p}$*. If* $Y(n,p)$ *converges to a nonzero limit* y*, then the number of posts in* $P_{n,p}$ *is asymptotically a Poisson random variable with mean* $2\pi y e^{\pi^2/6}$.

Theorem 1.2 is proved in section 2, along with other related results.

If we are in a regime where there are many posts, we can draw conclusions about various parameters of random graph orders. Most of the natural parameters of partial orders fall into one of two types, according to their behavior on linear sums.

A parameter f of partial orders is called *maximizing* if, whenever $P = P_1 \oplus P_2$, we have $f(P) = \max\{f(P_1), f(P_2)\}$. Examples of maximizing parameters include the width and the dimension. For "normal" maximizing parameters f, one would expect that in a random graph order $P_{n,p}$ breaking into many factors $f(P_{n,p})$ will almost always be much larger than the value of f for the majority of its factors. Theorem 1.1 is an example of this behavior in the case where f is the width. We shall not explore maximizing parameters further here.

A parameter f of partial orders is called *additive* if, whenever $P = P_1 \oplus P_2$, we have $f(P) = f(P_1) + f(P_2)$. Examples of additive parameters are the number of elements, the height, the logarithm of the number of linear extensions, the number of incomparable pairs of elements, and the jump number. Other parameters, such as the number of covering pairs and the bump number, can be modified slightly so as to make them additive.

It is essentially the case that in a random graph order with many posts the various factors are independent. For finite random graph orders, this statement is slightly complicated by edge effects, so let us turn for a moment to $P_{\mathbf{Z},p}$. In $P_{\mathbf{Z},p}$, x_1 is taken to be the first post strictly to the right of 0, and, with probability 1, the posts and factors form two-way infinite sequences. Let the (labeled) sequence of factors be $\dots Q_{-1}, Q_0, Q_1, Q_2, \dots$. It was noted in [2] that the various factors Q_i are independent random variables, identically distributed except that the distribution of Q_0 is distorted by the requirement that it contain 0. See also Theorem 3.1.

A consequence for us is that in a regime where we have many posts any additive parameter f of a random graph order is distributed as a sum of independent random variables, so it might be expected to have an asymptotically normal distribution. Of course, the number of variables being summed is the number of posts (plus one), which is itself a random variable that is not independent of the summands. In particular, if $f(P)$ is proportional to the number of elements of P, then $f(P_n)$ is a constant random variable. However, we shall show in section 3 that all other nonpathological additive parameters are asymptotically normally distributed, provided the expected number of posts is sufficiently large. This idea was first developed in Alon et al. [2], for constant p, using results from the theory of stopped random walks. Here we are interested in the case where p tends to 0 slowly, and we must use more elementary tools—in particular, we use the Berry–Esseen Theorem.

We prove a general result, covering several specific examples. For instance, we obtain the following theorem for the height $H_{n,p}$ of $P_{n,p}$.

THEOREM 1.3. *Let $H_{n,p}$ be the height of the random graph order $P_{n,p}$. Suppose $p \to 0$ and $p \geq (1+\epsilon)\pi^2/\log n$ for some $\epsilon > 0$; then there are functions $\alpha_H(p) = e(1+o(1))p$ and $\beta_H(p)$ such that $(H_{n,p} - \alpha_H(p)n)/\sqrt{n}\beta_H(p) \xrightarrow{d} N(0,1)$.*

One unfortunate feature of our methods is that we do not in general get good estimates for the mean and variance of our asymptotically normal random variables. In Theorem 1.3, the estimate for $\alpha_H(p)$ requires a separate argument—in fact, this is taken from a result of Newman [21]—while we have no sensible bounds at all for $\beta_H(p)$. The development of methods for obtaining bounds on the variance of additive parameters (especially lower bounds) seems to us to be an area worth further study.

2. Posts. Here we give more detail about the distribution of the set of posts. The significance of our results is that the appearance of posts marks a change in the structure of the partial order: if p is small, the partial order is reasonably homogeneous, and it makes sense to treat it as one structure, while if p is large enough that there are many posts, then the random graph order is better viewed as the linear sum of many rather smaller partial orders. In this latter regime, "local" parameters of the partial order, such as width and dimension, depend on the exceptional components of the linear sum and tell us little about the partial order as a whole. We have already seen one example of this, in that the behavior of the width of the partial order changes radically as p goes through $C/\log n$. Theorem 1.2 tells us that this is also the threshold for the appearance of posts in the partial order—indeed that something much sharper is true.

The problem we face in dealing with the set of posts is the lack of independence. For any two vertices x and y in $[n]$, the events that x and y are posts are dependent, in fact, positively correlated. If $|x-y|$ is large, this correlation is small, but if, in the extreme case, $|x-y|=1$, then the two events are fairly strongly correlated, as we shall see. Our approach is to define a slight variant of the notion of a post so that we do obtain independence in the case where $|x-y|$ is large.

Accordingly, for $M \geq 1$, we define an element x in the interval $[M+1, n-M]$ to be an M-post of $P_{n,p}$ if x is comparable with all the elements $x-M, x-M+1, \ldots, x-1, x+1, \ldots, x+M$ in $P_{n,p}$. Let $A_x = A_x(M)$ be the event that x is an M-post. Observe that if $|x-y| \geq 2M$, then the events A_x and A_y depend on disjoint sets of edges in the underlying random graph, and so are independent. Indeed A_x is independent of the set of events $\{A_y : |x-y| \geq 2M\}$.

Note that if $P_{n,p}$ is identified with the restriction of $P_{\mathbf{Z},p}$ to $[n]$, then we have that, for $x \in [M+1, n-M]$, "x a post in $P_{\mathbf{Z},p}$" implies "x a post in $P_{n,p}$" implies "x an M-post in $P_{n,p}$."

We shall choose M so that, almost surely, all M-posts are posts in $P_{n,p}$, or indeed posts in $P_{\mathbf{Z},p}$. We shall then prove that in a suitable range of p the number of M-posts in $P_{n,p}$ converges in distribution to a Poisson random variable.

Let us first find the probability that x is an M-post for $M < x \leq n-M$. For $1 \leq k \leq M$, the probability that the vertex $x+k$ is comparable with x, given that all of $x+1, \ldots, x+k-1$ are comparable with x, is just $1-q^k$. So the probability that x is comparable with all of $x+1, \ldots, x+M$ is

$$\prod_{i=1}^{M} \left(1-q^i\right) \equiv \eta_M(p).$$

The event that x is comparable with all of $x-1, \ldots, x-M$ is independent of this, and also has probability $\eta_M(p)$, so we have

$$\Pr(A_x) \equiv \Pr(x \text{ is an } M\text{-post}) = \eta_M(p)^2.$$

As $M \to \infty$, $\eta_M(p)$ tends to a limit $\eta(p)$, which is thus the square root of the probability that x is a post in the infinite random graph order $P_{\mathbf{Z},p}$, as noted in Alon et al. [2].

The function $\eta(p) = \prod_{i=1}^{\infty} \left(1-q^i\right)$ is better known as the reciprocal of the generating function for the partition function. As such, it has been studied extensively, and the following very precise estimate was found by Hardy and Ramanujan [18] in the course of their work on the asymptotic behavior of the partition function. The first estimate is quoted (essentially) from Hall [17, equation 4.2.11]. The second follows readily upon using the estimate $\log(1/q) = p + \frac{1}{2}p^2 + O(p^3)$.

LEMMA 2.1. *As $p \to 0$,*

$$\log \eta(p) = \frac{-\pi^2}{6\log(1/q)} - \frac{1}{2}\log\log(1/q) + \frac{1}{2}\log(2\pi) + o(1).$$

Thus

$$\eta(p)^2 = (1+o(1))2\pi e^{\pi^2/6} e^{-\pi^2/3p} p^{-1}. \qquad \square$$

Curiously, the functions $\eta_M(p)$ and $\eta(p)$ appear in several other places in the theory of random orders; see Brightwell [13] and our earlier paper [11]. The next estimate is taken from [11, Lemma 11].

Lemma 2.2. *Suppose* $0 < p = 1 - q < 1$ *and* v *is a positive integer with* $q^v \le 1/2$. *Then we have*

$$\frac{q^{v+1}}{\log(1/q)} \le \log \eta_v(p) - \log \eta(p) \le \frac{q^v + \frac{1}{2}q^{2v}}{\log(1/q)}. \qquad \Box$$

We now prove two useful results, which combine to show that the number of posts is almost certainly not too different from the number of M-posts in $P_{n,p}$.

Lemma 2.3. *Suppose* $1 \le M \le n/2$.

(i) *The expected number of* M*-posts in* $P_{n,p}$ *that are not posts of* $P_{\mathbf{Z},p}$ *is at most* $2n\eta_M(p)^2 q^M/p$. *In particular, for* $M \ge r \log n/p \ge r \log n/\log(1/q)$, *the expected number of* M*-posts that are not posts is at most* $2n^{1-r}$.

(ii) *The expected number of posts of* $P_{n,p}$ *in the set* $\{1, 2, \dots, M, n-M+1, \dots, n\}$ *is at most* $2M\eta_{n-M}(p)$.

Proof. (i) Fix a vertex $x \in [M+1, n-M]$, and, for $j \ge M$, let B_j be the event that, in $P_{\mathbf{Z},p}$, x is comparable with all of $x-M, \dots, x-1, x+1, \dots, x+j$, but incomparable with $x+j+1$. The probability of B_j is $\eta_M(p)\eta_j(p)q^j \le \eta_M(p)^2 q^j$. Hence the probability that x is an M-post, but is incomparable with $x+j+1$ for some $j \ge M$, is at most

$$\sum_{j=M}^{\infty} \eta_M(p)^2 q^j = \eta_M(p)^2 q^M/p.$$

Similarly, the probability that x is an M-post incomparable with $x-j-1$ for some $j \ge M$ is also at most $\eta_M(p)^2 q^M/p$, so the probability that x is an M-post but not a post in $P_{\mathbf{Z},p}$ is at most $2\eta_M(p)q^M/p$.

Note that $\eta_M(p) \le p$, so the above probability is certainly at most $2q^M$. Also, if $M \ge r \log n/\log(1/q)$, then we have $q^M \le n^{-r}$, proving the final assertion of (i).

Part (ii) is straightforward since the probability that an element $x \in [M]$ is comparable with all the elements $x+1, x+2, \dots, n$ is at most $\eta_{n-M}(p)$, and likewise for the "top" elements. $\Box$

We immediately deduce the following, establishing one of the assertions of Theorem 1.2.

Theorem 2.4. *Suppose* $n\eta(p)^2 \to 0$ *as* $n \to \infty$. *Then there are almost surely no posts in* $P_{n,p}$. *In particular, if*

$$\frac{1}{p} - \frac{3}{\pi^2}(\log n + \log\log n) \to \infty$$

as $n \to \infty$, *then there are almost surely no posts in* $P_{n,p}$.

Proof. Since the probability of having no posts is decreasing in p, we may (and shall) assume that $p \ge n^{-1/3}$. Set $M = \lfloor \sqrt{n} \rfloor$. The expected number of M-posts in $P_{n,p}$ is at most $z = n\eta_M(p)^2$, and the expected number of posts of $P_{n,p}$ among $\{1, \dots, M, n-M+1, \dots, n\}$ is at most $2M\eta_{n-M}(p) \le 2\sqrt{z}$ by Lemma 2.3(ii). The expected number of posts in $P_{n,p}$ is at most the sum of these two terms. Now we have

$$z = n\eta_M(p)^2 \le n\eta(p)^2 \exp\left(\frac{2q^M + q^{2M}}{\log(1/q)}\right)$$

by Lemma 2.2. The term $(2q^M + q^{2M})/\log(1/q)$ tends to 0, and $n\eta(p)^2 \to 0$ by assumption, so $z \to 0$, implying the first statement.

For the second statement, we suppose that

$$\frac{1}{p} = \frac{3}{\pi^2} \left(\log n + \log\log n + \omega(n)\right),$$

where $\omega(n) \to \infty$ with $\omega(n) \le \log\log n$, and use Lemma 2.1 to obtain that

$$\begin{aligned} n\eta(p)^2 &= O\left(np^{-1}e^{-\pi^2/3p}\right) \\ &= O\left(\exp\left(\log n + \log\log n + O(1) - (\log n + \log\log n + \omega(n))\right)\right) \\ &= O(\exp(-\omega(n))) = o(1), \end{aligned}$$

as required. □

Our next aim is to show that the number $A(M) = A(M,n,p)$ of M-posts in $P_{n,p}$ converges in distribution to a Poisson random variable with mean $(n-2M)\eta_M(p)^2$. We shall then be able to deduce a similar result for the number of posts in $P_{n,p}$.

We use the following result, due, in this form, to Arratia, Goldstein, and Gordon [3], although it is implicit in earlier work of Chen [15] and Barbour and Eagleson [6]. Recall that, for random variables X and Y taking values in a set T, the total variation distance $d_{\mathrm{TV}}(X,Y)$ between X and Y is the maximum, over all subsets S of T, of $|\Pr(X \in S) - \Pr(Y \in S)|$. Recall also that a sequence (X_n) of real-valued random variables is said to converge in distribution to a random variable X—we write $X_n \xrightarrow{d} X$—if $\Pr(X_i < x) \to \Pr(X < x)$ whenever $\Pr(X < x)$ is continuous at x. For an integer-valued random variable, we have that $X_n \to X$ iff $d_{\mathrm{TV}}(X_n, X) \to 0$ as $n \to \infty$. For positive real λ, let $\mathrm{Po}(\lambda)$ denote a Poisson random variable with mean λ.

THEOREM 2.5. *Let $(A_\alpha)_{\alpha\in I}$ be a family of Bernoulli random variables. For $\alpha,\beta \in I$, set $p_\alpha = \mathbf{E}A_\alpha$ and $p_{\alpha\beta} = \mathbf{E}A_\alpha A_\beta$. For $\alpha \in I$, let C_α be a subset of I such that A_α is independent of $\{A_\beta : \beta \in C_\alpha\}$ and set $B_\alpha = I \setminus C_\alpha$. Let*

$$b_1 = \sum_{\alpha\in I}\sum_{\beta\in B_\alpha} p_\alpha p_\beta, \quad b_2 = \sum_{\alpha\in I}\sum_{\beta\in B_\alpha, \beta\ne\alpha} p_{\alpha\beta}.$$

Let $A = \sum_{\alpha\in I} A_\alpha$ and $\lambda_0 = \mathbf{E}A = \sum_{\alpha\in I} p_\alpha$. Then

$$d_{\mathrm{TV}}(A, \mathrm{Po}(\lambda_0)) \le (b_1 + b_2)\frac{1-e^{-\lambda_0}}{\lambda_0}. \qquad \square$$

THEOREM 2.6. *Suppose that $M \le 2^{1/p-3}$. Let $A(M) = A(M,n,p)$ be the number of M-posts in $P_{n,p}$ and let $\lambda_M = (n-2M)\eta_M(p)^2$. Then*

$$d_{\mathrm{TV}}(A(M), \mathrm{Po}(\lambda_M)) \le 17p.$$

Proof. For $x \in [M+1, n-M]$, we let A_x be the event that x is an M-post in $P_{n,p}$, so $p_x \equiv \Pr(A_x) = \eta_M(p)^2$. Then $A(M) = \sum_{x=M+1}^{n-M} A_x$ and $\mathbf{E}A(M) = (n-2M)\eta_M(p)^2 = \lambda_M$.

For $x \in [M+1, n-M]$, let $B_x = [M+1, n-M] \cap \{x-2M, x-2M+1, \ldots, x-1, x+1, \ldots, x+2M\}$ and $C_x = [M+1, n-M] \setminus B_x$. As mentioned earlier, A_x is independent of $\{A_y : y \in C_x\}$. Now let b_1 and b_2 be as in Theorem 2.5. Note that, by Kleitman's

Lemma [19] (see, e.g., [10]), the events A_x are mutually positively correlated, so that $p_x p_y \leq p_{xy}$ for every x and y. Thus we have

$$b_1 \leq b_2 + \sum_{x=M+1}^{n-M} p_x^2 = b_2 + (n-2M)\eta_M(p)^4 = b_2 + \lambda_M \eta_M(p)^2,$$

so it remains to estimate b_2.

Suppose that x and $x+k$ are vertices with $M+1 \leq x < x+k \leq n-M$ and that $k \leq 2M$. Set $l = \lfloor k/2 \rfloor$. We want an upper bound for the probability that x and $x+k$ are both M-posts. This is certainly at most the probability that x is comparable with all the elements in $\{x-M, \ldots, x+l\}$ and $x+k$ is comparable with all the elements in $\{x+l, \ldots, x+k+M\}$. These two events are independent: the first has probability $\eta_M(p)\eta_l(p)$ and the second has probability $\eta_M(p)\eta_{k-l}(p)$. Set $f(k) = \eta_l(p)\eta_{k-l}(p)$, so that the probability that x and $x+k$ are both M-posts is at most $\eta_M(p)^2 f(k)$.

Some relatively crude bounds for $\eta_m(p)$ in the range $1 \leq m \leq M$ will suffice. We have

$$\eta_m(p) = \prod_{i=1}^{m} \left(1 - q^i\right) \leq \prod_{i=1}^{m} pi = m! p^m.$$

Hence, for $l = \lfloor k/2 \rfloor$, we have

$$f(k) = \eta_l(p)\eta_{k-l}(p) \leq p^k l!(k-l)! \equiv g(k).$$

Note that $g(1) = p$ (which is indeed the probability that x and $x+1$ are both M-posts, given that x is comparable with the elements below it and $x+1$ comparable with the elements above it). For $p(k+1) \leq 1$, we see that $g(k) \leq g(k-1)/2$, so $f(k) \leq g(k) \leq p2^{1-k}$. For $k \geq \lfloor 1/p \rfloor$, we simply note that

$$f(k) \leq f(\lfloor 1/p \rfloor - 1) \leq p2^{3-1/p}.$$

Thus, for $x \in [M+1, n-M]$, we have

$$\begin{aligned}
b_2 &= \sum_{x=M+1}^{n-M} \sum_{y \in B_x} \Pr(A_x \cap A_y) \\
&\leq 2(n-2M)\eta_M(p)^2 \sum_{k=1}^{2M} f(k) \\
&\leq 2(n-2M)\eta_M(p)^2 \left(\sum_{k=1}^{\lfloor 1/p \rfloor - 1} f(k) + 2Mp2^{3-1/p} \right) \\
&\leq 2(n-2M)\eta_M(p)^2 \left(2p + 16Mp2^{-1/p}\right) \\
&= 4(n-2M)p\eta_M(p)^2 \left(1 + 8M2^{-1/p}\right) \\
&\leq 8(n-2M)p\eta_M(p)^2 = 8p\lambda_M,
\end{aligned}$$

where at the end we used the assumption that $M \leq 2^{1/p-3}$.

We can now apply Theorem 2.5 to conclude that the number $A(M)$ of M-posts in $P_{n,p}$ satisfies

$$d_{\mathrm{TV}}(A(M), \mathrm{Po}(\lambda_M)) \leq (b_1 + b_2)\frac{1 - e^{-\lambda_M}}{\lambda_M} \leq \frac{2b_2 + \lambda_M \eta_M(p)^2}{\lambda_M} \leq 16p + \eta_M(p)^2 < 17p,$$

as required. □

It is a straightforward matter to convert our result about M-posts into the corresponding result for posts. The following result, combined with the estimates in Lemma 2.1, implies Theorem 1.2.

Theorem 2.7. *Suppose that $p \leq 1/2 \log\log n$. Let A be the number of posts in $P_{n,p}$ and set $\lambda = n\eta(p)^2$. Then, for sufficiently large n,*

$$d_{\mathrm{TV}}(A, \mathrm{Po}(\lambda)) \leq 21p.$$

Proof. Set $M = \lceil 2 \log n/p \rceil$. The upper bound on p then implies that, for n sufficiently large, $M \leq 2^{1/p-3}$. Hence, by Theorem 2.6, the number A_M of M-posts satisfies

$$d_{\mathrm{TV}}(A_M, \mathrm{Po}(\lambda_M)) \leq 17p.$$

From Lemma 2.3(i), the probability that any of these M-posts are not posts is at most $2/n$. Also, by Lemma 2.3(ii), the expected number C of posts in the set $\{1, \ldots, M, n - M + 1, \ldots, n\}$ is at most

$$2M\eta_{n-M}(p) \leq 2M \exp(-1/p).$$

Our bound on p implies that

$$e^{-1/p}p^{-2} \leq 4(\log\log n)^2(\log n)^{-2}.$$

Hence

$$C \leq 5 \log np^{-1} 4(\log\log n)^2(\log n)^{-2}p^2 \leq p$$

for sufficiently large n.

Therefore, we have

$$d_{\mathrm{TV}}(A, A(M)) \leq C + 2/n$$

and also $|\lambda_M - \lambda| \leq C + 2/n$, so that

$$d_{\mathrm{TV}}(\mathrm{Po}(\lambda_M), \mathrm{Po}(\lambda)) \leq C + 2/n.$$

Combining these bounds and using the fact that $C + 2/n \leq 2p$ for sufficiently large n, we have the result. □

If $n\eta(p)^2 \to \infty$ with $p \leq 1/2 \log\log n$, Theorem 2.7 certainly implies that we almost surely have posts in $P_{n,p}$. Of course, if p is even larger than $1/2 \log\log n$, then the probability of no posts in $P_{n,p}$ is even smaller, so we have proved the first assertion of Theorem 1.2 as well.

The bound of $O(p)$, for the total variation distance between the number of posts and a Poisson random variable, is the best possible since the events that various vertices are posts are moderately positively correlated. In the most extreme case, the probability that x and $x+1$ are both posts in $P_{\mathbf{Z},p}$ is clearly equal to $p\eta(p)^2$, so the probability that $x+1$ is a post, conditional on x being a post, is p, which is quite a high probability in this context. In the range we are concerned with, p is at least $1/\log n$, and we shall be interested in getting error probabilities of the form n^{-k}, so Theorem 2.7 cannot be used for this purpose. We prove the following result, showing that the number of posts is very unlikely to be very far from its mean. As with many similar "large deviation" results, the tool used will be a martingale inequality.

Lemma 2.8. *Let B be the number of posts of $P_{\mathbf{Z},p}$ in $[n]$ and take any $y > 1$. Then*

$$\Pr\left(|B - n\eta(p)^2| > 6y^{3/2}p^{-1}\sqrt{n}\log^{3/2} n\right) \leq n^{(1-y)/2}.$$

Proof. Set $M = y\log n/2p$ and let $A(M) = A(M,n,p)$ be the number of M-posts of $P_{n,p}$. Also let $a = 2y^{3/2}p^{-1}\sqrt{n}\log^{3/2} n$.

For $i = 1,\dots,n$, let H_i be the set of pairs of vertices (j,i) with $j < i$. Consider the effect on $A(M)$ of adding or removing pairs from H_i to/from the underlying graph. If $|k-i| > M$, this cannot affect whether or not k is an M-post, so $A(M)$ will only be changed by at most $2M+1$.

We are thus in a setting where we may apply the following result, based on a martingale inequality due to Azuma [4]. See Bollobás [8, 9] or McDiarmid [20] for further details.

Theorem 2.9. *Suppose that $H_1 \cup \cdots \cup H_m$ is a partition of $[n]^{(2)}$ into m parts. Let $Z(G)$ be a random variable depending on the random graph $G_{n,p}$ with vertex set $[n]$ such that $|Z(G) - Z(G')| \leq h$ whenever G and G' differ only on one of the H_i. Then, for any real a, we have*

$$\Pr(|Z(G_{n,p}) - \mathbf{E}Z(G_{n,p})| > a) \leq 2\exp(-a^2/2mh^2). \qquad \square$$

Applying this result with $Z = A(M)$, $m = n$, $h = 2M+1$, and, as above, $a = 2y^{3/2}p^{-1}\sqrt{n}\log^{3/2} n$, we obtain

$$\Pr\left(|A(M) - \mathbf{E}A(M)| > a\right) \leq 2\exp\left(-\frac{4y^3p^{-2}n\log^3 n}{2n(y\log np^{-1}+1)^2}\right) \leq n^{-y}.$$

By Lemma 2.3(i), we have

$$0 \leq \mathbf{E}A(M) - n\eta(p)^2 = \mathbf{E}(A(M) - B) \leq 2n^{1-y/2}.$$

Thus the probability that $A(M) - B$ is as large as a is at most $2n^{1-y/2}a^{-1} \leq \frac{1}{2}n^{(1-y)/2}$.

Combining these facts gives us that

$$\Pr(|B - n\eta(p)^2| > 3a) \leq n^{-y} + \frac{1}{2}n^{(1-y)/2} \leq n^{(1-y)/2},$$

as required. $\square$

A rather stronger version of the above lemma could doubtless be proved for the case where the expected number of posts is around n^α for some $\alpha < 1$.

Our final result in this section states that, for values of p rather larger than $\pi^2/3\log n$, there are almost surely no extraordinarily long gaps without posts. Again, the bound of $O(p)$ for the total variation distance in Theorem 2.7 is of no use to us, as we want the probability of no posts to be substantially smaller than p. However, it is not too hard to use Theorem 2.6 directly.

THEOREM 2.10. *Suppose $p \le 0.005$. Let $n_0 = n_0(p) = \lceil \eta(p)^{-2} \rceil$. Take any integer r with $2 \le r \le \frac{1}{60} 2^{1/p} p^2$ and set $M = \lceil 2r \log n_0/p \rceil$ and $n_1 = n_0 + 2M$. Then the probability that a given set of rn_1 consecutive vertices of $\mathbf{Z}$ contains no posts of $P_{\mathbf{Z},p}$ is at most 2^{1-r}.*

Proof. The upper bound on r ensures that we have $M \le 2^{1/p-3}$, so we can apply Theorem 2.6 to obtain that, for any range $\{y+1, y+2, \dots, y+n_0\}$, there is an M-post in the range with probability at least $1-e^{-1}-17p > 1/2$. Now, for any x, we consider the r distinct ranges $U_i = \{x + in_1 + 1, \dots, x + in_1 + n_0\}$, $i = 0, \dots, r-1$. Since there is a gap of size $n_1 - n_0 = 2M$ between each U_i, the events that the various U_i contain an M-post are independent, and each have probability at least 1/2. Hence the probability that there is no M-post in the range $X = \{x+1, \dots, x+rn_1\}$ is at most 2^{-r}. One may check that $M \ge (r + \log n_1)/p$. Lemma 2.3 now tells us that, with probability at least $1 - e^{-r}$, all M-posts in $P_{X,p}$ are indeed posts in $P_{\mathbf{Z},p}$, which implies the result. □

It does not seem easy (or particularly interesting) to extend the reasonably accurate bound of Theorem 2.10 to values of r larger than $\frac{1}{60} 2^{1/p} p^2$. We content ourselves with the following cruder argument, which uses an idea first presented in Alon et al. [2].

THEOREM 2.11. *Suppose $p \le 0.005$. Let $n_0 = n_0(p) = \lceil \eta(p)^{-2} \rceil$. Take any integer $s \ge p^{-1}$. Then the probability that a given set of $2s^2 n_0$ consecutive vertices of $\mathbf{Z}$ contains no posts of $P_{\mathbf{Z},p}$ is at most $3sp^{-1}q^s$.*

Proof. Without loss of generality, the set of $2s^2n_0$ vertices commences at 1. Consider the sn_0 vertices, starting at s and increasing in steps of size $2s$. The events that these various vertices are s-posts are independent and have probability at least $\eta(p)^2$. So the probability that none of these vertices is an s-post is at most $(1 - \eta(p)^2)^{sn_0} \le 2e^{-s}$.

Now the probability that among these vertices there is an s-post that is not a post is at most $sn_0 \eta_s(p)^2 q^s p^{-1}$, as in Lemma 2.3. The lower bound on p gives that $\eta_s(p)^2$ is almost equal to $1/n_0$, so the probability that we fail to find a post is at most

$$2e^{-s} + 2sq^s p^{-1} \le 3sq^s p^{-1},$$

as desired. □

We can use the last two results to prove that the tth moments of the gap between posts are not too large.

COROLLARY 2.12. *Suppose $p < 0.005$. Let the random variable Z be the least $x > 0$ such that x is a post in $P_{\mathbf{Z},p}$. For each fixed $t \le 1000$, there is a constant $C(t)$ such that*

$$\mathbf{E}Z^t \le C(t)\eta(p)^{-2t}.$$

Proof. Set $n_0 = \lceil \eta(p)^{-2} \rceil$ and $n_1(r) = n_0 + 4r \log n_0/p$, as in Theorem 2.10. Also set $r_0 = \lfloor \frac{1}{60} 2^{1/p} p^2 \rfloor$. Now we have

$$\begin{aligned}
\mathbf{E}Z^t \le (4tn_1(4t))^t &+ \sum_{r=4t+1}^{r_0} \Pr(Z > (r-1)n_1(r-1))(rn_1(r))^t \\
&+ \sum_{s=\lfloor \sqrt{r_0}/2 \rfloor}^{\infty} \Pr(Z > 2(s-1)^2 n_0)(2s^2 n_0)^t \\
\le (4tn_1(4t))^t &+ \sum_{r=4t+1}^{\infty} 2^{2-r}(rn_1(r))^t + \sum_{s=\lfloor \sqrt{r_0}/2 \rfloor}^{\infty} 3sp^{-1}q^s(2s^2n_0)^t,
\end{aligned}$$

where in the last step we used Theorems 2.10 and 2.11 to estimate the probabilities.

One can readily check that each term in the first sum is at most 9/10 of the previous one, and that, by our condition on p, $n_1(4t) \le 2n_0$. Also, the second sum is comfortably bounded above by 1. Therefore, $\mathbf{E}Z^t \le 10(4tn_1(4t))^t + 1 \le 10(8t)^t n_0^t$, as required. □

It follows once more from Kleitman's Lemma (again, see [10]) that if we condition on 0 being a post, then the probability that the next s vertices are not posts does not decrease. Therefore, Corollary 2.12 also applies, for instance, to the random variable Y, which is the gap between the first and second posts of $P_{\mathbf{Z},p}$ in $[n]$.

3. Normal convergence. In this section, we use the results of the previous section to show that if $p(n)$ is not too small, then "natural" additive parameters of the random graph order $P_{n,p(n)}$ have an asymptotically normal distribution.

For fixed constant p, consider the infinite random graph order $P_{\mathbf{Z},p}$. It was shown by Alon et al. [2] (and indeed it follows immediately from, for instance, Theorem 2.10) that there is, with probability 1, a two-way infinite sequence of posts in $P_{\mathbf{Z},p}$. Let the sequence of posts of this partial order in $\mathbf{N}$ be $X_1, X_2, \ldots$, so for instance X_1 is the first post of $P_{\mathbf{Z},p}$ to the right of 0 and the X_i are random variables taking values in $\mathbf{N}$.

For $i \ge 1$, let $Q_i = Q_i(p)$ be the partial order induced on the interval $(X_i, X_{i+1}]$. Also, let Q_0 be the (isomorphism class of the) partial order induced on $[1, X_1]$. The Q_i are random variables, taking values in the set $\mathcal{Q}$ of finite unlabeled partial orders with a unique maximum and no other post.

It was noted in [2] that the Q_i $(i \ge 0)$ are mutually independent random variables and that the Q_i with $i \ge 1$ are identically distributed. Indeed, we prove the following result.

THEOREM 3.1. *Let Q be an unlabeled partial order in $\mathcal{Q}$ with m minimal elements, a covering pairs, b incomparable pairs, ℓ linear extensions, and s automorphisms. Take any $i \ge 1$ and any event $\mathcal{E}$ concerning factors Q_j with $j < i$. Then*

$$\Pr(Q_i = Q \mid \mathcal{E}) = \ell p^{m+a} q^b / s.$$

Proof. For each fixed x, let us consider the event $\mathcal{B}(x)$ that $\mathcal{E}$ occurs with $X_i = x$ and $Q_i = Q$. We break $\mathcal{B}(x)$ up into the following two independent events. Let $\mathcal{B}_1(x)$ be the event that (i) x is comparable with every element to its left, (ii) there are $i-1$ elements of $[x-1]$ comparable to every element of $\mathbf{Z}$ to the left of x, and (iii) supposing these elements to be indeed posts, $\mathcal{E}$ occurs. Let $\mathcal{B}_2(x)$ be the event that (i) the unlabeled partial order induced on $[x+1, x+|Q|]$ is equal to Q, (ii) $x+|Q|$ is comparable with every element to its right, and (iii) x is comparable with every minimal element in $[x+1, x+|Q|]$. These events are independent because $\mathcal{B}_1(x)$ depends only on edges of the underlying random graph whose right-hand endpoint is at most x, whereas $\mathcal{B}_2(x)$ depends only on edges whose left-hand endpoint is at least x.

The probability of $\mathcal{B}_2(x)$ is equal to $h(Q,p) \equiv \left(\frac{\ell}{s} p^a q^b\right) \eta(p) p^m$: ℓ/s counts the number of order-preserving labelings of Q with the elements of $[x+1, x+|Q|]$ and $p^a q^b$ is the probability that the partial order induced on $[x+1, x+|Q|]$ is the labeled copy of Q. Thus the probability that $\mathcal{E}$ occurs and $Q_i = Q$ is $\sum_x \Pr(\mathcal{B}_1(x)) h(Q,p)$. Also, the probability that $\mathcal{E}$ occurs is the sum over x of the probability that $\mathcal{B}_1(x)$ occurs and that x is comparable with every element to its right, which is $\sum_x \Pr(\mathcal{B}_1(x))\eta(p)$. Therefore, the probability that $Q_i = Q$, conditioned on $\mathcal{E}$, is $h(Q,p)/\eta(p) = \ell p^{a+m} q^b / s$, as desired. □

Note that Q_0 has a different, but related, distribution.

The random order $P_{\mathbf{N},p}$ can be recovered as the linear sum of the Q_i $(i \geq 0)$. To recover $P_{n,p}$, we just need to truncate. Hopefully it is clear what is involved, and there is no need for us to be too precise. Thus the behavior of $P_{n,p}$ and $P_{\mathbf{N},p}$ can in principle be recovered from the distribution of $Q_i(p)$ and the related random variable $Q_0(p)$.

Our aim here is not to investigate in any more detail the distributions of the Q_i, but to make use of the fact that additive parameters of the random order, such as the height, are essentially sums of independent random variables. For instance, the height of $P_{n,p}$ is the sum of the heights of the Q_i. This idea was developed in [2] for fixed p—what is new here is that we shall use the Berry–Esseen Theorem, which provides an estimate for the rate of convergence of a sum of iid random variables to a normal distribution, together with our estimates for the number of posts, to show normal convergence also when $p \to 0$ sufficiently slowly to guarantee enough posts.

From now on, we let f be an additive parameter of partial orders, and let us suppose that $0 \leq f(P) \leq k|P|^s$ for some fixed k and $s \leq 1000$. This is satisfied, for instance, if $f(P)$ is the height, the logarithm of the number of linear extensions, or the number of incomparable pairs of P.

For fixed p and $i \geq 0$, set $N_i = N_i(p)$ equal to $|Q_i(p)|$ $(= X_{i+1} - X_i$, except for $i = 0)$ and $F_i = F_i(p)$ equal to $f(Q_i(p))$. Then $F_i(p) \leq kN_i(p)^s$, and the two-dimensional random variables (F_i, N_i) are mutually independent and, except for $i = 0$, identically distributed. Set

$$\alpha = \alpha(p) = \frac{\mathbf{E}F_i(p)}{\mathbf{E}N_i(p)}$$

and

$$L_i = L_i(p) = F_i(p) - \alpha(p)N_i(p) \quad (i \geq 1).$$

Note that the L_i are iid random variables with $\mathbf{E}L_i = 0$. We can estimate α. Note first that $\mathbf{E}N_i(p) = \eta(p)^{-2}$ and that $\mathbf{E}F_i(p) \leq k\mathbf{E}N_i(p)^s = O(\eta(p)^{-2s})$ by Corollary 2.12 and the remark after. Therefore, $\alpha = O(\eta(p)^{-2(s-1)})$.

The idea is that $f(P_{X_m,p})$ is given by

$$f(Q_0 \oplus \cdots \oplus Q_{m-1}) = \sum_{i=0}^{m-1} F_i = F_0 + \sum_{i=1}^{m-1} (L_i + \alpha N_i) = F_0 + \sum_{i=1}^{m-1} L_i + \alpha X_m.$$

The first term $F_0 \leq kN_0^s$ is almost surely not too large; the second is the sum of $m - 1$ iid random variables with mean 0 and so is, for large m, approximately a normal random variable with mean 0; the third is a fixed constant times the number of elements in the partial order.

It is a little more complicated to deal with the random order $P_{n,p}$ rather than $P_{X_m,p}$, but the principle is the same. The results of the previous section tell us that if $m = \lfloor n\eta(p)^2 \rfloor$, then the number of posts of $P_{\mathbf{Z},p}$ in $[n]$ is unlikely to be too different from m (and so n is unlikely to be very far from X_m). Indeed, fix n, set $m = \lfloor n\eta(p)^2 \rfloor$, and let B be the number of posts of $P_{\mathbf{Z},p}$ in $[n]$. Then, by Lemma 2.8 with $y = 3$, the probability that $|B - m|$ is larger than $32p^{-1}\sqrt{n}\log^{3/2} n$ is at most $1/n$. Let $Q'_B = Q'_B(p)$ be the partial order restricted to $(X_B, n]$ and set $F'_B = f(Q'_B)$ and N'_B

equal to $n - X_B$, the order of Q'_B. Proceeding as above, we have that

$$\begin{aligned} f(P_{n,p}) &= F_0 + \sum_{i=1}^{B-1} F_i + F'_B \\ &= F_0 + F'_B + \sum_{i=1}^{B-1} (L_i + \alpha N_i) \\ &= F_0 + F'_B + \alpha(n - N_0 - N'_B) + \sum_{i=1}^{m} L_i - \sum_{i=B}^{m} L_i. \end{aligned}$$

(We adopt the convention that, for $B > m$, $\sum_{i=B}^{m} L_i \equiv -\sum_{i=m+1}^{B-1} L_i$.) Therefore,

$$f(P_{n,p}) - \alpha n - \sum_{i=1}^{m} L_i = (F_0 - \alpha N_0) + (F'_B - \alpha N'_B) - \sum_{i=B}^{m} L_i. \tag{1}$$

We will show that the right-hand side of (1) is, with very high probability, not too large. Then we will use the Berry–Esseen Theorem to deduce that $f(P_{n,p})$ is close to a normal random variable with mean $\alpha(p)n$ and variance given by m times the variance $\sigma(p)^2$ of $L_i(p)$. (In a typical setting, we may have some estimates for $\alpha(p)$, but are unlikely to have much knowledge of $\sigma(p)$.)

From now on, we assume that $p \geq (2/3 + \epsilon)\pi^2/\log n$, for some $\epsilon > 0$, so that the expected gap between posts, $\eta(p)^{-2}$, is at most $n^{1/2-\epsilon}$, for large enough n, by Lemma 2.1. Applying Theorem 2.10, we see that, with probability at least $1 - 1/n$, there is almost surely no post-free gap in $[n]$ of length as great as $K = 3\log n(\eta(p)^{-2} + 12\log^2 n/p)$. We assume that this is indeed the case. In particular, the "edge terms" $F_0 - \alpha N_0$ and $F'_B - \alpha N'_B$ are both bounded above in absolute value by $kK^s + \alpha K \leq 2kK^s$.

We now turn our attention to the term $\sum_{i=B}^{m} L_i$. As we mentioned earlier, the probability that $|B - m|$ is at most $t \equiv 32p^{-1}\sqrt{n}\log^{3/2} n$ is at least $1 - 1/n$. Consider the sequence of partial sums $S_j = L_{m+1} + \cdots + L_{m+j}$ for $j = 0, \ldots, t$. The sequence (S_j) is a martingale, and the variance of S_t is $t\sigma(p)^2$. Hence, by the Doob–Kolmogorov inequality (see, for instance, [16]),

$$\Pr\left(\max_{1\leq j\leq t} |S_j| \geq d\right) \leq \frac{t\sigma(p)^2}{d^2}$$

for any $d > 0$. We apply this with $d = \sigma(p)n^{1/4+\delta}$, for any $\delta > 0$, and obtain that, with probability at least $1 - tn^{-1/2-2\delta} = 1 - o(1)$, the maximum of the $|S_j|$ is at most $\sigma(p)n^{1/4+\delta}$. If this is the case and $m < B \leq m + t$, then certainly

$$\left|\sum_{i=B}^{m} L_i\right| = \left|\sum_{i=m+1}^{B-1} L_i\right| = |S_{B-m-1}| \leq \sigma(p)n^{1/4+\delta}.$$

A similar argument works for the case where $m - t < B \leq m$, and we conclude that, almost surely, we have $\sum_{i=B}^{m} L_i \leq \sigma(p)n^{1/4+\delta} = o(\sigma(p)\sqrt{n}\eta(p))$, provided we choose $\delta < \epsilon/2$.

Now we come to the sum $\sum_{i=1}^{m} L_i$. We know very little about the random variables $L_i = L_i(p)$, except that they are iid with mean 0 and some finite variance $\sigma(p)^2$. The Berry–Esseen Theorem [7] (or see, for instance, [22]), which we now state, nevertheless gives us information about the rate of convergence of the sum to a normal random variable.

THEOREM 3.2. *Let the random variables* $X_1, \dots, X_m$ *be iid with mean* 0 *and variance* 1. *Suppose that* $\mathbf{E}|X_1|^3 < \rho$. *Then*

$$\sup_x \left| \Pr\left(\frac{1}{\sqrt{m}} \sum_{i=1}^m X_i < x \right) - \Phi(x) \right| \le \frac{2\rho}{\sqrt{m}}.$$

In particular, if $\rho = o(\sqrt{m})$, *then*

$$\frac{1}{\sqrt{m}} \sum_{i=1}^m X_i \xrightarrow{d} N(0,1).$$

As usual, $N(0,1)$ denotes a normal random variable with mean 0 and variance 1 and $\Phi(x)$ denotes its distribution function.

We shall apply the above with X_i, a normalized version of the L_i. For this, we need a bound on $\mathbf{E}|L_1|^3$, which we obtain, rather crudely, as follows:

$$\mathbf{E}|L_1|^3 \le \mathbf{E}L_1^2(2kK^s) + \Pr(L_1 > 2kK^s)\mathbf{E}(|L_1|^3 \mid L_1 > 2kK^s).$$

As in Corollary 2.12, the second term is negligible, and we obtain that $\mathbf{E}|L_1|^3 \le 3kK^s\sigma(p)^2$.

Now we let $X_i = L_i/\sigma(p)$, so $\mathbf{E}|X_i|^3 \le 3kK^s/\sigma(p)$, and apply Theorem 3.2 to obtain that

$$\sup_x \left| \Pr\left(\frac{1}{\sqrt{n\eta(p)}\sigma(p)} \sum_{i=1}^m L_i < x \right) - \Phi(x) \right| \le \frac{6kK^s}{\sigma(p)\sqrt{n\eta(p)}}.$$

The right-hand side above is $o(1)$ provided $K^s = o(\sigma(p)\sqrt{n\eta(p)})$. If this is the case, then we also have, almost surely, that the terms on the right-hand side of (1) are all $o\left(\sum_{i=1}^m L_i\right)$ since this sum is almost surely of order $\sigma(p)\sqrt{n\eta(p)}$. The condition on K^s is equivalent to the requirements that

$$\log^s n\eta(p)^{-(2s+1)} = o(\sigma(p)\sqrt{n})$$

and

$$\log^3 np^{-s} = o(\sigma(p)\sqrt{n\eta(p)}).$$

Thus we have the following result.

THEOREM 3.3. *Suppose* f *is an additive parameter of partial orders satisfying* $f(P) \le k|P|^s$ *for some* k *and* $s \le 1000$. *Take a function* $p = p(n)$ *satisfying* $(2/3 + \epsilon)\pi^2/\log n \le p(n) \le 0.005$ *for some* $\epsilon > 0$. *Let* $\alpha(p) = \mathbf{E}f(Q_1)/\mathbf{E}|Q_1|$ *and* $\sigma(p)^2$ *be the variance of* $L_1(p) = f(Q_1) - \alpha(p)|Q_1|$. *Suppose that* $\eta(p)^{-(2s+1)}\sigma(p)^{-1} = o(\sqrt{n}\log^{-s} n)$ *and* $\eta(p)^{-1}p^{-s}\sigma(p)^{-1} = o(\sqrt{n}\log^{-3s} n)$. *Then*

$$\frac{f(P_{n,p}) - \alpha(p)n}{\sqrt{n\eta(p)}\sigma(p)} \xrightarrow{d} N(0,1). \qquad \square$$

In particular, if $s = 1$ and $\sigma(p) \ge n^{-\epsilon}$ for every $\epsilon > 0$, then the condition $p(n) \ge (1+\epsilon)\pi^2/\log n$ suffices; i.e., we have normal convergence whenever there are at least $n^{2/3+\epsilon}$ posts in the random graph order.

The above condition on $\sigma(p)$ is far from demanding. Indeed one would expect that if the typical factor size $\eta(p)^{-2}$ is a power of n, then so is $\sigma(p)$, so that we should certainly expect normal convergence with just $n^{1/2+\epsilon}$ posts. We know that the probability that $|Q_1| = 1$ is just p, so if $f(P)$ is bounded away from $\alpha(p)$ for P, the one-element partial order, we certainly have $\sigma(p) \geq cp$ for some $c > 0$, which is enough to be able to apply Theorem 3.3.

Although part of the point of this work is to obtain a result as general as Theorem 3.3, it is also obviously important to see that it can be applied to the various familiar additive parameters we have mentioned along the way. We start with the height: this is just a restatement of Theorem 1.3.

Theorem 3.4. *Let $H_{n,p}$ be the height of the random graph order $P_{n,p}$. Suppose $p \to 0$ and $p \geq (1+\epsilon)\pi^2/\log n$ for some $\epsilon > 0$; then there are functions $\alpha_H(p) = e(1+o(1))p$ and $\beta_H(p)$ such that $(H_{n,p} - \alpha_H(p)n)/\sqrt{n}\beta_H(p) \xrightarrow{d} N(0,1)$.*

Proof. Newman [21] proved that in this range (and indeed whenever $pn \to \infty$ and $p \to 0$) the height of $P_{n,p}$ is almost surely $(1+o(1))epn$; i.e., $\alpha(p) = (1+o(1))ep$. Since the one-element partial order has height 1, we see that $\sigma(p) \geq p/2$. The result now follows upon applying Theorem 3.3 with $s = 1$. □

The lower bound $\sigma(p) \geq p/2$ translates to a lower bound on $\beta_H(p)$, but we are convinced this bound is far from the truth. We offer no nontrivial upper bound. The problem of obtaining tight bounds for $\beta_H(p)$ seems to us to be interesting, but probably quite difficult.

The case of p constant was dealt with in Albert and Frieze [1] and Alon et al. [2]. Again, we obtain normal convergence, but the function $\alpha(p)$ is unknown; see Albert and Frieze [1] for estimates.

Theorem 3.5. *Let $L_{n,p}$ be the natural logarithm of the number of linear extensions of the random graph order $P_{n,p}$. Suppose $p \to 0$ and $p \geq (1+\epsilon)\pi^2/\log n$ for some $\epsilon > 0$; then there are functions $\alpha_L(p) = \log(1/p) + O(1)$ and $\beta_L(p)$ such that $(L_{n,p} - \alpha_L(p)n)/\sqrt{n}\beta_L(p) \xrightarrow{d} N(0,1)$.*

Proof. Note first that the number $N(P)$ of linear extensions of a partial order P is bounded above by $|P|!$, so for any $\delta > 0$ we have $\log N(P) = O(|P|^{1+\delta})$, so we may apply Theorem 3.3 with $s = 1+\delta$. The required lower bound on $\sigma(p)$ again follows from a consideration of the one-element partial order.

It remains to justify the estimate given for $\alpha_L(p)$. As proved in Alon et al. [2], the expected value of $N(P_{n,p})$ is equal to $\eta_n(p)p^{-n}$, so $L_{n,p}$ is almost surely at most $n\log(1/p) + O(1/p)$. Rather crudely, any n-element partial order with height at most h has at least $(n/h)!^h \simeq (n/eh)^n$ linear extensions, so $L_{n,p}$ is almost surely at least $n\log(1/e^2p) = n(\log(1/p) - 2)$. □

Theorem 3.6. *Let $I_{n,p}$ be the number of incomparable pairs of elements of the random graph order $P_{n,p}$. Suppose $p \to 0$ and $p \geq (1+\epsilon)5\pi^2/3\log n$ for some $\epsilon > 0$; then there are functions $\alpha_I(p) = p^{-1}\log(1/p)(1+o(1))$ and $\beta_I(p)$ satisfying $(\pi/\sqrt{6})p^{-1} \leq \beta_I(p) \leq \big(2p^{-1}\log(1/p)\big)^{3/2}$ such that $(I_{n,p} - \alpha_I(p)n)/\sqrt{n}\beta_I(p) \xrightarrow{d} N(0,1)$.*

Proof. Here we need to apply Theorem 3.3 with $s = 2$: the required condition is satisfied by this slightly larger lower bound on p.

The estimates on α_I and β_I are derived entirely separately, based on results of Simon, Crippa, and Collenberg [23]. We merely sketch the proof.

Let the random variable X be the number of elements of $P_{\mathbf{N},p}$ incomparable with element 1. Then Simon, Crippa, and Collenberg prove that X has mean and variance given by

$$\mathbf{E}X = \sum_{r=1}^{\infty} \frac{q^r}{1-q^r}, \qquad \sigma^2 X = \sum_{r=1}^{\infty} \frac{q^r}{(1-q^r)^2}.$$

It is a straightforward exercise to estimate these sums in the case where $p \to 0$. Estimating the sum by the integral gives

$$\frac{\log(1/p)}{\log(1/q)} \leq \mathbf{E}X \leq \frac{\log(1/p)}{\log(1/q)} + \frac{q}{p}.$$

For the variance, for small r, the rth term of the sum is about $(rp)^{-2}$, and we obtain that

$$\sigma^2 X = \frac{\pi^2}{6p^2}\left(1 + O(p^{1/2})\right).$$

Ignoring edge effects, the number $I_{n,p}$ of incomparable pairs in $P_{n,p}$ is distributed as a sum of n dependent random variables distributed as X. We thus obtain that

$$\mathbf{E}I_{n,p} = n\frac{\log(1/p)}{\log(1/q)}(1+o(1)) = np^{-1}\log(1/p)(1+o(1)),$$

giving the desired estimate for $\alpha_I(p)$.

To estimate the variance, we need to take a closer look at the dependence among the random variables. Thus let X_i be the number of elements in the set $[i+1, i+2p^{-1}\log(1/p)]$ incomparable with element i. It is easy to see that the distribution of X_i is very close to that of X, so that $Z = \sum_{i=1}^{n} X_i$ is a good approximation to $I_{n,p}$. Now we have

$$\sigma^2 Z = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\mathbf{E}(X_i X_j) - (\mathbf{E}X_i)(\mathbf{E}X_j)\right).$$

For any $i \neq j$, X_i and X_j are monotone decreasing functions on the underlying space of random graphs, so by the FKG inequality (see, for instance, [10]) we have $\mathbf{E}(X_i X_j) \geq (\mathbf{E}X_i)(\mathbf{E}X_j)$. Therefore,

$$\sigma^2 Z \geq \sum_{i=1}^{n} \sigma^2 X_i \approx n\frac{\pi^2}{6p^2},$$

giving the required lower bound on $\beta_I(p) = \sqrt{\frac{1}{n}\sigma^2 I_{n,p}}$. For the upper bound, we note that X_i and X_j are independent whenever $|i-j| \geq 2p^{-1}\log(1/p)$. Furthermore, if i and j are close, then we certainly have $X_i X_j \leq \left(2p^{-1}\log(1/p)\right)^2$, so, extremely crudely, we get that

$$\sigma^2 Z \leq n\left(2p^{-1}\log(1/p)\right)^3,$$

as required. □

The true value of $\beta_I(p)$ is probably on the order of p^{-1} rather than $p^{-3/2}$. One can probably obtain better estimates, and indeed prove normal convergence for $I_{n,p}$ in a significantly wider range of p, without too much difficulty.

The next result is closely related, and one can probably obtain bounds for $\beta_C(p)$ in much the same way. Note, however, that the number of covering pairs (in a certain set) is *not* a monotone property of the underlying random graph.

THEOREM 3.7. *Let $C_{n,p}$ be the number of covering pairs in the random graph order $P_{n,p}$. Suppose $p \to 0$ and $p \geq (1+\epsilon)5\pi^2/3\log n$ for some $\epsilon > 0$; then there are functions $\alpha_C(p) = \log(1/p)(1+o(1))$ and $\beta_C(p)$ such that $(C_{n,p} - \alpha_C(p)n)/\sqrt{n}\beta_C(p) \xrightarrow{d} N(0,1)$.*

Proof. The number of covering pairs is not itself an additive parameter of partial orders. However, let $D(P)$ denote the number of covering pairs of P plus the number of minimal elements. This is still not an additive parameter, but it becomes one if we impose the condition that all factors in the linear sum have a unique maximal element. This condition is satisfied in our setting, where all the factors have a post as top element.

Thus we may apply Theorem 3.3 to $D(P_{n,p})$ with $s = 2$, and the number of minimal elements is almost surely at most $(1+o(1))p^{-1}$—a negligible contribution to $D(P_{n,p})$. We omit the remaining details. □

Normal convergence probably holds for all these parameters even when p is much smaller that $1/\log n$, and it may well be possible to prove much stronger versions of the above theorems by making use of further knowledge of the particular parameters. In particular, any good bounds on $\sigma(p)$ for the various parameters of study would automatically yield an improvement in our results.

We would like to finish by reiterating the qualitative interpretation of our results, which we feel are more important than any of the quantitative statements we prove. Essentially we have shown that if $p\log n \to \infty$ (or even if $p\log n$ is sufficiently large), then the structure of a random graph order is that of the linear sum of many smaller orders, and so additive parameters such as the height have an asymptotically normal distribution.

REFERENCES

[1] M. Albert and A. Frieze, *Random graph orders*, Order, 6 (1989), pp. 19–30.

[2] N. Alon, B. Bollobás, G. Brightwell, and S. Janson, *Linear extensions of a random partial order*, Ann. Appl. Probab., 4 (1994), pp. 108–123.

[3] R. Arratia, L. Goldstein, and L. Gordon, *Two moments suffice for Poisson approximation: The Stein–Chen method*, Ann. Probab., 17 (1989), pp. 9–25.

[4] K. Azuma, *Weighted sums of certain dependent random variables*, Tôhoku Math. J., 19 (1967), pp. 357–367.

[5] A. Barak and P. Erdős, *On the maximal number of strongly independent vertices in a random acyclic directed graph*, SIAM J. Algebraic Discrete Meth., 5 (1984), pp. 508–514.

[6] A. D. Barbour and G. K. Eagleson, *Poisson approximation for some statistics based on exchangeable trials*, Adv. Appl. Probab., 15 (1982), pp. 585–600.

[7] A. C. Berry, *The accuracy of Gaussian approximation to the sum of independent variables*, Trans. Amer. Math. Soc., 49 (1941), pp. 122–136.

[8] B. Bollobás, *Martingales, isoperimetric inequalities and random graphs*, in Combinatorics, A. Hajnal, L. Lovász, and V. T. Sós, eds., Colloq. Math. Sci. Janos Bolyai 52, North–Holland, 1988, pp. 113–139.

[9] B. Bollobás, *Sharp concentration of measure phenomena in random graphs*, in Random Graphs '87, M. Karonski, J. Jaworski, and A. Rucinski, eds., John Wiley, New York, 1990, pp. 1–15.

[10] B. Bollobás, *Combinatorics*, Cambridge University Press, London, 1986.

[11] B. Bollobás and G. Brightwell, *The width of random graph orders*, The Mathematical Scientist, 20 (1995), pp. 69–90.

[12] B. Bollobás and G. Brightwell, *The dimension of random graph orders*, in The Mathematics of Paul Erdős II, R. L. Graham and J. Nešetřil, eds., Springer-Verlag, 1996, pp. 51–69.

[13] G. Brightwell, *Linear extensions of random orders*, Discrete Math., 125 (1994), pp. 87–96.

[14] G. Brightwell, *Models of random partial orders*, in Surveys in Combinatorics 1993, Invited papers at the 14th British Combinatorial Conference, K. Walker, ed., Cambridge University Press, London, 1993, pp. 53–83.

[15] L. H. Y. Chen, *Poisson approximation for dependent trials*, Ann. Probab., 3 (1975), pp. 534–535.

[16] G. R. Grimmett and D. R. Stirzaker, *Probability and Random Processes*, Oxford University Press, London, 1982.

[17] M. Hall, *Combinatorial Theory*, 2nd ed., Wiley-Interscience Series in Discrete Mathematics, 1986.

[18] G. H. Hardy and S. Ramanujan, *Asymptotic formulae in combinatorial analysis*, Proc. London Math. Soc. (2), 17 (1918), pp. 75–115.

[19] D. J. Kleitman, *Families of non-disjoint subsets*, J. Combinatorial Theory, 1 (1966), pp. 153–155.

[20] C. J. H. McDiarmid, *On the method of bounded differences*, in Surveys in Combinatorics 1989, Invited papers at the 12th British Combinatorial Conference, J. Siemons, ed., Cambridge University Press, London, 1989, pp. 148–188.

[21] C. M. Newman, *Chain lengths in certain random directed graphs*, Random Structures Algorithms, 3 (1992), pp. 243–253.

[22] V. V. Petrov, *Sums of Independent Random Variables*, Ergebnisse der Mathematik und ihrer Grenzgebiete, Band 82, Springer, 1975.

[23] K. Simon, D. Crippa, and F. Collenberg, *On the distribution of the transitive closure in a random acyclic digraph*, Lecture Notes in Comput. Sci., 726 (1993), pp. 345–356.

SIAM J. Discrete Math.
Vol. 10, No. 3, pp. 337–358, August 1997

BOUNDING THE SIZE OF PLANAR INTERTWINES*

ARVIND GUPTA[†] AND RUSSELL IMPAGLIAZZO[‡]

Abstract. The proof of Wagner's conjecture by Robertson and Seymour gives a finite description of any family of graphs which is closed under the minor ordering. This description is a finite set of minimal graphs not in the family; these graphs are called the obstructions of the family. Since the intersection and union of two minor closed graph families is again a minor closed graph family, an interesting question regards computing the obstructions of the new family given the obstructions for the original two families. It is easy to compute the obstructions of the intersection, but nontrivial to compute those of the union. In this paper, we show that if the original families are planar then the planar obstructions of the union are no larger than $n^{O(n^2)}$, where n is the size of the largest obstruction of the original families.

Key words. intertwines, minors, topological embedding

AMS subject classification. 05C10

PII. S0895480192239931

1. Introduction.

Robertson and Seymour's proof of Wagner's conjecture [RSa] raises some interesting computational questions. An immediate corollary of the theorem is that for every family of graphs closed under minors (called lower ideals), the set of minimal graphs outside the family is finite. This set of graphs is called the obstruction set of the family. Robertson and Seymour also show that for every fixed graph G there is a polynomial-time algorithm that checks if a graph H contains G as a minor [RS95]. Therefore, every lower ideal has a polynomial-time membership test.

It would thus seem that the Robertson and Seymour proof should give powerful techniques for devising graph-theoretic algorithms. However, the nonconstructive nature of some parts of this work make it, in general, difficult to apply to some problems. For example, consider the problem of deciding whether a graph is embeddable in 3-space without a knot. It is easy to see that this class of graphs is closed under minors and therefore can be recognized in polynomial time. However, there is no specific polynomial-time algorithm or, in fact, even a recursive algorithm known for this problem.

Some research has centered on devising constructive proofs of certain aspects of the Robertson and Seymour work. A general format for such work is to start with a specific lower ideal or a class of lower ideals and, for these ideals, either compute their obstruction sets or find an alternate polynomial-time algorithm. Bodendiek and Wagner [BW89] gave upper bounds on the size of obstruction sets for fixed genus graphs. In [DR91], Djidjev and Reif gave improved bounds for the same problem.

[*] Received by the editors September 29, 1992; accepted for publication (in revised form) April 9, 1996.

http://www.siam.org/journals/www.siam.org/journals/sidma/10-3/23993.html

[†] School of Computing Sciences, Simon Fraser University, Burnaby, Canada V5A 1S6 (arvind@cs.sfu.ca). This research was partially performed while the author was a NSERC postdoctoral fellow in the Department of Combinatorics and Optimization, University of Waterloo. It was supported by the Natural Sciences and Engineering Council of Canada and the Advanced Systems Institute.

[‡] Department of Computer Science, University of California, San Diego, La Jolla, CA 92093 (rimpagliazzo@ucsd.edu). This research was partially performed while the author was a postdoctoral fellow in the Department of Computer Science, University of Toronto. It was supported by the Natural Sciences and Engineering Council of Canada.

In [RST95], Robertson, Seymour, and Thomas found the obstruction set for the problem of linkless embeddings of graphs in three-dimensional space. Like knotless embeddings, no algorithm for this problem was previously known.

Fellows and Langston have extensively studied the more general problem of determining when obstruction sets can be computed. In [FL94], they developed algorithms that use structural bounds based on knowledge about certain structures in the obstruction set and more general algorithms that rely on the existence of a polynomial-time self-reduction. Here the obstruction is constructed incrementally as needed—it is not possible to know whether the obstruction set has been entirely computed. These techniques seem widely applicable with few known natural problems to which they do not apply (knotlessness being one of the few such examples).

In [FL89], Fellows and Langston showed that it is possible to compute obstruction sets when the following three conditions are satisfied:

1. A bound on the treewidth of the obstructions is known.
2. A membership algorithm for the lower ideal is given.
3. A finite congruence for the lower ideal is known.

Since it is often the case that the first two conditions are satisfied, the difficulty often lies in constructing the congruence. Moreover, the algorithms generated by this approach are only known to halt—there is no bound known on their running time.

In this paper we are interested in operations under which lower ideals are closed. In particular, lower ideals are closed under both finite unions and intersections. In light of this, a natural problem is that of determining the obstructions of the new lower ideal in terms of the obstructions of the original ideals. For intersections the situation is quite easily resolved; the new obstructions are a subset of all the original obstructions.

For unions, the problem can be reduced to the following. *Given two graphs G_1 and G_2, compute all the minimal graphs under the minor ordering containing both G_1 and G_2 as minors.* These minimal graphs are called the *intertwines* of G_1 and G_2. Now every obstruction of the union is an intertwine of some pair of obstructions from the original families. In this paper we are interested in the intertwines of planar graphs.

A problem closely related to computing intertwines under the minor ordering is that of computing the topological embedding intertwines of two graphs. That is, given two graphs G_1 and G_2, what are the minimal graphs under the topological embedding relation that contain both G_1 and G_2 as topological embeddings? An upper bound on the size of the largest topological embedding intertwine gives an upper bound on the minor intertwines. The topological embedding intertwines have a number of properties that make their study simpler.

In 1976, Lovàsz conjectured that the number of topological embedding intertwines is finite for any pair of graphs. In 1978, Ungar [Ung78] independently also made the same conjecture and went on to compute the intertwines for a few specific examples. Robertson and Seymour's work directly implies that the number of minor intertwines for any two graphs must be finite. They also proved that the number of topological embedding intertwines is also finite [RSa].

Until recently, no recursive bounds on the sizes of either minor or topological embedding intertwines were known, even for very restricted cases. Fellows and Langston [FL89] described a congruence that can be used for this problem when a bound is known on the treewidth of the obstructions. This allows, for example, the intertwines of two planar graphs to be computed by a recursive procedure since the treewidth of the intertwines is bounded by a quadratic function of the size of the underlying obstructions.

Seymour and Thomas [ST91] gave a general bound on the sizes of topological embedding intertwines using techniques developed in the Robertson–Seymour work. In general, their bound is an iterated tower of 2's whose height is an iterated tower of 2's of height n with n the size of the two graphs. Moreover, they gave an upper bound that is doubly exponential in t and n for topological embedding intertwines having treewidth $\leq t$. This, together with other results of Robertson, Seymour, and Thomas, yields a bound of $2^{2^{2^{2^{poly(n)}}}}$ for the case of planar graphs.

Recently, Lagergren [La94] improved these bounds to being triply exponential in n^5, where n is the size of the planar graphs. He also gave bounds on the size of intertwines for trees and graphs of bounded pathwidth.

In this paper we obtain a bound of $n^{O(n^2)}$ for the size of the planar intertwines of two planar graphs of size n. This directly gives a doubly exponential-time algorithm for finding the planar obstructions of the union of two lower ideals of graphs. Unlike most of the other results mentioned above, our results do not draw upon the Robertson and Seymour work, but rather rely on studying properties of the planar embeddings of intertwines. Recently we also obtained a triply exponential bound on the size of the planar topological embedding intertwine of planar graphs [GI].

The outline of the paper is as follows. In the next section, we introduce basic definitions and results about graph minors. In section 3, we give a number of technical results which will be useful in the main theorem. Section 4 contains the main result. Finally, in section 5, we present open problems.

2. Preliminaries. We refer the reader to Bondy and Murty [BM76] for background material on graph theory. In this paper, we will deal only with undirected simple graphs; that is, we do not allow multiple edges or self-loops. Our results can easily be extended to nonsimple graphs. For a graph G, $V(G)$ and $E(G)$ will denote its vertex and edge set, respectively. For $a, b \in \mathbb{N}$, we will denote by $[a, b]$ the set $\{a, a+1, \ldots, b\}$.

We will mainly be concerned with the minor relation on graphs. A graph G is a *minor* of a graph H if by performing a sequence of vertex deletions, edge deletions, and edge contractions on H we obtain a graph isomorphic to G. We will use the following characterization of the minor relation.

LEMMA 2.1. *Let G and H be graphs. Then G is a minor of H if and only if there is an injective function $\mu : V(G) \to \{$subgraphs of $H\}$ such that*

1. *for every $v \in V(G)$, $\mu(v)$ is a connected nonnull subgraph of H,*
2. *for $v, w \in V(G)$, if $v \neq w$ then $\mu(v) \bigcap \mu(w) = \varnothing$, and*
3. *for each $e \in E(G)$, $e = \{v, w\}$, there is an $e' = \{x, y\} \in E(H)$ such that $x \in \mu(v)$ and $y \in \mu(w)$.*

We will call μ the *minor embedding* of G into H.

DEFINITION. A family of graphs $\mathcal{L}$ is a *lower ideal* if whenever a graph $H \in \mathcal{L}$ and $G \leq_m H$ then $G \in \mathcal{L}$. The *obstruction set* $\mathcal{O}$ of a lower ideal $\mathcal{L}$ is the minimal set of graphs (with respect to the minor ordering) not in $\mathcal{L}$. Then a graph $H \notin \mathcal{L}$ if and only if for some $G \in \mathcal{O}$, $G \leq_m H$.

We can characterize Robertson and Seymour's result on graph minors in terms of obstruction sets of lower ideals.

THEOREM 2.2 (Robertson–Seymour). *The obstruction set of every lower ideal is finite.*

Related to the minor relation is the topological embedding relation on graphs. Let us call an edge with one endpoint having degree at most 2 a 2-edge. A graph G is *topologically embedded* in a graph H, $G \leq_e H$, if by performing a sequence of vertex deletions, edge deletions, and 2-edge contractions on H we obtain a graph isomorphic to G. We use the following characterization of this relation.

Lemma 2.3. *Let G and H be graphs. Then G is topologically embedded in H if and only if there is a pair of injective functions (τ, τ') such that the following holds.*

1. *$\tau : V(G) \to V(H)$. We call $\tau(V(G))$ the* terminals *of G.*
2. *$\tau' : E(G) \to \{$simple paths in $H\}$.*
3. *If $e = \{x, y\} \in E(G)$ then $\tau'(e)$ has endpoints $\tau(x)$ and $\tau(y)$.*
4. *For $e, e' \in E(G)$, $e \neq e'$, $\tau'(e)$ and $\tau'(e')$ are internally vertex disjoint.*

We will call the pair (τ, τ') the *topological embedding* of G into H. Notice that the function τ is implicit from τ'; we will often only specify the edge mapping and speak of a function τ' as being a topological embedding of a graph G into a graph H.

Clearly if $G \leq_e H$ then $G \leq_m H$. The converse holds in general only if G is trivalent. The following well-known relationship between minors and topological embeddings is central to our results; we sketch the proof here.

Lemma 2.4. *For every graph G there is a finite family of graphs $G_1, G_2, \ldots, G_k$, $G \leq_m G_i$ for every i, such that for any graph H, $G \leq_m H$ if and only if for some i, $G_i \leq_e H$. Furthermore, every G_i has the same genus as G and $|V(G_i)| \leq O(|E(G)|)$.*

Proof (sketch). Suppose G and H are graphs such that $G \leq_m H$, and let μ be the minor embedding of G into H which, for every $v \in V(G)$, minimizes the size of $\mu(v)$. Then, for every $v \in V(G)$, $\mu(v)$ is a tree with leaves at most the degree of v in G. Furthermore, the number of internal vertices of $\mu(v)$ with degree at least 3 is bounded by the number of leaves of $\mu(v)$. Let $\mu'(v)$ be the tree obtained from $\mu(v)$ by contracting all 2-edges. Then the G_i we are looking for is G such that for every vertex v we substitute $\mu'(v)$ where the leaves of $\mu'(v)$ are used to construct the adjacencies of v. Now each of the G_i's is obtained by substituting for each vertex v of H a tree with at most degree of v leaves and no internal degree 2 vertices. Since the total number of possible trees for each degree is bounded, the result follows. The genus and size conditions are easy to verify. □

We will refer to $G_1, \ldots, G_k$ in Lemma 2.4 as the *expansions* of G.

2.1. Unions of lower ideals.

Clearly, lower ideals are closed under finite unions and intersections. Given the obstruction set of two lower ideals, one can ask for the obstruction set of their intersection or union. For intersection, the situation is straightforward.

Lemma 2.5. *Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be lower ideals with obstruction sets $\mathcal{O}_1$ and $\mathcal{O}_2$, respectively. Then the obstruction set of $\mathcal{L}_1 \bigcap \mathcal{L}_2$ is the set of minimal graphs in $\mathcal{O}_1 \bigcup \mathcal{O}_2$.*

The case for union is more complicated.

Definition. For graphs G_1 and G_2 the *intertwine set* of G_1 and G_2, $\mathcal{I}(G_1, G_2)$, is

$$\{G : G_1, G_2 \leq_m G \text{ and for every } H \leq_m G, H \neq G, \\ \text{either } G_1 \not\leq_m H \text{ or } G_2 \not\leq_m H\}.$$

Note that $\mathcal{I}(G_1, G_2)$ can, in general, have many elements; it is not difficult to construct examples where $|\mathcal{I}(G_1, G_2)| \geq 2^{O(|E(G_1)|+|E(G_2)|)}$.

LEMMA 2.6. *Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be lower ideals with obstruction sets $\mathcal{O}_1$ and $\mathcal{O}_2$, respectively. Then the obstruction set of $\mathcal{L}_1 \bigcup \mathcal{L}_2$ is a subset of $\bigcup\{\mathcal{I}(G_1, G_2) : G_1 \in \mathcal{O}_1, G_2 \in \mathcal{O}_2\}$.*

Proof. Let H be a graph, $H \notin \mathcal{L}_1 \bigcup \mathcal{L}_2$. Then $H \notin \mathcal{L}_1$ and $H \notin \mathcal{L}_2$; therefore, there is a $G_1 \in \mathcal{O}_1$ and a $G_2 \in \mathcal{O}_2$ such that $G_1, G_2 \leq_m H$. But then, by the definition of $\mathcal{I}(G_1, G_2)$, there is an $H' \in \mathcal{I}(G_1, G_2)$ such that $H' \leq_m H$. □

Notice that if $\{G_1\}$ and $\{G_2\}$ are the obstruction sets of $\mathcal{L}_1$ and $\mathcal{L}_2$, respectively, then $\mathcal{I}(G_1, G_2)$ is exactly the obstruction set of $\mathcal{L}_1 \bigcup \mathcal{L}_2$.

We can similarly define the *topological embedding intertwine* of two graphs $\mathcal{I}_e(G_1, G_2)$ as the set of minimal graphs (under topological embedding) which contains G_1 and G_2 as topological embeddings. Using Lemma 2.4 we obtain the following.

LEMMA 2.7. *Let G_1 and G_2 be graphs and $H \in \mathcal{I}(G_1, G_2)$. Then there are graphs G_1', G_2', expansions of G_1 and G_2, respectively, such that $H \in \mathcal{I}_e(G_1', G_2')$.*

Let G_1, G_2, G_1', G_2', H be as in Lemma 2.7. If (τ_1, τ_1') is the embedding of G_1' in H and (τ_2, τ_2') is the embedding of G_2' in H, then the *terminals* of H are $\tau_1(V(G_1')) \bigcup \tau_2(V(G_2'))$, that is, the terminals of G_1' plus the terminals of G_2'. The *nonterminals* of H are all vertices which are not terminals.

2.2. Grids and planar graphs. We denote a $k \times \ell$ grid by $\mathcal{G}_{k,\ell}$. We can label the vertices of $\mathcal{G}_{k,\ell}$ by ordered pairs $\mathcal{G}(i, j)$, where $1 \leq i \leq k$ and $1 \leq j \leq \ell$, and by (i, j) when the grid graph is understood.

It is not difficult to see that every planar graph is a minor of a sufficiently large grid; however, our bounds rely on making this grid as small as possible. We next show that for a graph on n nodes an $O(n) \times O(n)$ grid suffices. A similar result is inherent in the work of Robertson and Seymour in which they show that planar graphs are well quasi-ordered under minors [RS84].

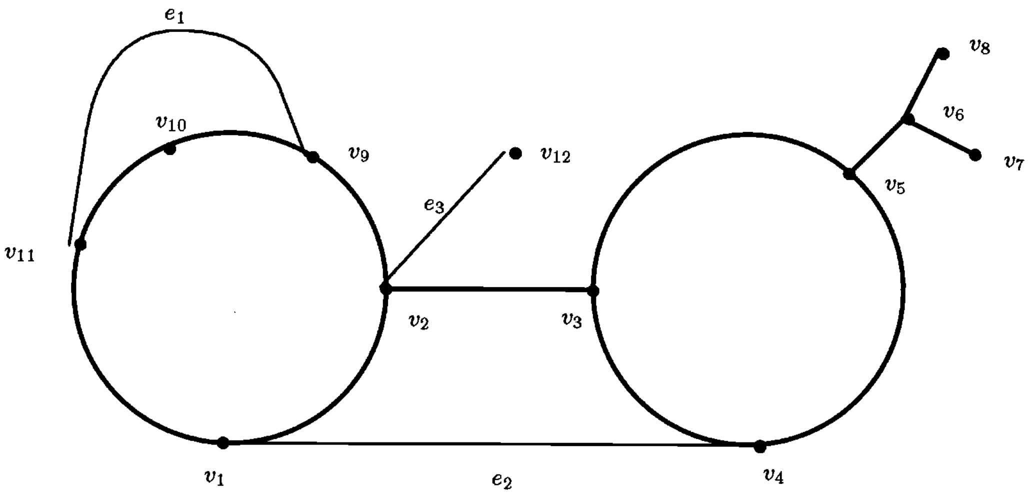
THEOREM 2.8. *For every planar graph G such that $|V(G)| = n$, $G \leq_m \mathcal{G}_{3n,2n}$.*

Proof. Without loss of generality, we can assume that the graph is connected. A planar graph (with a corresponding planar embedding) can be inductively built by adding, at each step, either an edge between two vertices on the outside face or an edge from a vertex on the outside face to a new vertex, where in either case the new edge is constrained to lie on the outside face of the resulting embedding.

Given a planar embedding, there is a unique closed walk along the outside face (in, say, the counter clockwise direction) (see Figure 1). Notice that since the graph is not necessarily two connected, some vertices can be encountered more than once on this walk. For example, for a tree, this walk corresponds to a listing of the vertices in a depth-first traversal of the tree.

Let $v_1, v_2, \ldots, v_k$ be the closed walk on the outside face of an n node planar graph with m edges, $m \leq 3n - 6$, where some of the v_i's might be the same vertex and $v_1 = v_k$. The induction hypothesis is that there is a minor embedding μ of G into a grid $\mathcal{G}_{m,2n}$ such that for each i, $1 \leq i \leq k$, (m, ℓ_i) is a vertex of $\mu(v_i)$ for some $1 \leq \ell_i \leq m$ and $\ell_1 < \ell_2 < \cdots < \ell_k$. Furthermore, for every vertex v of G, $\mu(v)$ does not contain any edge in row m of the grid $\mathcal{G}_{m,2n}$ (i.e., any edge of the form (m, j)).

We consider two cases. First suppose that G' is formed from G by adding an edge between two vertices x and y on the outside face of G (see edges e_1 and e_2 in Figure 1). Then, for the resulting walk $w_1, \ldots, w_{k'}$ on the outside face of G', there is an r, $1 \leq r \leq k'$ such that $w_1, \ldots, w_r$ is exactly the same as $v_1, \ldots, v_r$ and $w_{r+1}, \ldots, w_{k'}$ is exactly the same as $v_{k-k'+r+1}, \ldots, v_k$ with $w_r = x$ and $w_{r+1} = y$. Let μ be a minor embedding of G into an $m \times 2n$ grid $\mathcal{G}'_{m,2n}$, as in the induction hypothesis. We define μ' to be a minor embedding of G' into an $(m + 1) \times 2n$ grid $\mathcal{G}'_{m+1,2n}$ as follows. For



Walk on original graph (dark lines): $v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_6, v_8, v_6, v_5, v_3, v_2, v_9, v_{10}, v_{11}, v_1$

After addition of edge e_1: $v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_6, v_8, v_6, v_5, v_3, v_2, v_9, v_{11}, v_1$

After addition of edges e_1, e_2: $v_1, v_4, v_5, v_6, v_7, v_6, v_8, v_6, v_5, v_3, v_2, v_9, v_{11}, v_1$

After addition of edges e_1, e_2, e_3: $v_1, v_4, v_5, v_6, v_7, v_6, v_8, v_6, v_5, v_3, v_2, v_{12}, v_2, v_9, v_{11}, v_1$

FIG. 1. *A planar embedding with walks on the original graph (dark lines) and after addition of some new edges.*

vertices u of G' other than those in $w_1, \ldots, w_{k'}$, $\mu'(u) = \mu(u)$. For $1 \leq i \leq r-1$, $\mu'(w_i)$ consists of $\mu(v_i)$ plus the edge from (m, ℓ_i) to $(m+1, \ell_i)$. Similarly, for $r+1 \leq i \leq k'$, $\mu'(w_i)$ consists of $\mu(v_{k-k'+i})$ plus the edge from $(m, \ell_{k-k'+i})$ to $(m+1, \ell_{k-k'+i})$. Finally, $\mu'(w_r)$ consists of $\mu(v_r)$ plus the edge from (m, ℓ_r) to $(m+1, \ell_r)$ plus the path from (m, ℓ_r) to $(m, \ell_{k-k'+r+1} - 1)$. It is straightforward to verify that the induction hypothesis is satisfied.

Now suppose that G' is formed from G by adding an edge from some vertex x on the outside face of G to a new vertex y (see edge e_3 in Figure 1). Let us assume that x is v_1; the case where it is not is similar but more technical. Now the walk along the outside face of G' is $v_1, \ldots, v_k, v_{k+1}, v_{k+2}$, where $v_1 = v_k = v_{k+2} = x$ and $v_{k+1} = y$. Let μ be a minor embedding of G into an $m \times 2n$ grid $\mathcal{G}'_{m,2n}$, as in the induction hypothesis. Then we define μ' to be a minor embedding of G' into an $(m+1) \times (2n+2)$ grid $\mathcal{G}'_{m+1,2n+2}$. For vertices u of G other than the v_i, $\mu'(u) = \mu(u)$. For $1 \leq i \leq k$ and $v_i \neq x$, $\mu'(v_i)$ consists of $\mu(v_i)$ plus the edge from (m, ℓ_i) to $(m+1, \ell_i)$. Furthermore, $\mu'(y)$ consists of the single vertex $(m+1, 2n-1)$ and $\mu'(x)$ consists of the following pieces:

1. $\mu(x)$,
2. the edges from (m, j) to $(m+1, j)$, where $v_j = x$, and
3. the path from (m, ℓ_k) to $(m, 2n+2)$ and the edge from $(m, 2n+2)$ to $(m+1, 2n+2)$.

Again, it is straightforward to verify that the induction hypothesis is satisfied. □

Let G be a planar graph with some fixed embedding on the plane. Let C be a simple circuit of G. Then the plane with C removed divides into two regions—an infinite one and a finite one. We call the finite region the disc induced by C and denote it $\Delta(C)$. $\Delta(C)$ will not contain C itself.

3. Technical results.

3.1. Basic structure of intertwine graphs. Let G_1 and G_2 be graphs where $|E(G_1)| + |E(G_2)| = m$. Consider a minor intertwine H of G_1 and G_2 and let G'_1 and G'_2 be the topological expansions of G_1 and G_2 given in Lemma 2.7. Throughout this section assume that G_1, G_2, G'_1, G'_2, and H are all fixed and that for $i = 1, 2$, (τ_i, τ'_i) is the topological embedding of G'_i in H. We begin by defining a labeling on $E(H)$ and $V(H)$.

LEMMA 3.1. *Let $e \in E(H)$ such that neither endpoint is a terminal. Then there is exactly one $e' \in E(G'_1) \bigcup E(G'_2)$ such that e is in the image of e'.*

Proof. If there is no edge e' whose image contains e, then we can delete e from H, contradicting the minimality of H. Furthermore, clearly there cannot be two edges of $E(G'_1)$ (respectively, $E(G'_2)$) whose image contains e since τ'_1 (respectively, τ'_2) maps edges of G'_1 (G'_2) to internally vertex disjoint paths. Suppose $e'_1 \in E(G'_1)$ and $e'_2 \in E(G'_2)$ both contain e in their image. But then H with e contracted contains both G_1 and G_2 as minors, again contradicting the minimality of H. □

For e, e' as in Lemma 3.1, we denote e' by $l(e)$.

LEMMA 3.2. *Every nonterminal of H is in the image of exactly one edge of G'_1 and one edge of G'_2.*

Proof. Let v be a nonterminal of H. First, v can be in the image of at most one edge of G'_1 and one edge of G'_2 since the images of edges are internally vertex disjoint paths. If v is not in the image of an edge of G'_1 or G'_2 we can delete v, thereby reducing the size of H. If v is in the image of an edge of G'_1 but not G'_2 (or an edge of G'_2 but not G'_1) then v has degree 2 and we can contract one of the edges through v, again reducing the size of H. □

If v is a nonterminal, let $e_1 \in E(G'_1)$ and $e_2 \in E(G'_2)$ such that v is in the image of e_1 and e_2, as in Lemma 3.2. Then we write $l_1(v) = e_1$ and $l_2(v) = e_2$. We note the following corollary of Lemma 3.1 and Lemma 3.2.

COROLLARY 3.3. *Let v be a nonterminal of H. Then v has degree* 4.

3.2. Reroutings. We will frequently wish to show that an intertwine graph cannot contain certain structures as subgraphs. To do this, we will demonstrate a method of using the structure to reroute paths from the embedding of one graph along edges currently used only for the embedding of the other graph. If this rerouting of paths from the first graph is proper (i.e., does not use an edge used in the original embedding), we can omit that edge and still contain both graphs as embeddings. Thus, we get a contradiction to the minimality of the intertwine, so no intertwine can contain the structure. Formally, we have the following definition.

DEFINITION. Let H be a graph, and let $P_1, \dots, P_k$ be simple vertex disjoint paths in H. Let s_i and t_i be the endpoints of P_i. A *rerouting* of $P_1, \dots, P_k$ is a sequence of vertex disjoint simple paths of H, $R_1, \dots, R_k$ satisfying the following:

1. R_i has endpoints s_i and t_i,
2. every vertex x in R_i lies in P_j for some $1 \leq j \leq k$,
3. there is at least one edge $f \in \cup_{1 \leq i \leq k} P_i - \cup_{1 \leq i \leq k} R_i$.

DEFINITION. Let H be a graph containing G as a topological embedding via map τ. Let $P_1, \dots, P_k$ be simple vertex disjoint paths in H. We say that $P_1, \dots, P_k$ are *monochromatic* for τ if for every $1 \leq i \leq k$ there is an $e_i \in E(G)$ such that P_i is a subpath of $\tau(e_i)$.

The next lemma shows that reroutings are not possible in a topological embedding intertwine.

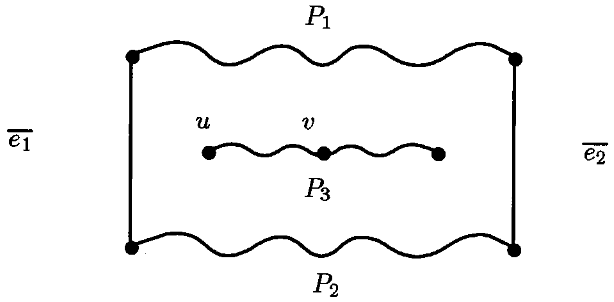


FIG. 2. *Regions of this form contain a vertex of H if and only if they contain a terminal vertex.*

LEMMA 3.4. *Let H be a topological embedding intertwine of G_1 and G_2. Let τ_1 and τ_2 be the corresponding embeddings, and let $P_1, \ldots, P_k$ be monochromatic paths for τ_1. Then H does not contain a rerouting of $P_1, \ldots, P_k$.*

Proof. Let $P_1, \ldots, P_k$ be monochromatic paths for τ_1 with s_i and t_i, the endpoints of P_i. Let $e_1, \ldots, e_k$ be edges of G_1 such that P_i is a subpath of $\tau_1(e_i)$, $1 \leq i \leq k$. Suppose $R_1, \ldots, R_k$ is a rerouting in H of $P_1, \ldots, P_k$. Let $f \in \cup_{1\leq i\leq k} P_i - \cup_{1\leq i\leq k} R_i$. We claim that $H' = H - \{f\}$ also contains G_1 and G_2 as embeddings. This is a contradiction since H properly contains H' as a topological embedding.

Since $f \in P_{i_0} \subseteq \tau_1(e_{i_0})$ for some i_0, $l(f) = e_{i_0} \in E(G_1)$, so H' contains G_2 as an embedding via the same embedding τ_2. Let τ be the following embedding of G_1 into H. For nodes and edges of G_1 other than $e_1, \ldots, e_k$, τ is the same as τ_1. $\tau(e_i)$ is $\tau_1(e_i)$ with all subpaths P_i replaced by R_i. (Note that we do not assume the e_i's are distinct in the definition of monochromatic, so in addition to P_i there may be some other $P_{i'}$ which is replaced by $R_{i'}$.)

To see that τ is an embedding, first note that each R_i and each $g \in E(G_1)$ shares no vertex other than s_i and t_i with any of the subpaths we get from $\tau_1(g)$ by deleting all P_j with $g = e_j$. This is because each vertex in R_i is in P_j for exactly one j, and if $e_j \neq g$, P_j is vertex disjoint from $\tau_1(g)$. Then, since the R_i are vertex disjoint, it follows that the $\tau(g)$'s are vertex disjoint for $g \in E(G_1)$. Using a similar argument, we can also show that the $\tau(e_i)$'s are simple. □

As an immediate consequence, we obtain the following.

LEMMA 3.5. *Let H be a topological embedding intertwine of G_1 and G_2. Let $e \in E(H)$ and $l(e) \in E(G_1)$, and let x and y be the endpoints of e, with neither a terminal. Then $l_2(x) \neq l_2(y)$.*

Proof. Assume $l_2(x) = l_2(y) = f \in E(G_2)$, and let P be the subpath of $\tau_2(f)$ connecting x to y. Then e is a rerouting of P. □

3.3. Circuits in planar intertwine graphs. From this point on, we assume that G_1, G_2, G_1', G_2', and H are all planar and that we have fixed some planar drawing of H. This induces a fixed planar drawing of G_1' and G_2'. We begin by studying circuits of H.

LEMMA 3.6 (refer to Figure 2). *Let $e_1, e_2 \in E(G_1')$ and $f_1, f_2 \in E(G_2')$. Suppose there is an edge $\overline{e_1}$ of $\tau_1'(e_1)$, an edge $\overline{e_2}$ of $\tau_1'(e_2)$, a subpath P_1 of $\tau_2'(f_1)$, and a subpath P_2 of $\tau_2'(f_2)$ such that $P_1, \overline{e_1}, P_2, \overline{e_2}$ forms a simple cycle C of H. If any vertex of H is in $\Delta(C)$ then there is a terminal of H in $\Delta(C)$.*

Proof. If $f_1 = f_2$, H is not minimal since we can reroute along $\overline{e_1}$. Suppose v is a nonterminal in $\Delta(C)$ such that no terminal of H is in $\Delta(C)$, and suppose $l_2(v) = g_2$.

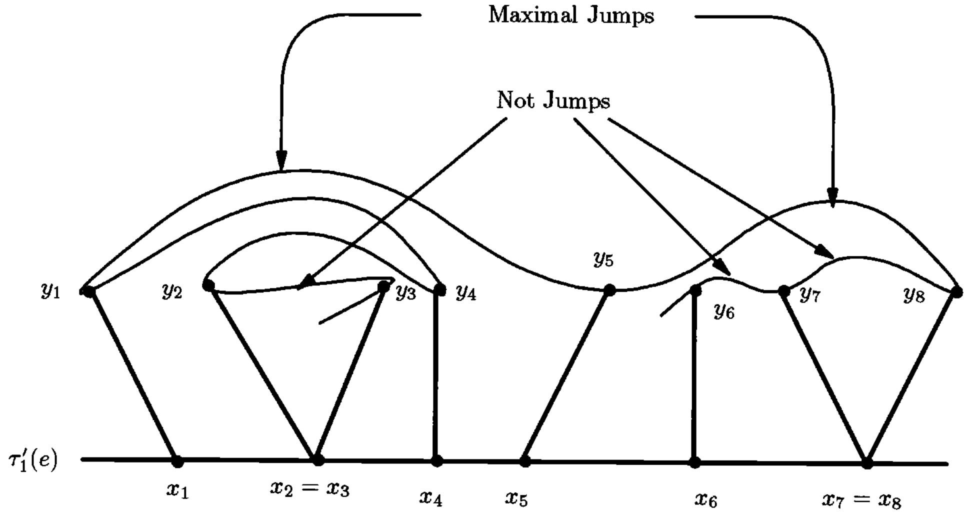


FIG. 3. *Jumps.*

Since H is planar and the endpoints of $\tau_2'(g_2)$ are not in $\Delta(C)$, the path $\tau_2'(g_2)$ must cross C (at least two times). If $g_2 \neq f_1, f_2$ then by Lemma 3.2 $\tau_2'(g_2)$ cannot cross C. If $g_2 = f_1$ or $g_2 = f_2$ then the path $\tau_2'(g_2)$ is not simple. □

3.4. Jumps. Here we consider edges of G_i' $(i = 1, 2)$ whose images are long paths in H. For the remainder of this section let $e \in E(G_1')$ and $f_1, \ldots, f_k \in E(H)$ such that

1. each f_i has endpoints x_i and y_i where the x_i are on $\tau_1'(e)$ (but not necessarily the y_i),
2. $x_1, x_2, \ldots, x_k$ occur in that order on $\tau_1'(e)$ (i.e., ordered by distance from one endpoint),
3. the y_i's are all distinct, and
4. if z and z' are the endpoints of $\tau_1'(e)$, then by viewing $\tau_1'(e)$ as a directed path in the plane from z to z' any edge of H meeting $\tau_1'(e)$ is in one of two orientations, which we will call "up" and "down." Then all edges f_i have the same orientation with respect to $\tau_1'(e)$.

If $e \in E(G_1')$, x is an endpoint of $\tau_1'(e)$, and u, v are two other vertices on $\tau_1'(e)$ then we say that relative to x, u is *left* (respectively, *right*) of v on $\tau_1'(e)$ if x, u, v (respectively, x, v, u) occur in that order on $\tau_1'(e)$.

DEFINITION. A *jump* on $f_1, \ldots, f_k$ is an interval $[j_1, j_2]$, $1 \leq j_1 < j_2 \leq k$, such that

1. $l_1(y_{j_1}) = l_1(y_{j_2}) = g$ for some $g \in E(G_1')$,
2. the subpath of $\tau_1'(g)$ with endpoints y_{j_1} and y_{j_2} contains no other y_i, $1 \leq i \leq k$, and
3. $j_2 > j_1 + 1$.

A jump $[j_1, j_2]$ is *maximal* if there is no other proper jump $[j_1', j_2']$ such that $[j_1, j_2] \subset [j_1', j_2']$ (see Figure 3).

With every jump $[j_1, j_2]$ we associate a disc $\Delta(j_1, j_2)$ bounded by the circuit which starts at y_{j_1}, follows $\tau_1'(g)$ to y_{j_2}, goes to x_{j_2}, follows $\tau_1'(e)$ to x_{j_1}, and returns to y_{j_1}. We make two observations about jumps and their associated discs.

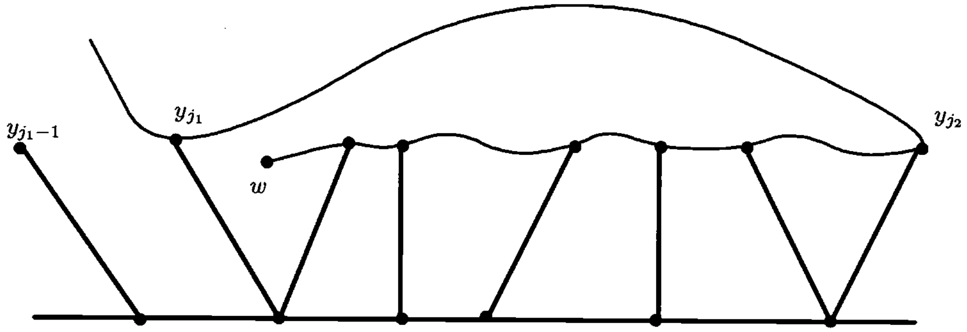


FIG. 4. *When $[j_1, j_2]$ contains no jump, the path $\tau_1'(g)$ starting at w goes through y_{j_1+1}, $y_{j_1+2}, \ldots, y_{j_2}, y_{j_1}$ in that order. Then $r = j_1 + 1$ satisfies the claim in the proof of Lemma* 3.9.

FACT 3.7. *Because of planarity, all maximal jumps must be disjoint (except possibly at their endpoints).*

FACT 3.8. *For any jump $[j_1, j_2]$, $\Delta(j_1, j_2)$ contains a terminal of H. If, in addition, $g = l_1(y_j) = l_1(y_{j_1})$ for $j_1 < j < j_2$, then $\Delta(j_1, j_2)$ contains an endpoint of $\tau_1'(g)$.*

LEMMA 3.9. *Suppose that for $1 \leq i \leq k$, $l_1(y_i) = g$ for some $g \in E(G_1')$. Then there is a j, $1 \leq j \leq k$ and $\ell \geq k/5$ such that $y_{j+1}, \ldots, y_{j+\ell}$ occur in that order on $\tau_1'(g)$.*

Proof. Suppose $[j_1, j_2]$ is a jump on $f_1, \ldots, f_k$. Since at least one endpoint of $\tau_1'(g)$ occurs in $\Delta(j_1, j_2)$ (Fact 3.8), there are at most two maximal jumps in $[1, k]$.

We first consider the structure of jumps that contain one endpoint of $\tau_1'(g)$ and those that contain two endpoints.

Let $[j_1, j_2]$ be a jump, $1 \leq j_1 < j_2 \leq k$, so that exactly one endpoint w of $\tau_1'(g)$ lies in $\Delta(j_1, j_2)$. Without loss of generality, assume that w, y_{j_2}, y_{j_1} occur in that order on $\tau_1'(g)$.

Claim. There is an r, $j_1 \leq r \leq j_2$, so that the path $\tau_1'(g)$ passes through the vertices $w, y_r, y_{r-1}, \ldots, y_{j_1}$ in that order and $\tau_1'(g)$ passes through $w, y_r, y_{r+1}, \ldots, y_{j_2}$ in that order.

Proof. The proof is by induction on $j_2 - j_1$. For the base case, $j_2 - j_1 = 2$. By choosing $r = j_1 + 1$ it is straightforward to verify that the claim is satisfied.

For the induction step, we consider two cases. If $[j_1, j_2]$ does not properly contain any jump then the path from w passes through $y_{j_1+1}, \ldots, y_{j_2}$ in that order and we let $r = j_1 + 1$ (see Figure 4).

Otherwise suppose $[j_1, j_2]$ contains at least one proper jump. Then, in the interval $[j_1, j_2]$, there is a unique maximal jump $[j_1', j_2']$. Here, uniqueness is guaranteed by Fact 3.8 since w must lie in any such maximal jump. Notice that there are no jumps in the interval $[j_2', j_2]$ since such a jump would necessitate a terminal other than w in the jump $[j_1, j_2]$. Therefore, on $\tau_1'(g)$, the vertices $w, y_{j_2'}, y_{j_2'+1}, \ldots, y_{j_2}$ occur in that order. Furthermore, $j_1' = j_1 + 1$, since otherwise the only way that $\tau_1'(g)$ could pass through, for example, y_{j_1+1} is if another endpoint of $\tau_1'(a)$ lies in $[j_1, j_2]$ but not in $[j_1', j_2']$, which is a contradiction. By induction there is an r in $[j_1', j_2']$ such that starting at w, the path $\tau_1'(g)$ passes through the vertices $w, y_r, y_{r-1}, \ldots, y_{j_1'}$ in that order and $\tau_1'(g)$ passes through $w, y_r, y_{r+1}, \ldots, y_{j_2'}$ in that order. Thus, $\tau_1'(g)$ passes through $w, y_r, \ldots, y_{j_2'}, \ldots, y_{j_2}$ in that order. Furthermore, $\tau_1'(g)$ starting at y_{j_1+1} and going to y_{j_1} can only pass through y_i for $i > r$, since otherwise it would intersect itself. Thus, $\tau_1'(a)$ starting at w passes through $y_r, y_{r-1}, \dots, y_{j_1+1}, y_{j_1}$ in that order. This proves the claim.

We now consider a jump $[j_1, j_2]$ that contains two endpoints. First suppose there are no maximal proper jumps inside $[j_1, j_2]$. Let w_1 and w_2 be the endpoints and suppose that $\tau_1'(g)$ passes through $w_1, y_{j_1}, y_{j_2}, w_2$ in that order. If y_r is the first y_i on the path from w_1 to y_{j_1}, then clearly $\tau_1'(g)$ passes through $w_1, y_r, y_{r-1}, \dots, y_{j_1}, y_{j_2}, y_{j_1-1}, \dots, y_{r+1}, w_2$ in that order.

If the jump $[j_1, j_2]$ contains maximal proper jumps then no proper maximal jump in $[j_1, j_2]$ contains both endpoints. Thus, we can apply the above claim to maximal jumps inside $[j_1, j_2]$ to yield an r_1, r_2, r_3, $j_1 \leq r_1 \leq r_2 \leq r_3 \leq j_2$, such that $y_{r_1}, y_{r_1-1}, \dots, y_{j_1}$, $y_{r_1}, y_{r_1+1}, \dots, y_{r_2}$, $y_{r_3}, y_{r_3-1}, \dots, y_{r_2}$, and $y_{r_3}, y_{r_3+1}, \dots, y_{j_2}$ each occur in that order.

Finally, we consider all y_i that are not inside any jump. Then $\tau_1'(a)$ must pass consecutively through these y_i. Thus, the interval $[1, k]$ can be partitioned into at most five subintervals such that the y_i's occur consecutively on $\tau_1'(a)$ in each piece. By choosing the largest subinterval, the result follows. □

We can generalize Lemma 3.9 to handle the case of arbitrary labels on the y_i.

Lemma 3.10. *There is a $g \in E(G_1')$ and $1 \leq i < j \leq k$ such that*

1. *$y_i, y_{i+1}, \dots, y_j$ occurs in that order on $\tau_1'(g)$,*
2. *$j - i$ is at least $\frac{k}{O(m^2)}$ (where $|E(G_1')| \leq m$), and*
3. *there are no vertices in $\Delta(\ell, \ell+1)$ for $i \leq \ell \leq j-1$.*

Proof. Since $|E(G_1')| \leq m$, there is an $\ell \geq \frac{k}{m}$ and a $g \in E(G_1')$ such that $y_{i_1}, y_{i_2}, \dots, y_{i_\ell}$ are all vertices on $\tau_1'(g)$. By Lemma 3.9, there are $j_1 < j_2 < \cdots < j_{\ell'}$, $\{j_1, \dots, j_{\ell'}\} \subseteq \{i_1, \dots, i_\ell\}$, such that $\tau_1'(g)$ passes through $y_{j_1}, \dots, y_{j_{\ell'}}$ in that order starting at one of the endpoints and $\ell' \geq \frac{k}{5m}$. We need only show that a sufficiently large subset of these j_i occur in consecutive order. Consider the discs $\Delta(j_r, j_{r+1})$ for $1 \leq r < \ell'$. By Lemma 3.6, at most $2m$ of these discs contain a vertex of H since each disc which contains a vertex of H contains a terminal vertex. Then there are r and s, $1 \leq r < s < \ell'$, such that $s - r \geq \frac{k}{5m(2m+1)}$ such that the discs $\Delta(j_r, j_r + 1)$ and $\Delta(j_s, j_s + 1)$ both contain terminals but no disc in between these contains a terminal. Then, by Fact 3.8, there cannot be any jump on $[j_{r+1}, j_s]$; therefore, $y_{j_r+1}, y_{j_r+2}, \dots, y_{j_s}$ must occur in that order along $\tau_1'(g)$. □

3.5. Intersections in planar drawings. We next look at the types of intersections two paths can make on the plane. In order to avoid overly cumbersome notation, our presentation in this section is slightly informal—it is not difficult to formalize these arguments.

For a planar graph G, suppose P_1 is any nontrivial path in G, $P_2 = v_1, v_2, v_3$ is a length 3 path in G, and P_1 and P_2 have only the vertex v_2 in common, where v_2 is not an endpoint of P_1. Let w be an endpoint of P_1 and consider a fixed planar drawing of G. Then, relative to w, there are exactly three different types of intersections between P_1 and P_2 with respect to the drawing (see Figure 5). We call these three types of intersections a $\bigvee$-intersection, a $+$-intersection, and a $\bigwedge$-intersection relative to w. Furthermore, we will call v_1 and v_3 the *ends* of the intersections relative to w, and, more specifically for a $\bigvee$-intersection, we will call v_1 the *left end* and v_3 the *right end*.

Lemma 3.11. *Let $e \in E(G_1')$ and w be an endpoint of $\tau_1'(e)$. Suppose a path $W = v_1, v_2, v_3$ in H is a $\bigvee$-intersection with $\tau_1'(e)$ relative to w and $l_1(v_1) = l_1(v_3) = f$. Consider the subpath P of $\tau_1'(f)$ between v_1 and v_3. Then, if the disc bounded by the*

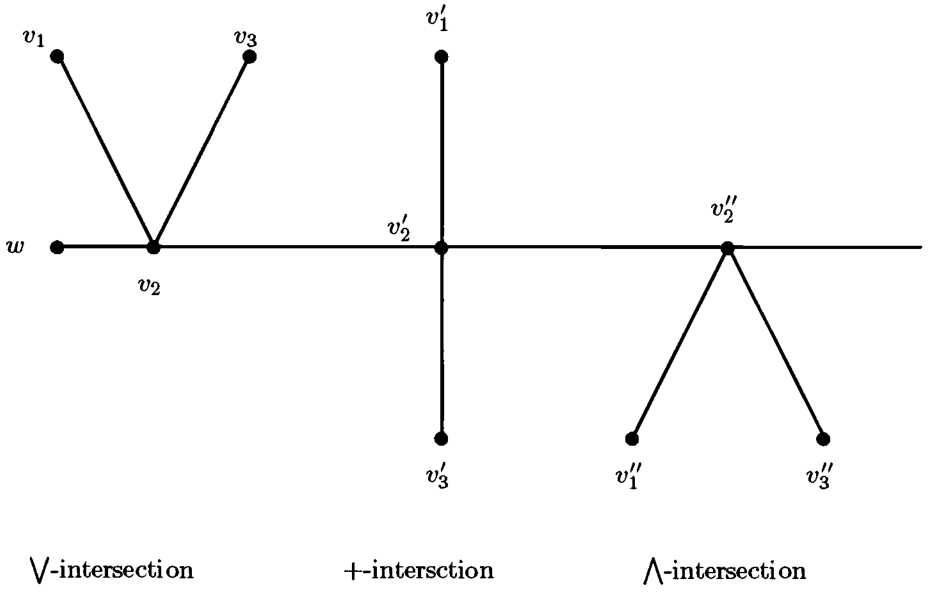


FIG. 5. *Types of intersections.*

circuit P, v_3, v_2, v_1 *contains no terminals, there is a path* $W' = w_1, u, w_2$ *in* H *such that* u *is on* P *and* W' *is a* $\bigvee$*-intersection with* P *relative to* v_1*. We call* W' *a* child of W (*see Figure* 6).

Proof. By Lemma 3.5, P is not a single edge. So $|P| > 2$ and there is a nonterminal u in the interior of P. Suppose $l_1(u) = g$, where $g \in E(G_1')$. Let $\tau_1'(g) = P_1, w_1, u, w_2, P_2$, where P_1 and P_2 are subpaths of $\tau_1'(g)$. Since by Lemma 3.6 no part of $\tau_1'(g)$ can be inside the disc bounded by the circuit P, v_3, v_2, v_1, either w_1, u, w_2 or w_2, u, w_1 is the required $\bigvee$-intersection. □

Referring to Lemma 3.11, we define a $\bigvee$-intersection P' to be a *descendant* of a $\bigvee$-intersection P if there is a sequence of $\bigvee$-intersections $P_1 = P, P_2, \ldots, P_k = P'$ such that for $1 < i \leq k$, P_i is a child of P_{i-1}.

4. Planar intertwines.

In this section we prove our following main result.

THEOREM 4.1. *For planar graphs* G_1 *and* G_2 *and* $H \in \mathcal{I}(G_1, G_2)$*, if* $|E(G_1)| + |E(G_2)| = m \geq 2$ *then* $|E(H)| \leq m^{O(m^2)}$.

This theorem will follow from Theorem 2.8 and the following lemma.

LEMMA 4.2. *For planar graphs* G_1 *and* G_2 *and* $H \in \mathcal{I}_e(G_1, G_2)$*, if* $|E(G_1)| + |E(G_2)| = m > 2$ *and* $|E(H)| > m^{O(m^2)}$ *then* $\mathcal{G}_{3m,3m} \leq_m H$.

We first give a proof of Theorem 4.1 using Lemma 4.2. The remainder of this section will then be devoted to a proof of Lemma 4.2.

Proof (Theorem 4.1). Let G_1 and G_2 be planar graphs and $H \in \mathcal{I}(G_1, G_2)$. Suppose $|E(H)| > m^{O(m^2)}$. Then there are G_1' and G_2' expansions of G_1 and G_2, respectively, such that $H \in \mathcal{I}_e(G_1', G_2')$. For $\bar{m} = |E(G_1')| + |E(G_2')|$, $\bar{m} \in O(m)$, so $|E(H)| > \bar{m}^{O(\bar{m}^2)}$. But then by Lemma 4.2, $\mathcal{G}_{3\bar{m},3\bar{m}} \leq_m H$. Since by Theorem 2.8 $G_1, G_2 \leq_m \mathcal{G}_{3m,2n}$ and $\mathcal{G}_{3m,2n}$ is a proper minor of $\mathcal{G}_{3\bar{m},3\bar{m}}$, it follows that H is not a minor minimal graph containing G_1 and G_2, which is a contradiction. □

To begin the proof of Lemma 4.2, we let G_1 and G_2 be planar graphs and H be

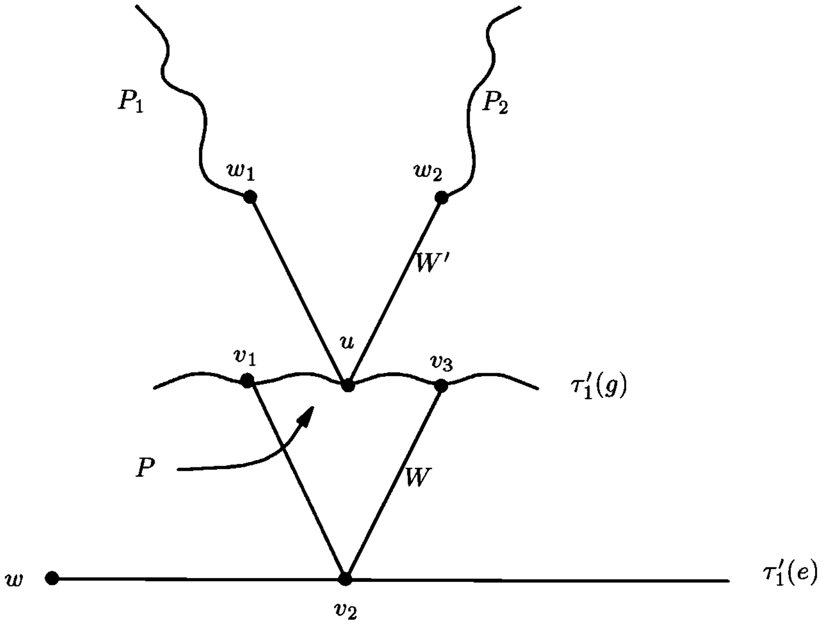


FIG. 6. *W' is a child of W.*

as in the statement of the lemma. Let (τ_1, τ_1') and (τ_2, τ_2') be, respectively, topological embeddings of G_1 and G_2 into H. We begin with an outline of the proof; formal details follow the outline.

Suppose H is large (where large is defined to mean size at least $m^{\Omega(m^2)}$). Then the image of some edge of G_1 is a long path in H. Let v be an endpoint of this path. Then every nonterminal of this path must have a $\bigvee$-, $+$-, or $\bigwedge$-intersection relative to v with a subpath of an edge of G_2. Suppose there are a large number of $\bigvee$-intersections relative to v. Then by applying Lemma 3.10 there is some edge e of G_1 such that $\tau_1'(e)$ goes through the consecutive endpoints of a large fraction of the $\bigvee$-intersections. Now, by Lemma 3.11, each $\bigvee$-intersection has a child $\bigvee$-intersection on $\tau_1'(e)$. Continuing, we construct a grid.

The only problem occurs if we encounter the same edge of G_1 more than once. In that case if we are, for the most part, using a different part of the image of that edge then we can continue constructing the grid. Otherwise, we show that the paths corresponding to the edges of G_1 that we have encountered so far can be rerouted, thus contradicting the minimality of H. The argument for the case where there are many $\bigwedge$-intersections is symmetric. If there is no edge whose image has many $\bigvee$-intersections or $\bigwedge$-intersections, then the image of some edge has many consecutive $+$-intersections. The basic idea is the same in this case with a few different technical details; these details are outlined after the case of $\bigvee$-intersection is handled.

We are now ready for a formal presentation of the proof. Assume that $|E(H)|$ is $m^{\Omega(m^2)}$. Since every nonterminal of H is labeled by some edge of G_1, there is an edge e of G_1 such that $\tau_1'(e)$ is a path of length $m^{\Omega(m^2)}$ in H. We begin by considering the case where there are $m^{\Omega(m)}$ $\bigvee$-intersections on $\tau_1'(e)$. The case of $\bigwedge$-intersections is similar, and we will not discuss it.

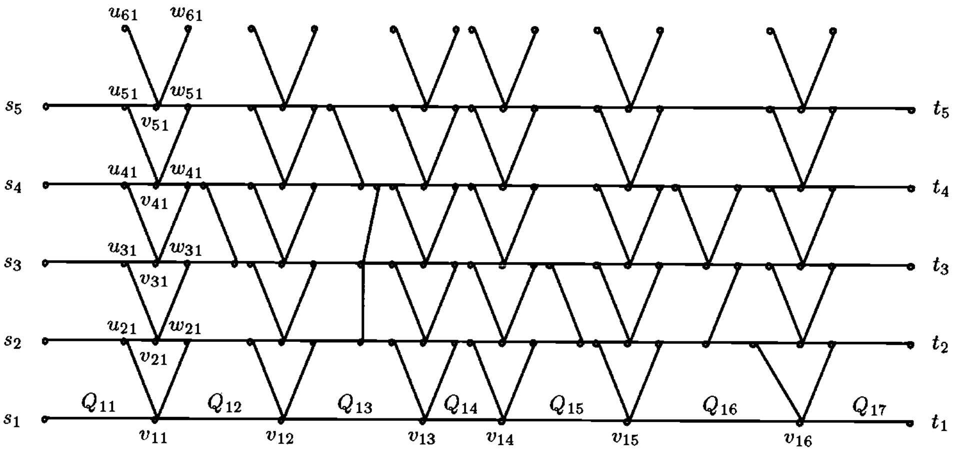


FIG. 7. *A* $\bigvee_{5,6}$*-grid.*

DEFINITION (refer to Figure 7). Let $k, \ell \in \mathbb{N}$. A planar graph G is a $k \times \ell$ $\bigvee$-grid, $\bigvee_{k,\ell}$, if the following holds: there are k vertex disjoint simple paths in G, $P_1, \ldots, P_k$, where P_i has endpoints s_i and t_i, so that

1. there are vertices $v_{i,j}$ for $(i,j) \in [1,k] \times [1,\ell]$, $u_{i,j}, w_{i,j}$ for $(i,j) \in [2,k+1] \times [1,\ell]$ which are all distinct,
2. $P_1 = s_1, Q_{1,1}, v_{1,1}, Q_{1,2}, v_{1,2}, \ldots, v_{1,\ell}, Q_{1,\ell+1}, t_1$, where $Q_{1,j}$ are arbitrary disjoint paths of G,
3. for $2 \le i \le k$, $P_i = s_i, Q_{i,1}, u_{i,1}, U_{i,1}, v_{i,1}, W_{i,1}, w_{i,1}, Q_{i,2}, \ldots, W_{i,\ell+1}, t_i$, where all $Q_{i,j}, U_{i,j}, W_{i,j}$ are arbitrary disjoint paths of G of length ≥ 1,
4. for $(i,j) \in [1,k] \times [1,\ell]$ there is an edge from $v_{i,j}$ to $u_{i+1,j}$ and $w_{i+1,j}$, and
5. there is no vertex v' on P_1, v' between $v_{1,1}$ and $v_{1,\ell}$ and distinct from the $v_{1,j}$ and s_1, t_1 such that there are two edges from v' to P_2.

DEFINITION. Suppose H is the planar intertwine of two graphs G_1 and G_2. A $\bigvee$-grid in H is *monochromatic* if for every P_i there is an edge $e_i \in E(G_1)$ such that P_i is a subpath of $\tau_1'(e_i)$.

A $\bigvee$-grid is illustrated in Figure 7. The idea behind the proof of Lemma 4.2 is to inductively find a $\bigvee_{3m,3m}$ as a subgraph of H. Since $\mathcal{G}_{3m,3m} \le_m \bigvee_{3m,3m}$, the lemma will follow. If we ever get stuck in building the $\bigvee$-grid, we show that H could not have been minimal.

Since H contains a monochromatic $1 \times m^{\Omega(m)}$ $\bigvee$-grid, Lemma 4.2 follows from the following lemma.

LEMMA 4.3. *Suppose that H contains a monochromatic $k \times \ell$ $\bigvee$-grid as a subgraph, $k \le m$. Then H contains a monochromatic $(k+1) \times \frac{\ell}{O(m^2)}$ $\bigvee$-grid as a subgraph.*

We now turn our attention to proving the above lemma. For $e \in E(G_1)$, suppose u, v are vertices on $\tau_1'(e)$. Then we denote the subpath of $\tau_1'(e)$ between u and v by $T_e(u,v)$. Suppose we have found a monochromatic $k \times \ell$ $\bigvee$-grid as a subgraph of H. Let $P_1, \ldots, P_k$ be the monochromatic paths in the $\bigvee$-grid such that P_i is in the image of $\tau_1'(e_i)$ for some $e_i \in E(G_1)$. Consider the vertices $u_{k+1,1}, w_{k+1,1}, u_{k+1,2}, w_{k+1,2}, \ldots, u_{k+1,\ell}, w_{k+1,\ell}$. By Lemma 3.10, there is an $f \in E(G_1)$ and there are p and q, $1 \le p < q \le \ell$, such that

1. the vertices $u_{k+1,p}, \ldots, w_{k+1,p}$ occur in that order on $\tau_1'(f)$,

2. $q - p \geq \frac{\ell}{O(m^2)} = \ell'$, and
3. there are no vertices of H in the discs bounded by $u_{k+1,j}, T_f(u_{k+1,j}, w_{k+1,j})$, $w_{k+1,j}, v_{k,j}$, $p \leq j \leq q$.

Let s_{k+1} be the vertex of $\tau_1'(f)$ adjacent to $u_{k+1,p}$ such that $s_{k+1}, u_{k+1,p}, w_{k+1,p}$ occur in that order in $\tau_1'(f)$, and similarly let t_{k+1} be the vertex of $\tau_1'(f)$ adjacent to $w_{k+1,q}$ such that $u_{k+1,q}, w_{k+1,q}, t_{k+1}$ occur in that order in $\tau_1'(f)$.

By Lemma 3.11, for $p \leq r \leq q$, each $\bigvee$-intersection $u_{k+1,r}, v_{k,r}, w_{k+1,r}$ has a child, say, $u_{k+2,r}, v_{k+1,r}, w_{k+2,r}$, where $u_{k+2,r}$ is the left endpoint of the child, $v_{k+1,r}$ occurs on $\tau_1'(f)$, and $w_{k+2,r}$ is the right endpoint of the child. If $T_f(s_{k+1}, t_{k+1}) \bigcap T_{e_i}(s_i, t_i) = \varnothing$ for $1 \leq i \leq k$ then the above construction yields the required $\bigvee$-grid.

Now suppose that $T_f(s_{k+1}, t_{k+1}) \bigcap T_{e_i}(s_i, t_i) \neq \varnothing$; let Z_i be this intersection. Then Z_i is a subpath of $\tau_1'(e_i)$. Clearly $i < k$; suppose $i > 1$. Consider the following closed circuit $\mathcal{C}$ in the $\bigvee$-grid:

$$\begin{aligned} &T_{e_i}(v_{i,1}, u_{i,1}), (u_{i,1}, v_{i-1,1}),\ T_{e_{i-1}}(v_{i-1,1}, v_{i-1,\ell}),\ (v_{i-1,\ell}, w_{i,\ell}),\\ &T_{e_i}(w_{i,\ell}, v_{i,\ell}),\ (v_{i,\ell}, w_{i+1,\ell}),\ T_{e_{i+1}}(w_{i+1,\ell}, u_{i+1}, 1),\ (u_{i+1,1}, v_{i,1}). \end{aligned}$$

Planarity ensures that Z_i can only intersect $\mathcal{C}$ at either $v_{i,1}$ or $v_{i,\ell}$; we can form the new $\bigvee$-grid by removing the leftmost (or rightmost) $\bigvee$-intersection from each row of the grid.

Finally suppose that $i = 1$. Let a_1 be the endpoint of $\tau_1'(e_1)$ closer to s_1 than to t_1 and let a_{k+1} be the endpoint of $\tau_1'(e_1)$ closer to s_{k+1} than to t_{k+1}. Consider the path

$$v_{1,1}, u_{2,1}, U_{2,1}, v_{2,1}, u_{3,1}, \ldots, U_{k+1,1}, v_{k+1,1}, T_{e_1}(v_{1,1}, v_{k+1,1}).$$

This forms a closed circuit containing both a_1 and a_{k+1}. Since exactly one endpoint of $\tau_1'(e_1)$ is inside this circuit (since the other is outside), it follows that $a_1 = a_{k+1}$.

Since the remainder of the proof only involves vertices $u_{i,j}, v_{i,j}, w_{i,j}$ such that $p \leq j \leq q$, we rename these vertices $u_{i,j-p+1}, v_{i,j-p+1}, w_{i,j-p+1}$. Notice that for $1 \leq j \leq \ell'$ (recall that $\ell' = q - p$) the $\bigvee$-intersection $u_{k+1,j}, v_{k,j}, w_{k+1,j}$ is a descendant of the $\bigvee$-intersection $u_{2,j}, v_{1,j}, w_{2,j}$. Furthermore, $u_{k+1,j}$ and $w_{k+1,j}$ are on $\tau_1'(e_1)$.

Without loss of generality, suppose that $u_{k+1,1} \in T_{e_1}(v_{1,1}, v_{1,\ell'})$. The case where $u_{k+1,\ell'} \in T_{e_1}(v_{1,1}, v_{1,\ell'})$ is symmetric. If, relative to s_1, $u_{k+1,k}$ occurs to the right of $v_{1,\ell'}$, then it is sufficient to ignore the first k $\bigvee$-intersections on each path $T_{e_i}(s_i, t_i)$ that was in our grid. Thus, we obtain our $\bigvee$-grid by taking our original grid and taking the subgraph induced by $u_{i+1,j}, v_{i,j}, w_{i+1,j}, T_{e_i}(u_{i+1,j}, u_{i+1,j+1})$, where $1 \leq i \leq k$ and $k + 1 \leq j \leq \ell'$.

Now suppose $u_{k+1,k}$ occurs to the left of $v_{1,\ell'}$. We will show that in this case H is not minimal. For each $\bigvee$-intersection $u_{k+1,i}, v_{k,i}, w_{k+1,i}$, $1 \leq i \leq k$, there is at least one $\bigvee$-intersection $u_{2,j}, v_{1,j}, w_{2,j}$ $(1 \leq j \leq \ell')$ such that $v_{1,j}$ occurs between $u_{k+1,i}$ and $w_{k+1,i}$. This follows from Lemma 3.11 and the fact that there are no $\bigvee$-intersections between $u_{2,j}, v_{1,j}, w_{2,j}$ and $u_{2,j+1}, v_{1,j+1}, w_{2,j+1}$, where $1 \leq j < \ell'$ (see the definition of a $\bigvee$-grid). For each $\bigvee$-intersection $u_{k+1,j}, v_{k,j}, w_{k+1,j}$ $(1 \leq j \leq k)$, choose a $\bigvee$-intersection $u_{2,r_j}, v_{1,r_j}, w_{2,r_j}$ such that v_{1,r_j} occurs between $u_{k+1,j}$ and $w_{k+1,j}$ on $\tau_1'(e_1)$.

We will now define a rerouting of $P_1, \ldots, P_k$, called $R_1, R_2, \ldots, R_k$ (see Figure 8). This will result in a contradiction to Lemma 3.4. The idea is that we can find k consecutive spirals in the $\bigvee$-grid each spiraling in the same direction. Having $u_{k+1,k}$ to the left of $v_{1,\ell'}$ ensures that each spiral goes through exactly one complete

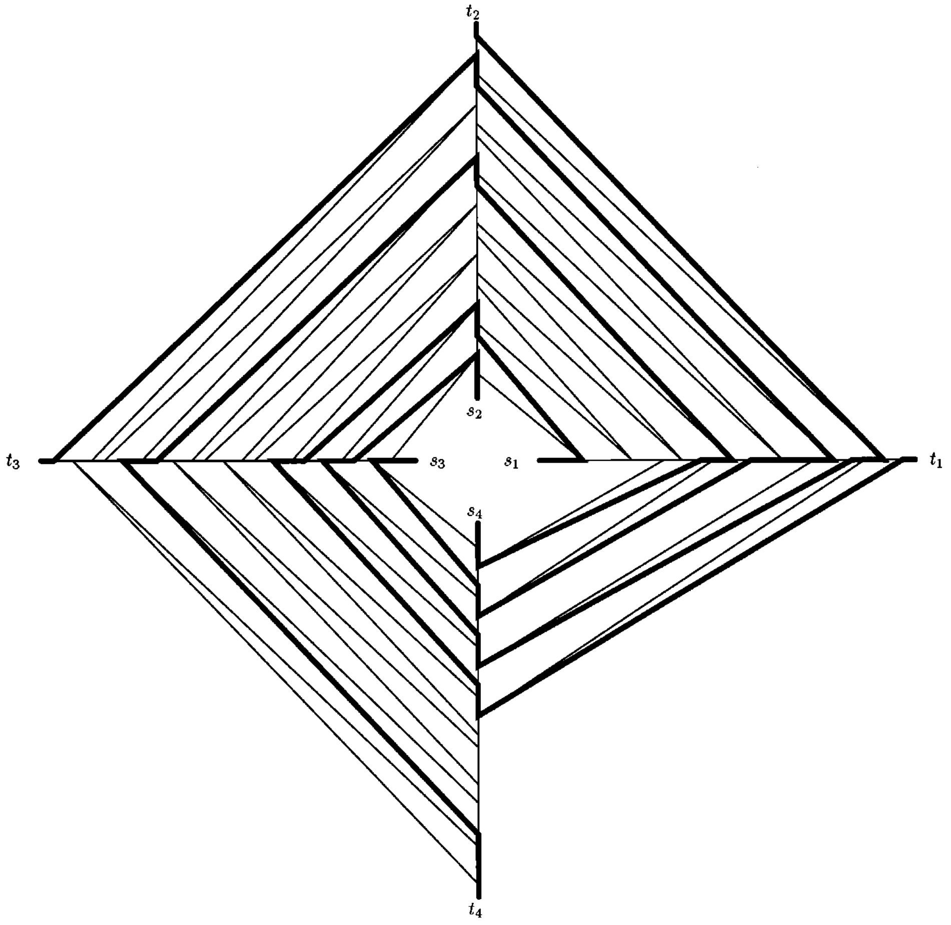


FIG. 8. *Rerouting on a $\bigvee$-grid. Dark edges indicate the rerouting.*

rotation with the ith spiral starting at s_i and finishing at t_i; let us call the ith spiral R_i. Then along P_i the order of intersection of the spirals (from s_i to t_i) is $R_i, R_{i-1}, \ldots, R_1, R_k, R_{k-1}, \ldots, R_{i+1}, R_i$. Since R_i occurs at the beginning and end of this sequence, it is the required rerouting. A formal presentation of these ideas follows.

For $2 \leq i \leq k+1$ and $1 \leq j \leq \ell'$, define the *successor* of vertex $w_{i,j}$, $S(w_{i,j})$, as follows:

$$S(w_{i,j}) = \begin{cases} w_{i+1,j+1} & \text{if } i < k+1 \text{ and } j < \ell', \\ w_{2,r_{j+1}} & \text{if } i = k+1 \text{ and } r_j < \ell', \\ v_{i,\ell'} & \text{otherwise.} \end{cases}$$

Furthermore, define the *link* of $w_{i,j}$, $L(w_{i,j})$, as follows:

$$L(w_{i,j}) = \begin{cases} T_{e_i}(w_{i,j}, v_{i,j+1}), w_{i+1,j+1}, & i < k+1, j < \ell', \\ T_{e_1}(w_{i,j}, v_{1,r_{j+1}}), w_{2,r_{j+1}}, & i = k+1, r_j < \ell', \\ T_{e_i}(w_{i,j}, v_{i,\ell'}) & \text{otherwise.} \end{cases}$$

We will denote s compositions of S by $S^{(s)}$. From the definition of successor and link, we can immediately conclude the following lemma.

LEMMA 4.4. *Let $1 \leq i, i' \leq k+1$ and $1 \leq j, j' \leq \ell'$. If $w_{i,j} \neq w_{i',j'}$ then $S(w_{i,j}) \neq S(w_{i',j'})$ and $L(w_{i,j})$ is internally vertex disjoint from $L(w_{i',j'})$.*

For $1 \leq i \leq k$ and $1 \leq t \leq \ell'$ define a path $R(i,t)$ as follows:

$$\begin{aligned} R(i,0) &= (v_{i,1}, w_{i+1,1}),\\ R(i,t+1) &= R(i,t), L(S^{(t)}(w_{i,1})),\\ R(i,k+1) &= R(i,k), T_{e_i} S^{(k+1)}(w_{i,1}). \end{aligned}$$

Now, let $R_i = R(i,k+1)T(v_{i,l'}, t_i)$.

We show that $R_1, \ldots, R_k$ satisfy the definition of a rerouting. By construction, all the vertices of R_i lie in P_j for some j and R_i has endpoints s_i and t_i. Thus, it remains to show that the R_i are vertex disjoint and that they omit some edge of the P_j's.

LEMMA 4.5. *For $1 \leq i \leq k$, let the ith vertex sequence be given by*

$$\begin{aligned} &w_{i,1},\ S(w_{i-1,1}), S^{(2)}(w_{i-2,1}), \ldots, S^{(j)}(w_{i-j,1}), \ldots, S^{(i-2)}(w_{2,1}),\\ &S^{(i-1)}(w_{k+1,1}), \ldots, S^{(k-1)}(w_{i-1,1}). \end{aligned}$$

Then the elements of the ith vertex sequence appear in order on P_i starting at s_i.

Proof. First notice that for any i and $1 \leq j < j' \leq \ell'$ the vertex $S(w_{i,j})$ is to the left of $S(w_{i,j'})$ with respect to $s_{(i \bmod k)+1}$. We prove the lemma simultaneously for all i and for prefixes of length t, $1 \leq t \leq k+1$, of the vertex sequences.

For $t = 1$, the claim is obvious. Now, suppose the lemma holds for $t = s$. Then, by the induction hypothesis for $i' = i-1$ if $i \neq 1$ and $i' = k$ when $i = 1$, the elements of the prefix of the i'th vertex sequence appear in order on $P_{i'}$ starting at $s_{i'}$. Therefore, their successors appear in order on P_i starting at s_i. But their successors are exactly the second through $(s+1)$st elements of the ith vertex sequence. Furthermore, w_{r_1} is the leftmost w on P_i with respect to s_i. □

From Lemma 4.4 it immediately follows that the R_i are vertex disjoint. Furthermore, by definition every vertex on these paths is a vertex of $T_{e_i}(s_i, t_i)$ for some i. Finally, if g is the edge of $\tau_1'(e_1)$ with one endpoint $v_{1,1}$ and the other endpoint to the right of $v_{1,1}$ relative to s_1, then g is not used in any of the R_i. Thus H is not minimal and we have proven Lemma 4.3.

Now suppose that there is no edge of either G_1 or G_2 whose image has $m^{\Omega(m)}$ $\bigvee$-intersections or $m^{\Omega(m)}$ $\bigwedge$-intersections. The idea is essentially the same as before—we attempt to construct a large $+$-grid using Lemma 3.10 and an observation analogous to Lemma 3.11. If we are successful in creating row after row of this grid we are done since we have a large grid in the graph. If not, we show that a rerouting is possible. We present the technical details of this construction that differ from those of the $\bigvee$-grid's construction.

Since there are less than $m^{O(m)}$ $\bigvee$-intersections and $\bigwedge$-intersections, there must be some edge of G_1 or G_2 whose image has at least $m^{\Omega(m^2)}$ $+$-intersections. Let e be an edge of G_1 such that $\tau_1'(e)$ has path length at least $m^{\Omega(m^2)}$ of which less than $m^{O(m)}$ are $\bigvee$-intersections and $\bigwedge$-intersections. Then there is a consecutive sequence of vertices $v_1, \ldots, v_r$ on $\tau_1'(e)$ such that each is the middle vertex of a $+$-intersection and r is $m^{\Omega(m)}$.

DEFINITION (refer to Figure 9). Let $k, \ell \in \mathbb{N}$. A planar graph G is a $k \times \ell$ $+$-grid, $+_{k,\ell}$, if the following holds:

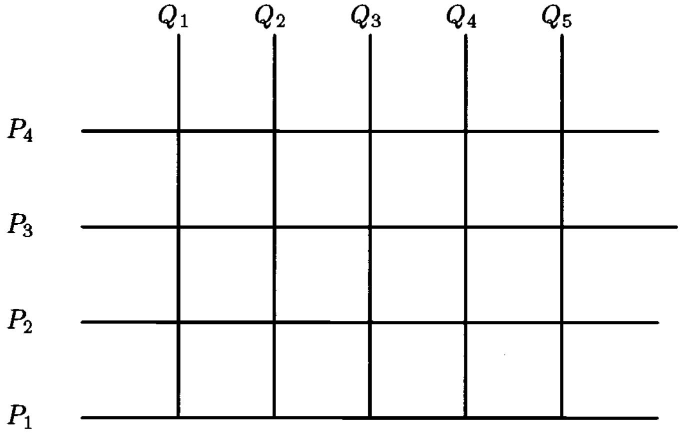


FIG. 9. *A* $+$*-grid.*

1. There are k vertex disjoint paths $P_1, \ldots, P_k$ in G such that P_i has endpoints s_i and t_i.
2. There are vertices $v_{i,j}$ for $(i,j) \in [1, k+1] \times [1, \ell]$ such that $s_i, v_{i,1}, v_{i,2}, \ldots, v_{i,\ell}, t_i$ occur in that order on P_i.
3. There are ℓ vertex disjoint paths $Q_1, \ldots, Q_\ell$ such that $v_{1,j}, v_{2,j}, \ldots, v_{k+1,j}$ occur in that order on Q_i.

We will call $v_{k+1,1}, \ldots, v_{k+1,\ell}$ the *offsprings* of the grid.

The notion of a *monochromatic* $+$-grid carries over from a $\bigvee$-grid. That is, for each P_i there is an edge $e_i \in E(G_1)$ such that P_i is a subpath of $\tau_1'(e_i)$ and for each Q_i there is a similar $f_i \in E(G_2)$.

The idea is to inductively build $\mathcal{G}_{3m,3m}$, from which the result will follow. This follows from the following analogue of Lemma 4.3.

LEMMA 4.6. *Suppose that H contains a monochromatic $k \times \ell$ $+$-grid as a subgraph, $k \geq 0$. Then H contains a monochromatic $(k+1) \times \frac{\ell}{O(m^2)}$ $+$-grid as a subgraph.*

We first notice that by Lemma 3.10 there is some $e \in E(G_1)$ such that $\tau_1'(e)$ passes consecutively through $\frac{\ell}{O(m^2)}$ of the offsprings of the $+$-grid. Furthermore, since there are at most m terminals, we can assume that all the offsprings are nonterminals. Denote these offsprings by $v_{k+1,j_1}, v_{k+1,j_1+1}, \ldots, v_{k+1,j_2}$, where $j_2 - j_1 \geq \frac{\ell}{O(m^2)}$. Since for $j_1 \leq j \leq j_2$ the path $\tau_2'(f_j)$ passes through $v_{k+1,j}$, there is either a $\bigvee$-intersection, a $+$-intersection, or a $\bigwedge$-intersection at $v_{k+1,j}$. But there can be at most $m^{O(m)}$ $\bigvee$-intersections or $\bigwedge$-intersections.

Thus there are j_3, j_4, $j_1 \leq j_3 \leq j_4 \leq j_2$, such that there is a $+$-intersection at $v_{k+1,j}$ for $j_3 \leq j \leq j_4$. Let $v_{k+2,j}$ be the vertex such that $v_{k+2,j}, v_{k+1,j}, v_{k,j}$ form a $+$-intersection centered at $v_{k+1,j}$. Let $e_{k+1} = e$. Let s_{k+1} be the vertex of $\tau_1'(e)$ adjacent to v_{k+1,j_3} such that $s_{k+1}, v_{k+1,j_3}, v_{k+1,j_3+1}$ occur in that order on $\tau_1'(e)$. Similarly, let t_{k+1} be the vertex of $\tau_1'(e)$ adjacent to v_{k+1,j_4} such that $s_{k+1}, v_{k+1,j_4}, t_{k+2}$ occur in that order on $\tau_1'(e)$.

If $T_{e_k+1}(s_{k+1}, t_{k+1}) \cap T_{e_1}(s_1, t_1) = \varnothing$ we have the required larger grid. Now suppose that $T_{e_k+1}(s_{k+1}, t_{k+1}) \cap T_{e_1}(s_1, t_1) \neq \varnothing$. In this case, we are interested only in the vertices $v_{i,j}$, where $1 \leq i \leq k+1$ and $j_3 \leq j \leq j_4$, so we rename these vertices

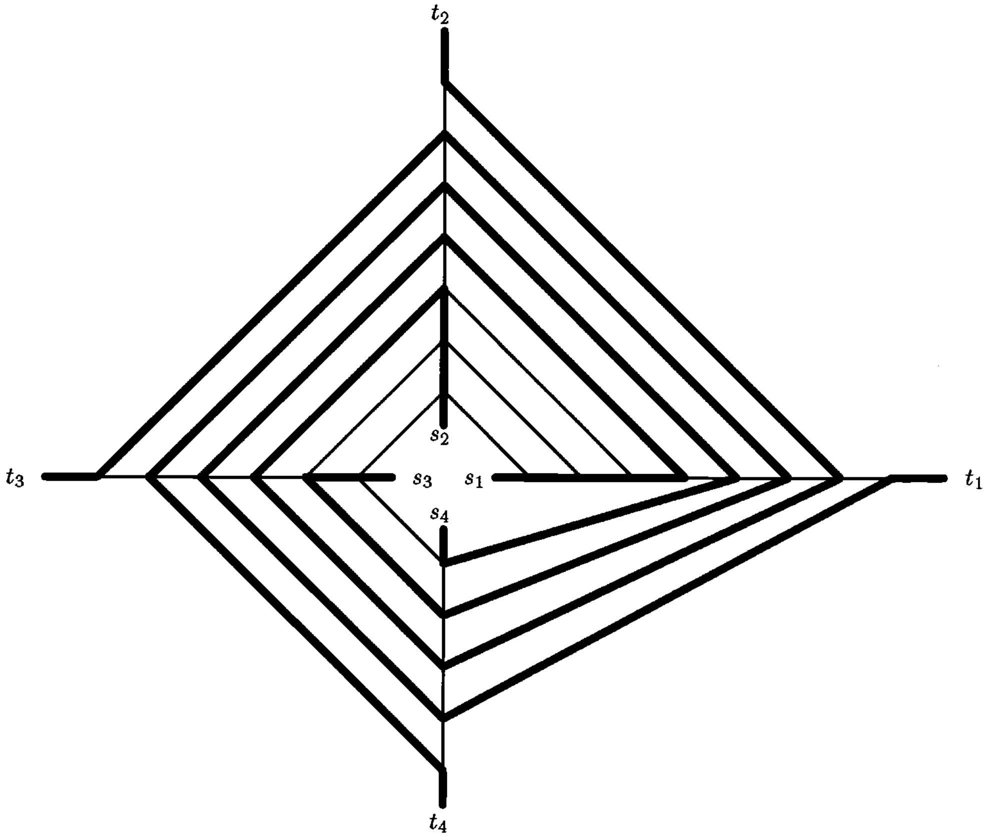


FIG. 10. *$v_{k+1,1}$ occurs to the right of $v_{1,k}$.*

$v_{i,j-j_3}$. Further, let $\ell' = j_4 - j_3 \geq \frac{\ell}{O(m^2)}$ and assume without loss of generality that $v_{k+2,1} \in T_{e_1}(v_{1,1}, v_{1,\ell'})$. Again, if relative to s_1, $v_{k+1,k}$ occurs to the right of $v_{1,\ell'}$, then we can simply ignore the k paths $Q_1, \ldots, Q_k$ in our grid.

If $v_{k+1,k}$ occurs to the left of $v_{1,\ell'}$, we must handle two further cases, namely, when $v_{k+1,1}$ occurs to the right of $v_{1,k}$ and when it occurs to the left of $v_{1,k}$.

First suppose $v_{k+1,1}$ occurs to the right of $v_{1,k}$ (see Figure 10). Then, for $1 \leq i \leq k$, let R_i be the path

$$T_{e_i}(s_i, v_{1,k-i+1}), v_{2,k-i+1}, v_{3,k-i+1}, \ldots, v_{k+1,k-i+1}, T_{e_i}(v_{k+1,k-i+1}, t_i).$$

Then it is clear that the R_i are vertex disjoint and that they are a rerouting of the P_i. Thus H is not minimal.

Now suppose that $v_{k+1,1}$ occurs to the left of or is the same as $v_{1,k}$ (see Figure 11). The key to solving this case is to notice that if we reverse the roles of the P_i and Q_j we can apply the previous case. In particular, we can reroute the Q_j using the P_i.

More formally, let $k' \leq k$ such that $v_{k+1,1} = v_{1,k'+1}$. Notice that for $1 \leq i \leq k'$, Q_i and $Q_{k'+i}$ are both subpaths of the same path $\tau_2'(f_i)$. Now, for $1 \leq i \leq k'$ let S_i be the subpath of $\tau_2'(f_i)$ which starts at a terminal and goes through $v_{1,i}, v_{2,i}, \ldots, v_{k'+2-i,i}$. Let S_i' be the subpath of $\tau_2'(f_i)$ starting at $v_{2k'+2-i,i}$, going through $v_{2k'+3-i,i}, \ldots,$

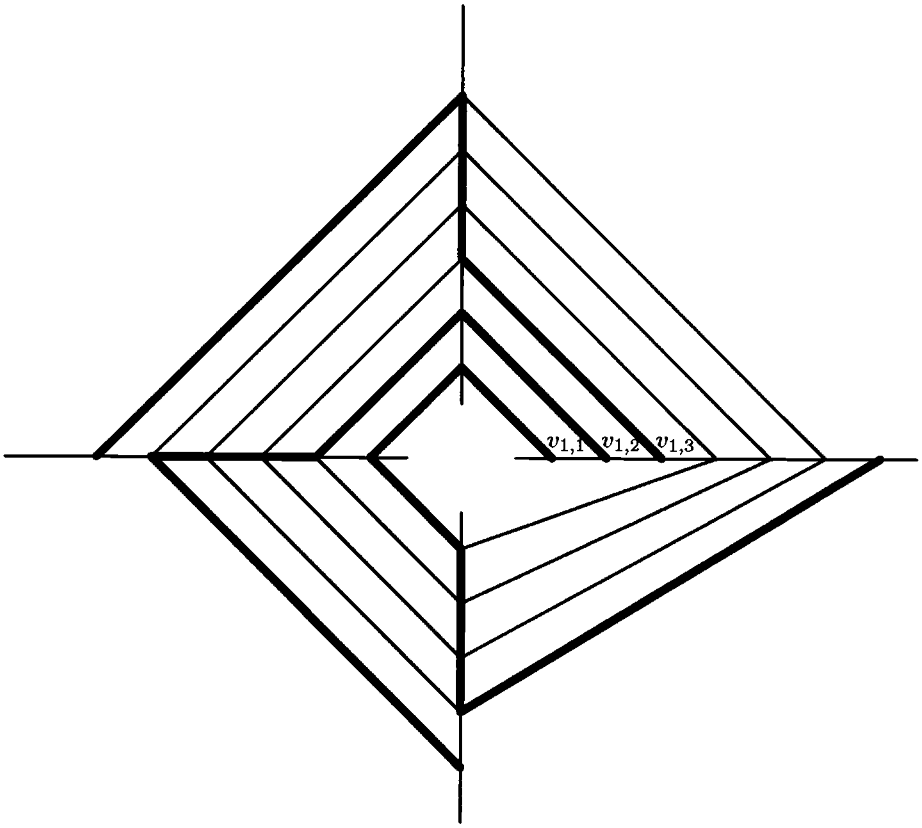


FIG. 11. $v_{k+1,1}$ *does not occur to the right of* $v_{1,k}$.

and ending at a terminal. Finally, let R_i be the path

$$S_i, T_{e_{k'+2-i}}(v_{k'+2-i,i}, v_{2k'+2-i}), S'_i.$$

Then, as before, it is straightforward to verify that the R_i are disjoint and are a rerouting of the Q_j. This contradicts the minimality of H.

We are now finished with the proof of Lemma 4.2.

5. Conclusions and open problems. There is a great amount of work which remains to be done in this area. There is a vast gap between the upper bounds and the known lower bounds. The best lower bound for even general graphs is polynomial (see, e.g., Figure 12), as compared to the upper bound of a tower of 2s for the general problem as given by Seymour and Thomas [ST91]. Although it is probably not difficult to obtain slight improvements in the lower bounds, no natural candidates for the asymptotic complexity of this function present themselves. Further investigation of intertwine bounds may give insight into the enormous constants found in parts of the Robertson and Seymour proof.

There are also a number of other natural operations on lower ideals. For example, consider the following problem suggested by Fellows and Langston. Let $\mathcal{F}$ be a lower ideal and define $\mathcal{F}+1$ to be

$$\{G : \exists v G \backslash v \in \mathcal{F}\}.$$

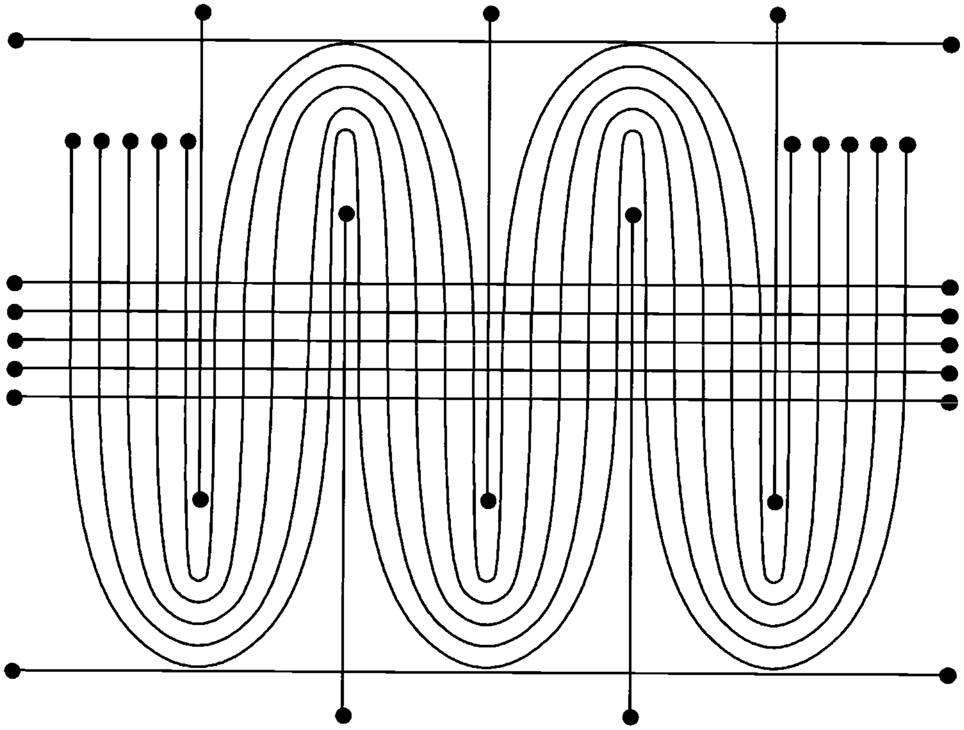


FIG. 12. *An $\Omega(n^3)$ node planar intertwine of two $O(n \log n)$ node graphs. Dark points on the endpoints of the lines represent distinct elements of a family of $O(\log n)$ node graphs which are independent under the minor ordering. The first intertwined graph consists of the $n+2$ horizontal lines and their corresponding endpoints. The second consists of the n vertical and n curved lines and their corresponding endpoints. The intertwine graph includes all of the points of intersection, so it has nodes from n distinct $n \times n$ grids for a total of n^3 points.*

Then $\mathcal{F}+1$ is a lower ideal, and it is interesting to compute its obstructions. For example, for the family of planar graphs $\mathcal{F}$, $\mathcal{F}+1$ is the set of *apex* graphs. It is possible to use the techniques of Fellows and Langston outlined in the introduction to compute these obstructions given those of $\mathcal{F}$. However, once again those techniques yield obstructions incrementally and do not give a bound on the size of the obstructions.

Acknowledgments. We thank Stephen Cook, Charles Rackoff, and Neil Robertson for a number of interesting and useful discussions. We thank the referees for their very careful reading of the manuscript and their many suggestions which led to a much clearer presentation.

REFERENCES

[BW89] R. BODENDIEK AND K. WAGNER, *Solution to König's graph embedding problem*, Math. Nachr., 140 (1989), pp. 251–272.

[BM76] J. BONDY AND U.S.R. MURTY, *Graph Theory with Applications*, North–Holland, Amsterdam, The Netherlands, 1976.

[DR91] H. DJIDJEV AND J. REIF, *An efficient algorithm for the genus problem with explicit construction of forbidden subgraphs*, in 31st Symposium on Foundations of Computer Science, St. Louis, MO, 1991, pp. 337–347.

[FL88] M. Fellows and M. Langston, *Nonconstructive tools for proving polynomial-time decidability*, J. Assoc. Comput. Mach., 35 (1988), pp. 727–739.

[FL94] M. Fellows and M. Langston, *On search, decision and the efficiency of polynomial-time algorithms*, J. Comput. System Sci., 49 (1994), pp. 769–779.

[FL89] M. Fellows and M. Langston, *An analogue of the Myhill–Nerode theorem and its use in computing finite-basis characterizations*, in 30th IEEE Symposium on Foundations of Computer Science, Research Triangle Park, NC, 1989, pp. 520–525.

[GI91] A. Gupta and R. Impagliazzo, *Computing planar intertwines*, in 32nd Symposium on the Foundations of Computer Science, San Juan, Puerto Rico, 1991, pp. 802–811.

[GI] A. Gupta and R. Impagliazzo, *Planar intertwines under topological embeddings*, manuscript.

[La94] J. Lagergren, *The size of an intertwine*, in 21st International Colloquium on Automata, Languages and Programming, Jerusalem, Israel, 1994, pp. 520–531.

[RS84] N. Robertson and P. Seymour, *Graph minors* III. *Planar tree-width*, J. Combin. Theory Ser. B, 36 (1984), pp. 49–64.

[RS95] N. Robertson and P. Seymour, *Graph minors* XIII. *The disjoint paths problem*, J. Combin. Theory Ser. B, 63 (1995), pp. 65–110.

[RSa] N. Robertson and P. Seymour, *Graph minors* XX. *Wagner's conjecture*, manuscript.

[RST95] N. Robertson, P. Seymour, and R. Thomas, *Sachs' linkless embedding conjecture*, J. Combin. Theory Ser. B, 64 (1995), pp. 185–227.

[ST91] P. Seymour and R. Thomas, personal communication, 1991.

[Ung78] P. Ungar, *Dissections and intertwinings of graphs*, Amer. Math. Monthly, 85 (1978), pp. 664–666.

SIAM J. Discrete Math.
Vol. 10, No. 3, pp. 359–372, August 1997

THE LENGTH OF A LEAF COLORATION ON A RANDOM BINARY TREE[*]

A. M. HAMEL[†] and M. A. STEEL[†]

Abstract. An assignment of colors to objects induces a natural integer weight on each tree that has these objects as leaves. This weight is called "parsimony length" in biostatistics and is the basis of the "maximum parsimony" technique for reconstructing evolutionary trees. Equations for the average value (over all binary trees) of the parsimony length of both fixed and random colorations are derived using generating function techniques. This leads to asymptotic results that extend earlier results confined to just two colors. A potential application to DNA sequence analysis is outlined briefly.

Key words. binary tree, Fitch's algorithm, maximum parsimony tree, DNA/RNA sequences, probability

AMS subject classifications. 05C05, 05A15, 92D20

PII. S0895480194271591

1. Introduction.

Let $\mathcal{B}(n),\ n \geq 2$, denote the set of (unrooted) trees with n leaves (vertices of degree 1) labeled $1, 2, \ldots, n$ and with all remaining vertices unlabeled and of degree 3. Such trees, which we will simply call *binary trees*, are useful representations of evolutionary relationships in taxonomy. In that case, the set $[n] = \{1, 2, \ldots, n\}$ represents the extant taxa being classified, while the remaining vertices in the tree represent ancestral taxa. It is often convenient to represent the (global) ancestral taxon of all these taxa by a root vertex obtained by subdividing an edge (the "most ancient" edge) of the tree. Let $\mathcal{R}(n),\ n > 1$, denote the set of all such edge-rooted binary trees on leaf set $[n]$. We define $\mathcal{R}(1)$ as the singleton set consisting of an isolated (root) vertex labeled 1. Note for $n \geq 2$ the bijection

$$\psi : \mathcal{B}(n) \to \mathcal{R}(n-1),$$

where, if $T \in \mathcal{B}(n),\ \psi(T)$ is the edge-rooted binary tree which results when leaf n and its incident edge are deleted, as in Figure 1. Edge subdivision also gives a bijection,

$$\psi' : \{(T, e) : T \in \mathcal{B}(n),\ e \in E(T)\} \to \mathcal{R}(n),$$

as in Figure 1. We let

$$B(n) := |\mathcal{B}(n)| \quad \text{and} \quad R(n) := |\mathcal{R}(n)|$$

for $n \geq 2$ and $n \geq 1$, respectively. Since $|E(T)| = 2n-3$, for each $T \in \mathcal{B}(n)$, it follows (from ψ and ψ') that, for $n \geq 3$,

$$\begin{aligned} R(n) = (2n-3)B(n) = (2n-3)!! &= 3 \times 5 \times \cdots \times (2n-3) \\ &= \frac{(2n-2)!}{2^{n-1}(n-1)!}, \end{aligned} \tag{1}$$

[*] Received by the editors July 22, 1994; accepted for publication (in revised form) May 31, 1996. The first author of this paper was supported by a postdoctoral fellowship from the Natural Sciences and Engineering Research Council of Canada.

http://www.siam.org/journals/sidma/10-3/27159.html

[†] Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand (physamh@cantua.canterbury.ac.nz, m.steel@math.canterbury.ac.nz).

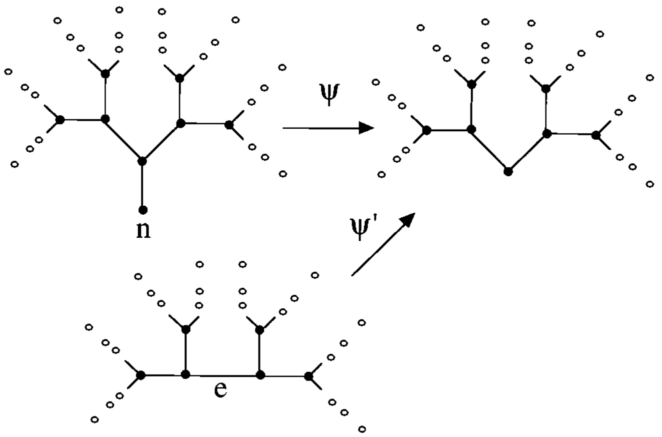


FIG. 1. *Bijections between rooted and unrooted binary trees.*

a result dating back at least as far as 1870 to a paper by Schröder [9]. Thus, by applying Stirling's formula to $R(n)$,

$$\frac{R(n)}{n!} \sim \frac{1}{2\sqrt{\pi}}\, 2^n n^{-3/2}. \tag{2}$$

(A definition of "asymptotic" ($\sim$) appears at the beginning of section 3.)

Let χ be a coloration of $[n]$ by a set $\mathcal{C}$ of $r \geq 2$ colors. For example, in phylogenetic analysis each site j in a collection of n aligned DNA/RNA sequences (where $r = 2$ or 4) gives a coloration $\chi = \chi^j$ of $[n]$ for which $\chi^j(i)$ is the actual nucleotide (when $r = 4$) or its purine/pyrimidine classification (when $r = 2$) that occurs at site j in the ith sequence.

Given a tree T in $\mathcal{B}(n)$ or $\mathcal{R}(n)$ and a coloration χ of $[n]$ let $\ell(T, \chi)$ be the minimal number of edges of T that need to be assigned differently colored ends in order to extend χ to a coloration of all the vertices of T (any such minimizing extension is called a *minimal extension* of χ for T). The number $\ell(T, \chi)$ is called the *parsimony length* of χ on T, and it is the basis of the widely used "maximum parsimony" technique for reconstructing evolutionary trees from aligned genetic sequences. This approach selects the tree(s) T which minimizes (minimize) the sum of $\ell(T, \chi^j)$ over all sites j in the sequences—this sum is the *length* of T on the sequences. Such a tree—a *maximum parsimony* tree—requires the fewest mutations to account for the variations in the aligned sequences.

The aim of this paper is to develop analytic techniques that would allow the length of the maximum parsimony tree on the original sequences to be compared with the average length of all binary trees on either (i) the original sequences or (ii) randomized versions of the original sequences (i.e., sequences generated randomly with the same expected frequencies of colors as the original sequences, as in Steel, Lockhart, and Penny [11]). These two average values are obtained by evaluating, respectively, certain functions μ_n and μ_n' (which we describe in section 2) at each sequence site and summing up the resulting values across the sites. An asymptotic formula for μ_n' is described in section 3 and since, as we show, μ_n and μ_n' are asymptotically equivalent, this provides an asymptotic formula for μ_n as well. Our results exploit some special properties of the generating functions which count various classes of leaf labeled trees

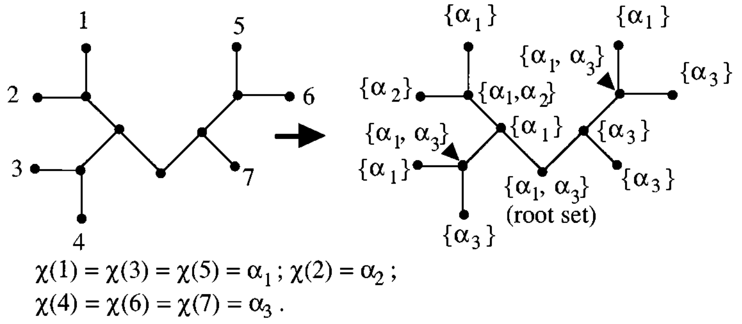


FIG. 2. *A rooted tree with sets assigned to vertices by the parsimony operation.*

according to their parsimony length. In this sense the exact and asymptotic analyses complement and extend the approaches of Carter et al. [2] and Butler [1], respectively, both of which analyzed similar systems of generating functions with just two colors (although the problems these authors considered were slightly different from ours).

First we describe a convenient technique for computing $\ell(T, \chi)$ known as the (first pass of) Fitch's algorithm (Fitch [3], Hartigan [5]). If $T' \in \mathcal{B}(n)$, subdivide any edge of T' to obtain a tree $T \in \mathcal{R}(n)$. Note that $\ell(T, \chi) = \ell(T', \chi)$. Now direct all edges of T away from the root and recursively assign nonempty subsets of colors to the vertices of T beginning with the leaves and progressing toward the root, as follows:

(1) leaf $i \in [n]$ is assigned the singleton set $\{\chi(i)\}$,

(2) once the descendants of vertex v have both been assigned sets A, B, then assign vertex v the set $A * B$, where $*$ is the (nonassociative, binary) "parsimony operation" defined on $2^{\mathcal{C}} - \phi$,

$$A * B = \begin{cases} A \cap B & \text{if } A \cap B \neq \phi, \\ A \cup B & \text{if } A \cap B = \phi. \end{cases}$$

The set assigned to the root of T is called the *root set.* (In the case $T \in \mathcal{R}(1)$ the root set is just $\{\chi(1)\}$.) These concepts are illustrated in Figure 2. A fundamental property of this procedure is the following.

LEMMA 1.1 (Hartigan [5]). *$\ell(T, \chi)$ is the number of times an empty intersection (option 2 in the above description of $*$) is encountered in this assignment of sets of colors to the vertices of T. Furthermore, the root set is precisely the set of those colors that occur in at least one minimal extension of χ for T.*

We will use both of these properties in section 2.

Notation.

(1) For convenience, we write

$$\underline{x} \quad \text{to denote} \quad (x_1, x_2, \ldots, x_r),$$

$$\underline{x}^{\underline{a}} \quad \text{to denote the monomial} \quad x_1^{a_1} x_2^{a_2} \ldots x_r^{a_r},$$

and

$$\underline{a}! \quad \text{to denote} \quad a_1! a_2! \ldots a_r!.$$

(2) We also write

$$[\underline{x}^{\underline{a}}] f(\underline{x}) \quad \text{to denote the coefficient of} \quad x_1^{a_1} x_2^{a_2} \ldots x_r^{a_r} \quad \text{in } f(\underline{x}),$$

as in Goulden and Jackson [4].

(3) $\mathcal{C} = \{\alpha_1, \ldots, \alpha_r\}$ will denote the set of colors which are assigned to the elements of the set $[n] = \{1, \ldots, n\}$. If $a_i = |\chi^{-1}(\alpha_i)|$, $i = 1 \ldots r$, we say χ is of *type* $\underline{a} = (a_1, \ldots, a_r)$. Thus, $a_i \geq 0$ and $\sum_{i=1}^{r} a_i = n$.

2. Calculations (exact). The aim of this paper is to calculate the two averages that we now define.

Definition 1 (μ_n and μ'_n). *Let $\mu_n(\underline{a})$ be the average, over all trees $T \in \mathcal{B}(n)$, of the length of a fixed coloration of $[n]$ of type $\underline{a}$ on T.*

For probability distribution $\underline{\phi} = (\phi_1, \phi_2, \ldots, \phi_r)$, $\phi_1 \geq 0, \phi_2 \geq 0, \ldots, \phi_r \geq 0$, $\sum_{i=1}^{r} \phi_i = 1$, let $\mu'_n(\underline{\phi})$ be the average, over all trees $T \in \mathcal{B}(n)$, of the expected length of a random coloration of $[n]$ on T. In this random coloration each element of $[n]$ is independently assigned color α_i with probability ϕ_i.

Note that $\mu'_n(\underline{\phi})$ is the average, over all trees $T \in \mathcal{B}(n)$, of

$$\sum_{\chi} \ell(T, \chi) \prod_{j=1}^{n} \phi_{\chi(j)},$$

and so

$$\mu'_n(\underline{\phi}) = \sum_{\underline{a}} \binom{n}{\underline{a}} \underline{\phi}^{\underline{a}} \, \mu_n(\underline{a}). \tag{3}$$

Here and elsewhere a summation over $\underline{a}$ ranges over all nonnegative r-tuples $a_1, \ldots, a_r$ with $\sum_{i=1}^{r} a_i = n$. Also, note that μ_n and μ'_n are symmetric functions in $a_1, \ldots, a_r$ and $\phi_1, \ldots, \phi_r$, respectively. The following generating function forms the basis for our calculations. For $\phi \neq A \subseteq \mathcal{C}$, let $T_A(\underline{x}, z) = \sum_{\underline{a},\ell} \frac{f_A(\underline{a},\ell)}{\underline{a}!} \; \underline{x}^{\underline{a}} z^{\ell}$, where $f_A(\underline{a}, \ell)$ is the number of trees in $\mathcal{R}(n)$, $n \geq 1$, of parsimony length $\ell \geq 0$ and root set A for a fixed r-coloration of $[n]$ of type $\underline{a}$. By Lemma 1.1, the set $\{T_A(\underline{x}, z),\ \emptyset \neq A \subseteq \mathcal{C}\}$ satisfies the system of simultaneous quadratic equations described in Steel [10],

$$\begin{aligned} &T_A(\underline{x}, z) = \\ &\sum_{(B,C): B \cap C = A} \frac{1}{2} T_B(\underline{x}, z) T_C(\underline{x}, z) + \sum_{(B,C):\, \substack{B \cap C = \emptyset \\ B \cup C = A}} \frac{z}{2} T_B(\underline{x}, z) T_C(\underline{x}, z) + \delta_A(\underline{x}), \end{aligned} \tag{4}$$

where

$$\delta_A(\underline{x}) = \begin{cases} x_i & \text{if } \ A = \{\alpha_i\}, \\ 0 & \text{if } \ |A| > 1. \end{cases} \tag{5}$$

For $r = 2$ this system can be treated by the multivariate Lagrange inversion formula (Goulden and Jackson [4]) to give an explicit closed-form expression for $f_A(\underline{a}, \ell)$—see Carter et al. [2], Steel [10].

Theorem 2.1 (exact formulae). *Let $T_A(\underline{x}) := T_A(\underline{x}, 1)$.*

(i)

$$\mu_n(\underline{a}) = \frac{\underline{a}!}{R(n)} \; [\underline{x}^{\underline{a}}] \sum_{(A,B): A \cap B = \emptyset} \frac{1}{2} \; T_A(\underline{x}) T_B(\underline{x}) \left(1 - 2 \sum_{i=1}^{r} x_i \right)^{-\frac{1}{2}}.$$

(ii)

$$\mu_n'(\underline{\phi}) = \frac{n!}{R(n)}[x^n] \sum_{(A,B):A\cap B=\emptyset} \frac{1}{2}\ T_A(\underline{\phi}x)T_B(\underline{\phi}x)(1-2x)^{-\frac{1}{2}},$$

where $\underline{\phi}x = (\phi_1 x, \ldots, \phi_r x)$.

(iii)

$$\mu_n'(\underline{\phi}) = \mu_{n-1}'(\underline{\phi}) + \sum_{\substack{i,A:\\ \alpha_i \notin A}} \phi_i \frac{(n-1)!}{R(n-1)}[x^{n-1}]T_A(\underline{\phi}x).$$

Proof of Theorem 2.1. Let

$$R(\underline{x}, z) := \sum_{A\neq\emptyset} T_A(\underline{x}, z), \tag{6}$$
$$R(\underline{x}) := R(\underline{x}, 1).$$

First observe that, from (4), we have the fundamental identity

$$R(\underline{x}, z) = \frac{1}{2}\ R^2(\underline{x}, z) + (z-1) \sum_{\substack{(B,C):\\ B\cap C=\emptyset}} \frac{1}{2}\ T_B(\underline{x}, z)T_C(\underline{x}, z) + \sum_{i=1}^{r} x_i. \tag{7}$$

Putting $z = 1$ in (7), we obtain

$$R(\underline{x}) = \frac{1}{2}R^2(\underline{x}) + \sum_{i=1}^{r} x_i, \tag{8}$$

so that

$$R(\underline{x}) = 1 - \sqrt{1 - 2\sum_{i=1}^{r} x_i}\ . \tag{9}$$

In particular,

$$R(\underline{\phi}x) = 1 - \sqrt{1-2x}, \tag{10}$$

and so

$$[x^n]R(\underline{\phi}x) = \frac{R(n)}{n!}. \tag{11}$$

Let

$$Q(\underline{x}) := \frac{\partial}{\partial z} R(\underline{x}, z)|_{z=1}. \tag{12}$$

Then, from (7),

$$Q(\underline{x}) = Q(\underline{x})R(\underline{x}) + \sum_{\substack{(B,C):\\ B\cap C=\emptyset}} \frac{1}{2}\ T_B(\underline{x})T_C(\underline{x}).$$

Hence

$$Q(\underline{x}) = \sum_{(B,C):B\cap C=\emptyset} \frac{1}{2}\ T_B(\underline{x})T_C(\underline{x}) \cdot (1-R(\underline{x}))^{-1}.$$

Applying (9) gives

$$Q(\underline{x}) = \sum_{(B,C):B\cap C=\emptyset} \frac{1}{2}\ T_B(\underline{x})T_C(\underline{x}) \left(1-2\sum_{i=1}^{r} x_i\right)^{-\frac{1}{2}}. \tag{13}$$

Now from (12) $\underline{a}![\underline{x}^{\underline{a}}]Q(\underline{x})$ is the sum $\sum_{\ell} \ell f(\underline{a},\ell)$, where $f(\underline{a},\ell) = \sum_{A\neq\emptyset} f_A(\underline{a},\ell)$, the total number of trees $T \in \mathcal{R}(n)$ of length ℓ for a coloration χ of $[n]$ of type $\underline{a}$. Thus $\frac{\underline{a}!}{R(n)}[\underline{x}^{\underline{a}}]Q(\underline{x})$ is the average length over all trees in $\mathcal{R}(n)$ of the length of χ. However, each edge rooting of a binary tree leads to an identical parsimony length (i.e., the position of the root is irrelevant to the length), so this quantity is also the average length over all trees in $\mathcal{B}(n)$ of the length of χ, which in view of (13) establishes part (i).

(ii) Applying part (i) to (3) we obtain

$$\mu_n'(\underline{\phi}) = \sum_{\underline{a}} \binom{n}{\underline{a}} \underline{\phi}^{\underline{a}} \frac{\underline{a}!}{R(n)}\ [\underline{x}^{\underline{a}}]F(\underline{x}), \tag{14}$$

where $F(\underline{x}) = \sum_{(A,B):A\cap B=\emptyset} \frac{1}{2}T_A(\underline{x})T_B(\underline{x})(1-2\sum_{i=1}^{r} x_i)^{-\frac{1}{2}}$.

Rewriting (14), we have

$$\mu_n'(\underline{\phi}) = \frac{n!}{R(n)} \sum_{\underline{a}} \underline{\phi}^{\underline{a}}[\underline{x}^{a}]F(\underline{x}) = \frac{n!}{R(n)}[x^n]F(\underline{\phi}x),$$

as required.

(iii) For any tree $T' \in \mathcal{R}(m)$ with root vertex ρ and subject to a random coloration of $[m]$ according to $\underline{\phi}$, let $S(T')$ denote the (random variable) root set of T' (as defined in section 1).

Suppose $T \in B(n)$ and χ is a coloration of $[n]$. By the bijection $\psi : \mathcal{B}(n) \to \mathcal{R}(n-1)$ and Lemma 1.1 (rooting T on the edge incident with the leaf labeled n), we have

$$\ell(T,\chi) = \ell(\psi(T),\chi') + \delta(T,\chi), \tag{15}$$

where

$$\delta(T,\chi) = \begin{cases} 1 & \text{if } \ \chi(n) \notin S(\psi(T)), \\ 0 & \text{otherwise,} \end{cases}$$

and where χ' is the restriction of χ to $[n-1]$.

Let $\mu(T)$ denote the expected value of $\ell(T,\chi)$ for a random χ (generated according to $\underline{\phi}$). Then from (15) we have

$$\mu(T) = \mu(\psi(T)) + \text{Prob}[\delta(T,\chi)=1]. \tag{16}$$

Now

$$\begin{aligned} \text{Prob}[\delta(T,\chi)=1] &= \text{Prob}[\chi(n) \notin S(\psi(T))] \\ &= \sum_{\substack{i,A:\\ \alpha_i\notin A}} \phi_i \text{Prob}[S(\psi(T))=A]. \end{aligned} \tag{17}$$

Also, by definition,

$$\mu'_n(\underline{\phi}) = \frac{1}{B(n)} \sum_{T\in\mathcal{B}(n)} \mu(T), \tag{18}$$

while

$$\begin{aligned}\mu'_{n-1}(\phi) = \frac{1}{B(n-1)} \sum_{T'\in\mathcal{B}(n-1)} \mu(T') &= \frac{1}{B(n)} \sum_{T'\in\mathcal{B}(n-1)} (2n-3)\mu(T') \\ &= \frac{1}{B(n)} \sum_{T\in\mathcal{B}(n)} \mu(\psi(T)).\end{aligned} \tag{19}$$

Thus, combining (16)–(19) we have

$$\mu'_n(\phi) = \mu'_{n-1}(\underline{\phi}) + \sum_{\substack{i,A:\\ \alpha_i\notin A}} \phi_i \frac{1}{B(n)} \sum_{T\in\mathcal{B}(n)} \mathrm{Prob}[S(\psi(T)) = A]. \tag{20}$$

Now

$$\frac{1}{B(n)} \sum_{T\in\mathcal{B}(n)} \mathrm{Prob}[S(\psi(T)) = A] = \frac{1}{R(n-1)} \sum_{T'\in\mathcal{R}(n-1)} \mathrm{Prob}[S(T') = A]. \tag{21}$$

Also, $n![x^n]T_A(\underline{\phi}x) = \sum_{\underline{a}} \binom{n}{\underline{a}} \underline{\phi}^{\underline{a}} f_A(\underline{a})$, and so

$$\begin{aligned} n![x^n]T_A(\underline{\phi}x) &= \sum_{\underline{a}} \underline{\phi}^{\underline{a}} \sum_{\substack{\chi:\chi \text{ has}\\ \text{type } \underline{a}}} \sum_{\substack{T\in\mathcal{R}(n)\\ S(T,\chi)=A}} 1 \\ &= \sum_{T\in\mathcal{R}(n)} \sum_{\underline{a}} \sum_{\substack{\chi:\chi \text{ has type } \underline{a}\\ \text{and } S(T,\chi)=A}} \underline{\phi}^{\underline{a}} \\ &= \sum_{T\in\mathcal{R}(n)} \sum_{\chi:S(T,\chi)=A} \mathrm{Prob}[\chi] \\ &= \sum_{T\in\mathcal{R}(n)} \mathrm{Prob}[S(T,\chi) = A]. \end{aligned}$$

Thus, the term on the right of (21) is just

$$\frac{(n-1)!}{R(n-1)} [x^{n-1}] T_A(\underline{\phi}x),$$

which, together with (20), establishes part (iii), thereby completing the proof of Theorem 2.1.

3. Calculations (asymptotic). In this section we obtain asymptotic results concerning $\mu'_n(\underline{\phi})$ and $\mu_n(\underline{a})$. Theorem 3.1 below shows that $\mu'_n(\underline{\phi})$ and $\mu_n(\underline{a})$ are asymptotically equivalent since they both grow linearly with n, and their difference (when $\underline{\phi} = \frac{1}{n}\underline{a}$) is bounded by a term of order $n^{\frac{1}{2}}$. The theorem also provides a prescription for calculating, in principle, their asymptotic values by solving a system of simultaneous quadratic equations involving real numbers. In the case of two colors this can be done analytically, but generally numerical techniques would seem to be required. However, in the case of equifrequency colorations ($\phi_i = \frac{1}{r}$) the resulting system is considerably simpler, being of dimension r rather than $2^r - 1$, and we solve this for $r \leq 4$ in Corollary 3.1. A second corollary provides a biology-oriented application.

We adopt the standard notation $f(n) \sim g(n)$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$ and $f(n) = O(g(n))$ if $\frac{f(n)}{g(n)}$ is bounded as $n \to \infty$.

THEOREM 3.1 (asymptotic formulae). (i) *$\mu'_n(\underline{\phi}) \sim \mu' n$, where $\mu' = \mu'(\underline{\phi})$ is given by*

$$\mu' = \sum_{(A,B):A\cap B=\emptyset} t_A t_B$$

and where the numbers $t_A = T_A(\underline{x})|_{\underline{x}=\frac{1}{2}\underline{\phi}}, \emptyset \neq A \in \mathcal{C}$ form the unique nonnegative solution to the simultaneous system

$$t_A = \sum_{(B,C):B*C=A} \frac{1}{2} t_B t_C + \frac{1}{2}\delta_A(\underline{\phi})$$

with δ_A given by (5).

(ii) *For $r = 2$ colors,*

$$\mu' = \frac{2}{3}\left(1 - \sqrt{1 - 3\phi_1\phi_2}\right).$$

(iii) *$\mu_n(\underline{a}) \sim n\mu'(\underline{\phi})$ for $\underline{\phi} = \frac{1}{n}\underline{a}$. Indeed, $|\mu_n(\underline{a}) - \mu'_n(\underline{\phi})| \leq \sqrt{n(r-1)}/2$ for all n.*

Proof of Theorem 3.1. (i) We first recall a special case of Lemma 1(i) of Meir, Moon, and Mycielski [6]: suppose $F(x)$ and $G(x)$ are power series in x and that

$$[x^n]F(x) = O(\rho^{-n} n^{-\frac{3}{2}}),$$

$$[x^n]G(x) \sim b\rho^{-n} n^{-\frac{1}{2}},$$

and $F(\rho) \neq 0$. Then

$$[x^n]F(x)G(x) \sim F(\rho)[x^n]G(x). \tag{22}$$

Taking $G(x) = (1 - 2x)^{-\frac{1}{2}}$ we have from (2) that

$$[x^n]G(x) = \frac{R(n+1)}{n!} = (2n-1)\frac{R(n)}{n!} \sim \frac{1}{\sqrt{\pi}}\rho^{-n} n^{-\frac{1}{2}}, \tag{23}$$

where $\rho = \frac{1}{2}$. Now take $F(x) = T_A(\underline{\phi}x)T_B(\underline{\phi}x)$ for any pair of nonempty sets $A, B \subseteq \mathcal{C}$. Since for all $C \subseteq \mathcal{C}$, $C \neq \emptyset$ the power series $T_C(\underline{\phi}x)$ has all nonnegative coefficients, we have

$$\begin{aligned} |[x^n]F(x)| &= |[x^n]T_A(\underline{\phi}x)T_B(\underline{\phi}x)| \\ &\leq [x^n]\left(\sum_{C\neq\emptyset} T_C(\underline{\phi}x)\right)^2 \end{aligned}$$

$$\begin{aligned}
&= [x^n]R(\underline{\phi}x)^2 \quad \text{from (6)}\\
&= [x^n](2R(\underline{\phi}x) - 2x) \;\; \text{from (8)}\\
&= 2\frac{R(n)}{n!} \quad \text{from (11)}\\
&\sim \frac{1}{\sqrt{\pi}}\rho^{-n}n^{-\frac{3}{2}}
\end{aligned}$$

for $\rho = \frac{1}{2}$ from (2).

Thus $[x^n]F(x) = O(\rho^{-n}n^{-\frac{3}{2}})$, and so, from (23) and the fact that F has nonnegative coefficients (so that $F(\rho) \neq 0$), we can apply (22) to Theorem 2.1 (ii) to deduce that

$$\begin{aligned}
\mu'_n &= \frac{n!}{R(n)} \sum_{(A,B):A\cap B=\emptyset} \frac{1}{2}T_A(\underline{\phi}x)T_B(\underline{\phi}x)_{|_{x=\frac{1}{2}}}[x^n](1-2x)^{-\frac{1}{2}}\\
&\sim n \sum_{(A,B):A\cap B=\emptyset} T_A\left(\frac{1}{2}\underline{\phi}\right)T_B\left(\frac{1}{2}\underline{\phi}\right),
\end{aligned}$$

as claimed.

The prescribed system for $t_A := T_A(\frac{1}{2}\underline{\phi})$ follows from (4) by putting $z = 1$ and $\underline{x} = \frac{1}{2}\underline{\phi}x$.

Now $t_A = T_A(\frac{1}{2}\underline{\phi})$ is clearly nonnegative. Thus, applying induction on $|A|$ to the simultaneous system described in (i) shows that, moreover, $t_A > 0$ for all A. We now show that there is only one such solution to this system.

Let $\underline{t} = [t_A]$ and $\underline{t}' = [t'_A]$ be two solutions of the system of simultaneous quadratic equations described in Theorem 3.1, with $t_A, t'_A > 0$ for all A. We wish to show that $\underline{t} = \underline{t}'$. First, note that $\sum_A t_A = \sum_A t'_A = 1$ since $\sum_A t_A = \frac{1}{2}\left(\sum_A t_A\right)^2 + \frac{1}{2}$.

Let $\epsilon = \sum_A |t_A - t'_A|$. We have

$$\begin{aligned}
\epsilon &= \frac{1}{2}\sum_A \left| \sum_{\substack{B,C:\\ B*C=A}} t_Bt_C - t'_Bt'_C \right|\\
&\leq \frac{1}{2}\sum_A \sum_{\substack{B,C:\\ B*C=A}} |t_Bt_C - t'_Bt'_C|\\
&\leq \frac{1}{2}\sum_A \sum_{\substack{B,C:\\ B*C=A}} t_B|t_C - t'_C| + t'_C|t_B - t'_B|\\
&\text{(since } \;|t_Bt_C - t'_Bt'_C| \leq |t_Bt_C - t_Bt'_C| + |t_Bt'_C - t'_Bt'_C|)\\
&= \frac{1}{2}\sum_{B,C} t_B|t_C - t'_C| + \sum_{B,C} t'_C|t_B - t'_B|\\
&= \frac{1}{2}\epsilon\left(\sum_B t_B + \sum_C t'_C\right)\\
&= \epsilon.
\end{aligned}$$

It follows that both inequalities in the above derivation must, in fact, be equalities. In particular, the second inequality becomes an equality only if, for all $B, C : B*C = A$, $\;t_Bt'_C$ lies between t_Bt_C and $t'_Bt'_C$. Since $\underline{t}$ and $\underline{t}'$ both have positive coordinates we either have

$$t_B \le t'_B \quad \text{and} \quad t_C \le t'_C$$

or

$$t_B \ge t'_B \quad \text{and} \quad t_C \ge t'_C \quad \text{whenever } B * C = A.$$

Thus, if we let

$$\Delta_A := \sum_{\substack{B,C:\\ B*C=A}} (t_B - t'_B)(t_C - t'_C)$$

we have $\Delta_A \ge 0$ with equality precisely if

$$(24) \qquad t_B = t'_B \quad \text{or} \quad t_C = t'_C \quad \text{for each} \quad (B,C): \quad B * C = A.$$

Now $t_A = \frac{1}{2}\sum_{\substack{B,C:\\ B*C=A}} t_B t_C + \frac{1}{2}\delta_A(\underline{\phi})$ and similarly for t'_A, so expanding out Δ_A we obtain

$$\Delta_A = 2t_A - \delta_A(\underline{\phi}) + 2t'_A - \delta_A(\underline{\phi}) - \sum_{\substack{B,C:\\ B*C=A}} (t'_B t_C + t_B t'_C),$$

so

$$\begin{aligned}\sum_A \Delta_A &= 2\sum_A t_A - 1 + 2\sum_A t'_A - 1 - \sum_{B,C}(t'_B t_C + t'_B t'_C)\\ &= 2 - \left(\sum_B t'_B\right)\left(\sum_C t_C\right) - \left(\sum_B t_B\right)\left(\sum_C t'_C\right)\\ &= 0.\end{aligned}$$

It follows that $\Delta_A = 0$ for all A (since $\Delta_A \ge 0$ for all A and $\sum_A \Delta_A = 0$). From (24) this implies that for *any* pair B, C we have

$$t_B = t'_B \quad \text{or} \quad t_C = t'_C.$$

In particular, taking $B = C$ we obtain that $t_B = t'_B$, and so $\underline{t} = \underline{t}'$, as claimed.

(ii) This result follows from part (iii) of Theorem 3.1, and the analogous result for $\mu_n(\underline{a})$ from Moon and Steel [7]. However, it can also be derived more directly from Theorem 3.1 (i). We have, for $\mathcal{C} = \{\alpha, \beta\}$,

$$(25) \qquad \mu' = 2T_{\{\alpha\}}T_{\{\beta\}},$$

where $T_{\{\alpha\}} = T_{\{\alpha\}}(\frac{1}{2}\underline{\phi})$, $T_{\{\beta\}} = T_{\{\beta\}}(\frac{1}{2}\underline{\phi})$, and $T_{\{\alpha,\beta\}} = T_{\{\alpha,\beta\}}(\frac{1}{2}\underline{\phi})$ satisfy the system

$$\begin{aligned}T_{\{\alpha\}} &= \frac{1}{2}T^2_{\{\alpha\}} + T_{\{\alpha\}}T_{\{\alpha,\beta\}} + \frac{\phi_1}{2},\\ T_{\{\beta\}} &= \frac{1}{2}T^2_{\{\beta\}} + T_{\{\beta\}}T_{\{\alpha,\beta\}} + \frac{\phi_2}{2},\\ T_{\{\alpha,\beta\}} &= \frac{1}{2}T^2_{\{\alpha,\beta\}} + T_{\{\alpha\}}T_{\{\beta\}}.\end{aligned}$$

Butler [1] solved this system, and from his equation (26) we have

$$\begin{aligned}T^2_{\{\alpha\}} &= \frac{1}{3}(-2 + 3\phi_1 + 2\sqrt{P}),\\ T^2_{\{\beta\}} &= \frac{1}{3}(1 - 3\phi_1 + 2\sqrt{P}),\end{aligned}$$

where $P = 1 - 3\phi_1\phi_2$, and from this we can obtain μ' directly from (25).

(iii) We first claim that

$$|\mu_n(\underline{a}) - \mu_n(\underline{a}')| \leq \frac{1}{2}|\underline{a} - \underline{a}'|_1, \tag{26}$$

where $|\ |_1$ denotes the l_1 norm on $\mathbf{R}^r$. Since the components of $\underline{a}$ and $\underline{a}'$ both sum to n, $|\underline{a} - \underline{a}'|_1 = 2k$ for some integer k. In this case we can find two colorations $\chi,\ \chi'$ of $[n]$ of types $\underline{a},\ \underline{a}'$, respectively, and such that χ and χ' agree on all but k elements of $[n]$.

Now, for any binary tree T, it is easily checked that $\ell(T, \chi') \leq \ell(T, \chi) + k$ since any minimal extension of χ for T produces an extension χ'' of T by just changing the colors of the (at most k) leaves of T for which χ and χ' disagree (and thereby increasing the number of edges of T with differently colored ends by at most k). Although χ'' may not be a minimal extension of χ' for T, we nevertheless obtain the claimed inequality. Conversely, $\ell(T, \chi) \leq \ell(T, \chi') + k$, and so

$$|\ell(T, \chi) - \ell(T, \chi')| \leq k,$$

which, upon averaging over all binary trees, gives

$$|\mu_n(\underline{a}) - \mu_n(\underline{a}')| \leq k,$$

which establishes (26).

Now from (3),

$$\mu'_n = E[\mu_n(\underline{A})],$$

the expected value of $\mu_n(\underline{A})$, where $\underline{A} = (A_1, \ldots, A_r)$ is drawn from a multinomial distribution with parameters n and $\phi_1, \ldots, \phi_r$.

Then

$$\begin{aligned} |\mu_n(\underline{a}) - \mu'_n| &= |E[\mu_n(\underline{a}) - \mu_n(\underline{A})]| \\ &\leq E[|\mu_n(\underline{a}) - \mu_n(\underline{A})|] \\ &\leq \frac{1}{2}E[|\underline{a} - \underline{A}|_1] \quad \text{from (26)} \\ &= \frac{1}{2}\sum_{i=1}^{r} E[|a_i - A_i|]. \qquad (27) \end{aligned}$$

Now A_i has a binomial distribution with parameters n and ϕ_i, and since

$$E[A_i] = n\phi_i = a_i,$$

we have, applying the convex version of Jensen's inequality (Rényi [8]),

$$\begin{aligned} E[|a_i - A_i|] &\leq \sqrt{E[(a_i - A_i)^2]} \\ &= \sqrt{Var[A_i]} \\ &= \sqrt{n\phi_i(1 - \phi_i)}, \end{aligned}$$

so that, from (27), $|\mu_n(\underline{a}) - \mu'_n(\underline{\phi})| \leq \frac{1}{2}\sum_{i=1}^{r}\sqrt{n\phi_i(1 - \phi_i)}$, which, by the concave version of Jensen's inequality, is at most $\sqrt{n(r-1)}/2$.

Corollary 3.1 (equifrequency colorations). *Suppose $\phi_i = \frac{1}{r}$ for $i = 1, \ldots, r$. Then*

$$\mu' = \sum_{(j,k): j+k \leq r;\ j,k \geq 1} \frac{r!}{j!k!(r-j-k)!} t_j t_k,$$

where the t_i satisfy the system

$$t_i = \sum_{(j,k)} \frac{1}{2} \pi_{ijk} t_j t_k + \delta_{i,1} \frac{1}{2r}$$

for $i = 1, \ldots, r$ and where $\delta_{i,1} = 1$ if $i = 1$ and 0 otherwise and where π_{ijk} is the number of sets of sizes j and k which, under the parsimony operation (*), give a specific root set of size i for the tree; i.e.,

$$\pi_{ijk} = \begin{cases} \binom{i}{j} & \text{if } j+k=i, \\ \binom{r-i}{j-i}\binom{r-j}{k-i} & \text{if } j,k \geq i, \\ 0 & \text{else.} \end{cases}$$

Examples. For $r = 2$ we have

$$\begin{aligned} t_1 &= \frac{1}{2}t_1^2 + t_1 t_2 + \frac{1}{4}, \\ t_2 &= \frac{1}{2}t_2^2 + t_1^2, \end{aligned}$$

which gives $t_1 = \frac{1}{\sqrt{6}}$, $t_2 = 1 - \frac{2}{\sqrt{6}}$, and so from Corollary 3.1 we obtain $\mu' = \frac{1}{3}$, which agrees with Theorem 3.1 (ii). For $r > 2$ it seems necessary to solve the system $\{t_i\}$ by numerical methods. For $r = 3$ and 4, the equations become

$$\begin{aligned} t_1 &= \frac{1}{2}t_1^2 + 2t_1t_2 + t_1t_3 + t_2^2 + \frac{1}{6}, \\ t_2 &= t_1^2 + \frac{1}{2}t_2^2 + t_2t_3, \\ t_3 &= 3t_1t_2 + \frac{1}{2}t_3^2 \end{aligned}$$

and

$$\begin{aligned} t_1 &= \frac{1}{2}t_1^2 + 3t_1t_2 + 3t_2^2 + 3t_1t_3 + 3t_2t_3 + t_1t_4 + \frac{1}{8}, \\ t_2 &= t_1^2 + \frac{1}{2}t_2^2 + 2t_2t_3 + t_3^2 + t_2t_4, \\ t_3 &= 3t_1t_2 + \frac{1}{2}t_3^2 + t_3t_4, \\ t_4 &= 4t_1t_3 + 3t_2^2 + \frac{1}{2}t_4^2, \end{aligned}$$

respectively, and we find that $t = (0.24855, 0.06755, 0.051705)$ and $\mu' = 0.4714$ for $r = 3$ and $t = (0.17656, 0.0339, 0.01843, 0.01660)$ and $\mu' = 0.5507$ for $r = 4$.

As a second and biologically-oriented application of Theorem 3.1, let us, as in section 1, regard a collection of n aligned DNA sequences of length c as a collection

$\chi^1, \ldots, \chi^c$ of r-colorations of $[n]$ (for $r = 2$ or 4). Let $\ell(T)$ denote the length of $T \in \mathcal{B}(n)$ for this data; that is,

$$\ell(T) = \sum_{j=1}^{c} \ell(\chi^j, T),$$

and let $\overline{\ell}$ be the average value of $\ell(T)$ over $\mathcal{B}(n)$. We also consider a randomized version of $\overline{\ell}$ as follows. Let $\ell^*(T)$ be the expected length of a given binary tree T on sequences randomly generated by assigning each of the c sites in sequence i a color α_j with probability ϕ^i_j, as in Steel, Lockhart, and Penny [11]. Let $\overline{\ell}^*$ denote the average value of $\ell^*(T)$ over $B(n)$. Finally, let $\mathcal{P}(n)$ denote the set of partitions of n into at most r-parts (thus $\mathcal{P}(n) = \{(p_1, \ldots, p_r) : p_1 \geq p_2 \geq \cdots \geq p_r \geq 0, \sum_{i=1}^{r} p_i = n\}$).

COROLLARY 3.2. *Asymptotically* (*as* $n \to \infty$),

(i)

$$\overline{\ell} \sim n \sum_{\underline{p} \in \mathcal{P}(n) : N(\underline{p}) > 0} \mu' \left(\frac{1}{n} \underline{p} \right) N\left(\underline{p}\right),$$

where $N(\underline{p})$ *is the number of sites* j *for which the type of* χ^j, *arranged in decreasing order, gives partition* $\underline{p}$.

(ii)

$$\overline{\ell}^* \sim cn\mu'(\underline{\phi}), \quad \text{where} \quad \underline{\phi} = \frac{1}{n} \sum_{i=1}^{n} \underline{\phi}^i.$$

Proof of Corollary 3.2.

(i)

$$\begin{aligned} \overline{\ell} &= \frac{1}{B(n)} \sum_{T \in \mathcal{B}(n)} \ell(T) \\ &= \frac{1}{B(n)} \sum_{T \in \mathcal{B}(n)} \sum_{j=1}^{c} \ell(\chi^j, T) \\ &= \sum_{j=1}^{c} \frac{1}{B(n)} \sum_{T \in \mathcal{B}(n)} \ell(\chi^j, T) \\ &= \sum_{j=1}^{c} \mu_n(\underline{a}^{(j)}), \end{aligned}$$

where $\underline{a}^j$ is the type of χ^j. Thus

$$\overline{\ell} = \sum_{\underline{p} \in \mathcal{P}(n) : N(\underline{p}) > 0} \mu_n(\underline{p}) N(\underline{p}),$$

and the result now follows from parts (i) and (iii) of Theorem 3.1.

(ii)

$$\overline{\ell}^* = cE'[\mu_n(\underline{A})], \tag{28}$$

where E' denotes expectation at a single site in the probability space described above. Since A_j is a sum of n independent (but not necessarily identical) $0-1$ random variables D_{ij}, with $\text{Prob}[D_{ij}=1]=\phi_j^i$, we have that

$$\frac{1}{n}\underline{A} \rightarrow_p \underline{\phi}, \tag{29}$$

where $\rightarrow_p$ denotes convergence in probability (as $n \rightarrow \infty$) and

$$\underline{\phi} = \frac{1}{n}\sum_{i=1}^{n} \underline{\phi}^i.$$

Now from (3), (26), and (29) it can be checked that

$$\left| \frac{1}{n}\mu_n'\left(\frac{1}{n}\underline{A}\right) - \frac{1}{n}\mu_n'(\underline{\phi}) \right| \rightarrow_p 0 \tag{30}$$

as $n \rightarrow \infty$.

Also, by Theorem 3.1, parts (i) and (iii), respectively,

$$\left| \frac{1}{n}\mu_n'(\underline{\phi}) - \mu'(\underline{\phi}) \right| \rightarrow 0,$$

$$\left| \frac{\mu_n(\underline{A})}{n} - \frac{1}{n}\mu_n'\left(\frac{1}{n}\underline{A}\right) \right| \rightarrow_p 0$$

as $n \rightarrow \infty$; thus

$$\frac{\mu_n(\underline{A})}{n} \rightarrow_p \mu'(\underline{\phi}) \tag{31}$$

as $n \rightarrow \infty$.

Now, from (28),

$$\overline{\ell}^* = cnE'\left[\frac{\mu_n(\underline{A})}{n}\right],$$

which, together with (31), establishes part (ii).

REFERENCES

[1] J. P. Butler, *Fractions of trees with given root traits; the limit of large trees*, J. Theoret. Biol., 147 (1990), pp. 265–274.

[2] M. Carter, M. D. Hendy, D. Penny, L.A. Székely, and N. C. Wormald, *On the distribution of lengths of evolutionary trees*, SIAM J. Discrete Math., 3 (1990), pp. 38–47.

[3] W. M. Fitch, *Towards defining the course of evolution: Minimum change for a specific tree topology*, Syst. Zool., 20 (1971), pp. 406–416.

[4] I. P. Goulden and D. M. Jackson, *Combinatorial Enumeration*, John Wiley, New York, 1983.

[5] J. A. Hartigan, *Minimum mutation fits to a given tree*, Biometrics, 29 (1973), pp. 53–65.

[6] A. Meir, J. W. Moon, and J. Mycielski, *Hereditary finite sets and identity trees*, J. Combin. Theory Ser. B, 35 (1983), pp. 142–155.

[7] J. W. Moon and M. A. Steel, *A limiting theorem for parsimoniously bicoloured trees*, Appl. Math. Lett., 6 (1993), pp. 5–8.

[8] A. Rényi, *Probability Theory*, North–Holland, Amsterdam, 1970.

[9] E. Schröder, *Vier combinatorische Probleme*, Zeitschrift für Mathematik und Physik, 15 (1870), pp. 361–376.

[10] M. A. Steel, *Decompositions of leaf-colored binary trees*, Adv. in Appl. Math., 14 (1993), pp. 1–24.

[11] M. A. Steel, P. J. Lockhart, and D. Penny, *Confidence in evolutionary trees from biological sequence data*, Nature, 364 (1993), pp. 440–442.

SIAM J. Discrete Math.
Vol. 10, No. 3, pp. 373–380, August 1997

CLASSES OF GRAPHS THAT ARE NOT VERTEX RAMSEY[*]

H. A. KIERSTEAD[†]

Abstract. Sauer [*Combinatorics*, 1 (1993), pp. 361–377] has conjectured that for any tree T and any clique K, the class Forb(T, K) of graphs that induces neither T nor K is not vertex Ramsey. This conjecture is implied by an even stronger conjecture of Gyárfás and independently by Sumner, that Forb(T, K) is χ-bounded. Until now, for all trees T, if Forb(T, K) was known to not be vertex Ramsey, then Forb(T, K) was also known to be χ-bounded. In this paper we introduce a new class of trees, spiders with toes, which includes all trees T such that Forb(T) is known to be χ-bounded as well as other trees for which it is not known to be χ-bounded. We show that for every spider with toes T, Forb(T, K) is not vertex Ramsey.

Key words. Ramsey, chromatic number, hypergraph

AMS subject classification. 05C35

PII. S0895480194264769

Introduction. For a positive integer c, let $[c]$ denote $\{1, \dots, c\}$. For a set V and a positive integer n, let $2^{[V]}$ denote the collection of subsets of V and $[V]^n$ denote the collection of n element subsets of V. A hypergraph is a pair $G = (V, E)$ such that $E \subset 2^{[V]}$. The elements of V are called vertices and the elements of E are called edges. In the case that $E \subset [V]^2$, G is just a graph and we abbreviate $\{x, y\}$ by xy. If xy is an edge, we say that x is adjacent to y and write $x \sim y$.

A function $f\colon V \to [c]$ is a proper c-coloring of a hypergraph $G = (V, E)$ if no edge of G is monochromatic; i.e., for all edges $e \in E$, $|\{f(x)\colon x \in e\}| > 1$. The chromatic number $\chi(G)$ of G is the least positive integer c such that G has a proper c-coloring. The chromatic number $\chi(\Gamma)$ of a class of hypergraphs Γ is the least positive integer c such that $\chi(G) \leq c$ for all hypergraphs $G \in \Gamma$, provided that such an integer exists; otherwise $\chi(\Gamma) = \infty$. If $\chi(\Gamma) < \infty$, we say that $\chi(\Gamma)$ is finite. A class of hypergraphs Γ is χ-bounded if there exists a function f such that for all hypergraphs $G \in \Gamma$, $\chi(G) \leq f(\omega(G))$, where $\omega(G)$ is the size of the largest complete subgraph of G.

Let $G = (V, E)$ be a graph and let $S \subset V$. The graph induced by S in G is the graph $G[S] = (S, E')$, where $E' = \{xy \in E\colon x, y \in S\}$. Another graph H is an induced subgraph of G if there exists $S \subset V$ such that H is isomorphic to $G[S]$. We call $G[S]$ an induced copy of H in G. Let Forb($H_1, \dots, H_n$) denote the class of graphs G such that none of the graphs $H_1, \dots, H_n$ are induced subgraphs of G. We denote $G[V - S]$ by $G - S$.

This article is motivated by the following conjecture due to Gyárfás [2] and independently Sumner [9].

Conjecture 1. *For every tree T,* Forb(T) *is χ-bounded.*

Let K_k denote the complete subgraph on k vertices. For our purposes it is convenient to reformulate Conjecture 1 as follows: for every tree T and every positive integer k, χ(Forb(T, K_k)) is finite. Erdös and Hajnal [1] have shown that there are graphs with arbitrarily large girth and chromatic number. Thus if H contains a cycle, then χ(Forb(H, K_3)) is infinite. It is also not hard to show that for a forest F with component trees $T_1, \dots, T_n$, χ(Forb(F, K_k)) is finite iff χ(Forb(T_i, K_k)) is finite for

[*] Received by the editors March 1, 1994; accepted for publication (in revised form) May 31, 1996. This research was partially supported by Office of Naval Research grant NOOO14-90-J-1206.
http://www.siam.org/journals/sidma/10-3/26476.html

[†] Department of Mathematics, Arizona State University, Tempe, AZ 85287 (kierstead@asu.edu).

all $i \in [n]$. Thus if true, the conjecture yields the best possible result. The conjecture has been shown to hold for certain trees. Gyárfás [3] proved that for all brooms B (the result of identifying the center of a star and a leaf of a path) $\mathrm{Forb}(B)$ is χ-bounded. A tree formed by identifying one leaf of each path in a collection of paths is called a spider. Of course, brooms are special cases of spiders. Very recently, Scott [8] proved that if T is a spider, then $\mathrm{Forb}(T)$ is χ-bounded. Spiders are the only trees T with radius greater than two such that $\mathrm{Forb}(T)$ is known to be χ-bounded. Gyárfás, Szemerédi, and Tuza [4] proved that for any radius two tree T, $\chi(\mathrm{Forb}(T, K_3))$ is finite. Kierstead and Penrice [6] later pushed their techniques further to show that for any radius two tree T, $\mathrm{Forb}(T)$ is χ-bounded.

Recently, working from the direction of the Ramsey theory, Sauer [7] introduced a weaker version of Conjecture 1. Before presenting Sauer's conjecture we must introduce some new concepts. A class of graphs Γ is said to be *vertex Ramsey* if for all integers c and graphs $G \in \Gamma$, there exists a graph $H = (V, E) \in \Gamma$ such that for all functions $f \colon V \to [c]$, there exists an induced monochromatic copy of G in H. Sauer's conjecture is the following.

CONJECTURE 2. *For every tree T with at least two edges and every positive integer $k \geq 3$, $\mathrm{Forb}(T, K_k)$ is not vertex Ramsey.*

Note that $K_2 = P_2$ and $\mathrm{Forb}(K_2)$ is the class of graphs with no edges. Thus by the pigeonhole principle, $\mathrm{Forb}(G, K_2) = \mathrm{Forb}(P_2, G)$ is vertex Ramsey for any graph G. The only tree with less than two edges is P_2. Thus the two restrictions in the conjecture are necessary.

Next we explore the connection between the two conjectures. For graphs G and $H = (V, E)$, let $\Lambda_G(H)$ be the hypergraph (V, E_G) such that $e \in E_G$ iff $H[e]$ is a copy of G. Let $\chi_G(H) = \chi(\Lambda_G(H))$. For a class of graphs Γ, let $\chi_G(\Gamma)$ be the least upper bound on $\chi_G(H)$ for $H \in \Gamma$. We say that f is a G-proper coloring of H if f is a proper coloring of $\Lambda_G(H)$. In other words, H does not contain a monochromatic induced copy of G. Similarly, H is G-critical if $\Lambda_G(H)$ is critical. We have the following reformulation.

PROPOSITION 0.1. *A class of graphs Γ is not vertex Ramsey iff there exists $G \in \Gamma$ such that $\chi_G(\Gamma)$ is finite.*

Notice that if $G = K_2$, then $\chi_G(H) = \chi(H)$. Also, if G is an induced subgraph of G', then for any graph H, $\chi_{G'}(H) \leq \chi_G(H)$. Thus using the proposition it is easy to see that Conjecture 1 implies Conjecture 2.

In this paper we introduce a new class of trees, called spiders with toes, and prove (Theorem 1) that for all spiders with toes S and all positive integers k, $\mathrm{Forb}(S, K_k)$ is not vertex Ramsey. The class of spiders with toes includes all trees T such that $\mathrm{Forb}(T)$ is known to be χ-bounded. It also includes the only examples of trees for which Conjecture 2 is known to hold, but Conjecture 1 is not known to hold. The proof is of technical interest because it uses the same template technique developed in [4], [5], and [6] but with much more general templates. The author strongly believes that some extension of this technique is likely to provide a proof of at least Conjecture 2.

Let P_r denote the path on r vertices and S_d denote the star on $d+1$ vertices. The root of P_r is defined to be one of its two leaves, while the root of S_d is defined to be its unique nonleaf. The (d, r)-*broom* $B_{d,r}$ is formed by identifying a leaf of P_r with the root of S_d. Thus the longest path from the root of $B_{d,r}$ has $r+1$ vertices. The remaining leaf of P_n is the root of $B_{d,r}$, except that in the case $r = 1$, the only leaf of P_1 is the root of $B_{d,1}$. Note that $B_{d,1} = S_d$ and that, except for their roots, S_{d+1} is the same as $B_{d,2}$. See Figure 1. A *spider* is the subdivision of a star or, alternatively,

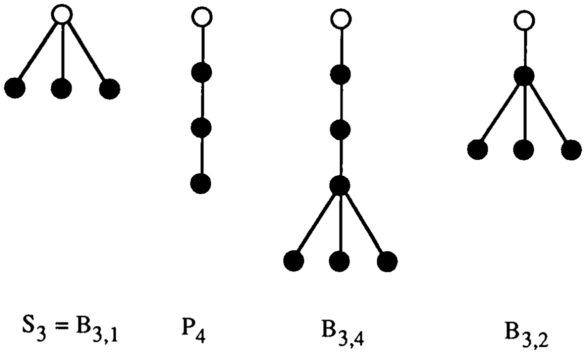


FIG. 1.

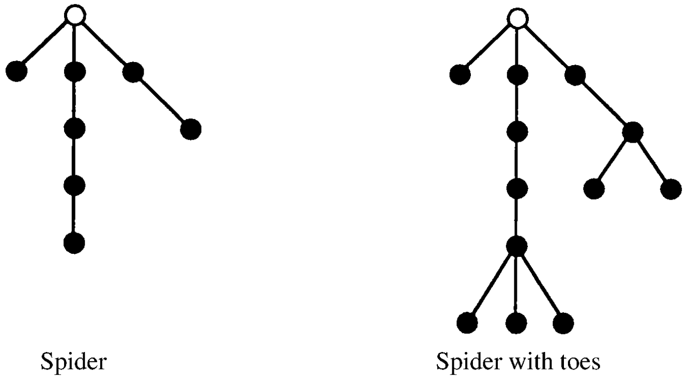


FIG. 2.

the result of identifying the roots of a collection of disjoint paths. A *spider with toes* S is the result of identifying the roots of a collection of disjoint brooms. The root v^* of S is the new vertex obtained by this identification. See Figure 2. A leg of S is a component of $S - v^*$.

Let $G = (V, E)$ be a graph, $v \in V$, and $S \subset V$. The open neighborhood $N(v)$ of v is the set $N(v) = \{u \in V\colon\ v \sim u\}$, and the closed neighborhood of v is the set $N[v] = N(v) \cup \{v\}$. The degree $\delta(v)$ of v is $|N(v)|$, and $\delta(v, S)$ denotes $|N(v) \cap S|$. Finally, let $R(k, b)$ denote the Ramsey function such that any graph on $R(k, b)$ vertices contains a clique of size k or an independent set of size b.

1. The template lemma. In this section we present some preliminary propositions and lemmas.

PROPOSITION 1.1. *If G and H are graphs such that $\chi_G(H) = q$ and G is connected, then for some connected component H' of H, $\chi_G(H') = q$.*

Proof. Since G is connected, no edge of $\Lambda_G(H)$ can contain vertices from two distinct components of H. Thus $\chi_G(H) = \max \chi_G(H')$ over all components H' of H. □

Lemma 1.1. *Let G and $H = (V, E)$ be graphs such that G is connected and $\chi_G(H) = c > ds$. If $S \subset V$ satisfies both $\chi_G(H[S]) \leq s$ and $\chi_G(H - S) < \chi_G(H)$, then there exists $v \in S$ such that $\delta(v, V - S) \geq d$.*

Proof. Suppose that for all $v \in S$, $\delta(v, V - S) < d$. By hypothesis, there exists a G-proper $(c - 1)$-coloring f of $H[V - S]$. Also there exists a G-proper s-coloring g of $H[S]$. For any $x \in S$, let $i(x)$ be the least nonnegative integer i such that $is + g(x) \neq f(y)$ for any $y \in N(x) \cap (V - S)$. Note that $i(x) < d$, since $\delta(v, V - S) < d$. Define a coloring h of H by $h(x) = i(x)s + g(s)$ if $x \in S$, and $h(x) = f(x)$ if $x \notin S$. Then $h(v) \leq ds < c$ for all $v \in V$.

We shall obtain a contradiction by showing that h is a G-proper coloring of H. The only possible monochromatic edges of $\Lambda_G(H)$ are edges that contain vertices from both S and $V - S$. However, since G is connected, any such edge e contains two vertices $u \in V - S$ and $v \in S$ such that $u \sim v$. By our choice of i, $h(u) \neq h(v)$, and thus e is not monochromatic. □

Lemma 1.2 (*the template lemma*). *There exists a function $q(r, d, k, s, t)$ such that for all nontrivial connected graphs G and $H = (V, E)$, for all subsets $X \subset V$, and for all vertices $v \in V - X$, if*

$$\chi_G(H - (N[v] - X)) < \chi_G(H), \tag{1}$$

$$\omega(H) < k, \tag{2}$$

$$|X| \leq t, \tag{3}$$

$$\chi_G(H[N[u]]) \leq s, \text{ for all } u \in V, \text{and} \tag{4}$$

$$\chi_G(H) > q(r, d, k, s, t), \tag{5}$$

then $H - X$ contains a copy of $B_{d,i}$ with root v for all $2 \leq i \leq r$.

Proof. We define q and show that the definition works by induction on r. Since q will be increasing in r, it suffices to find an induced $B_{d,r}$ with root v. First consider the base step $r = 2$. Let $q(2, d, k, s, t) = s(R(k, d) + t)$. Then, by Lemma 1 with $S := N[v] - X$, there exists a vertex $v' \in N[v] - X$ such that $\delta(v', V - (N[v] - X)) \geq R(k, d) + t$. Since $|X| \leq t$ and $\omega(H) < k$, $N(v') - (N[v] \cup X)$ contains an independent set L of size d. Clearly, $H[L \cup \{v, v'\}] \approx B_{d,2}$.

Next consider the induction step. For $0 \leq i \leq r$ define M_i and N_i inductively by $M_0 = \{v\} = N_0, M_{i+1} = N(M_i) - (N_i \cup X)$, and $N_{i+1} = N_i \cup M_{i+1}$. Note that if $v' \in M_i$ then there exists a path $P \approx P_{i+1}$ in N_i from v to v' such that $|P \cap M_j| = 1$ for all $j \leq i$. Let $q(r, d, k, s, t) = p(R(k, d) + t)$, where $p = s + \sum_{2 \leq i < r} q(i, d, k, s, t)$.

Case 1. There exists $i \in \{2, 3, \ldots, r - 1\}$ such that $\chi_G(H[M_i]) > q(r - i + 1, d, k, s, t)$. Choose the least such i. Let $D \subset M_i$ be such that $\chi_G(H[D]) > q(r - i + 1, d, k, s, t)$ and $H[D]$ is G-critical. Note that $D \cap X = \emptyset$. Let v' be any vertex in M_{i-1} that is adjacent to some vertex in D. Since $H[D]$ is G-critical and $N(v') \cap D \neq \emptyset$, $\chi_G(H[(D \cup \{v'\}) - N[v']]) < \chi_G(H[D \cup \{v'\}])$. By the induction hypothesis with H replaced by $H[D \cup \{v'\}]$ and v replaced by v', there exists $S \subset D \cup \{v'\}$ such that $H[S] \approx B_{d,r-i+1}$ with root v'. By the above remark, there exists an induced path $P \approx P_i$ from v to v' in N_{i-1} such that $H[P \cup S] \approx B_{d,r}$. Since $X \cap (P \cup S) = \emptyset$, we are done.

Case 2. For all $i \in \{2, 3, \ldots, r - 1\}$, $\chi_G(H[M_i]) \leq q(r - i + 1, d, k, s, t)$. Then $\chi_G(H[N_{r-1}]) \leq p$. Recall that $p(R(k, d) + t) = q(r, d, k, s, t)$. By hypothesis, $\chi_G(H[V-$

$N_{r-1}]) < \chi_G(H)$ and $q(r,d,k,s,t) < \chi_G(H)$. By Lemma 1 with $S := N_{r-1}$, there exists $v' \in N_{r-1}$ such that $d(v', V - N_{r-1}) \geq R(k,d) + t$. Since $|X| \leq t$ and $\omega(H) < k$, v' is adjacent to a set $I \subset V - (X \cup N_{r-1})$ of d independent vertices. Clearly, $v' \in M_{r-1}$. So there exists an induced path $P \approx P_r$ from v to v' in N_{r-1} such that $H[P \cup I] \approx B_{d,r}$. □

We note for later use that q is increasing in all arguments. We shall also need the following easy proposition.

Proposition 1.3. *Let D be a digraph on n vertices with maximum out degree at most s. Then $\chi(D) \leq 2s+1$. In particular, D contains an independent set of size $\lceil n/(2s+1) \rceil$.* □

2. The main lemma and theorem. In this section we state and prove our main result.

Theorem 2.1. *For every spider with toes S with at least two edges, and every positive integer $k \geq 3$, $\mathrm{Forb}(S, K_k)$ is not vertex Ramsey.*

Proof. Let S be a spider with toes on r vertices. Let $B_1, \ldots, B_b$ be the legs of S and v^* be the root of S. Let Γ_k denote $\mathrm{Forb}(S, K_k)$. We shall show by induction on k that there exist a connected $G' \in \Gamma_k$ and a number s' such that $\chi_{G'}(H) \leq s'$ for all $H \in \Gamma_k$. Thus, by Proposition 1, Γ_k is not vertex Ramsey.

If $k = 3$ (base step), let $G = K_2$. Since S has at least two edges, $G \in \Gamma_3$. Also, G is connected, and $\chi_G(H) \leq 1$ for all $H \in \Gamma_2$. If $k > 3$ (induction step), then by the induction hypothesis there exists $G \in \Gamma_{k-1}$ and a number s such that G is connected, and $\chi_G(G') \leq s$ for all $G' \in \Gamma_{k-1}$. In either case $G \in \Gamma_k$. If $\chi_G(H) \leq q = q(r,r,k,s,r^2R(k,b))$ for all $H \in \Gamma_k$, then we finish by setting $G' = G$ and $s' = q$. Otherwise, there exists $G' \in \Gamma_k$ such that $\chi_G(G') > q$. In this case the following technical lemma completes the proof.

Lemma 2.3. *If $G \in \Gamma_k$ is connected and $\chi_G(H) \leq s$ for all $H \in \Gamma_{k-1}$, but there exists a connected $G' \in \Gamma_k$ such that $\chi_G(G') > q = q(r,r,k,s,r^2R(k,b))$, then there exists s' such that $\chi_{G'}(H) \leq s'$ for all $H \in \Gamma_k$.*

Proof. Fix k. Define $d_1 < d_2 < d_3 < d_4 < d_5$ as follows. Let $d_1 = R(k,r)$, $d_2 = rd_1$, $d_3 = b + b^2R(k,b)$, $d_4 = d_3 + d_3^2R(k,rd_3)$, and $d_5 = d_3 + d_4 + (d_3 + d_4)^2R(k,(d_3+d_4)rb)$. For a vertex $v \in V$ and a subset $W \subset V$, we say that v is adjacent, strongly adjacent, and very strongly adjacent to W if v is adjacent to at least 1, d_1, and d_2 vertices in W, respectively. Similarly, $W' \subset V$ is adjacent to W if some element of W' is adjacent to W.

Choose G' as in the hypothesis so that G' is also G-critical. Then by Proposition 2, G' is connected. Recursively partition the vertices of $H = (V,E)$ into sets $Y_1, \ldots, Y_n$, L as follows. Set $Y_0 = \emptyset$. Now suppose we have constructed $Y_1, \ldots, Y_i$. Let $V_i = V - (Y_1 \cup \cdots \cup Y_i)$. If $H[V_i]$ does not induce G', then set $i = n$ and $V_i = L$. Otherwise there exists $Z_{i+1} \subset V_{i+1}$ such that $H[Z_{i+1}] \approx G'$. Set $N_{i+1} = \{v \in V_i\colon v$ is strongly adjacent to $Z_{i+1}\}$ and $Y_{i+1} = Z_{i+1} \cup N_{i+1}$. Finally, let $N = N_1 \cup \cdots \cup N_n$ and $Z = Z_1 \cup \cdots \cup Z_n$.

We shall call the Z_1's templates. Note that if $i < j$, then no vertex in Z_j is strongly adjacent to Z_i. Also, it is easy to see, using the fact that G' is G-critical and Ramsey's theorem, that for any vertex v either in Z_i or strongly adjacent to Z_j, there exists an induced $B_{r,1}$ with root v in $\{v\} \cup Z_j$. For $v \in V$, let $VSN(v) = \{j\colon v$ is very strongly adjacent to $Z_j\}$.

Claim 0. *For all $v \in V$, $|VSN(v)| < b$.*

Proof. Suppose Claim 0 is not true. Let $VSN(v) = \{j_1 > \cdots > j_b\}$. We shall show by induction on $h \leq b$ that there exist $A_1, \ldots, A_h$ such that for all $i \in [h]$,

$A_i \subset Z_{j_i}$, $H[A_i \cup \{v\}] \approx S[B_i \cup \{v^*\}]$ with v mapped to v^*, and A_i is not adjacent to A_j if $i \neq j$. Thus we will obtain the contradiction $H[\{v\} \cup \bigcup_{1 \leq i \leq b} A_i] \approx S$.

The base step $h = 0$ is trivial, so consider the induction step $h \geq 1$. Let $X = \{x \in Z_{j_h}\colon$ x is adjacent to $\bigcup_{1 \leq i < h} A_i\}$. Since no vertex in Z_{j_i}, $i < h$ is strongly adjacent to Z_{j_h}, $|X| < rd_1 = d_2$. Since v is very strongly adjacent to Z_{j_h}, $(Z_{j_h} - X) \cap N(v) \neq \emptyset$. If $|B_i| = 1$, we are done. Otherwise we would like to apply Lemma 2, with H replaced by $H[Z_{j_h} \cup \{v\}]$. Since $|X| < d_2$, (3) holds. Since G' is G-critical, $\chi_G(H[(Z_{j_h} \cup \{v\}) - (N[v] - X)]) < \chi_G(G')$, so (1) holds. Since $H[N(u)] \in \Gamma_{k-1}$, $\chi_G(H[N(u)]) \leq s$ for all $u \in V$. Thus (4) holds. By hypothesis, $\chi_G(G') > q(r, r, k, s, rd_2)$, so (5) holds. Thus by Lemma 2 we can find $A_h \subset Z_{j_h} - X$ such that $H[A_h \cup \{v\}] \approx S[B_h \cup \{v\}]$, with v mapped to v^*. □

For all $v \in V$, let $J(v) = \{j\colon$ v is adjacent to $Z_j\}$.

Claim 1. *For all $v \in V$, $|J(v)| < d_3 = b + b^2R(k, b)$.*

Proof. Suppose Claim 1 is not true. For all $j \in J(v)$, choose $v_j \in Z_j$ such that $v \sim v_j$. By Claim 0, each of the vertices v, and v_j with $j \in J(v)$ is very strongly adjacent to less than b templates. Thus there exists $J \subset J(v)$ such that $|J| = b^2R(k, b)$ and v is not very strongly adjacent to Z_j for all $j \in J$. By Proposition 3, there exists $J' \subset J$ such that $|J| = R(k, b)$ and v_i is not very strongly adjacent to Z_j for all distinct i, $j \in J'$. (Define a digraph on J by $i \rightarrow j$ iff v_i is very strongly adjacent to Z_j.) By Ramsey's theorem and $\omega(H) < k$, there exists $J'' \subset J'$ such that $|J''| = b$ and $\{v_j\colon j \in J''\}$ is independent. Let $J'' = \{j_1 > \cdots > j_b\}$.

We shall show by induction on $h \leq b$ that there exist $A_1, \ldots, A_h$ such that for all $i \in [h]$, $A_i \subset Z_{j_i}$, $H[A_i] \approx S[B_i]$, v_{j_i} is the root of $H[A_i]$, A_i is adjacent to neither A_j nor v_{j_m} for all distinct i, $j \in [h]$ and $h < m \leq b$, and v is not adjacent to A_i for all $i \in [h]$. Thus we will obtain the contradiction $H[\{v\} \cup \bigcup_{1 \leq i \leq d_0} A_i] \approx S$.

The base step $h = 0$ is trivial, so consider the induction step $h \geq 1$. Let $X = ((N(v) \cap Z_{j_h}) - \{v_{j_h}\}) \cup \{x \in Z_{i_h}\colon$ x is adjacent to $(\{v_{j_{h+1}}, \ldots, v_{j_b}\} \cup \bigcup_{1 \leq i < h} A_i)\}$. Note, using the induction hypothesis and the definition of J'', that $v_{j_h} \notin X$. We would like to apply Lemma 2 with H replaced by $H[Z_{j_h}]$ and v replaced by v_{j_h}. Since $X \subset Z_{j_h}$ and every vertex in X is adjacent to one of less than r vertices, none of which are very strongly adjacent to Z_{j_h}, $|X| < rd_2$. Thus (3) holds. Since G' is G-critical, (1) holds. Clearly (2), (4), and (5) hold. Thus by Lemma 2, we can find $A_h \subset Z_{j_h} - X$ such that $H[A_h] \approx B_h$ with root v_{j_h}. □

For all $v \in V$, let $Q(v) = \{j\colon$ v is adjacent to $N_j\}$.

Claim 2. *For all $v \in V$, $|Q(v)| < d_4 = d_3 + d_3^2R(k, rd_3)$.*

Proof. Suppose Claim 2 is not true. By Claim 1 there exists $Q \subset Q(v)$ such that $|Q| \geq d_3^2R(k, rd_3)$ and for all $j \in Q$, v is not adjacent to Z_j. For all $j \in Q$, choose $w_j \in N_j$ such that $v \sim w_j$. By Claim 1 and Proposition 3, there exists $Q' \subset Q$ such that $|Q'| \geq R(k, rd_3)$ and w_i is not adjacent to any Z_j for any two distinct i, $j \in Q'$. By Ramsey's theorem there exists $Q'' \subset Q'$ such that $|Q''| = rd_3$ and $\{w_j\colon j \in Q''\}$ is independent.

We shall show by induction on $h \leq b$ that there exist $j_1, \ldots, j_h \in Q''$ and $A_1, \ldots, A_h$ such that for all $i \in [b]$, $A_i \subset Z_{j_i} \cup \{w_{j_i}\}$, $H[A_i] \approx B_i$, the root of $H[A_i]$ is w_{j_i}, and A_i is not adjacent to A_j if $i \neq j$. Thus we obtain the contradiction $H[\{v\} \cup \bigcup_{1 \leq i \leq b} A_i] \approx S$.

The base step $h = 0$ is trivial, so consider the induction step $h \geq 1$. Let j_h be any index in $Q^* = Q'' - \{j\colon$ Z_j is adjacent to A_i for some $i < h\}$. By Claim 1, $|Q^*| \geq (b - h + 1)d_3 > 0$, so j_h exists. If B_h is a star, then we can find A_h using the fact that w_{j_h} is strongly adjacent to Z_{j_h}. Otherwise, A_h exists, as above, by Lemma 2 with H replaced by $H[Z_{j_h} \cup \{w_{j_h}\}]$, v replaced by w_{j_h}, and X replaced by $\emptyset$. □

For all $v \in V$, let $P(v) = \{j\colon v$ is adjacent to a vertex which is strongly adjacent to $N_j\}$.

CLAIM 3. *For all $v \in V$, $|P(v)| < d_5 = d_3 + d_4 + (d_3 + d_4)^2 R(k, (d_3 + d_4)rb)$.*

Proof. Suppose Claim 3 is not true. By Claims 1 and 2 there exists $P \subset P(v)$ such that $|P| \geq (d_3 + d_4)^2 R(k, (d_3 + d_4)rb)$ and for all $j \in P$, v is not adjacent to any vertex in Y_j. For all $j \in P$, choose w_j such that both $w_j \sim v$ and w_j is strongly adjacent to N_j. By Claims 1 and 2 and Proposition 3, there exists $P' \subset P$ such that $|P'| \geq R(k, (d_3 + d_4)rb)$ and w_i is not adjacent to any vertex in Y_j for any two distinct i, $j \in P'$. By Ramsey's theorem there exists $P'' \subset P'$ such that $|P''| = (d_3 + d_4)rb$ and $\{w_j\colon j \in P''\}$ is independent.

We shall show by induction on $h \leq b$ that there exist $j_1, \ldots, j_h \in P''$ and $A_1, \ldots, A_h$ such that for all $i \in [b]$, $A_i \subset Y_{j_i} \cup \{w_{j_i}\}$, $H[A_i] \approx B_i$, w_{j_i} is the root of $H[A_{j_i}]$, and A_i is not adjacent to A_j if $i \neq j$. Thus we obtain the contradiction that $H[\{v\} \cup \bigcup_{1 \leq i \leq h} A_i] \approx S$. The base step $h = 0$ is trivial, so consider the induction step $h > 0$. Let j_h be any index in $P^* = P'' - \{j\colon Y_j$ is adjacent to A_i for some $i < h\}$. By Claims 1 and 2, $|Q^*| \geq (r - h + 1)(d_3 + d_4)b > 0$, and thus j_h exists.

Let B'_h be B_h with its root deleted. If B'_h is a broom with root v^{**}, let $v' \in N_{j_h}$ such that $w_{j_h} \sim v'$. Then by Lemma 2 with G replaced by G, H replaced by $H[Z_{j_h} \cup \{v'\}]$, v replaced by v', and X replaced by $\emptyset$, there exists $A'_h \subset Z_{j_h} \cup \{v'\}$ such that $H[A'_h] \approx S[B'_i]$ with v' mapped to v^{**}. Then we finish by setting $A_i = A'_i \cup \{w_{j_h}\}$. Otherwise B'_h is an independent set of size less than r. Since w_{j_h} is strongly adjacent to N_{j_h}, w_{j_h} has at least $R(k, r)$ neighbors in N_{j_h}. Thus by Ramsey's theorem we can find the desired A_h.

We complete the proof of Lemma 3 by defining a G'-proper coloring f of H, using $s' = 1 + pd_2 + ps(2d_5 + 1)d_1 d_4$ colors, where $p = |G'|$. The coloring f will use disjoint sets of 1, pd_2, and $ps(2d_5 + 1)d_1 d_4$ colors on L, Z, and N, respectively. Moreover, $f|Z$ will be a proper coloring of $H[Z]$ and $f|N$ will be a G-proper coloring of N.

First, for all vertices $x \in L$, let $f(x) = 0$. Since $H[L]$ does not induce G', $f|L$ is G'-proper. Let $Z_i = \{z_i^1, \ldots, z_i^p\}$ for $i = 1, \ldots, p$. Define $m(i, j)$ by recursion on i so that $m(i, j)$ is the least positive integer such that if $z_i^j \sim z_{i'}^j$ and $i' < i$, then $m(i, j) \neq m(i, j')$. Let $f(z_i^j) = (i, m(i, j))$. Clearly, $f|Z$ is a proper coloring of Z. By Claim 1, $m(i, j) \leq d_2$. Thus $f|Z$ uses at most pd_2 colors.

It remains to color N. Let $N_i^j = \{v \in N_i\colon j$ is the least index such that $v \sim z_i^j\}$. Since $N_i^j \subset N(z_i^j)$, $H[N_i^j] \in \Gamma_{k-1}$. Thus $\chi_G(H[N_i^j]) \leq s$. Let f^j be a G-proper s-coloring of $H[N_i^j]$. Define an auxiliary digraph $D_j = ([n], E_j)$ by $i \to i'$ iff there exists $w \in N_i^j$ such that w is strongly adjacent to $N_{i'}$. By Claim 3, the out degree of D_j is less than d_5. By Proposition 3, there exists a proper $(2d_5 + 1)$-coloring g_j of D_j. For $v \in N$, let $f'(v) = (j, f^j(v), g_j(i))$, where $v \in N_i^j$. Then $f'|N_i$ is a G-proper coloring that uses at most $ps(2d_5 + 1)$ colors. By recursion on i define $m(v)$ to be the least integer m such that if $v \sim v'$, $f'(v) = f'(v')$, $v \in N_i$, $v' \in N_{i'}$, and $i' < i$, then $m(v) \neq m(v')$. Finally, let $f(v) = (f'(v), m(v))$.

Clearly, $f|N$ is a G-proper coloring. It remains to show that for all $v \in N$, $m(v)$ is at most $d_1 d_4$, and thus f uses at most $ps(2d_5 + 1)d_1 d_4$ colors. Suppose $v \in N_i^j$. Let $U = \{v' \in N\colon v \sim v', f'(v) = f'(v'), v' \in N_{i'}$ and $i' < i\}$. It suffices to show that $|U| < d_1 d_4$. Let $I = \{i' < i\colon v' \in N_{i'}^j$ for some $v' \in U\}$. By Claim 2, $|I| < d_4$.

Note that for all $i \in I$, v is not strongly adjacent to N_i: otherwise, i is adjacent to i' in D_j; thus $g_j(i) \neq g_j(i')$ and $f'(v) \neq f'(v')$. Thus $|U| < d_1|I| < d_1 d_4$. □

The proof of Lemma 3 is an example of the general template method that is explained more formally in [5] and used in [4] and [6]. The novelty of this application is that the form of the templates is never actually determined. Rather we establish the existence of the graphs G' that have the properties required of templates.

REFERENCES

[1] P. Erdös and A. Hajnal, *On chromatic number of graphs and set systems*, Acta Math. Sci. Hung., 17 (1966), pp. 61–99.

[2] A. Gyárfás, *On Ramsey covering-numbers*, Colloq. Math. Soc. János Bolyai, 10 (1975), pp. 801–816.

[3] A. Gyárfás, *Problems from the world surrounding perfect graphs*, Zastos. Mat., XIX (1985), pp. 413–441.

[4] A. Gyárfás, E. Szemerédi, and Zs. Tuza, *Induced subtrees in graphs of large chromatic number*, Discrete Math., 30 (1980), pp. 235–244.

[5] H. Kierstead, *Long stars specify weakly χ-bounded classes*, Colloq. Math. Soc. János Bolyai 60: Sets, Graphs, and Numbers, (1991), pp. 421–428.

[6] H. Kierstead and S. Penrice, *Radius two trees specify χ-bounded classes*, J. Graph Theory, 18 (1994), pp. 119–129.

[7] N. Sauer, *Vertex partition problems*, Combinatorics, Paul Erdös is eighty, 1 (1993), pp. 361–377.

[8] A. D. Scott, *Induced Trees in Graphs of Large Chromatic Number*, 1993, manuscript.

[9] D. P. Sumner, *Subtrees of a graph and chromatic number*, in The Theory and Applications of Graphs, G. Chartrand, ed., John Wiley & Sons, New York, 1981, pp. 557–576.

SIAM J. DISCRETE MATH.
Vol. 10, No. 3, pp. 381–387, August 1997

NEW RAMSEY BOUNDS FROM CYCLIC GRAPHS OF PRIME ORDER*

NEIL J. CALKIN[†], PAUL ERDŐS[‡], AND CRAIG A. TOVEY[§]

Abstract. We present new explicit lower bounds for some Ramsey numbers. All the graphs are cyclic and are on a prime number of vertices. We give theoretical motivation for searching for Ramsey graphs of prime order and provide additional computational evidence that primes tend to be better than composites.

Key words. Ramsey, cyclic graphs, probabilistic bounds

AMS subject classifications. 05D10, 05C55, 05C80

PII. S0895480196298378

1. Introduction.

A red–blue coloring of the edges of the complete graph K_n (which we will regard as having vertex set $\{0, 1, 2, 3, \dots, n-1\}$) is *cyclic* if it is invariant under the rotation $i \rightarrow i+1 \pmod n$. For integers $k, l \geq 2$, define the *cyclic Ramsey number* $C(k, l)$ to be the least N so that for all $n \geq N$, every cyclic coloring of K_n contains either a red K_k or a blue K_l. Clearly, $C(k, l) \leq R(k, l)$. We note, however, that not every $n < C(k, l)$ is such that there exists a cyclic coloring without a red K_k or a blue K_l.

Many authors have searched for lower bounds for Ramsey numbers amongst cyclic graphs, and most of the best known explicit lower bounds come either from cyclic graphs or from cyclic graphs together with a small number of additional vertices.

In this paper we present cyclic graphs which improve the previously best known bounds for the classical Ramsey numbers $R(4, 12)$, $R(4, 15)$, $R(5, 7)$, and $R(5, 9)$. All these graphs are of prime order. We were motivated to search for such graphs by theoretical considerations, which we present in the final section.

2. New bounds.

The graphs were found by implicit enumeration of cyclic 2-colorings. The program was written in Pascal and run on Sun SPARCstations (2, 10, or 20). We emphasize that the algorithm is straightforward and the hardware unexceptional even by 1991 standards. The advantage that we had was knowing to look at graphs of prime order. We suspect that in the past when a complete search revealed no cyclic Ramsey graphs of order n or $n+1$ that researchers did not continue the search over larger orders. We searched for cyclic graphs of order equal to the smallest prime greater than or equal to the best-known bound. The required CPU times varied from 25 minutes (for $R(5, 7)$) to 10 days (for $R(4, 15)$).

$R(5, 7) \geq 80$. This improves on the bound of 76 reported in [4]. In the 79 vertex graph, the following edge differences are present: $6, 10, 12, 14, 17, 20, 21, 22, 24, 25, 26, 28, 34, 36, 37, 38$. There is no such cyclic graph on 83 vertices.

*Received by the editors February 9, 1996; accepted for publication (in revised form) May 31, 1996.

http://www.siam.org/journals/sidma/10-3/29837.html

[†]School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332 (calkin@math.gatech.edu).

[‡]The author is deceased. Former affiliation: Hungarian Academy of Sciences, Mathematical Institute, Budapest, Hungary.

[§]College of Computing, Georgia Institute of Technology, Atlanta, GA 30021 (craig.tovey@isye.gatech.edu).

$R(5,9) \geq 114$. This appears to the be first bound reported [4]. In the 113 vertex graph, the following edge differences are present: $8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 28, 32, 34, 35, 39, 42, 43, 44, 46, 48, 52, 54, 55$.

$R(4,12) \geq 98$. This improves on the bound of 97 reported in [4]. In the 97 vertex graph, the following edge differences are present: $11, 19, 21, 22, 23, 29, 34, 35, 38, 39, 43, 44, 46, 47, 48$.

$R(4,15) \geq 128$. This improves on the bound of 123 reported in [4]. In the 127 vertex graph, the following edge differences are present: $14, 27, 28, 29, 38, 39, 41, 43, 44, 45, 47, 49, 51, 52, 58, 60, 62, 63$.

We are grateful to the referee for bringing to our attention that subsequent to our submitting this paper, Piwakowski [2], [3] has improved our bounds for $R(4,12)$ to 106 and for $R(4,15)$ to 134. This immediately suggests searching for cyclic graphs of order 107 and 137, respectively.

Primes do not always fare better than composites. Besides the trivial case of 3-vertex graphs for $R(3,3)$, the smallest example occurs for $R(4,5)$; there is no cyclic Ramsey graph on 23 vertices, but there is one on 24 vertices.

3. Computational evidence that primes do well. As we shall see, at several points in the theoretical analysis primes seem to show some advantage over composites. We checked empirically for advantages of primes over composites with regard to bounds for $R(4,4)$, $R(5,5)$, and $R(6,6)$. The results are given in Figures 1–3, where shading denotes that a cyclic Ramsey graph is known to exist, and unshaded areas indicate that no cyclic Ramsey graph is known. Primes show a slight advantage in the first two cases and a dramatic advantage in the third case: the largest known Ramsey graph of composite order has 74 vertices, while every prime number order through 101 yields a ramsey graph, with the possible exception of 97.

4. The standard probabilistic analysis suggests that primes do better. The standard probabilistic lower bounds for $R(k,l)$ are obtained as follows: let $0 < p < 1$, randomly 2-color the edges of K_n, red with probability p, and blue with probability $1-p$. Compute the expected number of red K_ks and blue K_ls. If this expectation satisfies

$$\sum_{|K|=k} \Pr(K \text{ is a red clique}) + \sum_{|L|=l} \Pr(L \text{ is a blue clique}) < 1,$$

then there exists a coloring of K_n with no red K_k and no blue K_l.

In the cyclic case, the existence of one monochromatic subgraph implies the existence of many since the image of a monochromatic clique under the rotation $i \to i+1 \pmod{n}$ is also a monochromatic clique. It is easy to see that in fact the existence of one monochromatic clique of order k implies the existence of at least $\frac{n}{(n,k)}$ distinct cliques and, in particular, if n is prime, at least n distinct cliques. Hence, if

$$\frac{(n,k)}{n} \sum_{|K|=k} \Pr(K \text{ is a red clique}) + \frac{(n,l)}{n} \sum_{|L|=l} \Pr(L \text{ is a blue clique}) < 1,$$

where the expectations are now computed over all random *cyclic* colorings, then there exists a cyclic coloring of the edges of K_n without a red K_k or a blue K_l. This dependence on the greatest common divisors (n,k) and (n,l) suggests that we may be slightly more successful in finding graphs of prime order.

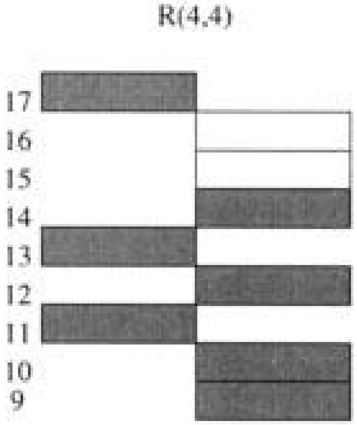


FIG. 1.

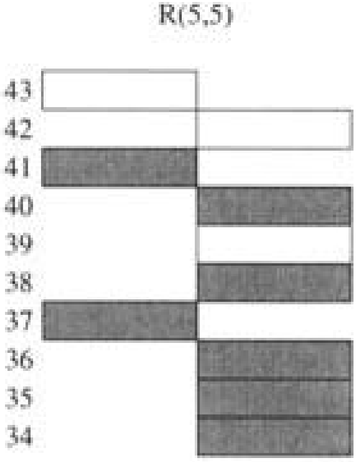


FIG. 2.

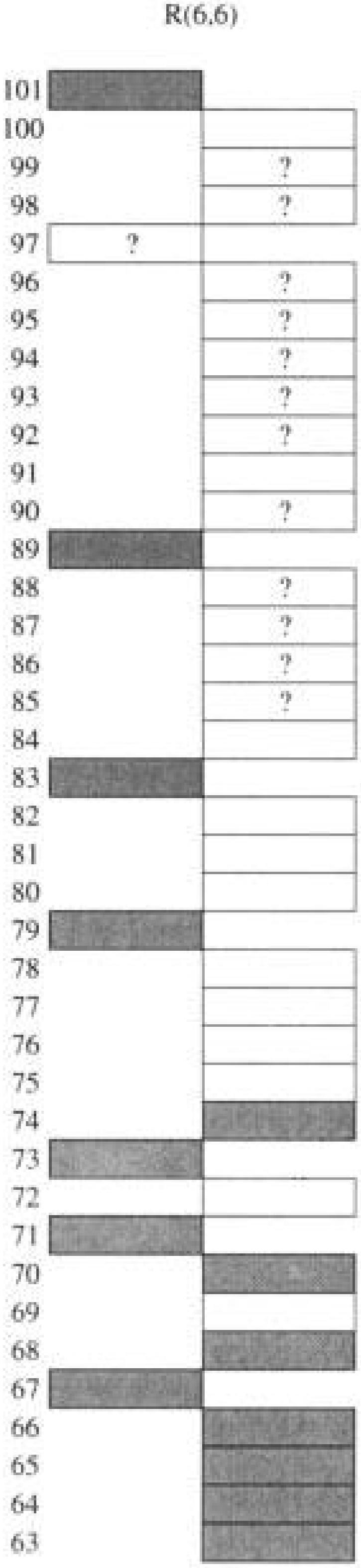


FIG. 3.

However, as we shall see, the computation of the expectation is not sufficient to obtain *any* bounds for $C(k, l)$: indeed, we shall see that the expression above grows at least as fast as $n/\sqrt{k}$ for large n.

We shall concentrate on the first part of the sum: fix k, n, and for now set $p = 1/2$. We wish to compute

$$\sum_{|K|=k} \Pr(K \text{ is a red clique}).$$

Define the *difference* of a pair of vertices i and j as $\min\{|i-j|, n-|i-j|\}$. Note that if a coloring is cyclic, then all edges with the same difference are the same color. The differences $D(K)$ of a set K of vertices are the differences between the pairs comprising $K \times K$.

If a set $K = \{x_1, x_2, \ldots, x_k\} \subseteq \{0, 1, 2, \ldots, n-1\}$ has exactly i distinct differences, i.e., $|D(K)| = i$, then the probability that K is a red clique in a random cyclic coloring is 2^{-i}. Define $N_{i,k,n}$ to be the number of k-subsets of $\{0, 1, 2, \ldots, n-1\}$ having exactly i distinct differences. Then the expected number of red k-cliques in a random cyclic coloring of K_n is

$$\sum_{i=\lfloor k/2 \rfloor}^{\binom{k}{2}} N_{i,k,n} 2^{-i}.$$

If $k \not| n$ then $N_{j,k,n} = 0$ for $j \leq k-2$. Since the 2^{-i} part of the summand is largest in the range $i \leq k-2$, this appears as another slight advantage for prime values of n.

PROPOSITION 4.1. *For n prime and $k < \sqrt{n/2}$,*

$$\sum_{i=k-2}^{\binom{k}{2}} N_{i,k,n} 2^{-i} = \Omega(n^2/\sqrt{k}).$$

Proof. Clearly,

$$\sum_{i=k-2}^{\binom{k}{2}} N_{i,k,n} 2^{-i} > 2^{-(2k-3)} \sum_{i=k-1}^{2k-3} N_{i,k,n}.$$

To bound the latter sum, consider the $\lfloor n/2 \rfloor$ arithmetic progressions mod n of $2k-2$ terms beginning at 0 with common difference $d: \ 0 < d < n/2$. Each of these sets has $2k-3$ distinct differences. From each progression we may remove $k-2$ nonzero elements in $\binom{2k-3}{k-2}$ ways to form a collection of k-subsets. We claim these are all distinct. It is obvious that those from the same progression are distinct, so it suffices to show that no two progressions of $2k-2$ terms, starting at 0, can contain the same k-subset. To see this, we first show that arithmetic progressions of integers with initial term 0 can't intersect in too many elements: let

$$A = \{0, a, 2a, \ldots, ka\}$$

and

$$B = \{0, b, 2b, \ldots, kb\}$$

be two arithmetic progressions with $a < b$ and $(a, b) = 1$ (otherwise just divide both a and b by their greatest common divisor); we will show that

$$|A \cap B| > \left\lfloor \frac{k}{b} \right\rfloor + 1.$$

Since $(a, b) = 1$, if an element x is in their intersection, it is of the form ja and lb, where $b|j$ and $a|l$. Thus the elements of A in the intersection are a subset of

$$0, ba, 2ba, 3ba, \ldots, b \left\lfloor \frac{k}{b} \right\rfloor a.$$

Hence there are at most $\lfloor \frac{k}{b} \rfloor + 1$ of them.

We now consider arbitrary arithmetic progressions. By translating both progressions, we may assume that

$$A = \{0, a, 2a, 3a, \ldots, ka\}$$

and

$$B = \{c, c + b, c + 2b, c + 3b, \ldots, c + kb\}.$$

Now, if $c > 0$ we can replace A by $A \backslash \{0\} \cup \{(k+1)a\}$ without decreasing the size of the intersection. Iterating this process, we see that we can translate until the arithmetic progressions both start with 0, and we are in the case handled above.

We now show that two arithmetic progressions taken modulo n have the same property, provided that k is much less than n (clearly it fails to be true if k is close to n).

Since n is prime, by multiplying both arithmetic progressions by a^{-1} mod p, and by rotating, we may assume

$$A = \{0, 1, 2, 3, \ldots, k\}$$

and

$$B = \{c, c + b, c + 2b, c + 3b, \ldots, c + kb\}.$$

Now, if we knew that B didn't wrap around modulo n, then we would be able to appeal to the statement for arithmetic progressions of integers above: we shall show that there is a value d mod n so that neither dA mod n nor dB mod n wrap around. Observe that since the progressions intersect in at least two elements then we have e, f, g, h so that $e = c + gb$ and $f = c + hb$, where each of e, f, g, h are at most k and we may assume $f > e$. Then

$$f - e = (h - g)b;$$

if $h < g$, then we will replace the arithmetic progression B by the reverse arithmetic progression (with common difference $n - b$ and initial term $c + kb$). Thus we are now in the situation where we have $0 \leq e < f \leq k$, $0 \leq g < h \leq k$, and

$$f - e = (h - g)b.$$

If we now let $d = h - g$, and consider the progressions $A' = dA$ and $B' = dB$, we see that

$$A' = \{0, d, 2d, \ldots, dk\}$$

and

$$B' = \{cd, cd + bd, cd + 2bd, \ldots, cd + kbd\}$$

(taken modulo n). Now, since $d \leq k$ and $bd = f - e \leq k \pmod{n}$, each of A' and B' has a small common difference. Indeed, the difference of A is $d \leq k$ and the difference of B is $bd \leq k$. Thus, provided that $k^2 < \frac{n}{2}$, A' doesn't wrap around (mod n), and B' wraps around at most once. Moreover, if B' wraps around we can rotate both arithmetic progressions so that B' starts at 0 and neither progression wraps around, reducing us to the cases handled above. Thus we have shown that if the arithmetic progressions modulo n intersect in many elements then they are the same arithmetic progression.

Now since n is prime and $d < n/2$ each subset may be rotated $n-1$ times to yield a total of

$$(\lfloor n/2 \rfloor)n\binom{2k-3}{k-2} \sim n^2 2^{2k-3}/\sqrt{k}$$

distinct k-subsets. Each of these subsets has at most $2k-3$ distinct differences, since each is a subset of a progression having $2k-3$ distinct differences. Therefore,

$$\sum_{i=k-1}^{2k-3} N_{i,k,n} = \Omega(n^2 2^{2k-3}/\sqrt{k}),$$

and the proposition follows.

From the proposition we see that a random cyclic graph has a large expected number of monochromatic k-cliques, even if the graph is of order k^2. Hence, standard expected value arguments cannot be used to give bounds on $R(k,l)$ that are exponential in $\min\{k,l\}$. However, Alon and Orlitsky [1] have shown by more sophisticated arguments that random cyclic graphs nonetheless give bounds on $R(k,k)$ of order $e^{c\sqrt{k}}$.

A final advantage of primes may be the following: A natural way to investigate bounds for $N_{i,k,n}$ is to "grow" a set K randomly, counting the number of new distinct differences when a vertex x is added to K. All $|K|$ differences will be distinct only if x does not satisfy any of a set of equations mod n derived from the vertices in K (e.g., x can not be the mean of two points in K). When n is prime these equations are solved over the field Z_n and have unique solutions. But when n is composite there can be multiple solutions, increasing the probability of duplicating a difference (e.g., both 3 and 0 are midpoints of 2 and 4, mod 6).

REFERENCES

[1] N. Alon and A. Orlitsky, *Repeated communication and Ramsey graphs*, IEEE Trans. Inform. Theory, 41 (1995), pp. 1276–1289.

[2] K. Piwakowski, *Applying Tabu search to determine new Ramsey graphs*, research paper R6, Electron. J. Combin., http://www.combinatorics.org (3 (1996)).

[3] K. Piwakowski, *Applying Algorithmic Techniques in Finding Lower and Upper Bounds for Ramsey Numbers*, Ph.D. thesis, Technical University of Gdańsk, Gdańsk, Poland, 1996 (in Polish).

[4] S. Radziszowski, *Small Ramsey numbers*, Dynamic Survey 1, Electron. J. Combin., http://www.combinatorics.org (1994).

SIAM J. Discrete Math.
Vol. 10, No. 3, pp. 388–398, August 1997

LINEAR STEINER TREES FOR INFINITE SPIRALS*

J. F. WENG†

Abstract. A full Steiner tree T for a given set of points P is defined to be linear if all Steiner points lie on one path called the trunk of T. A (nonfull) Steiner tree is linear if it is a degeneracy of a full linear Steiner tree. Suppose P is a simple polygonal line. Roughly speaking, T is similar to P if its trunk turns to the left or right when P does. P is a left-turn (or right-turn) polygonal spiral if it always turns to the left (or right) at its vertices. P is an infinite spiral if n tends to infinity. In this paper we first prove some results on nonminimal paths and the decomposition of Steiner minimal trees, and then, based on these results, we study the case in which an infinite spiral P has a Steiner minimal tree that is linear and similar to P itself.

Key words. Steiner tree, spiral

AMS subject classification. 05C05

PII. S0895480194274030

1. Linear Steiner trees and polygonal spirals.

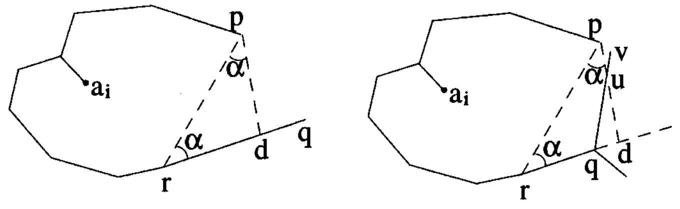
A Steiner minimal tree $SMT(A)$ for a set A of given points (called *regular points*) is a shortest network interconnecting these points with some additional points (called *Steiner points*) [3]. All angles in $SMT(A)$ are no less than 120°. A tree satisfying this angle condition is called a *Steiner tree.* By topology of a network we mean the graph structure of the network. Steiner trees can be classified as *full* or *nonfull.* A Steiner tree is called *full* if every regular point is of degree one. The importance of this classification is that any Steiner tree can be decomposed into a union of full subtrees. On the other hand, a nonfull Steiner tree can be regarded as a *degeneracy* of a full Steiner tree [4]; i.e., some Steiner points in the tree collapse into their adjacent regular points.

In this paper we give another natural classification of Steiner trees. Suppose a Steiner tree T for A is full. If every Steiner point is adjacent to three regular points, then there is only one Steiner point, and A is a three-point set. If every Steiner point is adjacent to two regular points, then there are only two Steiner points, and A is a four-point set. When A has five points, every Steiner point in T is at least adjacent to one regular point. However, when A has more than five regular points, there may be Steiner points that are not adjacent to any regular points. It is worth noticing that if every Steiner point is at least adjacent to one regular point in a full Steiner tree, then all Steiner points lie on one path. Such a full Steiner tree is defined to be *linear*, and the path joining all Steiner points in sequence is called its *trunk.* A nonfull Steiner tree is defined to be linear if it is a degeneracy of a full linear Steiner tree. Clearly, in a certain sense, linear trees are the simplest of all Steiner trees. First, we ask the following question: *What sets of points have linear Steiner minimal trees?*

A sequence of points $\{a_0, a_1, \dots, a_n\}$ $(n \geq 2)$ is called a (*simple*) *polygonal line* if no two nonconsecutive line segments $a_{i-1}a_i$ and a_ia_{i+1} meet. This line will be written as $P = a_0a_1 \cdots a_n$. Suppose a Steiner tree T for P is linear with trunk $s_1s_1 \cdots s_{n-1}$. (As mentioned above, some Steiner points may coincide with regular points if T is nonfull.) T is defined to be *similar* to P if s_i are adjacent to a_i for all i $(1 \leq i \leq n-1)$. The modifier "similar" comes from the fact that, roughly speaking, in this case the

*Received by the editors September 12, 1994; accepted for publication (in revised form) May 31, 1996. This research was supported by an Australian Research Council grant.
http://www.siam.org/journals/sidma/10-3/27403.html

†Department of Mathematics, University of Melbourne, Victoria 3052, Australia (weng@mundoe.maths.mu.oz.au).



(a) (b)

FIG. 1.

trunk turns to the right or left when P does. Now the second question arises: *What polygonal lines have Steiner minimal trees that are similar to themselves?*

There are two extreme types of polygonal lines. If a polygonal line $P = a_0 a_1 \cdots a_n$ always turns to the left or right at its vertices, then it is called a left-turn or right-turn polygonal *spiral.* (For simplicity, we often just say that P is a spiral and omit the modifier "polygonal.") If P turns left and right alternately in some way, then it is a *wave line.* In this paper we study only the spirals that have similar Steiner minimal trees and leave the discussion on wave lines with similar Steiner minimal trees to another paper. When P is a spiral, we are much more interested in its behavior when n tends to infinity; i.e., n is arbitrarily large. Such a spiral is referred to as an infinite spiral. In this paper we first prove some results on nonminimal paths and the decomposition of Steiner minimal trees, and then, based on these results, we study the case in which an infinite spiral P has a Steiner minimal tree that is similar to P itself.

2. Nonminimal paths and the decomposition of Steiner minimal trees. In this section we prove some general results that can be used to eliminate nonminimal Steiner trees.

A path in a Steiner tree that always turns left (or right) at its vertices is called a left-turn (or right-turn) path. An object involving more than two points (e.g., an angle, a path, a polygon, etc.) is usually written in counterclockwise order unless specially indicated. After Cockayne [1], by (ab) denote the third vertex of the equilateral triangle $(ab)ba$ based on ab and on the right side of ab looking from a to b. Furthermore, this notation can be used to represent a full Steiner tree. For example, $(ab)(cd)$ represents the full Steiner tree in which a, b join a Steiner point and c, d join another Steiner point. By $|x|$ we denote the length of x where x is a line segment, a polygonal line, or a tree. In this section suppose T is a Steiner tree on a set A.

LEMMA 2.1 (nonminimal paths). *Suppose $p \cdots rq$ is a path in T such that $\angle prq \leq 60°$. Let d be the point on rq (Figure 1(a)) or its extension (Figure 1(b)) such that $\angle dpr = \angle prq$. Suppose any regular point (e.g., a_i in Figure 1) in the polygon $p \cdots rd$ is connected to q via r, and suppose q is a Steiner point if q lies on rd. Then T is not minimal.*

Proof. If d lies strictly on rq then rq can be replaced by pd to shorten T since $|pd| < |rq|$. Now suppose q lies on rd. If $d = q$, then q is a Steiner point by our assumption. Hence, we obtain a new tree T' with the same length by replacing rq with pq. Since the angle condition is not satisfied at q, T' is not minimal. Finally, suppose d lies strictly on the extension of rq. Let the left edge of q looking from r to q end at v. If qv meets pd at a point u, then $|pu| \leq |rq|$. rq can be replaced by pu to shorten T. If qv does not meet pd, then the right turn path $rqv \cdots$ ends nowhere

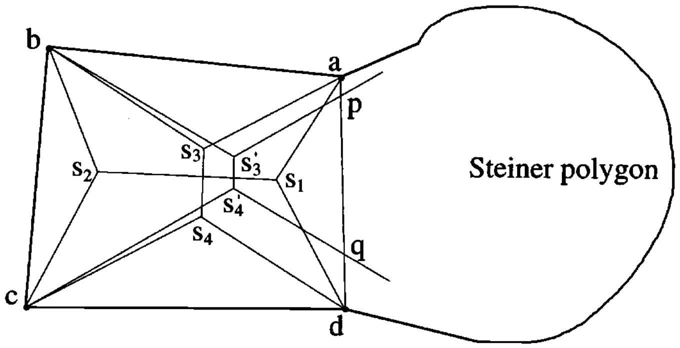


FIG. 2.

since all regular points in $p \cdots rd$ have been connected to q through r. □

Now we state a theorem which was first proved by Pollak [6], then improved in [2] and [8].

THEOREM 2.1 *Suppose abcd is a quadrilateral such that* $\angle(ab)cd \geq 120°$, $\angle cd(ab) \geq 120°$, $\angle(cd)ab \geq 120°$, $\angle ab(cd) \geq 120°$*. If the angle at the intersection of the diagonals subtending to ab is less than* $90°$*, then the full Steiner tree* $(ab)(cd)$ *exists and is the unique Steiner minimal tree.*

LEMMA 2.2. *Suppose* ps_1s_2q *is a convex path in* T *where* s_1, s_2 *are Steiner points. If both* $|s_1s_2|/|ps_1|$ *and*$|s_1s_2|/|s_2q|$ *are less than* $\sin(15°)/\sin(45°) = 0.3660$*, then* T *is not minimal.*

Proof. If the condition is satisfied, then there is a point p' on ps_1 and a point q' on qs_2 such that $p's_2$ is perpendicular to s_1q'. It follows that the angle at the intersection of ps_2 and s_1q subtending pq is less than $90°$. Clearly, the quadrilateral qps_1s_2 satisfies the conditions of Theorem 2.1. Therefore, $|ps_1|+|s_1s_2|+|s_2q|$ is longer than the Steiner minimal tree $(qp)(s_1s_2)$. T is not minimal. □

A *Steiner polygon* of a set A is a polygon such that all Steiner minimal trees for A lie in it [1], [9].

LEMMA 2.3. *Suppose*

(1) a, b, c, d *are four consecutive vertices of a Steiner polygon of* A *and* $\angle abc + \angle bcd \leq 180°$,

(2) *there is no regular point in abcd, and*

(3) *the full Steiner tree* $(da)(bc)$ *is the unique Steiner minimal tree on abcd.*

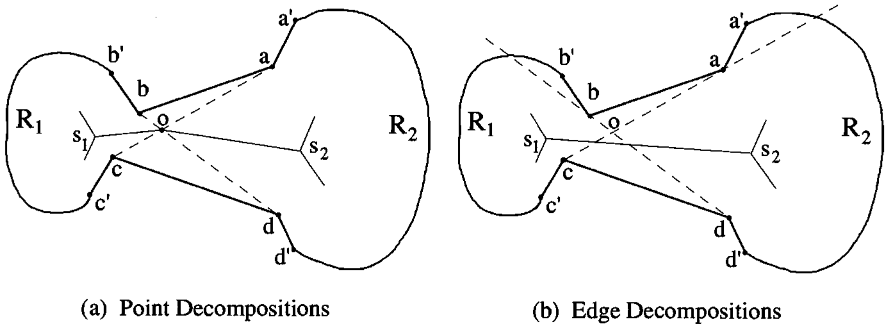
If there is more than one Steiner point lying on the path connecting b, c *in* T*, then* T *is not minimal.*

To prove this lemma we need the following embedding theorem [8].

THEOREM 2.2. *Suppose abcd is a quadrilateral embedded in another quadrilateral* $a'b'c'd'$ *with* a, d *on* $a'd'$ *and* b, c *on* $b'c'$*. If the topology* $(a'd')(c'b')$ *is optimal for* $a'b'c'd'$*, then the topology* $(ad)(cb)$ *is optimal for abcd.*

Proof of Lemma 2.3. Let $T_1 = (da)(bc)$, in which a, d join the same Steiner point s_1 and b, c join the same Steiner point s_2. Let T_2 be the Steiner tree whose topology is $(ab)(cd)$ or a degeneracy of it. Let a, b join the same Steiner point s_3 and c, d join the same Steiner point s_4 in T_2. By the assumption, $|T_2| > |T_1|$.

Since $\angle abc + \angle bcd \leq 180°$, there are at most two Steiner points, say s'_3 and s'_4, on the convex path connecting b, c in T. Because the left-turn Steiner path starting with bs'_3 cannot intersect ab and there is no regular point in $abcd$, the third edge of s'_3 must meet ad at a point p (Figure 2). Similarly, the third edge of s'_4 meets ad at



(a) Point Decompositions (b) Edge Decompositions

Fig. 3.

a point q. Applying the embedding theorem to $abcd$ and $pbcq$, we conclude that the subtree on $pbcq$, and consequently T, is not minimal. □

Let T be a Steiner minimal tree on a set A. Suppose A is decomposed into two subsets A_1 and A_2, and T can accordingly be decomposed into two subtrees T_1 and T_2 so that $T_i (i = 1, 2)$ connect the regular points in $A_i (i = 1, 2)$, respectively. If A_1 and A_2 share only one regular point a and T_1 joins T_2 at a, then T as well as A is defined to have a *point decomposition* at a. If A_1 and A_2 are disjoint and T_1 joins T_2 by an edge $s_1 s_2$ with $s_i (i = 1, 2)$ in T_i, then T as well as A is defined to have an *edge decomposition* with $s_1 s_2$. In such decomposition terms the above lemma can be stated as follows.

Lemma 2.3*. *Let $A_1 = \{b, c\}$ and let A_2 consist of a, d and other regular points of A. If the conditions in Lemma* 2.3 *are satisfied, then the Steiner minimal tree T on A has an edge decomposition with two subtrees spanning A_1 and A_2, respectively.*

Gilbert and Pollak [3] proved a property concerning the decomposition of Steiner minimal trees which they called the double wedge property.

Double Wedge Property. *Suppose two lines meet* 120° *at a point o so that all points of A lie in two* 60° *wedges R_1 and R_2.*

(1) *If o is a regular point, then the Steiner minimal tree T has a point decomposition at o with two subtrees spanning the subsets of A in R_1 and R_2, respectively.*

(2) *If o is not a regular point then T has an edge decomposition with two subtrees spanning the subsets of A in R_1 and R_2, respectively.*

Suppose $bb' \cdots c'cdd' \cdots a'a$ is a Steiner polygon of A. Let o be the intersection of ac and bd. If $\angle boa \geq 120°$ and no regular points lie in $\triangle boa$ and $\triangle doc$, then $abcd$ is called a *neck* of the Steiner polygon. Thus, the double wedge property can be improved as follows.

Lemma 2.4 (neck decomposition). *Suppose $bb' \cdots c'cdd' \cdots a'a$ is a Steiner polygon of A with a neck $abcd$. Then the Steiner minimal tree T has either a point decomposition* (*Figure* 3(a)) *or an edge decomposition* (*Figure* 3(b)) *as stated in the double wedge property.*

Proof. Suppose o is the intersection of ac and bd. Let the regions enclosed by $obb' \cdots c'c$ and $odd' \cdots a'a$ be R_1 and R_2, respectively. If o is a regular point, then we can delete $\triangle boa$ from the Steiner polygon of A since $\angle boa \geq 120°$. Similarly, we can delete $\triangle doc$. Thus we obtain a new Steiner polygon consisting of two small ones that join at o. Hence, T has a point decomposition at o.

Now suppose o is not a regular point. If there is a Steiner point s in $\triangle boa$, then there is a left- or right-turn path starting from s which intersects ab at an interior

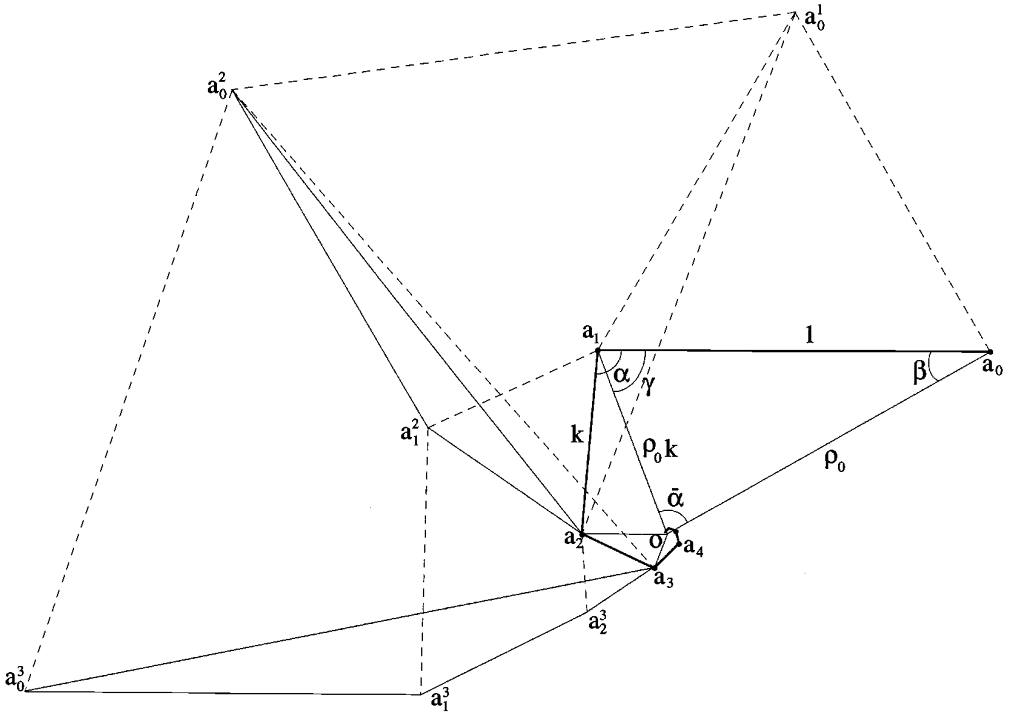


FIG. 4.

point and goes out of the Steiner polygon. This contradicts the definition of Steiner polygons. Similarly, there is no Steiner point in $\triangle doc$. Hence, all Steiner points lie in R_1 and R_2. Suppose there are two edges in T crossing the sides of $\triangle boa$ and/or $\triangle doc$ at p_1, q_1 in R_1 and p_2, q_2 in R_2, respectively. Since $\angle boa \geq 120°$, one of p_1p_2, q_1q_2 is the longest side of $p_1q_1q_2p_2$. Hence, T cannot contain both p_1p_2 and q_1q_2. We obtain a contradiction. □

Remark. In the lemma it is not required that no regular points lie in two angles $\angle boa$ or $\angle doc$. For example, a', b' lie in $\angle boa$ in Figure 3.

3. Logarithmic spirals. Suppose $P = a_0a_1a_2 \cdots a_n$ is a left-turn spiral. P can be characterized by two sets of parameters: edge ratios $k_i = |a_ia_{i+1}|/|a_{i-1}a_i|$ and turning angles $\alpha_i = \angle a_{i-1}a_ia_{i+1}$. It is naturally assumed that $0 < k_i < 1$ and $0 < \alpha_i < 180°$. Without loss of generality, assume $|a_0a_1| = 1$.

THEOREM 3.1. *If $k_i = k$ and $\alpha_i = \alpha$, where k and α are two constants, then all a_i lie on a logarithmic spiral curve with an asymptotic point o. All polar angles $\angle a_ioa_{i+1}$ between two adjacent points a_i, a_{i+1} $(0 \leq i < n)$ are equal.*

Proof. Let o be the intersection of the arcs $\overparen{a_0a_1}, \overparen{a_1a_2}$ such that $\angle a_2oa_1 = \angle a_1oa_0 = \bar{\alpha} = 180° - \alpha$. Since

$$\angle a_0a_1o = \alpha - \angle oa_1a_2 = 180 - \angle a_2oa_1 - \angle oa_1a_2 = \angle a_1a_2o,$$

the two triangles $\triangle oa_0a_1$ and $\triangle oa_1a_2$ are similar. Now because $k_i = k$ and $\alpha_i = \alpha$, all $\angle a_{i+1}oa_i$ equal $\bar{\alpha}$ and all $\triangle a_{i+1}oa_i$ are similar (Figure 4). Let o be the pole and oa_0 the polar axis. Let ρ denote the radius vector and θ the polar angle. Hence, the coordinates of a_i are $\rho(a_i) = |oa_i|$ and $\theta(a_i) = i\bar{\alpha}$. Since $|oa_{i+1}|/|oa_i| = k$, they satisfy

$$\rho(a_i) = |oa_i| = \rho_0 k^i = \rho_0(k^{1/\bar{\alpha}})^{\theta(a_i)},$$

where $\rho_0 = |oa_0|$. Hence, a_i $(0 \le i \le n)$ lie on the logarithmic spiral curve

$$\rho(\theta) = \rho_0 (k^{1/\bar{\alpha}})^\theta$$

and the polar angles between a_i and a_{i+1} $(i \ge 0)$ are all equal to $\bar{\alpha}$. $\square$

By this theorem, a (polygonal) spiral with constant edge ratio k and turning angle α is called a logarithmic (polygonal) spiral and is denoted by $L^n(k,\alpha)$. If n tends to infinity, then it is denoted by $L^\infty(k,\alpha)$. Let $\beta = \angle oa_ia_{i+1}, \gamma = \angle a_ia_{i+1}o$. It is easy to show that

$$\rho_0 = \frac{1}{\sqrt{1+2k\cos\alpha + k^2}},$$

$$\sin\beta = \rho_0 k \sin\alpha,$$

$$\sin\gamma = \rho_0 \sin\alpha.$$

If the Steiner minimal tree T for $L^\infty(k,\alpha)$ is similar to $L^\infty(k,\alpha)$ itself, then its topology is

$$\cdots((\cdots((a_0a_1)a_2)\cdots a_{i-1})a_i)\cdots.$$

Define

$$a_j^j = a_j,\ a_j^i = (a_j a_j^{i-1}),\ 0 \le j < i.$$

Assume we have expanded T to a_0^i by Melzak's method [5], and we denote the expanded tree by T^i. Then T^i is a Steiner minimal tree on the set

$$A^i = \{a_0^i, a_{i+1}, a_{i+2}, \ldots, a_n\}.$$

Clearly, $a_0^i a_1^i \cdots a_{i-1}^i a_i$ is a convex polygonal line with the same interior angle $\alpha + 60°$. It lies on the right (or left) side of $a_0^i a_i$ looking from a_0^i to a_i if $\alpha < 120°$ (or $> 120°$). When $\alpha = 120°$, all $a_0^i, a_1^i, \ldots, a_{i-1}^i, a_i$ lie on the extension of $a_{i+1}a_i$ (Figure 5), and

$$|a_0^i a_i| = 1 + k + k^2 + \cdots + k^{i-1} = \frac{1-k^i}{1-k}.$$

Theorem 3.2. *The Steiner minimal tree for $L^\infty(k,\alpha)$ cannot be similar to $L^\infty(k,\alpha)$ itself if $\alpha < 120°$.*

Proof. First note that if the Steiner minimal tree for $L^\infty(k,\alpha)$ is similar to $L^\infty(k,\alpha)$ itself, then the left-turn Steiner path starting from a_0 is an infinite path; that is, it has no end.

Now suppose $\alpha < 120°$; then $\angle a_{i-1}^i a_i a_0^i > 0°$. It is easily seen by geometric consideration that this angle is unlimited increasing. Consequently, $\angle a_0^i a_i a_{i+1} = \alpha + 60° - \angle a_{i-1}^i a_i a_0^i < 180°$, and it is unlimited decreasing when i goes to infinity (refer to Figure 4). Therefore, when i is large enough, $a_0^i a_i$ will intersect line segment $a_j a_{j+1}$ for certain $j > i$. It implies that the left-turn path starting from a_0 is not an infinite path and must end at a certain regular point. Thus, the theorem is proved. $\square$

Theorem 3.3. *If $k \le 0.4758$ and $\alpha \ge 120°$, then the Steiner minimal tree for $L^\infty(k,\alpha)$ is $L^\infty(k,\alpha)$ itself. Hence, the Steiner minimal tree is unique and similar.*

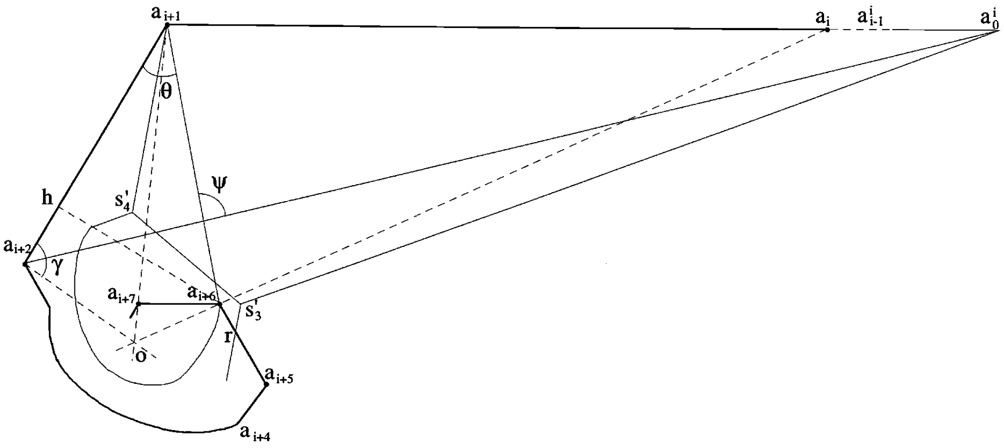


FIG. 5.

Proof. Suppose T is the Steiner tree for $L^\infty(k,\alpha)$ with the topology

$$\cdots((\cdots((a_0a_1)a_2)\cdots a_{i-1})a_i)\cdots.$$

If the theorem is true for $\alpha = 120°$, then all Steiner points must collapse into their adjacent regular points; that is, T is $L^\infty(k,\alpha)$ itself. Hence, the theorem also holds for $\alpha > 120°$. Below we prove the theorem is true for $\alpha = 120°$.

When $\alpha = 120°$, $a^i_0a_{i+1}\|a_{i+6}a_{i+7}\|a_{i+4}a_{i+3}$. By the formulae listed above,

$$\rho_0 = \frac{1}{\sqrt{1-k+k^2}}, \qquad \sin\beta = \frac{\sqrt{3}}{2}\rho_0 k, \qquad \sin\gamma = \frac{\sqrt{3}}{2}\rho_0.$$

Especially $1 < \rho_0 < 2/\sqrt{3}, \beta < 30°, \gamma > 90°$ when $k \le 0.4758$.

The theorem is proved if we can show that a^i_0 and a_{i+1} join the same Steiner point in T^i for all $i \ge 0$, where T^i is the Steiner minimal tree for $A^i = \{a^i_0, a_{i+1}, a_{i+2}, \dots, a_n\}$, as stated before. We prove by contradiction. That is, below we will show that T^i is not minimal if there are $m(\ge 2)$ Steiner points on the path connecting a^i_0 and a_{i+1} in T^i.

First, since the extension of $a_{i+4}a_{i+5}$ meets the line $a^i_0a_{i+1}$ at 60°, we have $\angle a^i_0a_{i+5}a_{i+4} > 120°$ and $\angle a_{i+5}a^i_0a_{i+1} < 60°$. Hence, $\angle a_{i+5}a^i_0a_{i+1} + \angle a^i_0a_{i+1}a_{i+2} < 180°$. It implies that $m \le 2$.

Now suppose $m = 2$ and s'_3, s'_4 are the two Steiner points on the path connecting a^i_0 and a_{i+1} in T^i (Figure 5). We want to show that $a_{i+6}a^i_0a_{i+1}a_{i+2}$ satisfy all three conditions of Lemma 2.3. We have shown that

$$\angle a_{i+6}a^i_0a_{i+1} + \angle a^i_0a_{i+1}a_{i+2} < \angle a_{i+5}a^i_0a_{i+1} + \angle a^i_0a_{i+1}a_{i+2} < 180°.$$

To satisfy the first condition, we need only to prove that a_{i+6} is on the Steiner polygon of A^i. It suffices to prove that the left-turn path starting with $a^i_0s'_3$ cannot meet $a_{i+5}a_{i+6}$. Suppose to the contrary they meet at a point r. Since the angle between $a_{i+5}a_{i+6}$ and $a^i_0a_{i+1}$ is 60°, r lies on the third edge of s'_3 and $\angle a_{i+6}rs'_3 < 60°$. Hence, by Lemma 2.1 the path connecting a_{i+6} and s'_3 cannot do so through r; that is, it should be through s'_4. It follows that

$$|s'_3s'_4| < |a_{i+6}r| \le |a_{i+6}a_{i+5}| = k^{i+5}.$$

Since $\angle a_{i+6} r s_3' < 60°$, we also have $|a_{i+6} s_3'| < |a_{i+6} a_{i+5}| = k^{i+5}$. Hence,

$$\begin{aligned}|s_4' a_{i+1}| &\geq |o a_{i+1}| - |o s_4'| > |o a_{i+1}| - |o a_{i+6}| - |a_{i+6} s_3'| - |s_3' s_4'| \\ &> \rho_0 k^{i+1} - \rho_0 k^{i+6} - 2k^{i+5}\end{aligned}$$

and

$$\begin{aligned}\frac{|s_3' s_4'|}{|s_4' a_{i+1}|} &< \frac{k^{i+5}}{\rho_0 k^{i+1} - \rho_0 k^{i+6} - 2k^{i+5}} \\ &= \frac{k^4}{\rho_0 (1-k^5) - 2k^4} < \frac{k^4}{(1-k^5) - 2k^4} < 0.3660.\end{aligned}$$

It is easily seen that $\angle a_0^i a_{i+1} s_4' \geq 60° \geq \angle s_3' a_0^i a_{i+1}$. Hence, we have $|s_3' a_0^i| \geq |a_{i+1} s_4'|$ and $|s_3' s_4'| / |s_3' a_0^i| < 0.3660$ too. By Lemma 2.2, T^i is not minimal. This contradicts the assumption that T^i is minimal. Hence, $a_{i+6}, a_0^i, a_{i+1}, a_{i+2}$ are four consecutive vertices of the Steiner polygon of the set A^i. The first condition of Lemma 2.3 is satisfied.

Let the distance of a_{i+6} from the line $a_0^i a_{i+1}$ be d_1 and the distance of a_{i+2} from the line $a_0^i a_{i+1}$ be d_2. Note that a_{i+6} lies on $o a_i$. Since $1 - k^6 > 1 - k + k^2$,

$$\begin{aligned}d_1 = |a_{i+6} a_i| \sin\beta = (|o a_i| - |o a_{i+6}|)\, \rho_0 k \sin\alpha &= \frac{\sqrt{3}}{2} \times \frac{k^{i+1}\left(1 - k^6\right)}{1 - k + k^2} \\ &> d_2 = |a_{i+2} a_{i+1}| \sin(180° - \alpha) = \frac{\sqrt{3}}{2} k^{i+1}.\end{aligned}$$

Hence, a_{i+2} lies between the parallel lines $a_0^i a_{i+1}$ and $a_{i+6} a_{i+7}$. It follows that there is no regular point in $a_0^i a_{i+1} a_{i+2} a_{i+6}$. The second condition of Lemma 2.3 is also satisfied.

It is easy to check that all the angles $\angle a_0^i a_{i+1} (a_{i+2} a_{i+6})$, $\angle (a_{i+2} a_{i+6}) a_0^i a_{i+1}$, $\angle (a_0^i a_{i+1}) a_{i+2} a_{i+6}$, $\angle a_{i+2} a_{i+6} (a_0^i a_{i+1})$ are no more than 120°. Let ψ be the angle at the intersection of $a_{i+6} a_{i+1}$ and $a_0^i a_{i+2}$ subtending $a_0^i a_{i+1}$ (Figure 5). By Theorem 2.1, the full Steiner tree $(a_0^i a_{i+1})(a_{i+2} a_{i+6})$ exists and is the Steiner minimal tree for $a_0^i a_{i+1} a_{i+2} a_{i+6}$ if $\psi < 90°$. Let $\theta = \angle a_{i+6} a_{i+1} a_{i+2}$ and let h be the point on $a_{i+1} a_{i+2}$ such that $a_{i+6} h \perp a_{i+1} a_{i+2}$. Since $\angle a_{i+1} a_{i+2} o = \gamma > 90°, \angle a_{i+2} o a_{i+6} = 120°$, we have $\angle o a_{i+6} h < 60°$. Hence,

$$\begin{aligned}\tan\theta &= \frac{|a_{i+6} h|}{|a_{i+1} h|} \\ &< \frac{|a_{i+2} o| + |a_{i+6} o|/2}{|a_{i+1} a_{i+2}| - |a_{i+6} o| \sqrt{3}/2} = \frac{\rho_0 (k^{i+2} + k^{i+6}/2)}{k^{i+1} - \rho_0 k^{i+6} \sqrt{3}/2} = \frac{\rho_0 (2k + k^5)}{2 - \sqrt{3} \rho_0 k^5}.\end{aligned}$$

It is easy to verify that the right-hand side is monotone increasing and is less than $1/\sqrt{3}$ when $k \leq 0.4758$. It follows that $\theta < 30°$, $\angle a_0^i a_{i+1} a_{i+6} = 120° - \theta > 90°$, and $\psi < 90°$. Hence, the third condition of Lemma 2.3 is satisfied.

Now, by Lemma 2.3, T^i is not minimal. This contradiction shows that $m = 1$; i.e., a_0^i and a_{i+1} always join the same Steiner point for any i. The proof is complete. □

4. Spirals with two asymptotic points.

LEMMA 4.1. *Suppose $L^\infty(k, 120°) = a_0 a_1 a_2 \cdots$, $k \leq 0.4758$, and v is any point on $a_0 a_1$. Then the Steiner minimal tree for $v a_1 a_2 \cdots$ is still $v a_1 a_2 \cdots$ itself.*

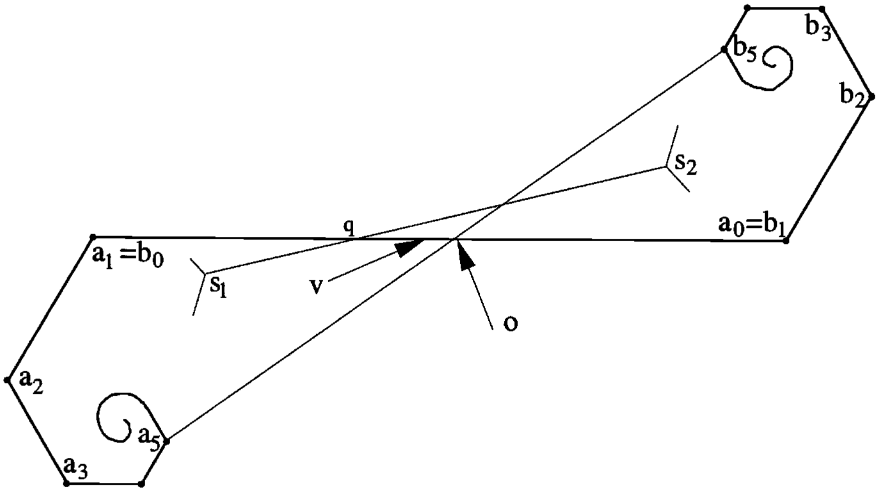


FIG. 6.

Proof. The proof is by Theorem 3.3 and the variational argument [7],[8]. □

Let $L_1 = L^\infty(k_1, 120°) = a_0a_1a_2\cdots$ with $k_1 \le 0.4758$, $L_2 = L^\infty(k_2, 120°) = b_0b_1b_2\cdots$ with $k_2 \le 0.4758$ be two left-turn logarithmic spirals. We combine them into one spiral $SL^\infty(k_1, k_2) = \cdots a_5a_4\cdots a_1b_1\cdots b_4b_5\cdots$ with two asymptotic points by setting $a_0 = b_1$ and $b_0 = a_1$ (Figure 6).

THEOREM 4.1. *The unique Steiner minimal tree for $SL^\infty(k_1, k_2)$ is $SL^\infty(k_1, k_2)$ itself.*

Proof. Obviously, $b_1b_2\cdots b_5a_1a_2\cdots a_5$ is a Steiner polygon of $SL^\infty(k_1, k_2)$. Let v be the midpoint of a_1b_1 and o be the intersection of a_1b_1 and a_5b_5. Then clearly $\angle a_0va_5 > 120°, \angle b_0va_5 > 120°$ since $k_1 < 1/2, k_2 < 1/2$. It follows that $\angle a_1ob_5 > 120°$. Hence, $a_1a_5b_1b_5$ is a neck of the Steiner polygon. Applying Lemma 2.4 to $a_1a_5b_1b_5$, the Steiner minimal tree for $SL^\infty(k_1, k_2)$ consists of two trees T_1 and T_2 connected by an edge s_1s_2. Let q be the intersection of s_1s_2 and b_1a_1. By Lemma 4.1, the Steiner minimal trees spanning $qa_1a_2\cdots$ and $qb_1b_2\cdots$ are $qa_1a_2\cdots$ and $qb_1b_2\cdots$ themselves. Hence, the unique Steiner minimal tree for $SL^\infty(k_1, k_2)$ is $SL^\infty(k_1, k_2)$ itself. □

LEMMA 4.2. *Suppose $L^\infty(k, 120°) = a_0a_1a_2\cdots$ with $k \le 0.4758$. Then $\angle a_5a_0a_1 < 30°$.*

Proof. First $\angle oa_0a_1 = \beta = \arcsin(\frac{\sqrt{3}}{2}\rho_0k) \le 28.58°$. Since $|oa_0| = \rho_0, |oa_5| = \rho_0k^5$ and $\angle a_0oa_5 = 60°$. Let $x = \angle a_5a_0o$. By the law of sines, $\sin x/\sin(120° - x) = |oa_5|/|oa_0| = k^5$. After simplification, $\tan x = \sqrt{3}k^5/\left(2 - k^5\right)$ and $x \le 1.26°$. Hence, $\angle a_5a_0a_1 = \angle a_5a_0o + \angle oa_0a_1 \le 29.84°$. □

Let $L_3 = L^\infty(k, 120°) = a_0a_1a_2\cdots$ be a left-turn spiral. Let $L_4 = b_0b_1b_2\cdots$ be a right-turn spiral with the same k and symmetrical to L_3. We combine them into one spiral $CL^\infty(k) = \cdots a_5a_4\cdots a_1b_1\cdots b_4b_5\cdots$ by setting $a_0 = b_1$ and $b_0 = a_1$.

THEOREM 4.2. *If $k \le 0.4655$, then the unique Steiner minimal tree for $CL^\infty(k)$ is $CL^\infty(k)$ itself.*

Proof. Obviously $a_1a_2\cdots a_5b_5\cdots b_2b_1$ is a Steiner polygon of $CL^\infty(k)$. Let o be the intersection of a_1b_5 and b_1a_5. By symmetry and the above lemma, $\angle a_1ob_1 = 180° - 2\angle a_5a_0a_1 > 120°$. Hence, $b_1a_1a_5b_5$ is a neck of the Steiner polygon of $CL^\infty(k)$. By Lemma 2.4, the Steiner minimal tree T for $CL^\infty(k)$ has an edge s_1s_1' which connects a subtree T_1 spanning $a_1a_2\cdots$ and a subtree T_2 spanning $b_1b_2\cdots$. Clearly,

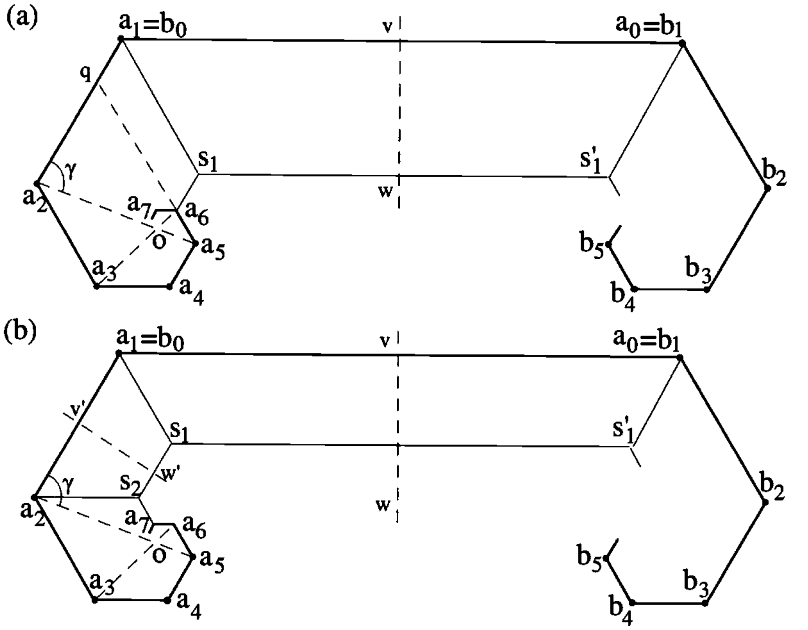


FIG. 7.

by symmetry there are either four or two Steiner points on the path connecting a_1 and b_1. However, the former case is impossible since $\angle b_2b_1a_1 = \angle b_1a_1a_2 = 120°$. It follows that the path is $a_1s_1s_1'b_1$ and $s_1s_1' \| a_1b_1$. Let the vertical bisector of a_1b_1 intersect a_1b_1 at v and intersect s_1s_1' at w. By symmetry, we need only consider T_1, the left part of T which spans w and $a_1a_2\cdots$. Suppose to the contrary that $s_1 \neq a_1$. There are two possibilities:

(1) The third edge of s_1 meets a_5a_6 (Figure 7(a)). It is easy to see for minimality that s_1 should end at a_6 and T_1 does not contain a_1a_2. Let the extension of a_5a_6 meet a_1a_2 at q. Since the asymptotic point o lies on a_2a_5, $\angle a_5qa_2 = 60°$, $\angle a_2a_5a_6 = \beta$, $\angle qa_2a_5 = \gamma$, and we have

$$|a_2q| = \rho_0^2 k\left(k^2 + k^5\right), \qquad |a_6q| = |a_5q| - |a_5a_6| = \rho_0^2\left(k^2 + k^5\right) - k^5.$$

Clearly, $a_1qa_6s_1$ is a parallelogram, $|s_1a_1| = |a_6q|$, and $\angle va_1s_1 = 60°$.

Define

$$\begin{aligned} f(k) &= |ws_1| + |s_1a_1| + |s_1a_6| - (|va_1| + |a_1a_2|) = \frac{1}{2}|s_1a_1| - |a_2q| \\ &= \frac{1}{2}\left(\rho_0^2\left(k^2 + k^5\right) - k^5\right) - \rho_0^2 k\left(k^2 + k^5\right) \\ &= \frac{1}{2}\left(\rho_0^2\left(k^2 + k^5\right)(1 - 2k) - k^5\right). \end{aligned}$$

$f(k)$ is positive when $k \leq 0.4655$, as shown in Figure 8. Hence, T is not minimal.

(2) The third edge of s_1 does not meet a_5a_6. Then it meets the bisector of $\angle a_1a_2a_3$; otherwise the right-turn path $a_1s_1s_2\cdots$ ends nowhere (Figure 7(b)). (Recall that we have shown a_2 is between the lines a_0a_1 and a_6a_7 in the proof of Theorem 3.3.) If s_1s_2 crosses the bisector of $\angle a_1a_2a_3$ but does not meet a_6a_7, then the right-turn path ends nowhere either. So, similarly to the argument in (1), for minimality, the

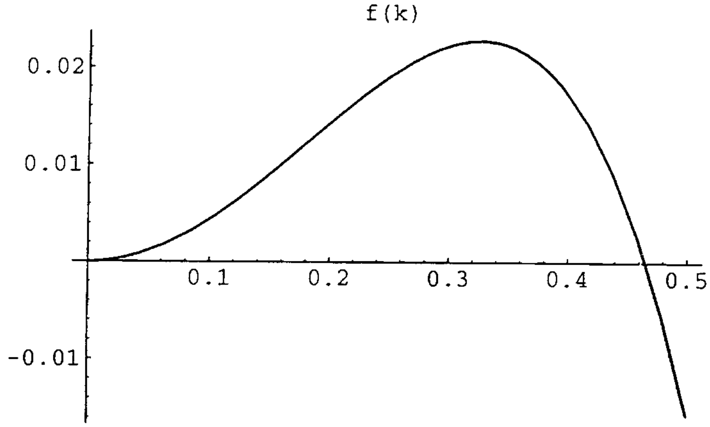


FIG. 8.

third edge of s_2 should end at a_7 and a_2a_3 is not in T_1. Let the bisector of a_1a_2 intersect a_1a_2 and s_1s_2 at v' and w'. Then $|ws_1|+|s_1a_1|+|s_1w'|=|va_1|+|a_1v'|$ and $|w's_2|+|s_2a_2|+|s_2a_7|>|v'a_2|+|a_2a_3|$, as argued in (1). Hence,

$$|ws_1|+|s_1a_1|+|s_1s_2|+|s_2a_2|+|s_2a_7|-(|va_1|+|a_1a_2|+|a_2a_3|)>0.$$

T is not minimal.

Thus, unlimitedly repeating such arguments results in T having to be linear and similar to $CL^\infty(k)$. In fact, $T=CL^\infty(k)$ since all angles at a_i equal 120°. Clearly, T is unique. The proof is complete. □

Acknowledgments. The author wishes to thank M. Brazil, J. H. Rubinstein, D. A. Thomas and N. C. Wormald for discussion in the preparation of this paper. He also wishes to thank an anonymous referee for the very careful reading and helpful suggestions.

REFERENCES

[1] E. J. COCKAYNE, *On the efficiency of the algorithm for Steiner minimal trees*, SIAM J. Appl. Math., 18 (1970), pp. 150–159.

[2] D. Z. DU, F. K. HWANG, G. D. SONG, AND G. Y. TING, *Steiner minimal trees on sets of four points*, Discrete Comput. Geom., 2 (1987), pp. 401–414.

[3] E. N. GILBERT AND H. O. POLLAK, *Steiner minimal trees*, SIAM J. Appl. Math., 16 (1968), pp. 1–29.

[4] F. K. HWANG AND J. F. WENG, *The shortest networks under a given topology*, J. Algorithms, 13 (1992), pp. 468–488.

[5] Z. A. MELZAK, *On the problem of Steiner*, Canad. Math. Bull., 4 (1961), pp. 143–148.

[6] H. O. POLLAK, *Some remarks on the Steiner problem*, J. Combin. Theory Ser. A, 24 (1978), pp. 278–295.

[7] J. H. RUBINSTEIN AND D. A. THOMAS, *A variational approach to the Steiner network problem*, Ann. Oper. Res., 33 (1991), pp. 481–499.

[8] J. F. WENG, *Variational approach and Steiner minimal trees on four points*, Discrete Math., 132 (1994), pp. 349–362.

[9] J. F. WENG, *Steiner polygons in the Steiner problem*, Geometriae Dedicata, 52 (1994), pp. 119–127.

SIAM J. Discrete Math.
Vol. 10, No. 3, pp. 399–430, August 1997

ASTEROIDAL TRIPLE-FREE GRAPHS[*]

DEREK G. CORNEIL[†], STEPHAN OLARIU[‡], AND LORNA STEWART[§]

Abstract. An independent set of three vertices such that each pair is joined by a path that avoids the neighborhood of the third is called an *asteroidal triple*. A graph is asteroidal triple-free (AT-free) if it contains no asteroidal triples. The motivation for this investigation was provided, in part, by the fact that the AT-free graphs provide a common generalization of interval, permutation, trapezoid, and cocomparability graphs. The main contribution of this work is to investigate and reveal fundamental structural properties of AT-free graphs. Specifically, we show that every connected AT-free graph contains a dominating pair, that is, a pair of vertices such that every path joining them is a dominating set in the graph. We then provide characterizations of AT-free graphs in terms of dominating pairs and minimal triangulations. Subsequently, we state and prove a decomposition theorem for AT-free graphs. An assortment of other properties of AT-free graphs is also provided. These properties generalize known structural properties of interval, permutation, trapezoid, and cocomparability graphs.

Key words. asteroidal triples, asteroidal triple-free graphs, interval graphs, permutation graphs, trapezoid graphs, cocomparability graphs, dominating pairs, graph decompositions, structural graph theory

AMS subject classifications. 05C75, 68R10

PII. S0895480193250125

1. Introduction.

The original motivation for this work was provided by the linear structure that is apparent in various families of graphs, including interval graphs, permutation graphs, trapezoid graphs, and cocomparability graphs. Somewhat surprisingly, the linearity of interval, permutation, trapezoid, and cocomparability graphs is described in terms of different and seemingly ad hoc properties of each of these classes of graphs. For example, in the case of interval graphs, the linearity property is traditionally expressed in terms of a linear order on the set of maximal cliques [3, 4]. For permutation graphs, the linear behavior is explained in terms of the underlying partial order of dimension two [1]. For cocomparability graphs, the linear behavior is expressed in terms of the well-known linear structure of comparability graphs [17], and so on. Our intention is to provide a unifying look at these classes in the hope of identifying the "agent" responsible for their linear behavior.

Before proceeding, it is perhaps appropriate to recall a few definitions. A graph is an *interval graph* if its vertices can be put in a one-to-one correspondence with a

[*] Received by the editors April 29, 1993; accepted for publication (in revised form) May 31, 1996. A preliminary version of this paper appeared as "On the linear structure of graphs: Asteroidal triple-free graphs" in *Graph Theoretic Concepts in Computer Science WG* '93, Utrecht, The Netherlands, 1993, Lecture Notes in Computer Science 790, J. van Leeuwen, ed., Springer-Verlag, Berlin, 1994, pp. 211–224.

http://www.siam.org/journals/sidma/10-3/25012.html

[†] Department of Computer Science, University of Toronto, Toronto, Ontario M5S 1A4, Canada (dgc@cs.toronto.edu). The research of this author was supported by financial assistance from the Natural Sciences and Engineering Research Council of Canada.

[‡] Department of Computer Science, Old Dominion University, Norfolk, VA 23529-0162 (olariu@cs.odu.edu). The research of this author was supported in part by National Science Foundation grants CCR-9407180 and CCR-9522093 and by ONR grant N00014-97-1-0526.

[§] Department of Computing Science, University of Alberta, Edmonton, Alberta T6G 2H1, Canada (stewart@cs.ualberta.ca). The research of this author was supported by financial assistance from the Natural Sciences and Engineering Research Council of Canada and in 1992 to 1993 by Laboratoire ARTEMIS, IMAG, Grenoble, and the Government of France.

set of intervals on the real line in such a way that two vertices are adjacent if and only if the corresponding intervals overlap. A graph is a *comparability graph* if the edges may be given a transitive orientation. A *cocomparability graph* is the complement of a comparability graph. A graph that is at the same time a comparability and a cocomparability graph is said to be a *permutation graph* [13].

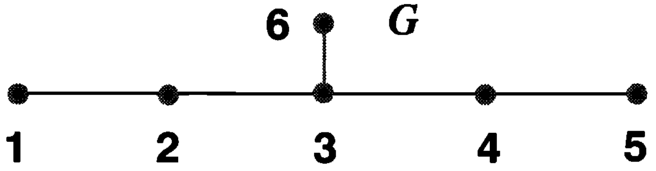


FIG. 1.1. *A graph G.*

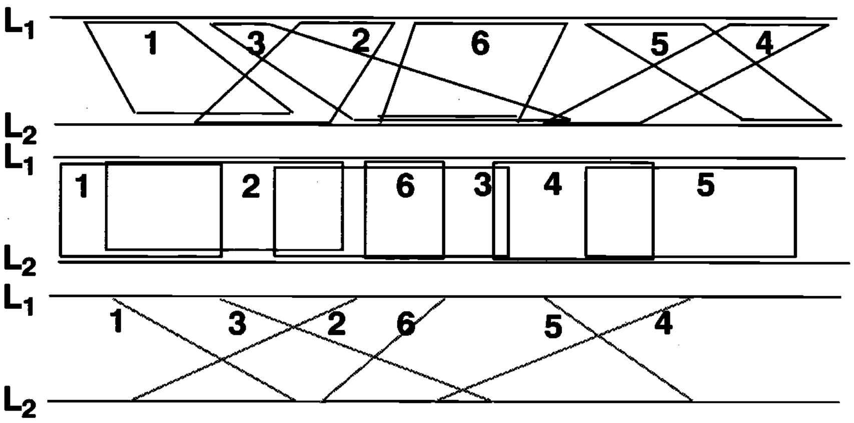


FIG. 1.2. *Trapezoid, interval, and permutation models of the graph in Figure* 1.1.

A *trapezoid representation* R consists of two parallel lines (denoted L_1 and L_2) and some trapezoids with two endpoints lying on L_1 and the other two lying on L_2. A graph G is a *trapezoid graph* if it is the intersection graph of such a representation. Specifically, the vertices of G are in one-to-one correspondence with the trapezoids in R and two vertices in G are adjacent if and only if their corresponding trapezoids intersect. If the trapezoids degenerate with the endpoints on L_1 (respectively, L_2) coinciding (i.e., the trapezoids become lines), then the intersection graph is a permutation graph. Similarly, if the intervals on L_1 are the mirror image of the intervals on L_2, then the intersection graph is an interval graph. The reader is referred to Figure 1.2 for an illustration of these notions for the graph presented in Figure 1.1. It is shown in [6] that permutation graphs and interval graphs are strictly contained in trapezoid graphs. Furthermore, trapezoid graphs are strictly contained in cocomparability graphs [5]. Cocomparability graphs, and thus trapezoid, permutation, and interval graphs, are *perfect* in the sense of Berge [15]; i.e., for every induced subgraph the chromatic number equals the clique number.

The trapezoid representation that provides the common thread with interval and permutation graphs also indicates that, in some sense, the graphs can only "grow" linearly. For example, referring to the graph in Figure 1.1 which is at the same time an interval, trapezoid, permutation, and cocomparability graph, we can add a new vertex adjacent to one of the vertices 1, 2, 3, 4, or 5 without destroying membership in any of these families; however, when looking at various intersection models of G featured in Figure 1.2, it seems as though we cannot add a new vertex adjacent to 6 without destroying membership in each family.

More than three decades ago Lekkerkerker and Boland [18] set out to identify the property that prevented a *chordal graph*, namely, a graph in which every cycle of length at least four has a chord, from "growing" in three directions at once. For this purpose, they defined an *asteroidal triple* to be an independent set of three vertices such that each pair of vertices is joined by a path that avoids the neighborhood of the third. For an illustration, the reader is referred to Figure 1.3, which features various instances of asteroidal triples.

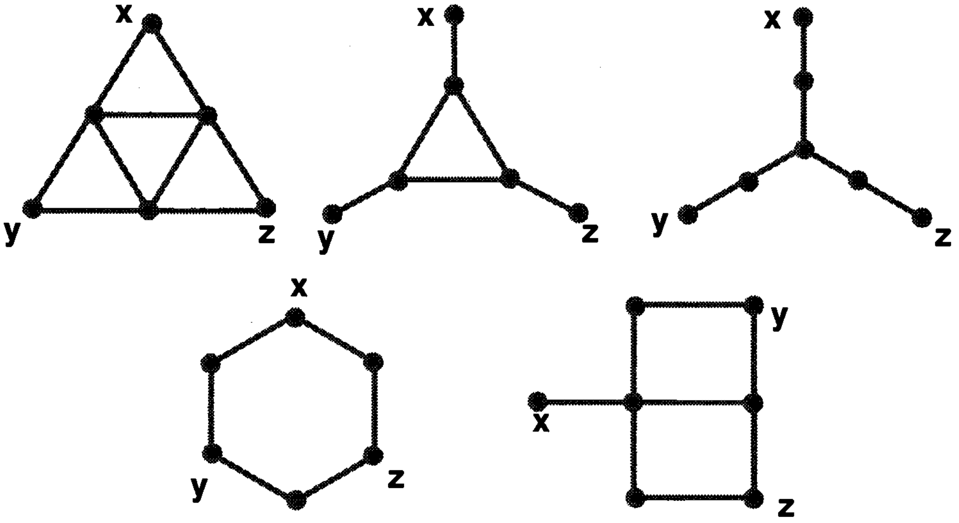


FIG. 1.3. *Various examples of asteroidal triples.*

Lekkerkerker and Boland [18] demonstrated the importance of asteroidal triples in the following theorem.

THEOREM 1.1 (see [18]). *A graph is an interval graph if and only if it is chordal and asteroidal triple-free.*

Thus, it appears that the condition of being asteroidal triple-free (AT-free) prohibits a chordal graph from growing in three directions at once. The top three graphs in Figure 1.3 are examples of chordal graphs that are not interval graphs.

Later, Golumbic, Monma, and Trotter Jr. [16] showed that cocomparability graphs (and, thus, permutation and trapezoid graphs) are also AT-free. Subsequently, it was shown that the perfect AT-free graphs strictly contain the cocomparability graphs [5]. Since C_5 is AT-free, the AT-free graphs need not be perfect. However, an easy argument shows that the celebrated Strong Perfect Graph conjecture is true for AT-free graphs [19].

Three decades ago Gallai [14], in his monumental work on comparability graphs, obtained the first characterization of AT-minimal graphs (i.e., graphs that contain an asteroidal triple and are minimal with this property) in terms of 15 families of subgraphs. Actually, Gallai's list is not complete. Since he was only interested in graphs with no induced C_5, all the AT-minimal graphs containing a C_5 are missing from [14]. For a full list of AT-minimal graphs the interested reader is referred to [7]. After Gallai's paper, little work was done on AT-free graphs.

The main contribution of this work is to provide a number of structural results concerning AT-free graphs. Our main results[1] are as follows.

1. We show that every connected AT-free graph has a dominating pair, that is, a pair of vertices such that every path joining them is a dominating set.
2. We provide properties of dominating pairs in AT-free graphs related to the concepts of connected domination and diameter.
3. We provide a characterization of AT-free graphs in terms of dominating pairs.
4. We provide a characterization of AT-free graphs in terms of minimal triangulations.
5. We provide a decomposition theorem for AT-free graphs.

The remainder of this work is organized as follows. Section 2 provides background material along with definitions of technical terms used throughout the paper. In section 3 we study the existence of dominating pairs in connected AT-free graphs. In section 4 we discuss properties of dominating pairs in the context of connected domination and show that some dominating pair achieves the diameter of the graph. In section 5 we offer two characterizations of AT-free graphs. Specifically, we provide characterizations of AT-free graphs in terms of dominating pairs and in terms of minimal triangulations. In section 6 we show that an AT-free graph may be extended to another AT-free graph by attaching, to each vertex in an appropriate dominating pair, a new vertex of degree one. This result leads to a decomposition theorem for AT-free graphs, whereby an AT-free graph is reduced to a single vertex by a sequence of contractions. In section 7 we show that in AT-free graphs of diameter greater than three, the sets of vertices that can be in dominating pairs are restricted to two disjoint sets, thus strengthening the intuition about the linear structure of this class of graphs. Finally, section 8 offers concluding remarks and poses some open problems.

2. Preliminaries.

All graphs in this paper are finite with no loops or multiple edges. We use standard graph-theoretic terminology compatible with [2], to which we refer the reader for basic definitions.

As usual, we shall write $G = (V, E)$ to denote a graph G with vertex set V and edge set E. The *complement* of a graph G is the graph $\overline{G}$ having the same vertex set as G; distinct vertices u and v are adjacent in $\overline{G}$ if and only if they are nonadjacent in G. For a vertex x in G, $N_G(x)$ denotes the set of all the vertices adjacent to x in G. The *degree* of vertex x in the graph G, denoted by $d_G(x)$, is the cardinality of $N_G(x)$. A vertex x will be said to be *pendant* if its degree is one. We let $N'_G(x)$ stand for the set of all the vertices adjacent to x in the complement $\overline{G}$ of G. The notation will be shortened to $N(x)$, $d(x)$, and $N'(x)$, respectively, whenever the context permits. If H is a subset of the vertex set V of G, then G_H will denote the subgraph of G induced by H. Occasionally, if no confusion is possible, we shall use H as a shorthand for G_H.

A *path* is a sequence $v_0, v_1, \ldots, v_p$ of distinct vertices of G with $v_{i-1}v_i \in E$ for all i $(1 \leq i \leq p)$. A *chord* in a path $v_0, v_1, \ldots, v_p$ is an edge v_iv_j with i and j differing by more than one. A *cycle* of length $p + 1$ is a sequence $v_0, v_1, \ldots, v_p$ of distinct vertices of G such that $v_{i-1}v_i \in E$ for all i $(1 \leq i \leq p)$ and $v_pv_0 \in E$. We let P_n and C_n denote the chordless path and cycle with n vertices, respectively. Unless stated otherwise, all paths in this work will be assumed to be chordless.

[1] For undefined terms the reader is referred to section 2.

A set S of vertices of graph G is said to be *dominating* if every vertex outside S is adjacent to some vertex in S. Among dominating sets S that induce connected subgraphs of G, one is often interested in those that have minimum cardinality. In the remainder of this paper such a dominating set will be referred to as an *mccds*. An mccds that induces a path will be referred to as a *path-mccds*.

A path joining vertices x and y is termed an x, y-path. A vertex u *misses* a path π if u is adjacent to no vertex on π; otherwise, u *intercepts* π. In a connected graph, a pair (u, v) of vertices is termed a *dominating pair* if all u, v-paths are dominating. For vertices u and v of graph G, we let $D(u, v)$ denote the set of vertices that intercept all u, v-paths. In this terminology, (u, v) is a dominating pair whenever $D(u, v) = V$. For vertices u, v, and x of graph G, we say that u and v are *unrelated with respect to* x if $u \notin D(v, x)$ and $v \notin D(u, x)$.

Given a connected graph $G = (V, E)$, the distance $d_G(u, v)$ (or $d(u, v)$, for short) between vertices u and v is the length of a shortest path in G joining u and v. The *diameter* of G is defined as

$$\mathrm{diam}(G) = \max_{u,v \in V} d_G(u, v).$$

Two vertices u and v such that $d(u, v) = \mathrm{diam}(G)$ are said to *achieve* the diameter.

3. Dominating pairs in AT-free graphs. The main purpose of this section is to prove a fundamental domination-related property of AT-free graphs. To state this property, recall that a pair of vertices (x, y) is a dominating pair in a graph G if all x, y-paths in G are dominating sets. As it turns out, connected AT-free graphs always contain dominating pairs. Although it is straightforward to see that connected interval, permutation, trapezoid, and cocomparability graphs all contain dominating pairs, it is somewhat surprising that, up to now, this property had not been noticed for these classes of graphs.

Throughout this section, we assume a connected AT-free graph $G = (V, E)$ along with an arbitrary vertex x of G. We are now in a position to state the main result of this section.

THEOREM 3.1. *Every connected AT-free graph contains a dominating pair.*

The conclusion of Theorem 3.1 is implied by the following stronger result.

THEOREM 3.2. *Let x be an arbitrary vertex of a connected AT-free graph G. Either (x, x) is a dominating pair or else for a suitable choice of vertices y and z in $N'(x)$, (y, x) or (y, z) is a dominating pair.*

Our proof of Theorem 3.2 relies on a number of intermediate results about connected AT-free graphs that we present next.

CLAIM 3.3. *Let u, v, and w be arbitrary vertices of G. If $u \in D(v, x)$, $w \in D(u, x)$, and u and w are not adjacent, then $w \in D(v, x)$.*

Proof. Suppose that w misses some v, x-path π: $v = v_0, v_1, \ldots, v_k = x$. Let j be the largest subscript for which u is adjacent to vertex v_j of π: since $u \in D(v, x)$, such a subscript must exist. But now, w misses the u, x-path, $u, v_j, v_{j+1}, \ldots, v_k = x$, contradicting that $w \in D(u, x)$. □

In the remainder of this section, we shall use "unrelated" as a shorthand for "unrelated with respect to x." The reader is referred to Figure 3.1 for an illustration. The paths confirming that vertices u and v are unrelated are drawn in heavy lines. We further assume that F is an arbitrary connected component of $N'(x)$.

CLAIM 3.4. *F contains no unrelated vertices.*

Proof. If u and v are unrelated vertices in F, then the connectedness of F implies that $\{u, v, x\}$ is an asteroidal triple. □

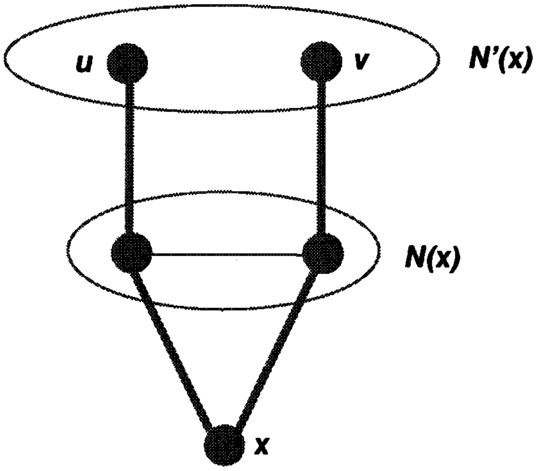


FIG. 3.1. *Illustrating unrelated vertices.*

CLAIM 3.5. *If u and v are vertices in F and if $v \notin D(u,x)$, then $D(u,x) \subset D(v,x)$.*

Proof. From Claim 3.4 it follows that $u \in D(v,x)$. Let w be an arbitrary vertex in $D(u,x) \backslash D(v,x)$. Clearly $w \notin N(x)$. If w and u are not adjacent, then Claim 3.3 guarantees that $w \in D(v,x)$; if w and u are adjacent, then clearly $w \in F$. If w misses some v,x-path then, in particular, v and w are not adjacent. Thus, with π standing for some u,x-path missed by v, $\pi \cup \{w\}$ contains a w,x-path missed by v. But now, v and w are unrelated, contradicting Claim 3.4. Consequently, $w \in D(v,x)$ and $D(u,x) \subseteq D(v,x)$; the inclusion is strict since $v \notin D(u,x)$. □

A vertex y in F is called *special* if $D(u,x) \subseteq D(y,x)$ for all vertices u in F. The following statement provides a characterization of special vertices.

CLAIM 3.6. *A vertex y in F is special if and only if $F \subseteq D(y,x)$.*

Proof. First, if the vertex y is special then, for every vertex v in F, $D(v,x) \subseteq D(y,x)$. In particular, $v \in D(v,x)$, implying that $F \subseteq D(y,x)$.

Conversely, suppose that $F \subseteq D(y,x)$. Let u be an arbitrary vertex in F and let w be an arbitrary vertex in $D(u,x)$. If w belongs to F then, since $F \subseteq D(y,x)$, $w \in D(y,x)$; if w does not belong to F, then u and w are not adjacent and Claim 3.3 guarantees that $w \in D(y,x)$, confirming that $D(u,x) \subseteq D(y,x)$. Since u is arbitrary, the claim follows. □

CLAIM 3.7. *F contains a special vertex.*

Proof. Choose a vertex y in F with $D(y,x) \subset D(t,x)$ for no vertex t in F. If y is not special then, by Claim 3.6, we find a vertex v in F with $v \notin D(y,x)$. By Claim 3.5, $D(y,x) \subset D(v,x)$, contradicting our choice of y. □

CLAIM 3.8. *Let v be an arbitrary vertex in $N'(x) \setminus F$. Either $v \in D(w,x)$ for all vertices w in F or $v \notin D(w,x)$ for all vertices w in F.*

Proof. Suppose not. For a suitable choice of vertices w and w' in F, we have $v \in D(w,x)$ and $v \notin D(w',x)$. Let π stand for a w',x-path missed by v, and let π' stand for a w,w'-path entirely within F. But now $\pi \cup \pi'$ contains a w,x-path missed by v, contrary to our assumption. □

CLAIM 3.9. *Let v be a vertex in $N'(x) \setminus F$. If $F \not\subseteq D(v,x)$ then, for a special vertex u^* in F, $u^* \notin D(v,x)$.*

Proof. Write $U = \{u \in F \mid u \notin D(v,x)\}$. Since $F \not\subset D(v,x)$, U is nonempty. Choose a vertex u^* in U such that $D(u^*,x) \subset D(u,x)$ for no vertex u in U. If u^* is not special then, by Claim 3.6, there exists some vertex w in $F \backslash D(u^*,x)$. In particular, u^* and w are not adjacent. By Claim 3.5, $D(u^*,x) \subset D(w,x)$; by our choice of u^*, w must belong to $F \setminus U$. This, however, implies that $w \in D(v,x)$. Since $w \notin D(u^*,x)$, Claim 3.4 implies that $u^* \in D(w,x)$. Since u^* and w are not adjacent, Claim 3.3 guarantees that $u^* \in D(v,x)$, which is the desired contradiction. □

Call a vertex u of $N'(x)$ *strong* if $N'(x) \subset D(u,x)$. It is easy to verify that if u is a strong vertex, then (u,x) is a dominating pair in G. From now on, we shall tacitly assume that $N'(x)$ contains no strong vertices. A pair (y,z) of vertices in distinct components of $N'(x)$ is an *admissible pair* if $D(y,x) \cup D(z,x) \subset D(t,x) \cup D(t',x)$ for no vertices t, t' in distinct components of $N'(x)$.

Notice that if $N'(x)$ is connected, Claim 3.7 implies that $N'(x)$ contains a special vertex which, by virtue of Claim 3.6, is strong. We shall, therefore, assume that $N'(x)$ is disconnected. Now, the absence of strong vertices in $N'(x)$ guarantees the existence of admissible pairs. As it turns out, admissible pairs play a crucial role in our arguments. We now study some of their properties.

CLAIM 3.10. *Let Y and Z be two distinct components of $N'(x)$ and let vertices y in Y and z in Z be an admissible pair. Then $Y \not\subset D(z,x)$ and $Z \not\subset D(y,x)$.*

Proof. Assume $Z \subset D(y,x)$. Then, in particular, $z \in D(y,x)$. To see that $D(z,x) \subseteq D(y,x)$, note that for an arbitrary vertex w in $D(z,x)$, $w \in D(y,x)$ whenever $w \in Z$ and that, by virtue of Claim 3.3, $w \in D(y,x)$ whenever $w \notin Z$.

Since y is not strong, we find a vertex y' in $N'(x) \backslash D(y,x)$. But now, either (z,y') or (y,y') contradicts our choice of (y,z). To see this, note that if y' belongs to Y then, by Claim 3.5, $D(y,x) \subset D(y',x)$, and so

$$D(y,x) \cup D(z,x) = D(y,x) \subset D(y',x) \subseteq D(y',x) \cup D(z,x).$$

If y' does not belong to Y, then $D(y,x) \cup D(z,x) = D(y,x) \subseteq D(y',x) \cup D(y,x)$. Since y' does not belong to $D(y,x)$, the inclusion is strict. The fact that $Y \not\subset D(z,x)$ follows by a similar argument. □

CLAIM 3.11. *If (y,z) is an admissible pair, then $N'(x) \subset D(y,x) \cup D(z,x)$.*

Proof. We assume, without loss of generality, that vertices y and z belong to distinct connected components Y and Z of $N'(x)$, respectively. If the claim is false, we find a vertex w in $N'(x) \backslash (D(y,x) \cup D(z,x))$. Clearly, $w \notin D(y,x)$ and $w \notin D(z,x)$.

Since G is AT-free, it is easy to verify that

(3.1) no distinct vertices t, t', t'' in $N'(x)$ are pairwise unrelated with respect to x.

We claim that

$$w \text{ does not belong to } Y \cup Z. \tag{3.2}$$

If the vertex w belongs to Y then, by Claim 3.5, $D(y,x) \subset D(w,x)$, and since $w \notin D(z,x)$,

$$D(y,x) \cup D(z,x) \subset D(z,x) \cup D(w,x),$$

contradicting that (y,z) is an admissible pair. The proof of the fact that $w \notin Z$ is similar and, thus, is omitted.

Further, we claim that for a suitable choice of vertices u and v in $N'(x)$

(3.3) $u \in D(y,x) \backslash (D(z,x) \cup D(w,x))$ and $v \in D(z,x) \backslash (D(y,x) \cup D(w,x))$.

To justify (3.3), observe that by (3.2) y, z, and w belong to distinct components of $N'(x)$. Since (y, z) is an admissible pair,

$$D(y,x) \cup D(z,x) \not\subset D(z,x) \cup D(w,x),$$

and, therefore, the required vertex u exists. A similar argument asserts the existence of vertex v.

Next, we claim that

$$y \in D(z,x) \cup D(w,x) \text{ and } z \in D(y,x) \cup D(w,x). \tag{3.4}$$

To see this, note that if $y \notin D(z,x) \cup D(w,x)$, then our choice of w guarantees that y and w are unrelated. Therefore, it must be that $z \in D(y,x) \cup D(w,x)$, for otherwise y, z, and w would be pairwise unrelated, contradicting (3.1). Consider the vertex v specified in (3.3); since $z \in D(y,x) \cup D(w,x)$ and $v \in D(z,x) \backslash (D(y,x) \cup D(w,x))$, Claim 3.3 implies that z and v are adjacent. But now $\{y, v, w\}$ is an asteroidal triple. This follows since y and w are unrelated, and both v, w and v, y are unrelated by (3.3) and Claim 3.8. Along similar lines, one can prove that $z \in D(y,x) \cup D(w,x)$. Thus, (3.4) must hold.

Further, we claim that

$$u \in Y \text{ and } v \in Z. \tag{3.5}$$

By (3.4), $y \in D(z,x) \cup D(w,x)$; by (3.3), $u \in D(y,x) \backslash (D(z,x) \cup D(w,x))$. It follows that u and y are adjacent, for otherwise we contradict Claim 3.3. The fact that $v \in Z$ is proved similarly.

To complete the proof of Claim 3.11, we first observe that (3.5), (3.3), and Claim 3.8 combined guarantee that $u \notin D(v,x)$ and $v \notin D(u,x)$, and so u and v are unrelated. Similarly, by (3.5), (3.3), and Claim 3.8, the vertices u and w are unrelated, as are v and w. But now, the vertices u, v, and w are pairwise unrelated, contradicting (3.1). With this, the proof of Claim 3.11 is complete. □

We are now in a position to give the proof of Theorem 3.2.

Proof (Theorem 3.2). If $N'(x)$ is empty, then (x, x) is a dominating pair. If $N'(x)$ is nonempty but contains a strong vertex y, then clearly (x, y) is a dominating pair. Otherwise, let (y, z) be an admissible pair in $N'(x)$. We assume, without loss of generality, that y and z belong to distinct connected components Y and Z of $N'(x)$, respectively. By Claims 3.10, 3.9, and 3.8 we find special vertices y^* in Y and z^* in Z such that $y^* \notin D(z^*, x)$ and $z^* \notin D(y^*, x)$. Put differently, y^* and z^* are unrelated. Furthermore, since y^* and z^* are special, we have $D(y,x) \cup D(z,x) \subseteq D(y^*,x) \cup D(z^*,x)$, implying that (y^*, z^*) is also an admissible pair.

We claim that

$$(y^*, z^*) \text{ is a dominating pair in } G.$$

By Claim 3.11, any vertex v that misses some y^*, z^*-path must be in $N(x)$. (Observe that v and x are distinct, since every y^*, z^*-path contains at least one vertex in $N(x)$.) Since y^* and z^* are unrelated, y^* misses some z^*, x-path π and z^* misses some y^*, x-path π'. But now we have reached a contradiction—$\{y^*, z^*, v\}$ is an asteroidal triple. To see this, note that, by assumption, v misses some y^*, z^*-path; in addition, y^* misses the z^*, v-path $\pi \cup \{v\}$ and z^* misses the y^*, v-path $\pi' \cup \{v\}$. □

It is perhaps interesting to note that Claim 3.4 suggests the following characterization of AT-free graphs. The proof is immediate and is left to the reader.

THEOREM 3.12. *A graph G is AT-free if and only if for every vertex x of G, no component F of $N'(x)$ contains unrelated vertices.*

4. Distance properties of dominating pairs. The purpose of this section is to examine various distance-related properties featured by dominating pairs in connected AT-free graphs. Specifically, we study the maximum distance between vertices of a dominating pair, as well as the relationship between dominating pairs and minimum cardinality-connected dominating sets. In particular, we show that in every connected AT-free graph some dominating pair achieves the diameter (Theorem 4.3) and some dominating pair forms the endpoints of a path-mccds (Theorem 4.6). To begin, we state a property of connected AT-free graphs that will be used throughout this section.

Claim 4.1. *A connected AT-free graph G is a clique if and only if it contains no nonadjacent dominating pair.*

Proof. The "only if" part is trivial. To prove the "if" part, note that if G is not a clique then, for some vertex x of G, $N'(x)$ is nonempty. By Theorem 3.2, there exist vertices $y, z \in N'(x)$ such that either (x, y) is a dominating pair (with x and y nonadjacent) or, failing this, (y, z) is a dominating pair. In the latter case, the vertices y and z belong to distinct connected components of $N'(x)$ and, consequently, must be nonadjacent. □

In the remainder of this section we assume a connected AT-free graph G which is not a clique. Claim 4.1 guarantees that we can find a nonadjacent dominating pair (x, y_0) in G. Let F be the connected component of $N'(x)$ containing y_0, and let Y be the set of vertices y in F for which (x, y) is a dominating pair in G. A vertex a in $F \setminus Y$ is called an *attractor* if $Y \subset D(a, x)$.

Claim 4.2. *F contains no attractors.*

Proof. If the statement is false then the set A of attractors in $F \setminus Y$ is nonempty. Let a^* be a vertex in A for which $D(a^*, x) \subset D(a, x)$ for no vertex a in A. We claim that (a^*, x) is a dominating pair in G. If the statement is false, we find a vertex t that misses some a^*, x-path π. However,

(i) $t \notin A$ by our choice of a^* and Claim 3.5 combined,
(ii) $t \notin Y$ because $Y \subset D(a^*, x)$,
(iii) $t \notin N'(x) \setminus F$, for otherwise t would miss a y_0, x-path (such a path is contained in the concatenation of π with a y_0, a^*-path in F),
(iv) $t \notin F \setminus (A \cup Y)$. Since $Y \subset D(a^*, x)$, t must be adjacent to every vertex in Y, implying that t belongs to A, which is a contradiction. □

The next result concerns the maximum distance between vertices in a dominating pair.

Theorem 4.3. *In every connected AT-free graph some dominating pair achieves the diameter.*

Our proof of Theorem 4.3 relies on the following intermediate result.

Lemma 4.4. *Let G be a connected AT-free graph and let vertices x and a of G be such that $d(x, a) = \operatorname{diam}(G)$. If (x, y) is a dominating pair with vertex y in $N'(x)$, then there exists a vertex z such that (x, z) is a dominating pair and $d(x, z) = \operatorname{diam}(G)$.*

Proof. Clearly, we may assume that $d(x, a) \geq 2$. Let Y be the set of vertices y in $N'(x)$ such that (x, y) is a dominating pair.

We assume that a does not belong to Y, for otherwise there is nothing to prove. Observe that Y is contained in the component of $N'(x)$ containing a; otherwise, $d(x, y) = 2$ and $d(x, a) = 2$, since every path joining x and y must dominate a.

By virtue of Claim 4.2, a cannot be an attractor; we find a vertex y in Y such that $y \notin D(a, x)$. In particular, a and y are nonadjacent. Consider an arbitrary shortest x,y-path $\pi(x,y)$: $x = u_0,\ u_1, \ldots,\ u_k = y$. Since (x,y) is a dominating pair, a must be adjacent to some vertex u_j. Since a and y are nonadjacent, $j < k$. But now, $\text{diam}(G) = d(x,a) \le d(x,u_j) + 1 \le d(x,y) \le \text{diam}(G)$, implying that (x,y) is a dominating pair with $d(x,y) = \text{diam}(G)$. This completes the proof of Lemma 4.4. □

We now give a proof of Theorem 4.3.

Proof (Theorem 4.3). Let vertices x and a be such that $d(x,a) = \text{diam}(G)$. Let C be the connected component of $N'(x)$ containing a. We may assume that x is in no dominating pair involving a vertex in $N'(x)$; otherwise we are done by Lemma 4.4. By the proof of Theorem 3.2, there exists a dominating pair (y,z) with vertices y and z belonging to distinct components of $N'(x)$. We observe that precisely one of y and z belongs to C; otherwise, $d(y,z) = 2$ and we are done. (To see this, note that if neither of y and z is in C, then a must be adjacent to a neighbor of x; therefore, $\text{diam}(G) = d(a,x) = 2$ and $2 \le d(y,z) \le \text{diam}(G)$, implying that (y,z) is a dominating pair of distance $\text{diam}(G)$.) Furthermore, we may assume that $d(y,z) < \text{diam}(G)$; otherwise, (y,z) is the desired dominating pair.

Assume without loss of generality that y belongs to C and that z belongs to some component C' ($\neq C$) of $N'(x)$. If there exists a shortest z,y-path $\pi(z,y)$: $z = u_0,\ u_1, \ldots,\ u_k = y$ such that a is adjacent to u_j, for some $j < k$, then $\text{diam}(G) = d(x,a) \le d(x,u_j) + 1 \le d(z,u_j) + 1 \le d(z,y) \le \text{diam}(G)$, and (y,z) is the required dominating pair. Otherwise, y is the only vertex on π adjacent to a and $\text{diam}(G) = d(x,a) \le d(a,z) \le d(y,z) + 1 \le \text{diam}(G)$. Therefore, $d(a,z) = \text{diam}(G)$ and the conclusion follows by Lemma 4.4. □

Thus, in a connected AT-free graph, some dominating pair achieves the diameter. We now consider shortest dominating paths and their relation to connected dominating sets. In the remainder of this section we shall find it convenient to make use of a special notation that we now introduce. When referring to a path π, we shall denote by $\pi - y$ the path obtained from π by removing y, one of its endpoints. Similarly, we let $\pi + x$ denote the path obtained from π by the addition of x as a new endpoint.

THEOREM 4.5. *Every connected AT-free graph has a path-mccds.*

Proof. Let G be a connected AT-free graph, let D be an arbitrary mccds, and let (x,y) be an arbitrary dominating pair in G. We may assume that $|D| \ \ge 3$; otherwise there is nothing to prove. We note that

$$\text{(4.1)} \qquad \text{if } \{x,y\} \subset D \text{ then } D \text{ induces a path.}$$

This follows from the fact that every x,y-path π in D is a connected dominating set, implying that $D = \pi$.

Next, we claim that

$$\text{(4.2)} \qquad \text{if } x \in D \text{ or } y \in D \text{ then some mccds induces a path.}$$

To justify (4.2) assume, without loss of generality, that $x \in D$. By (4.1), we may assume that $y \notin D$. Let Y consist of all the vertices in D adjacent to y. Since D is connected, we find a path π joining x and a vertex y' in Y such that all vertices in $\pi - y'$ are in $D \backslash Y$. Either $D = \pi$ or $\pi + y$ is a dominating path of cardinality at most $|D|$. Thus, (4.2) must hold.

By (4.1) and (4.2) combined we may assume that neither x nor y belongs to D. Let X and Y be the sets of vertices in D adjacent to x and y, respectively. Observe that X and Y must be disjoint, for otherwise with w standing for an arbitrary vertex in $X \cap Y$, $\{x,w,y\}$ induces a dominating path and there is nothing to prove. Connectedness of

D guarantees the existence of vertices x' in X, y' in Y, and of an x', y'-path π in D, all of whose internal vertices are in $D\backslash(X \cup Y)$. We claim that

$$|D\backslash\pi| \ = 1. \tag{4.3}$$

To see that this is the case, observe that if $D = \pi$ then we are done; if $|D\backslash\pi| \ > 1$, then $\pi + x + y$ is a dominating path of cardinality at most $|D|$. Thus, (4.3) must hold.

By (4.3) we write $\{z\}$=$D\backslash\pi$. Since the path $\pi + x$ is of cardinality $|D|$, we find a vertex u that misses $\pi + x$. Similarly, since the path $\pi + y$ is of cardinality $|D|$, we find a vertex v that misses $\pi + y$. The following are easily seen:

• $u \neq v$ and $uy,\ vx$ are edges (otherwise, we contradict that (x, y) is a dominating pair),

• u and v are not adjacent (else $\{u, x, y'\}$ is an AT in G),

• $u \neq z$, $v \neq z$, and both $uz,\ vz$ are edges (otherwise, we contradict that D is a connected dominating set),

• $x'z,\ y'z$ are both edges (if $x'z$ is not an edge, then $\{u, x', v\}$ is an AT). We claim that

$$\{u, z, v\} \text{ is an mccds.}$$

To see this, let w be a vertex that misses the path induced by $\{u, z, v\}$. Since D is dominating, w must be adjacent to some vertex on π. But now, it is easy to confirm that $\{u, v, w\}$ is an AT. □

Next, we show that Theorem 4.5 can be strengthened.

THEOREM 4.6. *In every connected AT-free graph the endpoints of some path-mccds are a dominating pair.*

Our proof of Theorem 4.6 relies on the following technical result.

LEMMA 4.7. *Let G be a connected AT-free graph and let $\pi(x, a)$ be a path-mccds in G with endpoints a and x. If x belongs to a dominating pair involving a vertex in $N'(x)$, then there exists a vertex y in $N'(x)$ such that (x, y) is a dominating pair and each shortest x, y-path is an mccds.*

Proof. Write $\pi(x, a)$: $x = u_0,\ u_1, \ldots, u_k = a$. We may assume that $k \geq 2$. Let C be the component of $N'(x)$ containing a. Observe that every vertex that forms a dominating pair with x must belong to C. To clarify this, suppose such a vertex t belongs to a component C' distinct from C. Then, since the path $\pi(x, a)$ is dominating, t is adjacent to u_1, implying that $d(x, t) = 2 \leq k$, and there is nothing to prove.

Let Y be the set of all special vertices in C. It is easy to see that x forms a dominating pair with every vertex in Y. Thus, we may assume that $a \notin Y$. Note that if some vertex in Y is adjacent to u_j with $j < k$ then we are done; otherwise, a is an attractor, contradicting Claim 4.2. This completes the proof of Lemma 4.7. □

Proof (Theorem 4.6). For convenience, we inherit the notation of Lemma 4.7. We may assume that $\pi(x, a)$ is a path-mccds and that x is in no dominating pair involving a vertex in $N'(x)$; otherwise we are done by Lemma 4.7. By the proof of Theorem 3.2, there exists a dominating pair (y, z) with y and z in distinct components of $N'(x)$. We observe that precisely one of the vertices y and z belongs to C; otherwise, $d(y, z) = 2 \leq k$ and we are done.

Assume without loss of generality that $y \in C$ and that z belongs to a component C' distinct from C. Note that since $\pi(x, a)$ is dominating, z is adjacent to u_1. Thus, y is adjacent to a and to no other vertex on $\pi(x, a)$, for otherwise $d(y, z) \leq k$.

We claim that at least one of the paths $\pi(x, a) - x + y$ or $\pi(x, a) - x + z$ is dominating. Observe that both of these paths are of length k and each of them is anchored at a vertex belonging to a dominating pair. Therefore, once we establish this claim the conclusion of Theorem 4.6 follows from Lemma 4.7. If neither of these paths is dominating then

- there exists a vertex v missing $\pi(x, a) - x + y$; trivially, both vx and vz are edges,
- there exists a vertex w missing $\pi(x, a) - x + z$; trivially, both wx and wy are edges.

But now we have reached a contradiction—$\{a, w, z\}$ is an AT, and the proof of the theorem is complete. □

5. Two characterizations of AT-free graphs. The goal of this section is to offer two characterizations of AT-free graphs. To motivate our first characterization, notice that Theorems 3.1 and 3.2 do not lead to a necessary and sufficient condition for a graph to be AT-free. For example, vertices achieving the diameter in the C_6 constitute a dominating pair. Furthermore, if we add a universal vertex to an arbitrary graph, we obtain a graph that has a dominating pair consisting of the universal vertex and any other vertex. Clearly, any attempt to provide a characterization of AT-free graphs involving dominating pairs must not only be based on induced subgraphs, but it must also restrict the types of dominating pairs. For example, the graph C_6 contains an AT, yet every induced subgraph has a dominating pair.

The first goal is to provide a characterization of AT-free graphs based on dominating pairs. As indicated previously, such a result must restrict the types of dominating pairs. In particular, we impose an adjacency condition on G with dominating pair (x, y), whereby the connected component of $G \setminus \{x\}$ containing y has a dominating pair (x', y) with x' adjacent to x. As illustrated in Figure 5.1, the graph C_6 fails this criterion. Here, (x, y) is a dominating pair in the graph, yet neither (x', y) nor (x'', y) is a dominating pair in the graph obtained by removing vertex x.

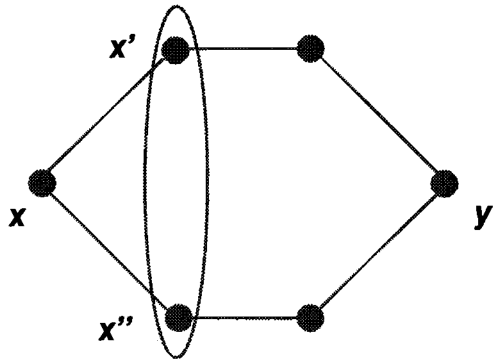


FIG. 5.1. C_6.

We begin by stating a simple property of vertices in AT-free graphs which is of independent interest.

CLAIM 5.1. *Let u, v, and y be vertices in a connected AT-free graph such that $v \notin D(u, y)$. If $D(u, y) \not\subset D(v, y)$ then, for some vertex w in $D(u, y)$, v and w are unrelated with respect to y.*

Proof. Let π be a u, y-path missed by v. Let w be an arbitrary vertex in the set $D(u, y) \setminus D(v, y)$. Since w does not belong to $D(v, y)$, w misses some v, y-path. Since w belongs to $D(u, y)$, w intercepts π and, moreover, $\pi \cup \{w\}$ contains a chordless

w, y-path missed by v, confirming that v and w are unrelated with respect to y. □

Let $\pi = u_1, u_2, \ldots, u_k$ and $\pi_1 = v_1, v_2, \ldots, v_l$ be two paths. We shall refer to the path $u_1, u_2, \ldots, u_i$ with $i \leq k$ as a *prefix* of π. A vertex w is a *cross* point of π and π_1 if $w = u_i = v_j$ and the four vertices u_{i-1}, v_{j-1}, u_{i+1}, and v_{j+1} are all defined and distinct.

For later reference, we now investigate properties of asteroidal triples. Let G be a graph containing an AT. Choose an induced subgraph H of G with the least number of vertices such that some triple $\{x, y, z\}$ is an AT in H. Let $\pi(x, y)$, $\pi(x, z)$, and $\pi(y, z)$ be paths in H demonstrating that $\{x, y, z\}$ is an AT. In the following we write $\pi(x, y) :\ x = u_1, u_2, \ldots, u_k = y$, $\pi(x, z) :\ x = v_1, v_2, \ldots, v_l = z$, and $\pi(z, y) :\ z = w_1, w_2, \ldots, w_t = y$. Clearly, the choice of H guarantees that x, y, and z have degree at most two.

Claim 5.2. *No pair of paths among $\pi(x, y)$, $\pi(x, z)$, and $\pi(y, z)$ has a cross point.*

Proof. Suppose that the paths $\pi(x, y)$ and $\pi(x, z)$ have a cross point w such that $w = u_i = v_j$. Observe that the definition of a cross point and the minimality of H combined guarantee that $3 \leq i$ and $3 \leq j$. Since the paths demonstrate that $\{x, y, z\}$ is an AT, $i \leq k - 2$ and $j \leq l - 2$. But now, in $H' = H \setminus \{v_{j-1}\}$, y misses the x, z-path $u_1, u_2, \ldots, u_i = v_j, v_{j+1}, \ldots, z$ and x misses the y, z-path $y, u_{k-1}, \ldots, u_i = v_j, v_{j+1}, \ldots, z$. Thus, $\{x, y, z\}$ is an AT in H', contradicting the minimality of H. □

Claim 5.3. *Let i be the largest subscript for which there exists a subscript j such that $u_i = v_j$ and $u_{i+1} \neq v_{j+1}$. Then $i = j$ and $u_t = v_t$ for all $1 \leq t \leq i$.*

Proof. Since y are z are distinct and $u_1 = v_1$, the subscript i in the statement of the claim always exists. Since, by Claim 5.2, u_i cannot be a cross point, we must have $u_{i-1} = v_{j-1}$. Let t be the least value for which $u_{i-t} \neq v_{j-t}$. We may assume that such a t exists, for otherwise there is nothing to prove.

Clearly, $u_1 = v_1$ implies that $t \leq \min\{i - 2, j - 2\}$. Consequently, we can remove vertex v_{j-t} from H, while still ensuring that $\{x, y, z\}$ is an AT in the remaining graph. This contradiction completes the proof of the claim. □

Lemma 5.4. *There exist unique vertices x', y', z' in H such that*

(i) *the unique path between x and x' is a prefix of both $\pi(x, y)$ and $\pi(x, z)$,*

(ii) *the unique path between y and y' is a prefix of both $\pi(y, x)$ and $\pi(y, z)$,*

(iii) *the unique path between z and z' is a prefix of both $\pi(z, x)$ and $\pi(z, y)$.*

Proof. Claim 5.3 guarantees that one can associate with x a unique vertex x' corresponding to the largest subscript for which $u_i = v_i$. Put differently, the path $x = u_1, u_2, \ldots, u_i = x'$ in H is the common prefix of both $\pi(x, y)$ and $\pi(x, z)$. In a perfectly similar way one can define vertices y' and z'. □

As it turns out, vertices x', y', z' have a number of interesting properties. We present some of them next.

Claim 5.5. *The vertices x', y', and z' are either all distinct or else they coincide.*

Proof. Suppose that exactly two of the vertices x', y', z' coincide. Symmetry allows us to assume that $x' = y'$. Write $x' = u_i$ and $y' = w_{t-k+i}$. Since $x'(= y')$ cannot be a cross point of $\pi(x, z)$ and $\pi(z, y)$, we must have $v_{i+1} = w_{t-k+i-1}$. Now an argument similar to that of the proof of Claim 5.3 guarantees that the subpaths of $\pi(x, z)$ and $\pi(z, y)$ between z and x' coincide, which is a contradiction. □

Claim 5.5 and the minimality of H combined imply the following result.

Corollary 5.6. *Vertices x', y', and z' coincide if and only if H is isomorphic to the graph in Figure* 5.2.

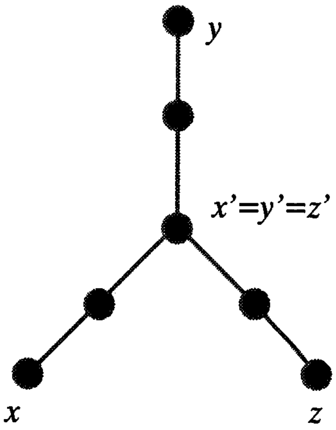


FIG. 5.2. *Illustrating Corollary* 5.6.

CLAIM 5.7. *Vertex x' is distinct from x if and only if $d_H(x) = 1$. Furthermore, if x', y', and z' are distinct and $x' \neq x$ then xx' is an edge.*

Proof. First, observe that if $x' = x$ then, by Claim 5.3, $d_H(x) = 2$. Conversely, if vertices x and x' are distinct, then $\pi(x, y)$ and $\pi(x, z)$ have at least one edge in common, confirming that $d_H(x) = 1$.

To settle the second part of the claim, assume that $x' = u_i$ with $3 \leq i$. Since x', y', z' are distinct, u_{i-1} misses the path $\pi(y, z)$ and, thus, $\{u_{i-1}, y, z\}$ is an AT in $H \setminus \{x\}$. The conclusion follows. ◻

For reasons that will become clear later, we shall say that a connected graph H with a dominating pair satisfies the *spine property* if for every *nonadjacent* dominating pair (α, β) in H there exists a neighbor α' of α such that (α', β) is a dominating pair of the connected component of $H \setminus \{\alpha\}$ containing β. We are now in a position to state the first main result of this section.

THEOREM 5.8 (The Spine theorem). *A graph G is AT-free if and only if every connected induced subgraph H of G satisfies the spine property.*

Proof. To settle the "only if" part, let G be an AT-free graph and let H be any connected induced subgraph of G. We may assume that H is not a clique (complete), since otherwise it has the spine property. By Claim 4.1, H has a nonadjacent dominating pair (α, β). Let C_β denote the connected component of $H \setminus \{\alpha\}$ that contains β. Let A denote $N(\alpha) \cap C_\beta$. We choose a vertex $\tilde{\alpha}$ in A such that $D(\tilde{\alpha}, \beta) \subset D(t, \beta)$ for no vertex t in A.

We claim that

$$(5.1) \qquad (\tilde{\alpha}, \beta) \text{ is a dominating pair in } C_\beta.$$

To see that (5.1) holds, suppose that a vertex t in C_β misses some $\tilde{\alpha}, \beta$-path. Observe that t must belong to A, for otherwise this path extends to an α, β-path in H missed by t, contradicting that (α, β) is a dominating pair. Our choice of $\tilde{\alpha}$ guarantees that $D(\tilde{\alpha}, \beta) \not\subset D(t, \beta)$. By Claim 5.1 we find a vertex w in $D(\tilde{\alpha}, \beta)$ such that t and w are unrelated with respect to β. Note that w belongs to A; otherwise the t, β-path missed by w would extend to an α, β-path missed by w. But now, w and t are in the same component of $N'(\beta)$ and are unrelated with respect to β, contradicting Claim 3.4. This completes the proof of the "only if" part.

To prove the "if" part, let H be an induced subgraph of G with the least number of vertices in which some set $\{x, y, z\}$ is an AT. Further, let $\pi(x, y)$, $\pi(x, z)$, and $\pi(y, z)$ be (chordless) paths in H demonstrating that $\{x, y, z\}$ is an AT.

CLAIM 5.9. *If H has an adjacent dominating pair, it also has a nonadjacent dominating pair.*

Proof. Suppose that (a, b) is an adjacent dominating pair in H and let $A = \{v \mid av \in E, bv \notin E\}$ $B = \{v \mid bv \in E, av \notin E\}$, and $C = \{v \mid av, bv \in E\}$. By the minimality of H, every vertex of $H\backslash\{x, y, z\}$ is on at least one π path. If $x = a$, then y and z are in B and $H\backslash\{b\}$ contains an AT on $\{x, y, z\}$. Thus we may assume that $\{a, b\} \cap \{x, y, z\} = \emptyset$. Furthermore, it is easy to see that A and B each contain at least one of $\{x, y, z\}$; otherwise one of a or b can be removed from H without destroying the AT. We now have two cases.

Case 1: $x \in A$, $y \in B$, $z \in C$. Since a and b must be on at least one π path, $\pi(x, z) = x, a, z$ and $\pi(y, z) = y, b, z$. Consider $\pi(x, y) = v_1(= x), v_2, \ldots, v_k(= y)$. First we note that none of $v_2, \ldots, v_{k-2}$ can be in A since such a vertex together with y and z would form an AT in $H\backslash\{x\}$. Similarly, none of $v_3, \ldots, v_{k-1}$ can be in B. Thus all of $v_3, \ldots, v_{k-2}$ (if they exist) must be in C. If v_2 is in C, then $B = \{y\}$ and (a, y) is a nonadjacent dominating pair; if v_{k-1} is in C, then (x, b) is a nonadjacent dominating pair. Thus v_2 is in B, v_{k-1} is in A, and all of $v_3, \ldots, v_{k-2}$ are in C. Now if $k > 4$, then $\{v_2, v_{k-1}, z\}$ forms an AT in $H\backslash\{x, y\}$; otherwise (x, b) is a nonadjacent dominating pair.

Case 2: $x \in A$, $y, z \in B$. Since each of a and b must belong to some π path, we may assume that $a \in \pi(x, y)$ and $\pi(y, z) = y, b, z$. Furthermore, we may assume that the degree of x is two since otherwise (x, b) would be a nonadjacent dominating pair. We now study $\pi(x, y) = v_1(= x), v_2(= a), \ldots, v_k(= y)$ and note by the fact that $\pi(x, y)$ is chordless that the only vertex of $\pi(x, y)$, other than x, that could be in A is v_3. Similarly, we let $\pi(x, z) = u_1(= x), u_2, \ldots, u_j(= z)$ and note that no vertex on $\pi(x, z)$ other than x and possibly u_2 may be adjacent to a since otherwise an x, z-path through a contradicts the minimality of H. We distinguish two subcases.

Case 2.1: $v_3 \in A$. First we show that $k = 4$ (i.e., v_3 is adjacent to y). To see this, note that if (x, b) is not a dominating pair then there exists a chordless x, b-path, P, and a vertex w in A missing P. Furthermore, v_4 must be adjacent to b. If $w = v_3$, then we have an AT on $\{x, v_3, z\}$ in $H\backslash\{y\}$; for the x, z-path consider the induced path on P and the edge bz. If $w \neq v_3$, then w is on $\pi(x, z)$ and we have $\{v_3, y, z\}$ being an AT in $H\backslash\{x\}$; now the v_3, z-path consists of the subpath of $\pi(x, z)$ from z to w together with the edges wa and av_3. Thus $k = 4$.

Now look at $\pi(x, z)$. Since the degree of x is two, a is not on $\pi(x, z)$. If u_2 is in A, then $j = 3$ (i.e., u_2 is adjacent to z); otherwise $\{u_2, y, z\}$ would be an AT in $H\backslash\{x\}$. Now if u_2v_3 is an edge, then (x, b) is a nonadjacent dominating pair; otherwise (u_2, v_3) is a nonadjacent dominating pair.

Thus we may assume that u_2 is not in A and therefore is adjacent to b. If $j = 3$, then (y, u_2) is a nonadjacent dominating pair. Suppose v_3 is not adjacent to some u_i, $2 < i < j$. Then $\{u_i, x, y\}$ forms an AT in $H\backslash\{z\}$. If u_2 is not adjacent to v_3, then $\{x, v_3, z\}$ forms an AT in $H\backslash\{y\}$; otherwise, (x, b) is a nonadjacent dominating pair.

Case 2.2: $v_3 \notin A$. Thus all of $v_3, \ldots, v_k$ are adjacent to b. Hence (x, b) is a nonadjacent dominating pair since b is adjacent to all vertices of H except x and possibly u_2, which is adjacent to x. □

We now assume that H has a nonadjacent dominating pair (a, b).

CLAIM 5.10. *Vertices a and b are distinct from x, y, z, x', y', and z'.*

Proof. To begin, we show that a and b are distinct from x, y, and z. Suppose not. We may assume, without loss of generality, that $a = x$. Since (a, b) is a dominating pair, b must belong to $\pi(y, z)$. Consider the x, b-path contained in the concatenation of $\pi(x, y)$ with the $y - b$ portion of $\pi(y, z)$. This path is missed by z unless vertices b and z are adjacent. A mirror argument shows that b and y are also adjacent.

Since, by assumption, H satisfies the spine property and vertices a and b are nonadjacent, we should be able to find a neighbor b' of b such that (a, b') is a dominating pair in $H \setminus \{b\}$. However, if b' belongs to $\pi(x, y)$, then z misses the corresponding b', a-path; if b' belongs to $\pi(x, z)$, then y misses a b', a-path. The fact that a is distinct from x' follows by an identical argument, whose details are omitted. □

Claim 5.10 has the following interesting corollary.

CLAIM 5.11. *Each pair of vertices x and x', y and y', and z and z' must coincide.*

Proof. First, observe that the vertices x', y', z' are distinct, for otherwise, by Corollary 5.6, H is isomorphic to the graph in Figure 5.2 which does not satisfy the spine property.

If the statement is false, then we may assume, without loss of generality, that x and x' are distinct. By Claim 5.7, x has degree one in H. By Claim 5.10, a (respectively, b) is distinct from both x and x', implying that x misses some a, b-path, which is a contradiction. □

By virtue of Claims 5.11 and 5.7 combined, x, y, and z have degree exactly two in H and, moreover, H is biconnected. Without loss of generality, let vertices a and b belong to $\pi(x, y)$ and to $\pi(x, z)$, respectively. Observe that vertices a and y must be adjacent, for otherwise the a, b-path through x is missed by y. Similarly, vertices b and z are also adjacent; otherwise the a, b-path through x is missed by z. Further, either a or b is adjacent to x, for if not, the a, b-path through y and z is missed by x. Symmetry allows us to assume, without loss of generality, that a and x are adjacent.

We claim that

$$\text{vertices } b \text{ and } x \text{ are adjacent.} \tag{5.2}$$

Since vertices a and b are not adjacent and H is biconnected, the spine property guarantees that we can find a neighbor a' of a such that (a', b) is a dominating pair of $H \setminus \{a\}$. Clearly, a' cannot be x; if b and x are not adjacent, then a' cannot be y. Therefore, a' must belong to $\pi(y, z)$. But now, x misses the a', b-path containing z, which is a contradiction. Thus, (5.2) must hold.

To complete the proof of the "if" part, we claim that

$$(b, y) \text{ is a dominating pair.} \tag{5.3}$$

It is clear that once (5.3) is proved, we have reached a contradiction: by Claim 5.10, y cannot be in a dominating pair.

To prove (5.3) consider a vertex c that misses a path π joining b and y. Since (a, b) is a dominating pair, π does not involve a. Trivially, c must belong to $\pi(y, z)$. But now, $\{c, x, y\}$ is an AT in $H \setminus \{a\}$. To see this, note that $\pi + x$ is an x, y-path missed by c; the y, c-path consisting of the portion of $\pi(y, z)$ from y to c is missed by x; finally, $\pi(x, z)$ concatenated with the $c - z$ portion of $\pi(y, z)$ contains a c, x-path missed by y. This completes the proof of Theorem 5.8. □

Let $G = (V, E)$ be a connected AT-free graph and let (x, y) be an arbitrary nonadjacent dominating pair in G. Construct a sequence $x_0, x_1, \ldots, x_k$ of vertices of G and a sequence $G_0, G_1, \ldots, G_k$ of subgraphs of G defined as follows:

(i) $G_0 = G$ and $x_0 = x$,

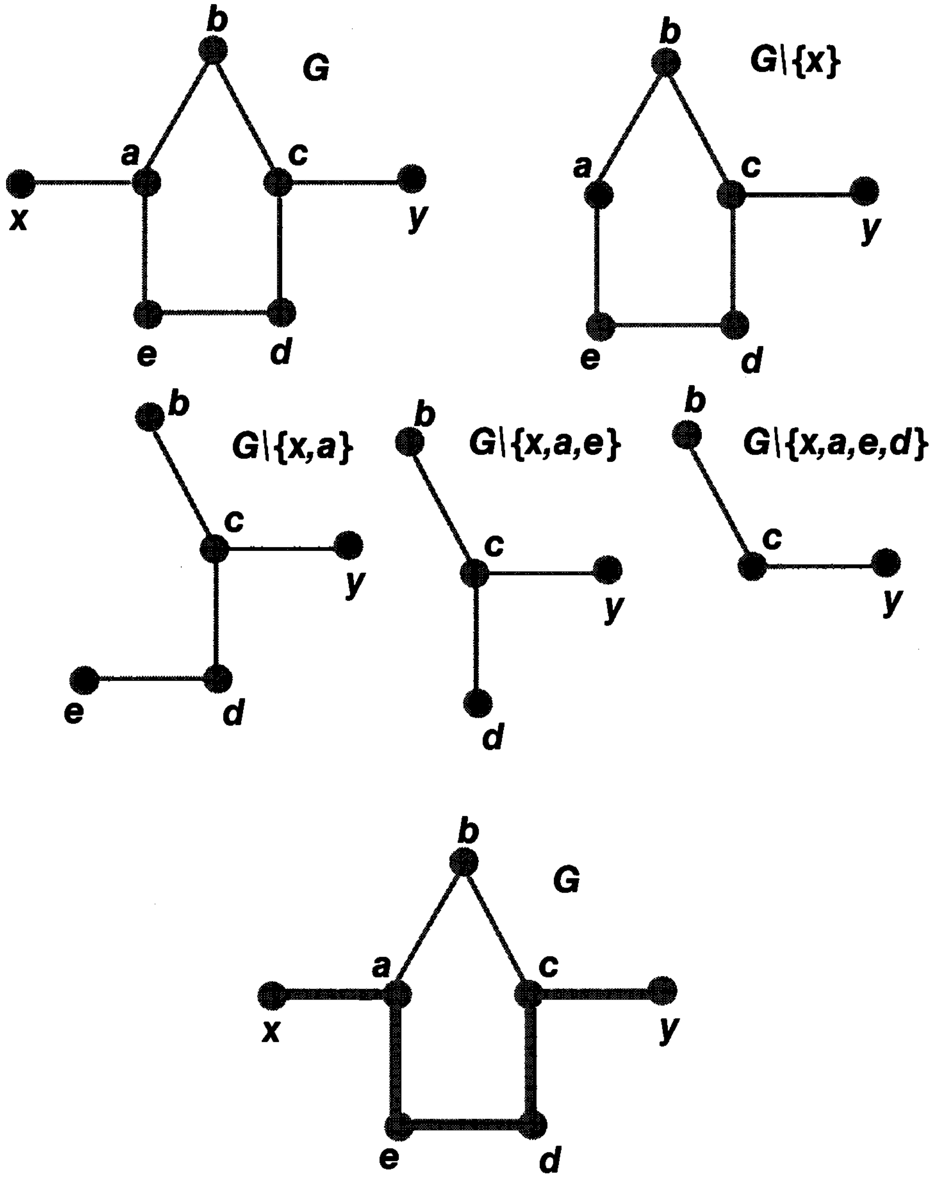


FIG. 5.3. *Illustrating the Spine theorem.*

(ii) for all i $(0 \leq i \leq k-1)$, $x_i y \notin E$ and $x_k y \in E$,

(iii) for all i $(1 \leq i \leq k)$, let G_i stand for the subgraph of G_{i-1} induced by the component of $G_{i-1} \setminus \{x_{i-1}\}$ containing y,

(iv) for all i $(1 \leq i \leq k)$, let x_i be a vertex in G_i adjacent to x_{i-1} and such that (x_i, y) is a dominating pair in G_i.

The existence of the sequence x_0, $x_1, \ldots,$ x_k is guaranteed by the Spine theorem. The sequence x_0, $x_1, \ldots,$ x_k, y will be referred to as a *spine* of G. For an illustration of the Spine theorem the reader is referred to Figure 5.3. The sequence of graphs featured in Figure 5.3 begins with a graph G with vertex set $\{a,\ b,\ c,\ d,\ e,\ x,\ y\}$

and dominating pair (x, y). The sequence continues with the graph $G \setminus \{x\}$ with dominating pair (a, y), and so on. The spine of the graph G is featured in heavy lines.

Note that the existence of a sequence of vertices and a sequence of subgraphs, as defined in (i) through (iv) above, does not necessarily imply that the graph is AT-free. For example, let (x, y) be the dominating pair $(1, 4)$ of the graph G of Figure 5.4. The vertex sequence $1, 7$ and the subgraph sequence $G, G \setminus \{1\}$ satisfy (i)–(iv) above; nevertheless, G is not AT-free ($\{2, 4, 6\}$ is an AT). However, the Spine theorem is not contradicted since the induced subgraph $G \setminus \{7\}$ has a dominating pair (1,4), yet $G \setminus \{1, 7\}$ has no dominating pair consisting of 4 and a neighbor of 1.

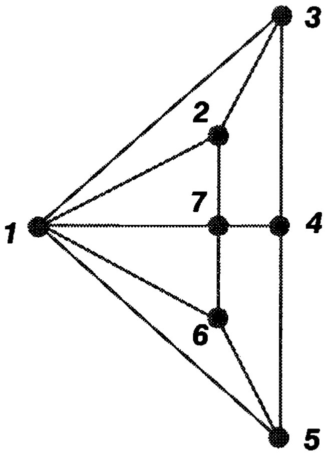


FIG. 5.4. *A graph G.*

The second goal of this section is to give a characterization of AT-free graphs in terms of minimal triangulations. Let $G = (V, E)$ be an arbitrary graph. A *triangulation* $T(G)$ of G is a set of edges such that the graph $G' = (V, E \cup T(G))$ is chordal. A triangulation $T(G)$ is *minimal* when no proper subset of $T(G)$ is a triangulation of G. Recently, Möhring [20] proved the following result.

THEOREM 5.12 (see [20]). *If G is an AT-free graph, then for every minimal triangulation $T(G)$ of G, the graph $G' = (V, E \cup T(G))$ is an interval graph.*

The remainder of this section is devoted to proving the converse of Theorem 5.12. A different proof of the converse was obtained independently by Parra [23].

THEOREM 5.13. *A graph G is AT-free if and only if, for every minimal triangulation $T(G)$ of G, the graph $G' = (V, E \cup T(G))$ is an interval graph.*

Our arguments rely, in part, on the following result which is of independent interest.

LEMMA 5.14. *Let G be an arbitrary graph and let $H = (V(H), E(H))$ be an induced subgraph of G. Let $T(H)$ be an arbitrary minimal triangulation of H. There exists a minimal triangulation $T(G)$ of G such that the only edges in $T(G)$ joining vertices in H are those in $T(H)$.*

Proof. If the statement is false, then we select a minimal triangulation $T(G)$ of G that adds as few new edges to H as possible. Since $T(H)$ is a triangulation of H, some edge uv with both u and v in H, present in $T(G)$ but not in $T(H)$, must be the unique chord of a set $\mathcal{C}$ of C_4's, each having (at least) one vertex outside H. Let w and w' be the remaining vertices of such a C_4 with w outside H. The removal of the edge uv from $T(G)$ and the addition of the ww' edge(s) will triangulate all C_4's in $\mathcal{C}$, but may create new cycles, each of which contains at least one vertex (such as w) that is not in H. Each such cycle will be triangulated by adding all possible chords incident with a particular vertex outside H. The addition of these edges may create new cycles that will be triangulated in a similar fashion. Since the graph is finite, we eventually have a triangulation $T'(G)$ that has one fewer H edge than $T(G)$. Any minimal triangulation in $T'(G)$ also has one fewer H edge than $T(G)$, thereby contradicting our choice of $T(G)$. □

Proof (Theorem 5.13). The "only if" part follows from Theorem 5.12.

To prove the "if" part, let G be a graph containing an AT. Choose an induced subgraph $H = (V(H), E(H))$ of G with the least number of vertices such that some triple $\{x, y, z\}$ is an AT in H. Let $\pi(x, y)$, $\pi(x, z)$, and $\pi(y, z)$ be paths in H demonstrating that $\{x, y, z\}$ is an AT, and write $\pi(x, y): \ x = u_1, u_2, \ldots, u_k = y$, $\pi(x, z): \ x = v_1, v_2, \ldots, v_l = z$, and $\pi(z, y): \ z = w_1, w_2, \ldots, w_t = y$. Clearly, the choice of H guarantees that x, y, and z have degree at most two.

Our plan is to exhibit a minimal triangulation $T(H)$ of H that results in a noninterval graph $H' = (V(H), E(H) \cup T(H))$. For this purpose, let x', y', and z' be the vertices specified in Lemma 5.4 and consider the triangulation $T(H)$ of H returned by the following procedure.

Step 1. If $x' = y' = z'$ then set $T(H) \leftarrow \emptyset$ and return.

Step 2. Let F be the graph obtained from H by removing vertices x, y, z and by adding the edges u_2v_2 (in case $x = x'$), $u_{k-1}w_{t-1}$ (in case $y = y'$), and $v_{l-1}w_2$ (in case $z = z'$). Let $T(F)$ be an arbitrary minimal triangulation of F. Return $T(H) \leftarrow T(F) \cup \{xu_2, xv_2, yu_{k-1}, yw_{t-1}, zv_{l-1}, zw_2\}$ (in case $x \neq x'$ one adds the edge xx' instead of the edges xu_2 and xv_2, etc.).

Now Claim 5.5 along with an easy ad hoc argument shows that $T(H)$ is a minimal triangulation of H and that $\{x, y, z\}$ is still an AT in the graph $H' = (V(H), E(H) \cup T(H))$. By Lemma 5.14, there must exist some minimal triangulation $T(G)$ of G such that H' is an induced subgraph of $G = (V, E \cup T(G))$. The conclusion follows. □

6. Augmenting AT-free graphs. The purpose of this section is twofold. First, we exhibit a structural property of AT-free graphs that naturally allows one to "stretch" an AT-free graph to a new AT-free graph. This in turn provides a condition under which two AT-free graphs can be "glued together" to form a new AT-free graph (Corollary 6.10). Next, we provide a decomposition theorem for AT-free graphs.

To begin, we address the issue of creating new AT-free graphs out of old ones. Specifically, we show how to "augment" an arbitrary AT-free graph G to obtain a new AT-free graph. This augmentation will be accomplished by finding a particular dominating pair (x, y) and by adding new vertices x' and y' adjacent to x and y, respectively. This augmentation of G again confirms our intuition about the linear structure of AT-free graphs, since the dominating pair (x, y) has been stretched to a new dominating pair (x', y').

In preparation for stating the first main result of this section, we need to define a few terms. A vertex v of an AT-free graph G is called *pokable* if the graph G' obtained from G by adding a pendant vertex adjacent to v is AT-free; otherwise, it is called *unpokable*. For example, referring to Figure 6.1, vertex u is pokable since the addition of a pendant vertex u' does not create an AT in the graph. At the same time, vertex v is unpokable, for the addition of the vertex v' creates the AT $\{a, b, v'\}$. A dominating pair (x, y) is referred to as *pokable* if both x and y are pokable. For further reference, we take note of the following simple observation whose proof is routine.

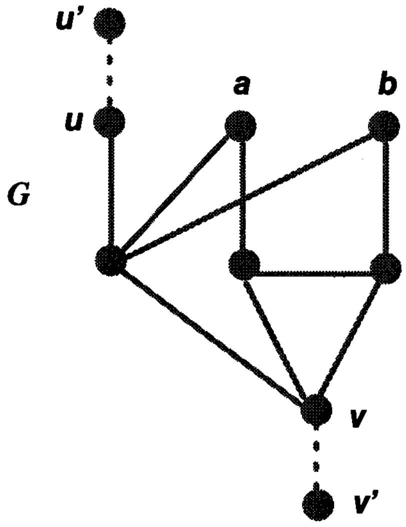


FIG. 6.1. *Illustrating pokable and unpokable vertices.*

OBSERVATION 6.1. *A vertex v of an AT-free graph G is unpokable if and only if there exist vertices u and w in G such that u and w are unrelated with respect to v and there is a u, w-path in G that does not contain v.*

Whenever we have a vertex v for which there exist vertices u and w unrelated with respect to v, we shall refer to the following induced paths, which must exist by the definition of unrelated vertices: a v, u-path $v = u_0, u_1, \ldots, u_p = u$ missed by w and a v, w-path $v = w_0, w_1, \ldots, w_q = w$ missed by u. We are now in a position to make the previous discussion precise.

THEOREM 6.2. *Every connected AT-free graph contains a pokable dominating pair; furthermore, every connected AT-free graph which is not a clique contains a nonadjacent pokable dominating pair.*

Proof. The theorem is trivial for cliques. We shall assume therefore that G is not a clique. Now, Claim 4.1 guarantees the existence of a nonadjacent dominating pair (x, y_0) in G. Let F be the connected component of $N'(x)$ containing y_0, and let Y stand for the set of vertices y in F for which (x, y) is a dominating pair in G. The conclusion of Theorem 6.2 is implied by the following technical result that will be proved later.

LEMMA 6.3. *Y contains a vertex y such that G has no unrelated vertices with respect to y.*

Let us examine how Theorem 6.2 follows from Lemma 6.3. Note that Lemma 6.3, together with Observation 6.1, implies that Y contains a pokable vertex. Let β be a pokable vertex in Y and let X denote the set of vertices x' in the same component of $N'(\beta)$ as x, for which (β, x') is a dominating pair. Clearly x belongs to X, and so X is not empty. By applying Lemma 6.3 again, with β as the "anchor," we find a pokable vertex α in X. The proof of Theorem 6.2 is established by noting that (α, β) is the desired nonadjacent pokable dominating pair. □

Proof (Lemma 6.3). The proof is by induction on the number of vertices in G. Assume that the lemma is true for all connected AT-free graphs with fewer vertices than G. We now present various facts that are used in the proof.

Claim 6.4. *Let v be a vertex in Y such that vertices u and w are unrelated with respect to v in G. Then all vertices u_i and w_j $(1 \leq i \leq p; 1 \leq j \leq q)$ belong to F.*

Proof. Without loss of generality let i be the smallest subscript for which u_i lies outside F. Trivially, u_i must belong to $N(x)$. Since w cannot miss the v, x-path, $v = u_0, u_1, \ldots, u_i, x$, and since w is adjacent to no vertex on the path $v = u_0, u_1, \ldots, u_i$, it follows that w belongs to $N(x)$.

Similarly, since u cannot miss the v, x-path, $v = w_0, w_1, \ldots, w_q = w, x$, and since u is adjacent to no vertex on the path $v = w_0, w_1, \ldots, w_q$, it follows that u belongs to $N(x)$. But now, $\{u, v, w\}$ is an AT, contradicting G being AT-free. □

It is important to note that, by virtue of Claim 6.4, Lemma 6.3 is established as soon as we exhibit a vertex y in Y such that there are no unrelated vertices with respect to y in the subgraph of G induced by F. If F and Y coincide, then by the induction hypothesis such a vertex must exist. Therefore, from now on, we shall assume that

$$F \setminus Y \neq \emptyset. \tag{6.1}$$

Let $Y_1, Y_2, \ldots, Y_k$ $(k \geq 1)$ be the connected components of the subgraph of $\overline{G}$ induced by Y.

Claim 6.5. *Let t be a vertex in $F \setminus Y$. If some vertex z in Y_i satisfies $z \in D(t, x)$, then $Y_i \subset D(t, x)$.*

Proof. If the claim is false, then we find vertices z, z' in Y_i such that $z \in D(t, x)$ and $z' \notin D(t, x)$. Since Y_i is a connected subgraph of $\overline{G}$, there exists a chordless path $z = s_1, s_2, \ldots, s_r = z'$ joining z and z' in $\overline{G}$, with all internal vertices in Y_i.

Let j be the smallest subscript for which $s_j \notin D(t, x)$. Since $z' \notin D(t, x)$, such a subscript must exist. But now, in G, s_{j-1} and s_j are nonadjacent and s_j misses some t, x-path, while s_{j-1} intercepts all such paths. It follows that s_j misses a s_{j-1}, x-path, which is a contradiction since s_{j-1} belongs to Y. □

Claim 6.6. *Y induces a disconnected subgraph of $\overline{G}$.*

Proof. First, we claim that

$$|Y| \ \geq \ 2. \tag{6.2}$$

If (6.2) is false, then $Y = \{y_0\}$. Let U stand for the set of all vertices in F adjacent to y_0. Note that (6.1), along with the connectedness of F, guarantees that U is nonempty. But now, for every u in U, $Y = \{y_0\} \subset D(u, x)$. Thus, u is an attractor, contradicting Claim 4.2. Therefore, (6.2) holds. Note that by virtue of (6.2) it makes sense to talk about Y being disconnected in the complement.

We now continue the proof of Claim 6.6. If $Y = Y_1$, then (6.1) and the connectedness of F imply the existence of a vertex z in Y adjacent to some vertex t in $F \setminus Y$. Note, in particular, that z belongs to $D(t, x)$ and so, by Claim 6.5, $Y \subset D(t, x)$. However, now t is an attractor, which is a contradiction. With this, the proof of Claim 6.6 is complete. □

Claim 6.7. *Let v be a vertex in Y such that vertices u and w are unrelated with respect to v in G. Then*

- *for all i $(1 \leq i \leq p)$, v belongs to $D(u_i, x)$ and*
- *for all j $(1 \leq j \leq q)$, v belongs to $D(w_j, x)$.*

Proof. Since v is adjacent to u_1, it follows that $v \in D(u_1, x)$. Let i be the smallest subscript for which v does not belong to $D(u_i, x)$. Let π be a u_i, x-path missed by v. Note that w must intercept π, for otherwise w would miss a v, x-path contained in $\{v, u_1, \dots, u_i\} \cup \pi$. However, now $\{u, v, w\}$ is an AT. The proof that v belongs to $D(w_j, x)$ follows by a mirror argument. □

For every i $(1 \leq i \leq k)$, let T_i stand for the set of vertices t in $F \setminus Y$ with the property that $Y_i \subset D(t, x)$. By renaming the Y_i's, if necessary, we ensure that

$$|T_1| \ \leq \ |T_2| \ \leq \cdots \leq \ |T_k|.$$

CLAIM 6.8. *Every vertex in T_1 is adjacent to all vertices in Y_1.*

Proof. The statement is vacuously true if T_1 is empty. Now assume that T_1 is nonempty and let t be a vertex in T_1 nonadjacent to some z in Y_1. Since, by Claim 4.2, t cannot be an attractor, we find a subscript j $(j \geq 2)$ such that for some z' in Y_j, z' does not belong to $D(t, x)$. Thus $t \in T_1 \setminus T_j$. Now, $|T_1| \ \leq \ |T_j|$ implies that there must exist a vertex t' in $T_j \setminus T_1$. By Claim 6.5, z does not belong to $D(t', x)$. Note that t does not belong to $D(t', x)$; otherwise, by Claim 3.3, z would belong to $D(t', x)$, which is a contradiction.

Since z' does not belong to $D(t, x)$, in particular, z' is not adjacent to t. The fact that t does not belong to $D(t', x)$ implies the existence of a t', x-path π' missed by t. Since $z' \in D(t', x)$, z' intercepts π' and thus $\pi' \cup \{z'\}$ contains a z', x-path missed by t, contradicting that z' is in Y. □

We now continue the proof of Lemma 6.3. Let Z be a connected component of the subgraph of G induced by Y_1. By the induction hypothesis, Z contains a vertex v such that Z has no unrelated vertices with respect to v. To complete the proof of Lemma 6.3, we need show only that F has no unrelated vertices with respect to v. Suppose u and w in F are unrelated with respect to v. By Claims 6.5 and 6.7 combined, all the vertices u_i and w_j $(1 \leq i \leq p;\ 1 \leq j \leq q)$ belong to Y or to T_1. By Claims 6.6 and 6.8 and the fact that the paths $v = u_0, u_1, \dots, u_p = u$ and $v = w_0, w_1, \dots, w_q = w$ are chordless, it follows that at most u_1 and w_1 belong to $T_1 \cup Y \backslash Y_1$. However, if either u_1 or w_1 is in $T_1 \cup Y \backslash Y_1$ then, by Claims 6.6 and 6.8, the edge $u_1 w$ or the edge $w_1 u$ must be present, contradicting the fact that u and w are unrelated with respect to v. Thus, all the u_i's and w_j's belong to Y_1. In fact, since Z is a connected component of Y_1, all the u_i's and w_j's must belong to Z, which is a contradiction. This completes the proof of Lemma 6.3. □

Theorem 6.2 implies the following results that are interesting in their own right.

COROLLARY 6.9. *Every AT-free graph is either a clique or contains two nonadjacent pokable vertices.*

COROLLARY 6.10 (The Composition theorem). *Given two AT-free graphs G_1 and G_2 and pokable dominating pairs (x_1, y_1) and (x_2, y_2) in G_1 and G_2, respectively, let G be the graph constructed from G_1 and G_2 by identifying vertices x_1 and x_2. Then G is an AT-free graph.*

The reader is referred to Figure 6.2 for an illustration of the Composition theorem.

We now show that the existence of a pokable dominating pair in a connected AT-free graph leads to a natural decomposition scheme. In preparation for stating the second main result of this section, we first give a necessary and sufficient condition for a vertex in a dominating pair to be pokable. Specifically, we have the following result.

CLAIM 6.11. *Let G be a connected AT-free graph with a dominating pair (x, y). Then x is pokable if and only if there are no unrelated vertices with respect to x.*

Proof. The "if" part is easily seen. To prove the "only if" part, consider unrelated vertices u and v with respect to x. In particular, we find a v, x-path missed by u and a u, x-path missed by v. Since (x, y) is a dominating pair, u and v intercept every

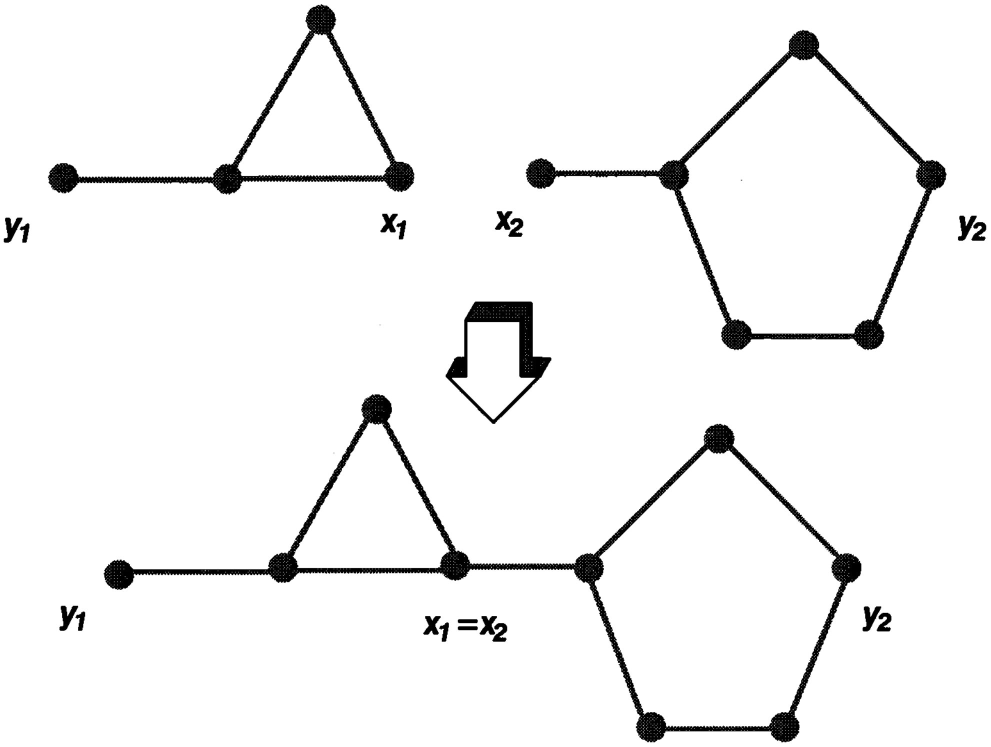


FIG. 6.2. *Illustrating the Composition theorem.*

path joining x and y. Let π be such a path and let u' and v' be vertices on π adjacent to u and v, respectively. Trivially both u' and v' are distinct from x. But now, there exists a u, v-path in G that does not contain x (this path contains vertices u', v' and a subpath of π), implying that x is not pokable. □

Let $G = (V, E)$ be a connected AT-free graph with at least two vertices and let (x, y) be a pokable dominating pair in G. Define a binary relation R on G by writing for every pair u, v of vertices,

$$u \; R \; v \iff D(u, x) = D(v, x). \tag{6.3}$$

Clearly, R is an equivalence relation; let $C_1, C_2, \ldots, C_k$ $(k \geq 1)$ be the equivalence classes of G/R. A class C_i is termed *nontrivial* if $|C_i| \geq 2$. The existence of nontrivial equivalence classes with respect to R is not immediately obvious. In what follows, we assume that the pokable dominating pair (x, y) is chosen to be nonadjacent whenever possible. The following result guarantees that nontrivial equivalence classes always exist.

CLAIM 6.12. *G/R contains at least one nontrivial equivalence class.*

Proof. If $N'(x)$ is empty then the class containing y, $C(y)$, is equal to V and is therefore nontrivial. Otherwise, Theorem 6.2 and our choice of x and y combined guarantee that x and y are nonadjacent. Let F be the connected component of $N'(x)$ containing y and let Y stand for the subset of F consisting of all the vertices that are in a dominating pair with x. Clearly, $y \in Y$, and so Y is nonempty. If F contains at least two vertices then (6.2) guarantees that Y itself contains at least two vertices, and so the equivalence class containing y is nontrivial.

We may assume, therefore, that $F = \{y\}$. Let y' be an arbitrary neighbor of y in $N(x)$. Clearly, $D(y', x) = V$, for otherwise if some vertex z does not belong to $D(y', x)$, then z must miss the y, x-path consisting of y, y', and x. Consequently, the equivalence class containing y is nontrivial and the proof of Claim 6.12 is complete. □

Remark. In fact, the proof of Claim 6.12 also tells us that the class $C(y)$ containing y is always nontrivial as long as the original graph has at least two vertices.

A nontrivial class C of G/R is said to be *valid* if C induces a connected subgraph of G. As before, the existence of valid equivalence classes is not immediately obvious. As we shall prove next, such classes always exist. Specifically, we propose to show that $C(y)$ is valid. As it will turn out, *all* valid classes of G/R enjoy very interesting properties that will allow us to select an arbitrary one for the purpose of decomposing the original graph. This freedom of choice opens the door to parallel decomposition algorithms for AT-free graphs.

CLAIM 6.13. *G/R contains at least one valid equivalence class.*

Proof. If $N'(x)$ is empty, then $C(y) = V$ and there is nothing to prove. We may therefore assume that $N'(x)$ is nonempty. As before, we may also assume that y belongs to $N'(x)$. Let F be the connected subgraph of $N'(x)$ containing y, let Y stand for the subset of F consisting of all the vertices that are in a dominating pair with x, and let $C(y)$ be the equivalence class containing y.

Notice that every vertex w that belongs to $N(x)$ and to $C(y)$ must be adjacent to all the vertices in F. In particular, if such a vertex exists, then $C(y)$, which by Claim 6.12 is nontrivial, must be connected and, thus, valid.

We will assume, therefore, that $N(x)$ and $C(y)$ are disjoint. In turn, this implies that $C(y) = Y$. Recall that, by Claim 6.6, Y induces a disconnected subgraph of $\overline{G}$, confirming that $C(y)$ is connected as a subgraph of G. The conclusion follows. □

Let S be a set of vertices of G. The graph G' is said to arise from G by an *S-contraction* if G' contains all the vertices in $G \setminus S$ along with a new vertex s adjacent, in G', to all the vertices in $G \setminus S$ that were adjacent, in G, to some vertex in S. Our next result states a fundamental property of valid equivalence classes, namely, that contracting any of them will result in an AT-free graph. The details are spelled out as follows.

LEMMA 6.14. *Let C be an arbitrary valid equivalence class of G/R. The graph G' obtained from G by a C-contraction is AT-free.*

Proof. Let c be the vertex in G' obtained by contracting C. To begin, we claim that

$$\text{there are no vertices } u,\ v \text{ in } G' \text{ such that } \{u, v, c\} \text{ is an AT.} \tag{6.4}$$

To justify (6.4) note that if $\pi(u, v)$ is a u, v-path missed by c, then the same path is missed, in G, by all the vertices in C. Let $\pi(u, c)$ be a u, c-path in G' missed by v. Then there exists a vertex c_1 in C such that v misses the path $\pi(u, c) - c + c_1$. Similarly, let $\pi(v, c)$ be a v, c-path in G' missed by u. There must exist a vertex c_2 in C such that u misses the path $\pi(v, c) - c + c_2$. Since C induces a connected subgraph of G, there exists a path joining c_1 and c_2, all of whose internal vertices are in C. By a previous observation, both u and v miss this path. Therefore, for a suitably chosen vertex c' in C, $\{u, v, c'\}$ is an AT in G, which is a contradiction. Thus, (6.4) must hold.

To complete the proof of Lemma 6.14, let $\{u, v, w\}$ be an arbitrary AT in G'. By (6.4), c is distinct from u, v, w. Let $\pi(u, v)$, $\pi(u, w)$, and $\pi(v, w)$ be paths in G'

confirming that $\{u, v, w\}$ is an AT. If c belongs to none of these paths, then $\{u, v, w\}$ is an AT in G. We may therefore assume without loss of generality that c belongs to $\pi(u, v)$. Since w misses $\pi(u, v)$, it is clear that w is adjacent to no vertex in C.

We claim that there exists a path $\pi'(u, v)$ in G missed by w. This path contains the same vertices as $\pi(u, v)$ outside of C. Inside C it contains a path between two vertices c' and c'' of C such that

- w misses a u, c'-path consisting of a subpath of $\pi(u, v)$,
- w misses a c'', v-path consisting of the remaining vertices in $\pi(u, v) - c$.

This completes the proof of Lemma 6.14. □

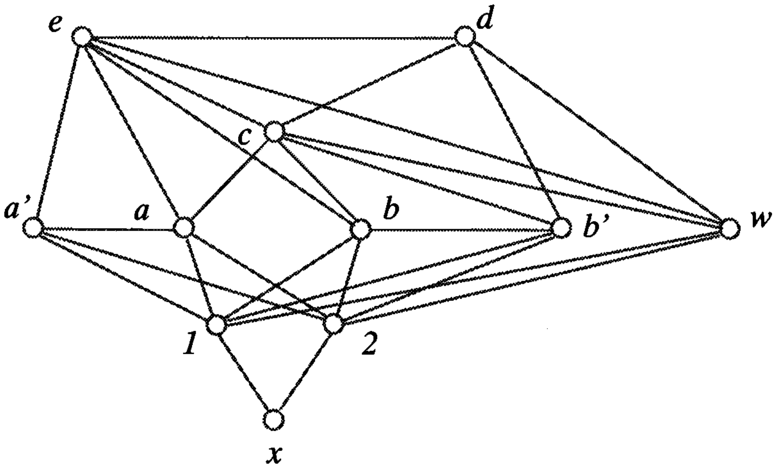


FIG. 6.3. *A graph G.*

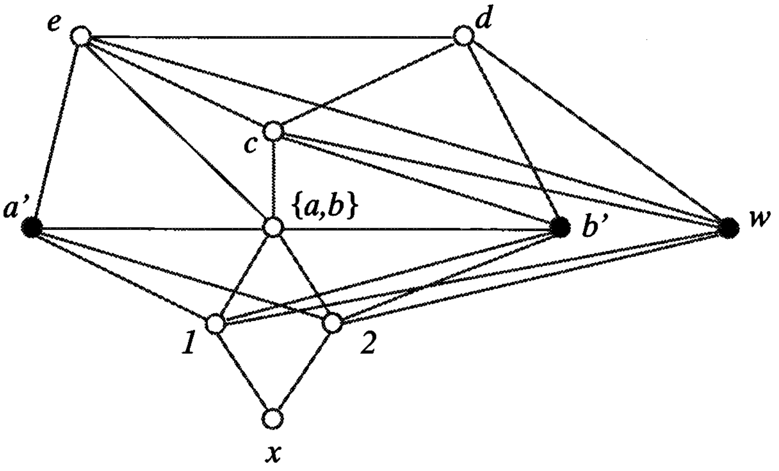


FIG. 6.4. *The graph G' obtained by contracting $\{a, b\}$.*

The example in Figures 6.3 and 6.4 shows that the connectivity of the equivalence class C in Lemma 6.14 is required if we are to guarantee that the resulting graph is AT-free. To wit, the graph G featured in Figure 6.3 is AT-free with a pokable dominating pair (x, e). The contraction of the equivalence class $\{a, b\}$ yields the graph G' in

Figure 6.4, which has the AT $\{a', b', w\}$. For the reader's benefit, the various values of the $D(*, x)$ sets, along with the equivalence classes corresponding to the graph in Figure 6.3, are summarized in Table 6.1.

Let $C(y)$ be the equivalence class containing y. Let G' be the graph obtained from G by a $C(y)$-contraction. Recall that the proof of Claim 6.13 guarantees that $C(y)$ is valid, and so Lemma 6.14 asserts that the graph G' is also AT-free. Let y' be the vertex of G' obtained by contracting $C(y)$. We now show that, in fact, more can be said about G'. Specifically, we have the following result.

LEMMA 6.15. *(x, y') is a pokable dominating pair in G'.*

Proof. To begin, we establish that (x, y') is a dominating pair in G'. For this purpose, suppose that there exists some path $\pi(x, y')$ joining x and y', missed by a vertex w. Clearly, w is adjacent, in G, to no vertex in $C(y)$. In particular, w is not adjacent to y. Since $C(y)$ is valid, w misses, in G, a y, x-path consisting of all the vertices in $\pi(x, y')$, along with a suitable path in $C(y)$. Therefore, (x, y') must be a dominating pair in G'.

Next, we show that both x and y' are pokable vertices of G'. Suppose that x is not pokable. Now Claim 6.11 guarantees the existence of unrelated vertices u and v (with respect to x). This, in turn, implies the existence of paths $\pi(v, x)$ and $\pi(u, x)$ in G', missed by u and v, respectively. Since (x, y') is a dominating pair in G', y' belongs to neither of these paths. But now, these paths must have been paths in G, which is a contradiction.

Finally, suppose that y' is not pokable. By virtue of Claim 6.11 this implies the existence of vertices u and v and paths $\pi(v, y')$ and $\pi(u, y')$ in G', missed by u and v, respectively. In particular, neither u nor v is adjacent to y'. In turn, this implies that neither u nor v is adjacent to a vertex in $C(y)$. But now, in G, there exists a u, y-path missed by v and a v, y-path missed by u, contradicting that y is pokable. This completes the proof of Lemma 6.15. □

TABLE 6.1
Illustrating the various equivalence classes.

Equivalence class	$D(*, x)$
x	$\{x, 1, 2\}$
1,2	$V \setminus \{c, d, e\}$
a'	$V \setminus \{c, d\}$
a, b:	$V \setminus \{d\}$
b'	$V \setminus \{e\}$
w, c, d, e	V

At this stage, the reader may wonder whether the class $C(y)$ is the only one whose contraction leaves x pokable. The answer is provided by the following result that complements Lemma 6.15.

LEMMA 6.16. *Let C be an arbitrary valid equivalence class in an AT-free graph G, and let G' be the graph obtained from G by a C-contraction. If C is distinct from $C(x)$ and $C(y)$, then (x, y) is a pokable dominating pair in G'.*

Proof. Let c be the vertex of G' obtained by the C-contraction. By assumption, c is distinct from y and x. We begin by showing that (x, y) is a dominating pair in G'. Suppose that there exists some path $\pi(x, y)$ joining x and y in G', missed by a vertex w. Clearly, c must belong to $\pi(x, y)$. Notice that w is adjacent, in G, to no vertex in C. Since C is valid, w misses, in G, a y, x-path consisting of all the vertices in $\pi(x, y) - c$, along with a suitable path in C. Thus, (x, y) must be a dominating

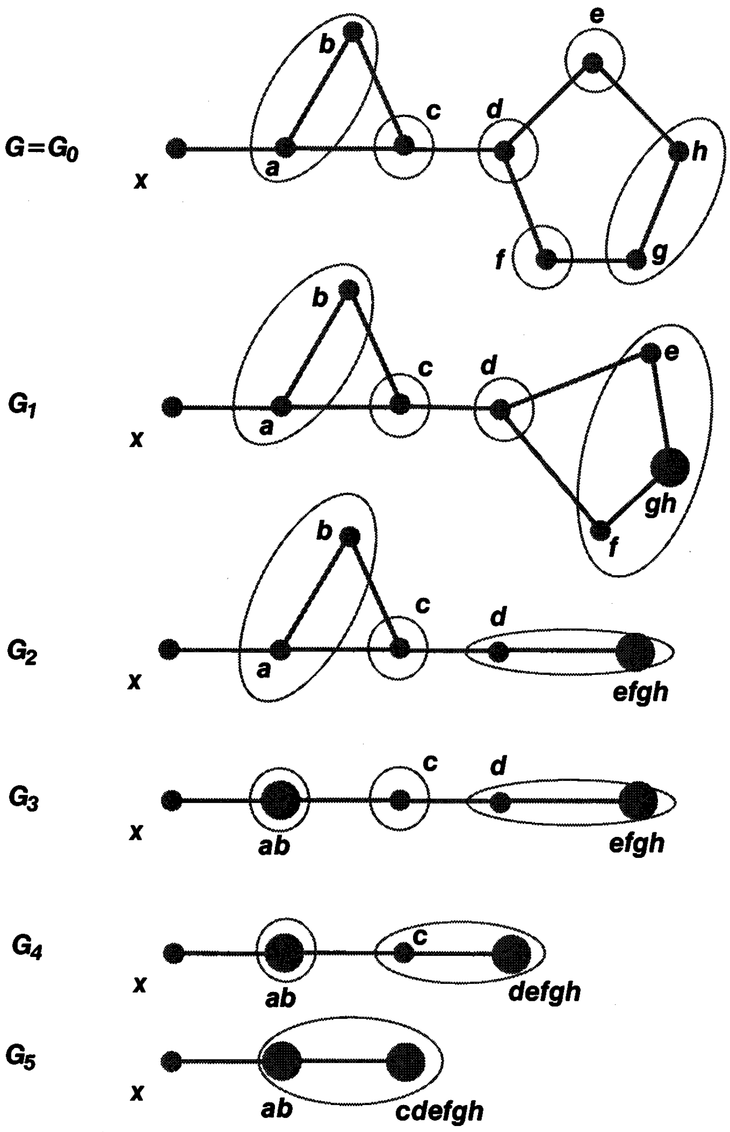


FIG. 6.5. *Illustration of an involutive sequence.*

pair in G'.

Next, we show that both x and y are pokable vertices of G'. If x is not pokable, Claim 6.11 guarantees the existence of vertices u and v unrelated with respect to x. In turn, this implies the existence of paths $\pi(v,x)$ and $\pi(u,x)$ in G', missed by u and v, respectively. Since x is pokable in G, c must belong to (at least) one of these paths. Symmetry allows us to assume, with no loss of generality, that c belongs to $\pi(u,x)$. The fact that v misses $\pi(u,x)$ guarantees that v is adjacent, in G, to no vertex in C. But now, we have reached a contradiction: v misses a u,x-path in G consisting of all the vertices of $\pi(u,x)$ outside C, along with a suitably chosen path in C. Thus, x must be pokable in G'.

A perfectly similar argument, whose details are omitted, asserts that y is also pokable. With this, the proof of Lemma 6.16 is complete. □

Lemmas 6.15 and 6.16 combined set the stage for a decomposition theorem for AT-free graphs. Consider an AT-free graph $G = (V, E)$ and let (x, y_0) be a pokable dominating pair in G. Let $G_0, G_1, \ldots, G_k$ be a sequence of graphs defined as follows.

(i) $G_0 = G$.

(ii) For all i $(0 \leq i \leq k-1)$, let R_i be the equivalence relation defined on G_i by setting $uR_iv \iff D(u,x) = D(v,x)$, and let C be an arbitrary valid equivalence class of G_i/R_i. Let G_{i+1} be the graph obtained from G_i by a C-contraction (i.e., G_{i+1} contains all the vertices in $G_i \backslash C$ as well as a new vertex c which is adjacent to all vertices in $G_i \backslash C$ that were adjacent to at least one vertex in C).

(iii) G_k consists of a single vertex.

Such a sequence $G_0, G_1, \ldots, G_k$ is called *involutive*. The reader is referred to Figure 6.5, which features the first five graphs in an involutive sequence of the given graph. Note that in the transition from G_2 to G_3 in Figure 6.5 two equivalence classes could be contracted, namely, $\{a, b\}$ and $\{d, efgh\}$. We have selected to contract the class $C = \{a, b\}$.

The obvious question is whether every connected AT-free graph has such an involutive sequence. This fundamental question is answered in the affirmative in the following theorem.

THEOREM 6.17. *Every connected AT-free graph G has an involutive sequence.*

Proof. We shall assume that G is not a clique, since otherwise there is nothing to prove. By Theorem 6.2, we find a nonadjacent pokable dominating pair (x, y_0) in G. Consider the transition from G_i to G_{i+1} for some i $(0 \leq i \leq k-1)$. Let C be an arbitrary valid equivalence class in G_i/R_i, and let (x, y_i) be a pokable dominating pair in G_i. Define y_{i+1} to be y_i in case C is distinct from $C(y_i)$ and to be the vertex obtained by contracting $C(y_i)$ otherwise. Clearly, G_{i+1} is connected whenever G_i is. By Lemmas 6.14, 6.15, and 6.16 combined, G_{i+1} is AT-free and (x, y_{i+1}) is a pokable dominating pair in G_{i+1}. This completes the proof of Theorem 6.17. □

We close with the obvious question: Can such an involutive sequence be constructed efficiently?

7. Dominating pairs in high diameter AT-free graphs. The purpose of this section is to show that, in a connected AT-free graph with diameter larger than three, the set of vertices that can be in dominating pairs is restricted to two disjoint sets. Specifically, we have the following result.

THEOREM 7.1. *Let G be a connected AT-free graph with diameter at least four. There exist nonempty, disjoint sets X and Y of vertices of G such that (x, y) is a dominating pair if and only if $x \in X$ and $y \in Y$.*

We note that Theorem 7.1 is the best possible in the sense that for AT-free graphs of diameter less than four, the sets X and Y are not guaranteed to exist. To wit, C_5 and the graph of Figure 7.1 provide counterexamples of diameter two and three, respectively.

Proof. Let (x_0, y_0) be a dominating pair in G achieving the diameter. (The existence of such a pair follows from Theorem 4.3.) Let Y stand for the set of all the vertices y in G such that (x_0, y) is a dominating pair, and let X be the set of all the vertices x in G for which (x, y_0) is a dominating pair. We propose to show that X and Y are the sets with the property specified in Theorem 7.1. Our proof relies on a number of intermediate results that we present next.

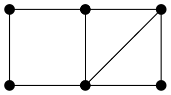


FIG. 7.1. *An AT-free graph of diameter three.*

To begin, we note that

(7.1) $$x_0 \in X \text{ and } y_0 \in Y.$$

In addition, by Claim 6.6,

(7.2) $$\text{both } X \text{ and } Y \text{ are disconnected in } \overline{G}.$$

Our choice of x_0 and y_0 guarantees that

(7.3) $$x_0 \text{ (respectively}, y_0) \text{ is adjacent to no vertices in } Y \text{ (respectively}, X).$$

Otherwise, (7.1) and (7.2) would imply that $d(x_0, y_0) \leq 3$.

Note that (7.2) and (7.3) combined guarantee that

(7.4) $$X \text{ and } Y \text{ are disjoint.}$$

The following argument justifies (7.4). If $z \in X \cap Y$ then, in particular, $z \in X$ and so (z, y_0) is a dominating pair. By (7.2), there exists a z, y_0-path contained in Y. By (7.3), x_0 misses this path, contradicting the fact that (z, y_0) is a dominating pair.

Let x and y be arbitrary vertices in X and Y, respectively. We claim that

(7.5) $$(x, y) \text{ is a dominating pair.}$$

To justify (7.5), suppose that some vertex u misses an x, y-path π. Observe that (7.2) guarantees the existence of an x_0, y-path contained in $\pi \cup X$. Since (x_0, y) is a dominating pair, this path is dominating. By (7.3), y_0 must be adjacent to a vertex of $\pi \setminus \{x\}$. Thus, $\pi \cup \{y_0\}$ contains an x, y_0-path. This path must be dominating and so u must be adjacent to y_0. A perfectly similar argument shows that u is adjacent to x_0, contradicting that x_0 and y_0 achieve the diameter.

Next, let x be an arbitrary vertex in X. We claim that

(7.6) $$\text{if } (x, z) \text{ is a dominating pair then } z \in Y.$$

Trivially, $z \notin X$; since $\text{diam}(G) \geq 4$, x and z are not adjacent. If $z \notin Y$, there exists an x_0, z-path π missed by some vertex u. Note that $\pi \cup X$ contains an x, z-path. Since, by assumption, (x, z) is a dominating pair, this path is dominating and so y_0 must intercept it. By (7.3) y_0 intercepts $\pi \setminus \{x_0\}$. Since (x_0, y_0) is a dominating pair it follows that u is adjacent to y_0. Trivially, u is not adjacent to x; otherwise the path y_0, u, x which is dominating implies that x and x_0 are adjacent and so $d(x_0, y_0) \leq 3$. Further, u and x being nonadjacent guarantees that x and x_0 are also nonadjacent; otherwise u misses the x, z-path contained in $\pi \cup \{x\}$. Now, (7.2) guarantees that some x' in X is adjacent to both x_0 and x. Since (x, z) is a dominating pair, u must be adjacent to x'. However, this implies that $d(x_0, y_0) \leq 3$, which is a contradiction.

Let y be an arbitrary vertex in Y. As above, we can prove that

(7.7) $$\text{if } (y, z) \text{ is a dominating pair then } z \in X.$$

Note that by virtue of (7.4), (7.5), (7.6), and (7.7), to complete the proof of Theorem 7.1 we only need to prove that if (v, w) is a dominating pair then $v \in X$ and $w \in Y$ (or $v \in Y$ and $w \in X$). Suppose not.

By (7.5), (7.6), and (7.7) it must be that $v \notin X \cup Y$ and $w \notin X \cup Y$. Let F be the component of $N'(x_0)$ that contains Y. (Observe that $\mathrm{diam}(G) \geq 4$ guarantees that Y is restricted to a unique component of $N'(x_0)$.) We claim that

$$(7.8) \qquad v \text{ or } w \text{ belongs to } F.$$

To justify (7.8), consider a shortest v, w-path in G. By assumption, this path is dominating and so both x_0 and y_0 must intercept it. Assume, without loss of generality, that y_0 intercepts the path "closer" to w than x_0 at a vertex t. Trivially, x_0 is adjacent to no vertex on this path from t to w, and the conclusion follows.

Let H be the component of $N'(y_0)$ that contains X. By virtue of (7.8) we may assume, without loss of generality, that $w \in F$ and that $v \in H$. Now, observe that y_0 can miss no w, x_0-path since such a path extends inside H to a w, v-path missed by y_0. Similarly, no vertex $y \in Y$ nonadjacent to y_0 can miss a w, x_0-path; otherwise, y would miss a y_0, x_0-path, which is a contradiction. Let $y \in Y$ be a vertex that misses some w, x_0-path π. By the previous argument, y and y_0 are adjacent. However, since (w, v) is a dominating pair, y must intercept every w, v-path contained in $\pi \cup H$, implying that y is adjacent to some neighbor x' of x_0. But now we have reached a contradiction—x_0 and y_0 are joined by a path of length three.

With this the proof of Theorem 7.1 is complete. □

8. Concluding remarks and open problems.

Many families of graphs, including interval graphs, permutation graphs, trapezoid graphs, and cocomparability graphs, demonstrate a type of linear ordering on their vertex sets. It is precisely this linear order that is exploited, in one form or another, in a search for efficient algorithms for these classes of graphs. The classes mentioned are known to have wide-ranging practical applications. In addition, they are all subfamilies of the class of graphs called *AT-free graphs.*

This work is the first attempt, known to us, to investigate structural properties of the AT-free graphs. In this direction our contributions are as follows.

1. We showed that every connected AT-free graph has a dominating pair, that is, a pair of vertices such that every path joining them is a dominating set.
2. We provided properties of dominating pairs in AT-free graphs related to the concept of connected domination and diameter.
3. We provided a characterization of AT-free graphs in terms of dominating pairs.
4. We provided a characterization of AT-free graphs in terms of minimal triangulations.
5. We provided a decomposition theorem for AT-free graphs.

The authors have also addressed some algorithmic questions with respect to AT-free graphs. Specifically, in [9], $O(|V| + |E|)$ time algorithms are given for finding a pokable dominating pair in a connected AT-free graph $G = (V, E)$ and for finding all dominating pairs in a connected AT-free graph $G = (V, E)$ of diameter greater than three. Included in the latter algorithm is an efficient procedure for computing all of the "D" sets, with respect to a particular pokable dominating pair vertex. An extended abstract of [9] can be found in [11]. Some preliminary results and an alternative approach to the dominating pair problem can be found in [10] and [12], respectively.

Many other questions are still open. For example, it is well known [17] that cocomparability graphs have a linear ordering; this ordering exemplifies the linear structure we observe in interval graphs, permutation graphs, and trapezoid graphs. It would be interesting to see whether the AT-free graphs also possess some linear ordering. Such an ordering could, conceivably, be exploited for algorithmic purposes.

A further natural question to ask is "What are the roles of dominating pairs and pokable vertices in the subfamilies of AT-free graphs?" It is clear that the extreme vertices of any intersection representation, for a connected graph in any of the subfamilies, form a dominating pair. Some additional partial answers to this question have been given, in a slightly different setting, in [21] and [22]. Investigating further properties of dominating pairs and pokability in each of these particular families promises to be a fruitful area for further research.

Recently Möhring [20] has added to the understanding of the linear structure of AT-free graphs by showing that the pathwidth of an AT-free graph equals its treewidth.

Just as there are many families of perfect AT-free graphs, one would expect to see a rich hierarchy of families of nonperfect AT-free graphs. So far nothing is known here. Since perfect AT-free graphs strictly contain cocomparability graphs, it would be interesting to study the perfect AT-free graphs.

The fastest recognition algorithm known to us runs in $O(n^3)$ time with an n-vertex graph as input. It is a tantalizing open problem to produce a recognition algorithm that is more efficient, perhaps even optimal.

Acknowledgments. The authors are indebted to A. Parra for pointing out an error in the original proof of Lemma 5.14 and to E. Koehler for pointing out an error in the original proof of Theorem 5.8. In addition, the authors wish to thank two anonymous referees for an exceptionally thorough reviewing job. Their constructive comments have greatly contributed to improving the presentation.

REFERENCES

[1] K. A. Baker, P. C. Fishburn, and F. S. Roberts, *Partial orders of dimension* 2, Networks, 2 (1972), pp. 11–28.

[2] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, North–Holland, New York, 1976.

[3] K. S. Booth and G. S. Lueker, *Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms*, J. Comput. System Sci., 13 (1976), pp. 335–379.

[4] K. S. Booth and G. S. Lueker, *A linear time algorithm for deciding interval graph isomorphism*, J. Assoc. Comput. Mach., 26 (1979), pp. 183–195.

[5] F. Cheah, *A Recognition Algorithm for II-Graphs*, Ph.D. thesis, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada, 1990.

[6] D. G. Corneil and P. A. Kamula, *Extensions of permutation and interval graphs*, Congr. Numer., 58 (1987), pp. 267–275.

[7] D. G. Corneil, S. Olariu, and L. Stewart, *Asteroidal Triple-free Graphs*, Technical report 262/92, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada, 1992.

[8] D. G. Corneil, S. Olariu, and L. Stewart, *On the linear structure of graphs: Asteroidal triple-free graphs*, in Graph Theoretic Concepts in Computer Science WG '93, Utrecht, The Netherlands, 1993, Lecture Notes in Computer Science 790, J. van Leeuwen, ed., Springer-Verlag, Berlin, 1994, pp. 211–224.

[9] D. G. Corneil, S. Olariu, and L. Stewart, *Linear time algorithms for dominating pairs in asteroidal triple-free graphs*, SIAM J. Comput., submitted.

[10] D. G. Corneil, S. Olariu, and L. Stewart, *A linear time algorithm to compute a dominating path in an AT-free graph*, Inform. Process. Lett., 54 (1995), pp. 253–257.

[11] D. G. Corneil, S. Olariu, and L. Stewart, *Linear time algorithms for dominating pairs in asteroidal triple-free graphs (Extended Abstract)*, in Twenty-second International Colloquium on Automata, Languages, and Programming ICALP '95, Lecture Notes in Computer Science 944, Z. Fülöp and F. Gécseg, eds., Springer-Verlag, Berlin, 1995, pp. 292–302.

[12] D. G. Corneil, S. Olariu, and L. Stewart, *Computing a dominating pair in an asteroidal triple-free graph in linear time*, in Algorithms and Data Structures WADS '95, Lecture Notes in Computer Science 955, S. G. Akl, F. Dehne, J.-R. Sack, and N. Santoro, eds., Springer-Verlag, Berlin, 1995, pp. 358–368.

[13] S. Even, A. Pnueli, and A. Lempel, *Permutation graphs and transitive graphs*, J. Assoc. Comput. Mach., 19 (1972), pp. 400–410.

[14] T. Gallai, *Transitiv orientierbare Graphen*, Acta Math. Acad. Sci. Hungar., 18 (1967), pp. 25–66.

[15] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[16] M. C. Golumbic, C. L. Monma, and W. T. Trotter Jr., *Tolerance graphs*, Discrete Appl. Math., 9 (1984), pp. 157–170.

[17] D. Kratsch and L. Stewart, *Domination on cocomparability graphs*, SIAM J. Discrete Math., 6 (1993), pp. 400–417.

[18] 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.

[19] F. Maffray, *private communication*, 1992.

[20] R. H. Möhring, *Triangulating graphs without asteroidal triples*, Discrete Appl. Math., 64 (1996), pp. 281–287.

[21] S. Olariu, *On the homogeneous representation of interval graphs*, J. Graph Theory, 15 (1991), pp. 65–80.

[22] S. Olariu, *On sources in comparability graphs, with applications*, Discrete Math., 110 (1992), pp. 289–292.

[23] A. Parra, *private communication*, 1994.

SIAM J. Discrete Math.
Vol. 10, No. 3, pp. 431–435, August 1997

A LOWER BOUND FOR ADJACENCIES ON THE TRAVELING SALESMAN POLYTOPE[*]

A. SARANGARAJAN[†]

Abstract. We study adjacency of vertices on T_n, the asymmetric traveling salesman polytope of degree n. Applying a result of G. Boccara [*Discrete Math.*, 29 (1980), pp. 105–134] to permutation groups, we show that T_n has $\Omega((n-1)(n-2)!^2 \log n)$ edges, implying that the degree of a vertex of T_n is $\Omega((n-2)! \log n)$. We conjecture the degree to be $\Omega((n-2)!(\log n)^k)$ for any positive integer k. We compute the density function δ_n given by the fraction of the total number of vertices adjacent to a given vertex for small values of n, and conjecture that it decreases with n.

Key words. asymmetric traveling salesman polytope, adjacency

AMS subject classification. 52B12

PII. S0895480195283798

1. Introduction.

The *asymmetric traveling salesman polytope* (ATSP) is one of the widely studied polytopes in combinatorial optimization for its intrinsic relation to the traveling salesman problem. Many results are known about the facets of this polytope (see chapter 8 of [4] for a detailed survey), but not much is known about adjacency of vertices on this polytope. From an optimization point of view, studying adjacency helps in estimating the size of exact neighborhoods for local search algorithms. Such estimates have been carried out in [7] for the symmetric TSP.

The most common relaxation of the ATSP is the Birkhoff (or assignment) polytope B_n. We study the relationship between the faces of B_n and T_n, specifically the edges of T_n arising from certain two-dimensional faces of B_n. These edges are counted using a result of Boccara [2] giving us the lower bound for the number of edges of T_n. In particular, we show that T_n has $\Omega((n-1)(n-2)!^2 \log n)$ edges and thus each vertex of T_n has degree $\Omega((n-2)! \log n)$.

We define some terms that will be used for the rest of this paper. Let $\mathcal{S}_n$ be the *symmetric group* of degree n, i.e., the set of all permutations of $[n] := \{1, 2, \ldots, n\}$. We call a permutation even (odd) if it can be expressed as a product of an even (odd) number of transpositions. Two permutations are said to have different parity if one is even and the other odd. Given $\sigma \in \mathcal{S}_n$, we define the corresponding $n \times n$ permutation matrix $X_\sigma \in \mathbf{R}^{n^2}$ by

$$(X_\sigma)_{ij} := \begin{cases} 1 & \text{if } \sigma(i) = j, \\ 0 & \text{otherwise.} \end{cases}$$

We denote by B_n the *Birkhoff polytope* of degree n, given by

$$B_n := \operatorname{conv}\{X_\sigma : \sigma \in \mathcal{S}_n\}.$$

Let

$$\mathcal{T}_n := \{\sigma \in \mathcal{S}_n : \sigma \text{ is a cycle of length } n\} \subset \mathcal{S}_n.$$

[*] Received by the editors March 27, 1995; accepted for publication (in revised form) May 31, 1996.
http://www.siam.org/journals/sidma/10-3/28379.html

[†] Department of Mathematics, University of Michigan, Ann Arbor, MI 48109-1003 (sarang@math.lsa.umich.edu).

The ATSP of degree n is defined by

$$T_n := \operatorname{conv}\{X_\sigma : \sigma \in \mathcal{T}_n\},$$

so that

$$T_n \subset B_n \subset \mathbf{R}^{n^2}.$$

Thus if F is a face of B_n, then $F \cap T_n$ is a face of T_n induced by F.

We call two vertices *adjacent* on a polytope P if they form an edge of P. The *graph* of P is a graph whose nodes are the vertices of P with two nodes adjacent if the corresponding vertices are adjacent on P.

A *partition* of n is a sequence of positive integers $\lambda := (\lambda_1, \ldots, \lambda_k)$, with $\sum \lambda_i = n$ and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$ and we indicate this by $\lambda \vdash n$. We call λ an even (respectively, odd) partition if $n-k$ is even (respectively, odd). Let $\hat{e} := (1, 1, \ldots, 1)$ be the identity partition. If $\pi \in \mathcal{S}_n$ is a product of k disjoint cycles of lengths $l_1, \ldots, l_k$ (including cycles of length 1) in nonincreasing order, then $(l_1, \ldots, l_k)$ is a partition of n. We call $(l_1, \ldots, l_k)$ the *cycle type* of π. A *composition* of n is a sequence of positive integers $\lambda := \langle \lambda_1, \ldots, \lambda_k \rangle$ with $\sum \lambda_i = n$ and we indicate this by $\lambda \models n$. Hence a partition of n into k parts can define up to $k!$ distinct compositions by permuting the parts of the partition. A k-partition (respectively, a k-composition) of n is a partition (respectively, a composition) of n into k parts.

If $f(n)$ and $g(n)$ are two positive valued functions, then we say $f(n) = \Omega(g(n))$ if there exists a positive constant c such that $g(n) \leq cf(n)$, for all allowable values of n.

2. The edges of the ATSP.

We will denote the matrix X_σ by σ. We study faces of B_n induced by a pair of vertices σ, π. The following result found in [1, Proposition 2.1] and in [3] shows that these faces are in fact cubes.

PROPOSITION 2.1. *If $\sigma^{-1}\pi = \prod_{i=1}^{k} C_i \in \mathcal{S}_n$ where $C_1, \ldots, C_k$ are disjoint cycles, then the smallest face $F_{\sigma,\pi}$ of B_n containing both σ and π is a k-cube, where $k \leq \lfloor \frac{n}{2} \rfloor$. The vertices of $F_{\sigma,\pi}$ are given by $\sigma \, \Pi_{i \in S} C_i$, for $S \subseteq [k]$.*

The convex hull of the vertices of $F_{\sigma,\pi}$ that correspond to cycles of length n is a face of T_n. In particular, if $\sigma, \pi \in \mathcal{T}_n$, and $\sigma^{-1}\pi = C_1C_2$ is a product of two cycles of even length, then $F_{\sigma,\pi}$ is a 2-cube. Since σC_1 and σC_2 have parity different from that of σ, neither can be n-cycles. Thus σ and π are adjacent on T_n. We now find the number of such representations. To do this, we need the following result [2, Corollary 4.8].

PROPOSITION 2.2. *Let $l = (l_1, \ldots, l_k) \vdash n$. Let $g(l)$ be the number of ways of writing a permutation of cycle type l as a product of two n-cycles. Then*

$$g(l) = \frac{2(n-1)!}{n+1} \sum_{I \subseteq \{2,\ldots,k\}} (-1)^{|I|+s(I)} \binom{n}{s(I)}^{-1}, \tag{2.1}$$

if l is an even partition and zero otherwise. Here $s(I) = \sum_{i \in I} l_i$.

Thus if $l = (l_1, l_2)$, and n is even, then

$$g(l) = \frac{2(n-1)!}{n+1} \left(1 - (-1)^{l_1} \binom{n}{l_1}^{-1} \right), \tag{2.2}$$

and if $l = (l_1, l_2, l_3)$, and n is odd, then

$$g(l) = \frac{2(n-1)!}{n+1} \left(1 - (-1)^{l_1} \binom{n}{l_1}^{-1} - (-1)^{l_2} \binom{n}{l_2}^{-1} - (-1)^{l_3} \binom{n}{l_3}^{-1} \right). \tag{2.3}$$

This result is generalized in [6] to give the number of ways of writing a permutation as a product of an arbitrary number of n-cycles.

THEOREM 2.3. *Let e_n be the number of edges of T_n, $n > 3$. Then*

$$e_n = \Omega((n-1)(n-2)!^2 \log n).$$

Proof. Suppose $n = 2m$ is even. Let $\eta \in \mathcal{S}_n$ have cycle type $\lambda_r = (n-2r, 2r)$, $1 \le r \le m/2$. If $\sigma, \pi \in \mathcal{T}_n$, and $\sigma^{-1}\pi = \eta$, then by the argument before Proposition 2.2 σ and π are adjacent on T_n. By (2.2), the number of ways of writing η as a product of two n-cycles is

$$g(\lambda_r) = \frac{2(n-1)!}{n+1}\left(1 - \binom{n}{2r}^{-1}\right) \ge \frac{(n-1)!}{n+1} \qquad \text{as } \binom{n}{2r} \ge 2,$$

and every such pair of n-cycles induce an edge in T_n. Now the number of permutations of cycle type λ_r is at least $n!/(4r(n-2r))$. Hence, counting each edge exactly once,

$$\begin{aligned} e_n &\ge \frac{(n-1)!}{2(n+1)} \sum_{r=1}^{\lfloor m/2 \rfloor} \frac{n!}{4r(n-2r)} = \frac{(n-1)!^2}{4(n+1)} \sum_{r=1}^{\lfloor m/2 \rfloor} \left(\frac{1}{2r} + \frac{1}{n-2r}\right) \\ &\ge \frac{(n-1)!^2}{8(n+1)} \sum_{r=1}^{m-1} 1/r \ge \frac{(n-1)!^2}{8(n+1)} \log m = \Omega((n-1)(n-2)!^2 \log n). \end{aligned}$$

If $n = 2m+1$ is odd, then let $\eta \in \mathcal{S}_n$ have cycle type $\lambda'_r = (2m-2r, 2r, 1)$, $1 \le r \le m/2$. By (2.3), the number of pairs of n-cycles whose product is η is

$$g(\lambda'_r) = \frac{2(n-1)!}{n+1}\left(1 - \binom{n}{2m-2r}^{-1} - \binom{n}{2r}^{-1} + \frac{1}{n}\right) \ge \frac{(n-1)!}{n+1},$$

and each such pair induce an edge in T_n. Hence the bound for e_n follows as before. □

The lower bound for the degree now follows from observing that T_n is a vertex symmetric polytope. If $\sigma, \pi \in \mathcal{T}_n$, then there exists $\gamma \in \mathcal{S}_n$ such that $\sigma = \gamma\pi\gamma^{-1}$. Hence vertices π and π_1 are adjacent on T_n if and only if σ is adjacent to $\gamma\pi_1\gamma^{-1}$. As a result the degree of each vertex of T_n is the same value deg (n) and

$$\text{deg}\,(n) = \frac{2e_n}{(n-1)!} \ge \frac{(n-1)!}{4(n+1)} \log m = \Omega((n-2)! \log n). \tag{2.4}$$

3. Further discussions. The degree bound shows the graph of T_n to be fairly dense. This may not seem very surprising considering that the diameter of T_n is 2 as shown in [5]. For the Birkhoff polytope B_n, the degree of a vertex is known to be $\sum_{k=0}^{n-2} \binom{n}{k}(n-k-1)!$, while for the symmetric TSP it is $\Omega(\lfloor \frac{n-1}{2} \rfloor !)$ as shown in [7].

An expression for the number of edges of T_n can be written as

$$e_n = \sum_{\substack{\lambda \vdash n,\, \lambda \ne \hat{e} \\ \lambda \text{ even}}} \frac{n_\lambda\, g(\lambda)\, h(\lambda)}{2}, \tag{3.1}$$

where n_λ is the number of permutations of cycle type λ and $h(\lambda)$ is the fraction of the pairs of n-cycles (σ, π) which are adjacent on T_n and such that $\sigma^{-1}\pi$ has cycle type λ. Hence $h(\lambda) = 1$ if λ corresponds to a cycle or a product of two cycles of even length. It is natural to ask how large e_n would be if this summation is taken over all k-partitions λ for a fixed k. We estimate this partially.

CONJECTURE 3.1. *For any positive integer k,*

$$deg(n) = \Omega((n-2)!(\log n)^k).$$

The rationale for this conjecture stems from the following argument. From (2.1) it follows that for an even partition $l = (l_1, \ldots, l_k) \vdash n$,

$$g(l) \geq \frac{2(n-1)!}{n+1}\left(1 - \frac{2^{k-1}}{n}\right) \geq \frac{(n-1)!}{n+1} \qquad \text{for } n \geq 2^k,$$

since each term in the summation in (2.1) is at least $-1/n$ except for the term corresponding to the empty set which is 1. Let n_i be the number of permutations that can be expressed as a product of i disjoint cycles (including cycles of length 1). We estimate the asymptotic growth of n_i with *i fixed.* We have

$$n_i = \sum_{\langle l_1,\ldots,l_i\rangle \models n} \frac{n!}{i! l_1 l_2 \cdots l_i} = \Omega((n-1)!(\log n)^{i-1}).$$

The above sum is taken over all i-compositions of n. The last equality follows from the lemma below.

LEMMA 3.2. *Let*

$$f_i(n) := \sum_{\langle l_1,\ldots,l_i\rangle \models n} \frac{n}{l_1 l_2 \cdots l_i},$$

the sum being taken over all i-compositions of n. Then $f_i(n) = \Omega((\log n)^{i-1})$. In particular, we show that if $n \geq 2^i$, then $f_i(n) \geq c_i(\log n)^{i-1}$, $c_i = 2^{-(i-1)(i-2)/2}$.

Proof. We prove this by induction on i. The result is straightforward for $i = 1$. Then for $i > 1$ and $n \geq 2^i$,

$$f_i(n) \geq \sum_{l_1=1}^{\lfloor n/2 \rfloor} \frac{n}{l_1(n-l_1)} f_{i-1}(n-l_1).$$

As $l_1 \leq n/2$, we have by induction $f_{i-1}(n-l_1) \geq c_{i-1}(\log(n-l_1))^{i-2} \geq c_i(\log n)^{i-2}$ since $\log(n-l_1) \geq \log(n/2) \geq 1/2 \log n$. Hence

$$f_i(n) \geq c_i(\log n)^{i-2} \sum_{l_1=1}^{\lfloor n/2 \rfloor} \left(\frac{1}{l_1} + \frac{1}{n-l_1}\right) \geq c_i(\log n)^{i-1} = \Omega((\log n)^{i-1}),$$

proving the result. □

The conjecture amounts to showing that for each k, there exists a positive constant h_k such that $h(\lambda) \geq h_k$ for any k-partition λ of n and any n such that $n - k$ is even. If this were true, then summing (3.1) over all k-partitions of n yields

$$e_n \geq \frac{h_k n_k (n-1)!}{2(n+1)} = \Omega((n-1)(n-2)!^2(\log n)^{k-1})$$

when $n-k$ is even. If $n-k$ is odd, then we sum (3.1) over all $(k+1)$-partitions of n to get a bound of $\Omega((n-1)(n-2)!^2(\log n)^k)$ for e_n. This yields the conjectured bound for $\deg(n)$.

We define the *density* δ_n of T_n to be the fraction of the total number of vertices adjacent to a given vertex, i.e., $\delta_n := \deg(n)/((n-1)!-1)$. Our bounds on $\deg(n)$ show that $\delta_n = \Omega(\log n/n)$. It would be desirable to bound this number either away from 0 or below 1 as $n \to \infty$. Since T_3 and T_4 are simplices and T_5 is 2-neighborly, they have a density of 1. Using MAPLE, some other densities were computed by constructing the cube $F_{\sigma,\pi}$ for a fixed n-cycle σ and examining when the n-cycle π was adjacent to σ. These are tabulated below:

n	$\deg(n)$	δ_n
6	110	0.92
7	628	0.87
8	4174	0.83
9	32433	0.80

We observe that δ_n decreases with n for $n \leq 9$. We conjecture that this holds in general.

Acknowledgments. I thank Louis J. Billera and the referees for their comments and criticisms that have greatly improved the exposition of this paper.

REFERENCES

[1] L. J. Billera and A. Sarangarajan, *The combinatorics of permutation polytopes*, in Algebraic Combinatorics, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, L. Billera, C. Greene, R. Simion, and R. Stanley, eds., American Mathematical Society, Providence, RI, 1994.

[2] G. Boccara, *Nombre de représentations d'une permutation comme produit de deux cycles de longueurs données*, Discrete Math., 29 (1980), pp. 105–134 (in French).

[3] J. Heller, *Neighbor relations on the convex hull of cyclic permutations*, Pacific J. Math., 6 (1956), pp. 467–477.

[4] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, eds., *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, John Wiley, New York, 1985.

[5] M. W. Padburg and M. R. Rao, *The traveling salesman problem and a class of polyhedra of diameter two*, Math. Programming, 7 (1974), pp. 32–45.

[6] R. P. Stanley, *Factorization of permutations into n-cycles*, Discrete Math., 37 (1981), pp. 255–262.

[7] P. Weiner, S. L. Savage, and A. Bagchi, *Neighborhood search algorithms for guaranteeing optimal traveling salesman tours must be inefficient*, J. Comput. Systems Sci., 12 (1976), pp. 25–35.

SIAM J. DISCRETE MATH.
Vol. 10, No. 3, pp. 436–446, August 1997

COUNTING PROBLEMS ASSOCIATED WITH STEINER TREES IN GRAPHS[*]

J. SCOTT PROVAN[†] AND MANOJ K. CHARI[‡]

Abstract. This paper considers counting problems associated with K-spanning and K-disconnecting sets for a specified terminal set K in an undirected graph G. In particular, we consider the problems of computing the number of *Steiner trees* and *min K-cuts* for G, as well as K-spanning and K-disconnecting sets of cardinality close to the minimum values. Among other things, these numbers are critical to the efficient approximation of K-connected reliability measures in stochastic networks. Although the counting problems considered in this paper are NP-hard in general, a large number of methods for *finding* shortest paths, min cuts, and Steiner trees in graphs can be extended to efficiently *count* K-spanning and K-disconnecting sets in important special cases.

Key words. Steiner tree, cut, counting, enumeration

AMS subject classifications. 05C05, 05C30, 90B25

PII. S0895480194262862

1. Introduction.

The K-connectedness problem considered in this paper has as input undirected graph $G = (V, E)$ along with subset K of *terminal* vertices. (Parallel edges and loops are allowed and are in fact necessary for some of the algorithms given in the paper.) We denote by n, m, and k the number of vertices, edges, and terminals of G, respectively. A subset S of edges of G is called *K-connected* if every pair of vertices in K can be connected by a path in S. S is called a *K-disconnecting set* if $E \backslash S$ is not K-connected. The minimum cardinality K-connected and K-disconnecting sets in G are called, respectively, K-Steiner trees and min K-cuts. We are concerned in this paper with the problems of counting the following numbers:

$\tau(G, K)$ = the number of K-Steiner trees in G;
$\gamma(G, K)$ = the number of min K-cuts in G;
$\kappa(G, K, r)$ = the number of K-connected sets of cardinality r in G.
$= \binom{m}{r}-$the number of K-*disconnecting* sets of cardinality $m - r$ in G.

The numbers $\tau(G, K)$, $\gamma(G, K)$, and $\kappa(G, K, r)$ are important measures in the assessment of reliability and vulnerability with respect to K-connectivity in networks. The numbers $\kappa(G, K, 0), \kappa(G, K, 1), \ldots, \kappa(G, K, m)$ are in fact the coefficients of the *K-connectedness reliability* polynomial, as given in [4]. The terms $\tau(G, K)$ and $\gamma(G, K)$ comprise the "extreme" values for $\kappa(G, K, r)$; that is, if we denote by l and c the cardinalities of a K-Steiner tree and min K-cut, respectively, then $\tau(G, K) = \kappa(G, K, l)$ and $\gamma(G, K) = \binom{m}{c} - \kappa(G, K, m - c)$. Any values of $\kappa(G, K, r)$ outside the range $t, t + 1, \ldots, m - c$ are trivial, being either 0 or $\binom{m}{r}$. Thus the values $\tau(G, K)$ and $\gamma(G, K)$ provide an important measure of the distribution of small and large K-connected sets, respectively, and in fact the values of $\kappa(G, K, r)$ for r close to t and to $m - c$ provide the most significant terms of the reliability polynomial for edge

[*] Received by the editors February 4, 1994; accepted for publication (in revised form) June 19, 1996.
http://www.siam.org/journals/sidma/10-3/26286.html

[†] Department of Operations Research, University of North Carolina, Chapel Hill, NC 27599-3180 (scott_provan@unc.edu). The work of this author was partially supported by NSF grant CCR-9200572.

[‡] Mathematics Department, Louisiana State University, Baton Rouge, LA 70803-4918 (chari@marais.math.lsu.edu). The work of this author was partially supported by LEQSF grants (92-94)-RD-A-09 and (95-98)-RD-A-08 from the Louisiana Board of Regents.

operating probabilities close to 0 and 1, respectively. These measures play a critical role in the approximation scheme developed in [3], where the theory of *Steiner complexes* is used to develop bounds on K-connectedness reliability in graphs using these measures.

Although much study has been devoted to *finding* Steiner trees and min cuts, relatively little has been done with regard to *counting* these objects. Almost all of the counting problems considered here are NP-hard—technically #P-complete [17]—in the general case, and hence are unlikely to be computable in polynomial time. The purpose of this paper is to give algorithms for computing $\tau(G, K)$, $\gamma(G, K)$, and $\kappa(G, K, r)$ and to identify some important classes of problems where this can be accomplished in polynomial time. The methodology takes advantage of several known counting techniques in graph theory and also points out the interesting interplay between the optimization algorithms for these problems and the associated counting algorithms.

2. Review of some known counting algorithms. In this section we review several important algorithms in the literature which, in addition to giving values for $\tau(G, K)$, $\gamma(G, K)$, and $\kappa(G, K, r)$ for $K = \{s, t\}$ and $K = V$, will also be used as subroutines for computing these measures for general K. We will restrict ourselves here to giving only the basic results; for proofs and details of the algorithms the reader is referred to the associated references. All complexities in this paper are given in terms of the number of arithmetic operations. As a technical matter, it should be pointed out that in many of the procedures given in this paper the associated numbers can be of the order of 2^m. Thus in a logarithmic computational model the actual complexity could involve another factor of m.

2.1. Counting shortest (s, t)-paths and min (s, t)-cuts. The paper [1, section III-C] gives a method for computing $\tau(G, \{s, t\})$, the number of shortest paths between two vertices s and t in G. The method is based on a simple modification of a breadth-first shortest-path algorithm and gives the following result [1].

LEMMA 2.1. *The number $\tau(G, \{s, t\})$ of shortest (s, t)-paths in G—and in fact the number of shortest (s, v)-paths for all $v \in V$ in G—can be determined using $O(n + m)$ arithmetic operations.*

Procedures for counting cuts often involve first explicitly *listing* the collection of cuts, and keeping a count as the listing proceeds. Algorithms for listing the collection of min $\{s, t\}$-cuts in a graph have been given by several authors [1], [7], [11], [14]. All algorithms are based on solving a max flow problem on G, and then using the resulting flow to identify the min $\{s, t\}$-cuts. The results of these papers can be summarized in the following lemma.

LEMMA 2.2. *The collection of min $\{s, t\}$-cuts of G can be listed in time $O(m\gamma + \alpha)$, where $\gamma = \gamma(G, \{s, t\})$ is the number of min cuts and α is the time to find an $\{s, t\}$-cut.*

Note that this is *not* a polynomial-time method for computing $\gamma(G, \{s, t\})$, since $\gamma(G, \{s, t\})$ can grow exponentially in the size of G. Indeed, the problem of computing $\gamma(G, \{s, t\})$ in general graphs is NP-hard [13]. However, for certain cut-listing problems, for example, those associated with $\gamma(G, V)$, the size of the associated collection of $\{s, t\}$-cuts *is* polynomially bounded in the size of the graph. Thus the cut-listing algorithm provides a polynomial-time method for computing $\gamma(G, V)$ (see [1]) as well as certain values of $\gamma(G, K)$ and $\kappa(G, K, r)$, as we show later in this paper.

There is one interesting situation for which it is possible to count the number of min $\{s, t\}$-cuts without explicitly listing them. This occurs when G is an (s, t)-*planar* *graph*—that is, G has a plane layout with s and t lying on the exterior face. In this case, the number of min $\{s,t\}$-cuts of G is equal to the number of shortest (s,t)-paths in the *(s,t)-planar dual graph* of G. Using Lemma 2.1 we have the following.

Lemma 2.3. *The number of min $\{s,t\}$-cuts in an (s,t)-planar graph G can be computed using $O(n)$ arithmetic operations.*

2.2. Counting spanning and disconnecting sets. The value $\tau(G,V)$ is simply the number of *spanning trees* in G and the computation of $\tau(G,V)$ reduces by the matrix tree theorem [9], [16] to computing the determinant of a $(n-1)\times(n-1)$ matrix, giving us the following result.

Lemma 2.4. *The number of spanning trees of G can be computed using $O(n^3)$ arithmetic operations.*

In spite of having a polynomial algorithm for computing $\kappa(G,V,n-1)=\tau(G,V)$, the complexity of computing even the value of $\kappa(G,V,n)$ on general graphs is still an open problem. When G is *planar*, however, the paper [10] gives a recursive formula for computing the value of $\kappa(G,V,n-1+d)$, starting with the matrix tree theorem. Although the complexity of the recursion was not explicitly analyzed, it can be easily derived, and results in the following lemma.

Lemma 2.5. *For any planar graph G, $\kappa(G,V,n-1+d)$ can be computed using $O(n^3m^d)$ arithmetic operations.*

Finally, the enumeration of "almost minimum cuts"—or, equivalently, the computation of $\kappa(G,V,r)$ for values of r close to $m-c$—is studied in [15]. In that paper (s,t)-disconnecting sets of cardinality $c+d$ are counted by a careful "factoring" on the edges of G. The results are summarized in the following lemma.

Lemma 2.6. *For any fixed d, the number $\kappa=\kappa(G,V,m-c+d)$ of disconnecting sets of G of cardinality $c+d$ can be computed in time $O(m\alpha\kappa)=O(\alpha m^d n^{d+2})$, where α is the time to find a min V-cut for G.*

3. Counting min *K*-cuts. In this section we use the material in section 2 to develop algorithms for computing $\gamma(G,K)$. The essential features of the procedures presented here were given in [1] for the case $K=V$. They are based on the fact that a min K-cut in G is always a min $\{s,t\}$-cut for some pair s and t of vertices in K. Hence the cardinality c of a min K-cut can be computed in polynomial time for any graph by simply computing the cardinality of a min $\{s,t\}$-cut for each pair s,t of elements of K and then taking the minimum over all these values. Further, the set of min K-cuts is the union of the set of min $\{s,t\}$-cuts for all pairs s and t for which the cardinality of a min $\{s,t\}$-cut is equal to c. What remains is to process the collections of $\{s,t\}$-cuts for the appropriate pairs s,t in such a way as to avoid repeating cuts over different pairs of s and t. This can be accomplished by using the following result, whose proof is routine.

Lemma 3.1. *Let c be the cardinality of a min K-cut in G, and let s and t be a pair of elements of K for which the cardinality of a min $\{s,t\}$-cut in G is equal to c. Let $\hat{G}$ be the graph obtained from G by identifying t with s and $\hat{K}=K\setminus\{t\}$ the corresponding terminal set. If the cardinality of a min $\hat{K}$-cut in $\hat{G}$ is also c, then*

$$\gamma(G,K)=\gamma(G,\{s,t\})+\gamma(\hat{G},\hat{K}),$$

otherwise $\gamma(G,K)=\gamma(G,\{s,t\})$.

Using Lemma 3.1 and the cut-listing result of Lemma 2.2, we have the following theorem.

Theorem 3.2. *For any graph G and terminal set K, the value of $\gamma=\gamma(G,K)$ can be computed in time $O(m\gamma+k\alpha)$, where α is the time to find a min $\{s,t\}$-cut.*

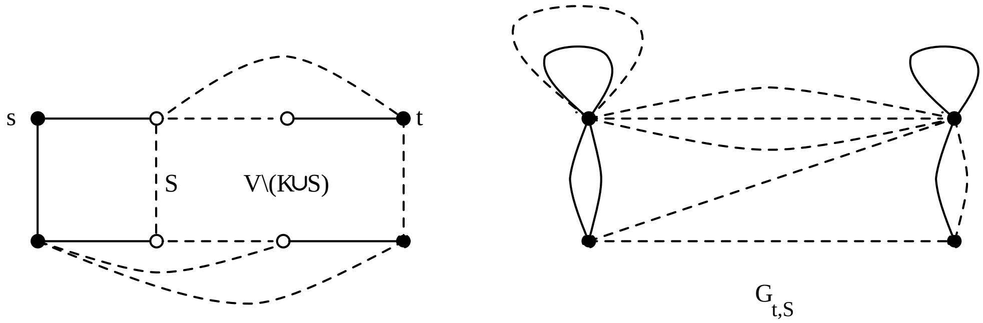


FIG. 1. *Construction of $G_{t,S}$.*

Again note that this does not constitute a polynomial time algorithm, since $\gamma(G,K)$ can grow exponentially in the size of G.

There are two interesting special instances when a polynomial bound on $\gamma(G,K)$ can be given. Bixby [2] proves that $\gamma(G,V)$ is at most $\binom{m}{2}$. We extend this results to K-cuts by showing that K is "close to" V—in particular, if K differs from V by no more than a fixed number—then we can also get a polynomial bound on $\gamma(G,K)$ and hence on the time to compute it. Begin by fixing an arbitrary element $s \in K$. Next, for every $t \in K, t \neq s$ and every subset $S \subseteq V \setminus K$ we define the graph $G_{t,S} = (V_{t,S}, E_{t,S})$ as the graph obtained from G by identifying the vertices of S with s and identifying the vertices of $V \setminus (K \cup S)$ with t. Figure 1 illustrates the construction of $G_{t,S}$ with the solid vertices being the elements of K, and S and $V \setminus (K \cup S)$ as marked.

Now it is straightforward to show that if X is a K-disconnecting set of G, then X is also $V_{t,S}$-disconnecting set in the graph $G_{t,S}$, where $t \in K$ is some vertex in the component of the subgraph $(V, E \setminus X)$ which does not contain s and S is the subset of vertices of $V \setminus K$ which are in the same component as s. In particular, the set of min K-cuts of G is contained in the union of the minimum cardinality $V_{t,S}$-cuts of graphs $G_{t,S}$, taken over all choices of t and $S \subseteq V \setminus K$. There are at most $k2^{n-k}$ such graphs and by applying Bixby's bound to each of these graphs we have the following.

THEOREM 3.3. $\gamma(G,K) \leq \binom{m}{2} k2^{n-k}$.

By applying Theorem 3.2 we obtain the following result.

COROLLARY 3.4. *The value of $\gamma(G,K)$ can be computed in time $O(m^3k2^{n-k} + k\alpha)$, where α is the time to find a min $\{s,t\}$-cut. In particular, for fixed d, $\gamma(G,K)$ can be computed in polynomial time for any $G = (V,E)$ and K with $n - k \leq d$.*

A min K-cut counting procedure based on Lemma 3.1 can also be implemented in polynomial time in the case where the graph G is *K-planar*, that is, has a plane layout with the elements of K lying entirely on the exterior boundary. In this case if the pair s and t of Lemma 3.1 is chosen to be the *closest* pair of eligible terminals with respect to their distance along the exterior boundary of G, then the graph $\hat{G}$ will also be $\hat{K}$-planar. In particular, the value of $\gamma(G,\{s,t\})$ required in Lemma 3.1 can be computed in $O(m)$ arithmetic operations by Lemma 2.3, since G is (s,t)-planar for each of the intermediate graphs. We thus have Theorem 3.5.

THEOREM 3.5. *If G is a K-planar graph, then $\gamma(G,K)$ can be computed using $O(km + k\alpha)$ arithmetic operations, where α is the time to find a min $\{s,t\}$-cut.*

4. Counting Steiner trees in graphs. In this section, we consider the problem of computing $\tau(G,K)$, the number of K-Steiner trees in G. The problem of counting K-Steiner trees is NP-hard [17], and in fact even the problem of *finding* a K-Steiner tree is also NP-hard. There are several instances, however, where the Steiner tree problem does have a polynomial time solution, and it turns out that in each of these instances the algorithm for finding the Steiner tree can be modified to also count the number of such trees.

4.1. The Hakimi method. The method given by Hakimi [8] finds K-Steiner trees by essentially solving a series of spanning tree problems. It can be modified to compute $\tau(G, K)$ in polynomial time for instances where almost all of the vertices are terminal vertices. It is based on the fact that a K-Steiner tree T in G is simply a spanning tree on its incident set of vertices. Thus one way to find the cardinality l of a K-Steiner tree is to determine the minimum cardinality of a set of vertices S containing K for which the subgraph G_S induced by S is connected. That is,

$$l = \min\{|S| : \ K \subseteq S \subseteq V,\ G_S \text{ connected}\}.$$

Further, for each such minimum cardinality S *every* spanning tree in G_S is a K-Steiner tree of G, and these are distinct for distinct S. We can therefore compute the *number* of K-Steiner trees by the analogous formula

$$\tau(G, K) = \sum \{\tau(G_S, V) : \ K \subseteq S \subseteq V, |S| = l,\ G_S \text{ connected}\}.$$

By applying the above formula, and using Lemma 2.4, we obtain the following result.

THEOREM 4.1. *$\tau(G, K)$ can be computed using $O(n^3\, 2^{n-k})$ arithmetic operations. In particular, for any fixed d there exists a polynomial-time algorithm for computing $\tau(G, K)$ over all G and K with $n - k \leq d$.*

4.2. The Dreyfus–Wagner method. The method of Dreyfus and Wagner [5] builds K-Steiner trees by recursively constructing and "patching together" smaller Steiner trees. The method can be applied to yield a polynomial time method of finding the cardinality of K-Steiner trees when the terminal set is small and also when G is K-planar. We give a more exacting version of their method which can be modified to count the number of Steiner trees as well. For $S \subseteq K$ and $u \in V$ define

$$\mathcal{T}(u, S) = \{T : \ T \text{ is an } (S + u)\text{-Steiner tree}\},$$

and for $S \subseteq K$, $|S| \geq 2$, and $v \in V$ define

$$\mathcal{T}_2(v, S) = \left\{ \begin{matrix} T\text{: } T \text{ is a minimum cardinality } (S{+}v)\text{-connected set for which } v \\ \text{separates at least two vertices of } S \text{ (including possibly } v \text{ itself)} \end{matrix} \right\}.$$

(Throughout this section we will use the notation $S{+}v = S \cup \{v\}$ and $S{-}v = S \setminus \{v\}$.) Also, for $u, v \in V$ let $\mathcal{P}(u, v)$ be the collection of minimum cardinality (u, v)-paths in G. Computing the cardinality and number of K-Steiner trees involves, respectively, the recursive evaluation of the functions

$$c\mathcal{T}(u, S) \text{ etc.} = \text{the cardinality of an element in } \mathcal{T}(u, S) \text{ etc.}$$
$$\#\mathcal{T}(u, S) \text{ etc.} = \text{the number of elements in } \mathcal{T}(u, S) \text{ etc.}$$

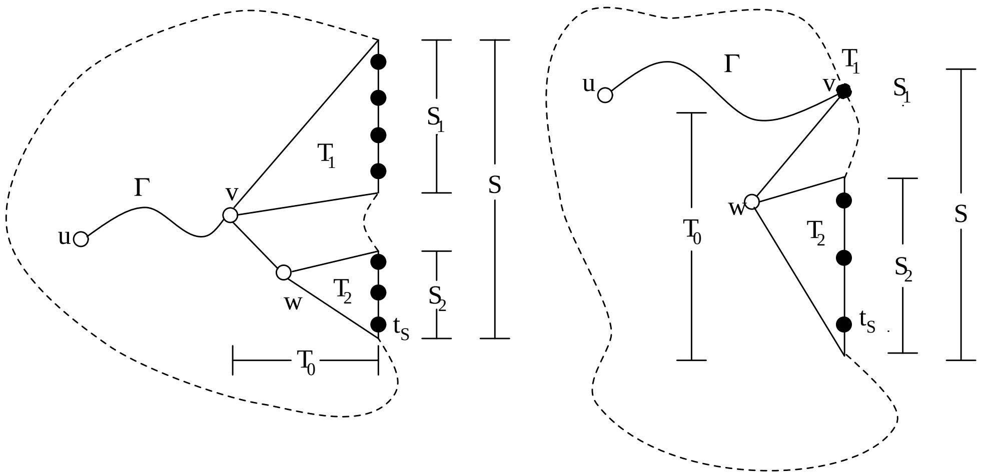


FIG. 2. *Steiner tree decompositions.*

It follows that $\tau(G,K) = \#\mathcal{T}(s, K-s)$, where s is some arbitrarily chosen element of K. The Dreyfus–Wagner method essentially computes $c\mathcal{T}(s, K-s)$; to extend this method to counting the number of these sets, we need to first give a canonical way to uniquely edge-partition an element of $\mathcal{T}(u,S)$ or $\mathcal{T}_2(u,S)$ into smaller pieces, each of which is recursively in one of these classes of sets. This will enable recursive equations to be given for $c\mathcal{T}(u,S)$ and $\#\mathcal{T}(u,S)$.

For the starting cases, note that if S consists of the single element v, then $\mathcal{T}(u,S) = \mathcal{P}(u,v)$. Assume then that S has at least two elements and arbitrarily identify one of these elements t_S as an "anchor" element. First let T be an element of $\mathcal{T}(u,S)$. Starting at u, follow the unique path Γ in T from u until the first vertex v is reached whose removal disconnects at least two elements of S. In particular, v is the first vertex which is either in S itself or is adjacent to at least two distinct edges on paths continuing to elements of S. Figure 2 gives examples of these two cases; the solid vertices in the figure are the terminals in S. Note that v could be u itself in the case where u is in S or has degree greater than 1. The vertex v therefore partitions the edges of T uniquely into the path $\Gamma \in \mathcal{P}(u,v)$ and the remaining tree T_0, which must in turn be in $\mathcal{T}_2(v,S)$. Next, let T be an element of $\mathcal{T}_2(v,S)$. Note that since T is an edge-minimal $(S+v)$-connected set, then there is a unique path from v to each element of S and every edge out of v lies on a path from v to at least one element of S. It follows that either v is t_S itself or there exists a unique edge (v,w) adjacent to v whose removal disconnects v from t_S in T. In the first case we have that $T \in \mathcal{T}(t_S, S-t_S)$. If $v \neq t_S$, let T_1 and T_2 be the two trees obtained by removing (v,w) from T, with $v \in T_1$ and $t_S \in T_2$. Let S_1 and S_2 be the elements of S lying in T_1 and T_2, respectively, and note that neither S_1 nor S_2 is empty. Thus T has a unique partition as $T_1 \cup T_2 \cup \{(v,w)\}$, with $T_1 \in \mathcal{T}(v,S_1)$ and $T_2 \in \mathcal{T}(w,S_2)$.

From the above discussion we can determine the cardinality measures $c\mathcal{T}(u,S)$ and $c\mathcal{T}_2(v,S)$ as follows. Clearly, $c\mathcal{T}(u,\{v\}) = c\mathcal{P}(u,v) =$ the length of a shortest (u,v)-path. For $|S| \geq 2$, the above discussion translates into the following set of equations for $c\mathcal{T}(u,S)$ and $c\mathcal{T}_2(v,S)$:

$$c\mathcal{T}(u,S) = \min\{c\mathcal{P}(u,v) + c\mathcal{T}_2(v,S) : v \in V\}, \tag{1}$$

$$c\mathcal{T}_2(v,S) = \begin{cases} c\mathcal{T}(t_S, S - t_S) & v = t_S, \\ \min \left\{ \begin{array}{l} c\mathcal{T}(v,S_1) + c\mathcal{T}(w,S_2) + 1: \\ (v,w) \in E,\ S_1, S_2 \text{ a nontrivial} \\ \text{partition of } S \text{ with } t_S \in S_2 \end{array} \right\} & v \neq t_S. \end{cases} \tag{2}$$

This results in the following procedure for computing the cardinality of a K-Steiner tree.

```
K-Steiner tree procedure
  compute $c\mathcal{T}(u,\{v\}) = c\mathcal{P}(u,v)$ for all pairs $u$ and $v$ of vertices
  for $S \subseteq K$ of cardinality at least 2 in increasing order of cardinality do
    compute $c\mathcal{T}_2(u,S)$ for all $u \in V$ using (2)
    compute $c\mathcal{T}(v,S)$ for all $v \in V$ using (1)
  end for
```

The evaluation of equation (2) dominates the complexity, with the total number of disjoint subset pairs S_1 and S_2 considered in all of the evaluations of equation (2) being bounded above by 3^k and used in turn with each of the n vertices. With $O(n^3)$ being the complexity to compute all of the shortest path lengths $c\mathcal{P}(u,v)$, we have the following result.

Theorem 4.2 (see [5]). *The cardinality of a K-Steiner tree in any graph can be computed using $O(n^3 + n3^k)$ arithmetic operations.*

From the previous argument, the actual collections $\mathcal{T}(u,S)$ and $\mathcal{T}_2(u,S)$ can be given the following characterizations:

$$\mathcal{T}(u,S) = \dot\cup\{\mathcal{P}(u,v) \times \mathcal{T}_2(v,S) :\ v \in V \ s.t.\ c\mathcal{T}(u,S) = c\mathcal{P}(u,v) + c\mathcal{T}_2(v,S)\},$$

$$\mathcal{T}_2(v,S) = \begin{cases} \mathcal{T}(t_S, S - t_S) & v = t_S, \\ \dot\cup \left\{ \begin{array}{l} \mathcal{T}(v,S_1) \times \mathcal{T}(w,S_2) \times \{(v,w)\}\text{: } (v,w) \in E, \\ S_1, S_2 \text{ a nontrivial partition of } S \text{ with } t_S \in S_2, \\ \text{and } c\mathcal{T}_2(v,S) = c\mathcal{T}(v,S_1) + c\mathcal{T}(w,S_2) + 1 \end{array} \right\} & v \neq t_S, \end{cases}$$

where $\dot\cup$ = disjoint union and $\times$ = Cartesian product. From this characterization we can immediately give the associated recursive equations for $\#\mathcal{T}(v,S)$ and $\#\mathcal{T}_2(v,S)$ by simply replacing $\mathcal{T}(v,S)$ by $\#\mathcal{T}(v,S)$, $\mathcal{T}_2(v,S)$ by $\#\mathcal{T}_2(v,S)$, $\dot\cup$ by $\sum$ and $\times$ by $\cdot$ (standard multiplication). This leads to the following pair of recursive formulas:

$$\begin{aligned} &\#\mathcal{T}(u,S) \\ &\quad = \sum\{\#\mathcal{P}(u,v) \cdot \#\mathcal{T}_2(v,S) :\ v \in V \text{ s.t. } c\mathcal{T}(u,S) = c\mathcal{P}(u,v) + c\mathcal{T}_2(v,S)\}, \end{aligned} \tag{3}$$

$$\#\mathcal{T}_2(v,S) = \begin{cases} \#\mathcal{T}(t_S, S - t_S) & v = t_S, \\ \sum \left\{ \begin{array}{l} \#\mathcal{T}(v,S_1) \cdot \#\mathcal{T}(w,S_2)\text{: } (v,w) \in E,\ S_1, S_2 \\ \text{a nontrivial partition of } S \text{ with } t_S \in S_2, \\ \text{and } c\mathcal{T}_2(v,S) = c\mathcal{T}(v,S_1) + c\mathcal{T}(w,S_2) + 1 \end{array} \right\} & v \neq t_S. \end{cases} \tag{4}$$

The equations above are processed exactly as in the K-Steiner Tree Procedure, with equations (3) and (4) replacing equations (1) and (2). The complexity analysis is the same. The value of $\#\mathcal{P}(u,v) = \tau(G,\{u,v\})$ can be computed using $O(m)$ arithmetic operations (Lemma 2.1) per initial vertex u, and so we summarize the computational complexity of the counting version of the Dreyfus–Wagner method in the following result.

THEOREM 4.3. *$\tau(G,K)$ can be computed using $O(nm + m3^k)$ arithmetic operations. In particular, for any fixed d, there exists a polynomial-time algorithm for computing $\tau(G,K)$ over all G and K with $k \leq d$.*

4.3. Counting Steiner trees in *K*-planar graphs. It turns out that the Dreyfus–Wagner method can be modified to give a polynomial algorithm for finding a K-Steiner tree in any K-planar graph. This has been given in [6] and [12] and follows closely the presentation given in the previous section.

Assume that G is *biconnected*, that is, contains no vertex whose removal disconnects G. (If not, then the collection of K-Steiner trees in G is the Cartesian product of the collection of Steiner trees in each of the biconnected components of G, and we can compute the cardinality and number of these trees in G from that of each of the components.) We therefore have that the exterior boundary of G is a polygon which contains every terminal. Define an *interval* to be the set of terminals found in any path along the boundary of G. In [6] (Theorem 4), it was established that removal of any edge of a K-tree results in two subtrees both of which span an interval of the terminal set. This means that if S is an interval and the anchor element t_S is always chosen to be the *clockwise-most* element of S, then the tree partitioning given in the section 4.2 for elements of $\mathcal{T}_2(v,S)$, $v \neq t_S$ will result in terminal subsets S_1 and S_2 which are also intervals of K and the partition of elements of $\mathcal{T}_2(v,S)$, $v = t_S$ will likewise result in terminal sets which are intervals. Again, see Fig. 2, where the dotted lines represent the boundary of G. Since the initial terminal set $K - s$ is itself an interval, then equations (1), (2), (3), and (4) need only be computed for intervals of $K - s$. There are at most k^2 such intervals (defined by their endpoints), so that at most nk^2 equations of each kind need to be evaluated, with equations (1) and (3) requiring $O(n)$ arithmetic computations to compute and (2) and (4) requiring $O(nk)$ arithmetic computations. We therefore have the following result.

THEOREM 4.4. *For G and K such that G is K-planar, $\tau(G,K)$ can be computed using $O(n^2k^3)$ arithmetic operations.*

This is a factor of k greater than the complexity of the algorithms given in [6] and [12] due to the more precise breakdown of the Steiner trees. It is an interesting question as to whether an $O(n^2k^2)$ algorithm is possible, as is obtained in those papers.

5. Counting *K*-spanning sets. The problem of computing $\kappa(G,K,r)$ for general r is understandably more difficult than that of computing either $\tau(G,K)$ or $\gamma(G,K)$. Here we give two extensions of the results in sections 3 and 4 which allow $\kappa(G,K,r)$ to be computed for values of r close to either of its extreme values and special instances of G and K. The first method extends the results of sections 3 and the method in [15] to compute $\kappa(G,K,r)$ for k close to n and r close to $m - c$. We start by extending the results in [2] and [15] to provide an upper bound for $\kappa(G,K,r)$.

LEMMA 5.1. *$\kappa(G,K,r) \leq kn^2 2^{n-k} m^{m-r-c}$.*

Proof. We recall the definition of the graphs $G_{t,S}$ in the discussion before Theorem 3.3. The collection of K-disconnecting sets of cardinality $\bar{r} = m - r$ (which corresponds to the collection of K-connected sets of cardinality r) is contained in the union of the $V_{t,S}$-disconnecting sets of cardinality $\bar{r}$ in each of the graphs $G_{t,S} = (V_{t,S}, E_{t,S})$, taken over all $S \subseteq V \setminus K$ and $t \in K$. This means that the number

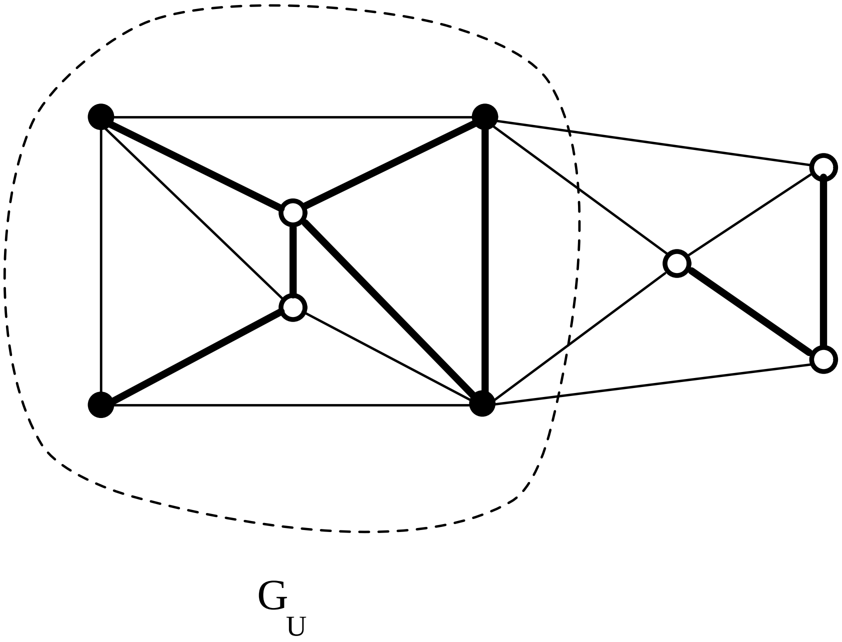


FIG. 3. *A K-spanning set.*

of K-disconnecting sets of cardinality $\bar{r}$ in G is bounded above by $k2^{n-k}$ times the maximum number of $V_{t,S}$-disconnecting sets of cardinality $\bar{r}$ in a graph of the type $G_{t,S}$. Using the bound $\gamma(G_{t,S}, V_{t,S}) \leq |V_{t,S}|^2$ given in [2] it is easy to see that the number of $V_{t,S}$-disconnecting sets of $G_{t,S}$ of cardinality $\bar{r}$ is bounded above by $|E_{t,S}|^{\bar{r}-c}\gamma(G_{t,S}, V_{t,S}) \leq m^{m-r-c}n^2$. The lemma follows. □

Finally, we note that the algorithm given in [15] to determine $\kappa(G, V, m-c+d)$ remains valid even when V is replaced by a general subset K. In particular, we have that $\kappa = \kappa(G, K, r)$ can be computed in time $O(m\alpha\kappa)$, where α is the time to find a min K-cut. This result in combination with Lemma 5.1 gives us the following.

THEOREM 5.2. *$\kappa(G, K, r)$ can be computed using $O(\alpha k n^2 2^{n-k} m^{m-r-c+1})$ arithmetic operations, where α is the time to find a min K-cut. In particular, for any fixed d, there is a polynomial-time algorithm for computing $\kappa(G, K, r)$ for G, K, and r, with $k \geq n-d$ and $r \geq m-c-d$.*

The second case where $\kappa(G, K, r)$ can be computed efficiently is where G is planar, with r close to k. In this case we can use section 2.2 to count the appropriate spanning sets of G to obtain $\kappa(G, K, r)$. In particular, let S be a K-spanning set of cardinality r. Then S has a unique connected component S_0 which spans K. Let $i = |S_0|$, let U be the set of vertices spanned by S_0, and let $G_U = (U, E_U)$ be the subgraph of G spanned in turn by U. Figure 3 gives an example of this having $r = 8$ and $i = 6$, with K represented by solid vertices and S by bold edges. Then G_U is comprised of the vertices and edges inside the circled region, with S_0 the set of bold edges in G_U. It follows that S_0 is a spanning set for G_U and, further, no edge of $S \setminus S_0$ is incident to a vertex of G_U. From this we get that the total number of K-spanning sets of G of cardinality r whose K-spanning component has vertex set U and edge set of cardinality n_0 is $\kappa(G_U, U, i) \cdot \binom{\bar{n}}{r-i}$, where $\bar{n}$ is the number of edges not incident to a vertex of U. By summing the K-spanning sets of G whose K-spanning component is U, over all relevant component edge-cardinalities i and vertex sets U, we obtain number of K-spanning sets of cardinality r in G. The details are given below.

K-SPANNING SET COUNTING PROCEDURE

```
set $\kappa = 0$
for $K \subseteq U \subseteq V$ such that $G_U$ is connected set
    $$\kappa = \kappa + \sum_{i=|U|-1}^{r} \kappa(G_U, U, i) \binom{\bar{n}}{r-i},$$
        where $\bar{n}$ = the number of edges not incident to a vertex of $U$
return $\kappa(G, K, r) = \kappa$.
```

THEOREM 5.3. *For planar graph G and $r = k - 1 + d$, the K-Spanning Set Counting Procedure computes $\kappa(G, K, r)$ using $O(m^d n^{d+3})$ arithmetic operations. In particular, for any fixed d there exists a polynomial time algorithm which computes $\kappa(G, K, r)$ for any planar graph G and terminal set K such that $r \leq k - 1 + d$.*

Proof. The value of each $\kappa(G_U, U, i)$ can computed using the method given in section 2.2, with the following inequalities holding:

$$k \leq |U| \leq i + 1 \leq r + 1 \leq k + d.$$

By Lemma 2.5, each such computation can be done in time $O(|U|^3 |E_U|^{i-|U|+1}) = O(n^3 m^d)$. Further, the sets U considered in the *for* loop must contain the set K, and can contain no more than $r+1$ vertices. Thus there can be no more than $\binom{n-k}{r+1-k} \leq n^d$ such sets. The total time for the K-Spanning Set Counting Procedure is therefore $O(m^d n^{d+3})$. □

As a final comment, we note that each of the index restrictions given in Theorems 5.2 and 5.3 is necessary in that dropping any of them will render the associated problem NP-hard. An interesting open problem here is the generalization of Theorems 4.3 and 4.4 to computing almost min cardinality K-spanning sets when K is small or G is K-planar. We leave this for future research.

REFERENCES

[1] M. O. BALL AND J. S. PROVAN, *Calculating bounds on reachability and connectedness in stochastic networks*, Networks, 13 (1983), pp. 253–278.

[2] R. E. BIXBY, *The minimum number of edges and vertices in a graph with edge connectivity N and M N-bonds*, Networks, 5 (1975), pp. 259–298.

[3] M. K. CHARI AND J. S. PROVAN, *Calculating K-connectedness reliability using Steiner bounds*, Math. Oper. Res., 21 (1996), pp. 905–921.

[4] C. J. COLBOURN, *The Combinatorics of Network Reliability*, Oxford University Press, Oxford, 1987.

[5] S. E. DREYFUS AND R. A. WAGNER, *The Steiner problem in graphs*, Networks, 1 (1971), pp. 195–207.

[6] R. E. ERICKSON, C. L. MONMA, AND A. F. VEINOTT, *The send-and-split method for minimum-concave-cost network flows*, Math. Oper. Res., 12 (1987), pp. 634–644.

[7] D. GUSFIELD AND D. NAOR, *Extracting maximal information about sets of minimum cuts*, Algorithmica, 10 (1993), pp. 64–89.

[8] S. L. HAKIMI, *Steiner's problem in graphs and its applications*, Networks, 1 (1971), pp. 113–133.

[9] G. KIRCHOFF, *Über die auflösung der gleichungen auf welche man sei der untersuchung der linearen verteilung galvanischer strome gefuhrt wind*, Poggendorg's Ann. Phys. Chem., 72 (1847), pp. 497–508 (in German). *On the solution of equations obtained from the investigation of the linear distribution of galvanic currents*, IRE Trans. Circuit Theory CT-5, (1958), pp. 4–8 (in English).

[10] C. J. LIU AND Y. CHOW, *On operator and formal sum methods for graph enumeration problems*, SIAM J. Alg. Discrete Methods, 5 (1984), pp. 384–406.

[11] J.-C. Picard and M. Queyranne, *On the structure of all minimum cuts in a network and applications*, Math. Programming Study, 13 (1980), pp. 8–16.

[12] J. S. Provan, *Convexity and the Steiner tree problem*, Networks, 18 (1988), pp. 55–72.

[13] J. S. Provan and M. O. Ball, *The complexity of counting cuts and of computing the probability that a graph is connected*, SIAM J. Comput., 12 (1983), pp. 777–788.

[14] J. S. Provan and D. R. Shier, *A paradigm for listing (s, t)-cuts in graphs*, Algorithmica, 15 (1996), pp. 351–372.

[15] A. Ramanathan and C. J. Colbourn, *Counting almost minimum cutsets with reliability applications*, Math. Programming, 39 (1987), pp. 253–261.

[16] W. T. Tutte, *The dissection of equilateral triangles into equilateral triangles*, Proc. Cambridge Philos. Soc., 44 (1948), pp. 203–217.

[17] L. G. Valiant, *The complexity of enumeration and reliability problems*, SIAM J. Comput., 8 (1979), pp. 410–421.

SIAM J. Discrete Math.
Vol. 10, No. 3, pp. 447–459, August 1997

OPTIMAL CYCLE CODES CONSTRUCTED FROM RAMANUJAN GRAPHS*

JEAN-PIERRE TILLICH[†] AND GILLES ZÉMOR[‡]

Abstract. We aim here to show how some known Ramanujan Cayley graphs yield error-correcting codes that are asymptotically optimal in the class of cycle codes of graphs.

The main reason why known constructions of Ramanujan graphs yield good cycle codes is that the number of their cycles of a given length behaves essentially like that of random regular graphs. More precisely, we show that for actual constructions of Ramanujan graphs of degree Δ which are bipartite, and for the double cover of known Ramanujan graphs which are not bipartite, the number of cycles of length $2l$ is $\mathcal{O}_\varepsilon(\Delta - 1 + \varepsilon)^{2l}$ (for every $\varepsilon > 0$), which is about what one could expect from a random regular graph of degree Δ. Furthermore, it is possible to show that this property guarantees the highest possible error probability p that the corresponding cycle codes can sustain, among the class of cycle codes of Δ-regular graphs. This gives a constructive answer to an early problem in coding theory, namely, determining what is asymptotically the best possible performance of cycle codes of graphs when submitted to the binary symmetric channel.

Key words. Ramanujan graph, cycle code, error probability

AMS subject classifications. 05C25, 05C38, 05C80, 11Z05, 68R10, 94A24, 94B25, 94B70

PII. S0895480195292065

1. Cycle codes of graphs.

Let $\mathbf{F}_2 = \{0,1\}$ denote the field on two elements. For any set S denote by 2^S the set of subsets of S. If $\mathbf{x}, \mathbf{y} \in 2^S$, $\mathbf{x} + \mathbf{y}$ will denote the symmetric difference of $\mathbf{x}$ and $\mathbf{y}$. 2^S is in a natural correspondence with $\mathbf{F}_2^s$, the vector space of binary s-tuples where $s = \#S$, and we shall identify subsets of S with their characteristic vectors in $\mathbf{F}_2^s$.

Let Γ be a finite graph. Denote by V and E the set of vertices and the set of edges of Γ, respectively. Let $v = \#V$ and $n = \#E$ denote the cardinalities of V and E. An edge of Γ is an element of 2^V containing exactly two vertices. For any edge $e \in E$, define its *boundary* $\partial e \in 2^V$ as the union of its endpoints. ∂ is naturally extended to a mapping of 2^E to 2^V, where

$$\partial : \mathbf{x} \mapsto \sum_{e \in \mathbf{x}} \partial e.$$

A (homological) *cycle* is a set of edges with zero boundary. Its connected components correspond to closed paths, and we refer to them as *elementary cycles*. The set of cycles of Γ, denoted by $\mathcal{C}(\Gamma)$, is a linear code (i.e., a vector space) over $\mathbf{F}_2$ referred to as the *cycle code* of Γ. If the graph Γ is connected, which we shall always suppose in what follows, $\mathcal{C}(\Gamma)$ has dimension $k = \dim \mathcal{C}(\Gamma) = n - v + 1$. We shall consider from now on only Δ-regular graphs, i.e., graphs such that every vertex has exactly Δ neighbors. In this case, $k = \dim \mathcal{C}(G) = n(1 - 2/\Delta) + 1$. The size of the smallest cycle in Γ is called the *girth* of Γ by graph theorists and is the *minimum distance* of $\mathcal{C}(\Gamma)$ for coding theorists; denote it by $d(\Gamma)$, or simply d.

*Received by the editors September 18, 1995; accepted for publication (in revised form) June 19, 1996.

http://www.siam.org/journals/sidma/10-3/29206.html

[†] Computer Science Department, Université de Caen, 14000 Caen, France (tillich@info.unicaen.fr). This work was done while the author was visiting the Department of Mathematics of the University of British Columbia, Vancouver V6T 1Z2, Canada.

[‡] Network Department, Ecole Nationale Supérieure des Télécommunications, 75634 Paris 13, France (zemor@res.enst.fr).

Error probabilities. We are interested in the probability $f_\Gamma(p)$ that a random set of edges $\mathbf{x}$ contains half the edges of some cycle, when $\mathbf{x}$ is obtained by choosing every edge independently with probability p. More precisely, define

$$\begin{aligned}&[0,1] \to [0,1],\\ &\quad p \mapsto f_\Gamma(p) = \sum_{\mathbf{x}\in W} p^{|\mathbf{x}|}(1-p)^{n-|\mathbf{x}|},\end{aligned}$$

where $|\mathbf{x}|$ denotes the weight (cardinality) of $\mathbf{x}$ and where

$$W = \{\mathbf{x} \in 2^E \mid \exists \mathbf{c} \in \mathcal{C}(\Gamma), \mathbf{c} \neq \mathbf{0}, |\mathbf{x} \cap \mathbf{c}| \geq |\mathbf{c}|/2\}.$$

In other words, W is the set of vectors that are closer, for the Hamming distance, to some nonzero codeword (cycle) than to the origin.

From the coding point of view, we are submitting codewords of $\mathcal{C}(\Gamma)$ to the binary symmetric (communication) channel with error probability p. This means that each transmitted binary symbol is transformed into the complementary symbol independently with probability p. One can assume, by linearity and without loss of generality, that the submitted codeword is the $\mathbf{0}$ vector. The received vector is then some random error vector $\mathbf{x}$, which is decoded by choosing the codeword closest to it for the Hamming distance. Whenever decoding produces a codeword different from $\mathbf{0}$, or a choice between $\mathbf{0}$ and one (or more) other closest codewords, we shall say that a *decoding* (or residual) *error* occurs. The probability that a decoding error occurs is therefore exactly the probability that $\mathbf{x} \in W$, i.e., equals $f_\Gamma(p)$.

Cycle codes of graphs were among the first families of graphs to be investigated during the early days of coding theory; see, e.g., [10]. They quickly became obsolete because of their poor minimal distance properties; namely, for growing n and fixed rate k/n (equivalently for fixed degree Δ), d must be upperbounded by a logarithmic function of n. However, they remain of theoretical interest because they can provide, for fixed rate k/n, infinite families of codes for which $f_{\Gamma_n}(p)$ tends to 0 when $n \to \infty$, for any $p < p_0$, for some fixed p_0. For instance, we have the following.

Proposition 1.1. *If (Γ_n) is a family of Δ-regular graphs whose girths satisfy*

$$d(\Gamma_n) \geq c \log_{\Delta-1} n,$$

then $\lim_{n\to\infty} f_{\Gamma_n}(p) = 0$ for any $p < p_0$, where

$$p_0 = \frac{1}{2}\left(1 - \sqrt{1 - \frac{1}{(\Delta-1)^{2(1+2/c)}}}\right).$$

Proof. Let Ω_n be a subset of the edge set of Γ_n. If Ω_n contains half the edges of some cycle, then there must exist a vertex x of Γ_n and a path of length $m = \lfloor d/2 \rfloor$ rooted at x with at least half its edges in Ω_n. (To find such a vertex x, travel around the cycle). Consider now that Ω_n is obtained by choosing randomly each edge of Γ_n with independent probability $p < 1/2$. We can upperbound the probability that Ω_n contains half the edges of a cycle by the probability that such a vertex x exists, so that

$$f_{\Gamma_n}(p) \leq v\Delta(\Delta-1)^{m-1} \sum_{m/2 \leq i \leq m} \binom{m}{i} p^i (1-p)^{m-i},$$

which gives, since $p < 1/2$,

$$f_{\Gamma_n}(p) \leq Cn \left[2(\Delta - 1)\sqrt{p(1-p)}\right]^m,$$

where C is a constant. It is now straightforward to check that $f_{\Gamma_n}(p) \leq Cn^{-\alpha}$ for some positive α whenever $p < p_0$. □

Infinite families of graphs (Γ_n) satisfying $d \geq c \log_{\Delta-1} n$ were first constructed in [16].

For a family $\mathcal{G} = (\Gamma_n)$ of Δ-regular graphs, denote by

$$\theta(\mathcal{G}) = \sup\{p \mid \lim_{n\to\infty} f_{\Gamma_n}(p) = 0\}.$$

Few constructive classes of codes that achieve vanishing residual error probability for positive p are known. Besides constructions that use concatenation [7, 13], one can quote essentially low-density parity check codes, a generalization of cycle codes of graphs [8], taken up again in [20], and product-type codes originating in [6]. For both these classes of codes it is a difficult problem to determine, for given rate k/n, the largest p for which decoding error probability vanishing with n can be achieved. Hence the motivation for solving one of the remaining open problems for cycle codes of graphs, namely,

(i) determining the largest possible $\theta(\mathcal{G})$ for families of Δ-regular graphs $\mathcal{G} = (\Gamma_n)$,

(ii) finding actual constructions of families $\mathcal{G} = (\Gamma_n)$ achieving this value of θ.

In [3] it is proved that for any family of Δ-regular graphs, one must have

$$\theta \leq \frac{1}{2}\left(1 - \sqrt{1 - \frac{1}{(\Delta-1)^2}}\right).$$

In this paper we show that some families of known Ramanujan Cayley graphs achieve the above value of θ and in this sense are optimal among the class of cycle codes of graphs.

This will be ensured by estimating the number A_i of cycles of length i of the graphs under consideration and using the following.

Proposition 1.2. *If $\mathcal{G} = (\Gamma_n)$ is a family of Δ-regular graphs such that*

1. $\lim_{n\to\infty} d(\Gamma_n) = \infty$;
2. *for any $\varepsilon > 0$ there exists c_ε such that the number A_i of elementary cycles of length i of any member of $\mathcal{G}$ satisfies*

$$A_i \leq c_\varepsilon(\Delta - 1 + \varepsilon)^i,$$

then

$$\theta(\mathcal{G}) = \frac{1}{2}\left(1 - \sqrt{1 - \frac{1}{(\Delta-1)^2}}\right).$$

Proof. Consider that Ω_n is a subset of the edge set of Γ_n obtained by randomly choosing each edge with independent probability $p \leq 1/2$. Let X_n be the number of subsets of edges of Ω_n that consist of at least half the edges of a cycle. The expected value of X_n is

$$\mathbf{E}_p(X_n) = \sum_{i \geq d(\Gamma_n)} A_i \sum_{j=i/2}^{i} \binom{i}{j} p^j (1-p)^{i-j},$$

where A_i is the number of elementary cycles of length i. Hence,

$$\mathbf{E}_p(X_n) \leq \sum_{i \geq d(\Gamma_n)} A_i 2^i [p(1-p)]^{i/2}$$

for any $p \leq 1/2$. Therefore,

$$\mathbf{E}_p(X_n) \leq c_\varepsilon \sum_{i \geq d(\Gamma_n)} \left((\Delta - 1 + \varepsilon) 2 \sqrt{p(1-p)} \right)^i .$$

It is routinely checked that whenever

$$p < \frac{1}{2}\left(1 - \sqrt{1 - \frac{1}{(\Delta - 1 + \varepsilon)^2}} \right),$$

then $(\Delta - 1 + \varepsilon)2\sqrt{p(1-p)} < 1$, so that $\lim_{n\to\infty} \mathbf{E}_p(X_n) = 0$ whenever $d(\Gamma_n) \to \infty$. And necessarily, if $\lim_{n\to\infty} \mathbf{E}_p(X_n) = 0$, then $\lim_{n\to\infty} f_{\Gamma_n}(p) = 0$. □

Remark. It can be checked easily enough that the expected number of homological cycles of length $2i$ of a randomly chosen Δ-regular bipartite graph is $(\Delta - 1)^{2i}$. Note that this means that random Δ-regular graphs have cycles of constant length. This must be avoided to obtain the conclusion of Proposition 1.2. Hence condition 1 in the proposition, which is satisfied by the Ramanujan graphs we consider.

2. Ramanujan graphs. There are several ways to define the actual explicit constructions of Ramanujan graphs (given in [1, 14, 15, 17, 18]). All these constructions can be described as $q+1$-regular Cayley graphs over $PGL_2(\mathbf{F}_{q'})$ or $PSL_2(\mathbf{F}_{q'})$, where q and q' are two prime powers, and $\mathbf{F}_{q'}$ is the finite field with q' elements.

For our purposes it will be more convenient to use the quaternion description of these graphs. As a matter of fact, by using the latter description we can relate the problem of counting the number of cycles of a given length to the problem of estimating the number of solutions of some diophantine equation.

Basically, the construction of those Ramanujan graphs is done in two steps.

1. The first step consists of constructing the $q+1$-regular infinite tree in an arithmetic way by using quaternions.
2. One obtains finite Ramanujan graphs from this tree by taking suitable finite quotients of this tree which do not create small cycles.

Let us see these constructions in more detail.

2.1. The construction of the infinite tree of degree $q+1$. The construction of the infinite tree starts by considering the following set of quaternions: $\mathcal{S} = \mathcal{A}\mathbf{1} + \mathcal{A}\mathbf{i} + \mathcal{A}\mathbf{j} + \mathcal{A}\mathbf{ij}$, where $\mathcal{A}$ is a Euclidean domain which will be either $\mathbb{Z}$ or $\mathbf{F}_q[X]$. We will denote by $\overline{x}$ the conjugate of the element $x \in \mathcal{S}$ and by $N(x) = x\overline{x} \in \mathcal{A}$ the norm of x. Then a prime π is chosen in $\mathcal{A}$; this is a prime number equal to q when $\mathcal{A} = \mathbb{Z}$, X when $\mathcal{A} = \mathbf{F}_q[X]$ for odd q, and $X+1$ for even q.

The basic step consists of setting up a set of $q+1$ quaternions $\alpha_1, \alpha_2, \ldots, \alpha_{q+1}$ of norm π such that

1. every quaternion α of norm π^n has a unique factorization

$$\alpha = u\pi^r \alpha_{i_1}\alpha_{i_2}\cdots\alpha_{i_m},$$

where u is a unit (an element of norm 1 here) and $2r + m = n$, and where the product of two consecutive terms of the product α_{i_j} and $\alpha_{i_{j+1}}$ never belongs to $\mathcal{A}$.

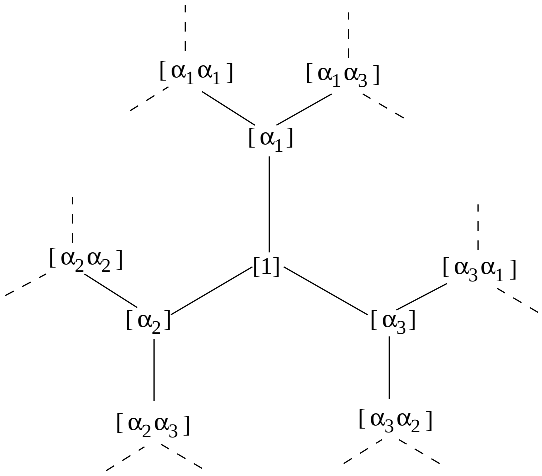


Fig. 2.1. *The infinite tree.*

2. for every α_i, $\overline{\alpha_i}$ is equal to some $\pm\alpha_j$.

We refer to [15, 17, 18, 19] to see how this set of quaternions is obtained. This set now enables us to construct the infinite $q+1$ regular tree as a Cayley graph. The group G from which this graph is constructed is just the set of quaternions generated by the α_is and we identify the quaternions which differ by a multiplication of some $\pm\pi^i$. Let us denote by $[\alpha]$ the equivalence class associated to α. This group is clearly generated by the $[\alpha_i]$s and the inverse of $[\alpha_i]$ is $[\alpha_j]$, where α_j is the quaternion such that $\alpha_j = \pm\overline{\alpha_i}$ (since $[\alpha_i][\pm\overline{\alpha_i}] = [\pm\alpha_i\overline{\alpha_i}] = [\pm\pi] = [1]$). That the infinite Cayley graph over G with generator set $[\alpha_1], [\alpha_2], \dots, [\alpha_{q+1}]$ is indeed the $q+1$-regular infinite tree is just a consequence of the fact that every quaternion of norm π^n has a unique factorization over the α_is.

We have depicted such an example in Fig. 2.1, when there are three generators $[\alpha_1], [\alpha_2], [\alpha_3]$ and we have assumed that $\overline{\alpha_1} = \alpha_2$ and $\overline{\alpha_3} = \alpha_3$; in other words $[\alpha_1]^{-1} = [\alpha_2]$ and $[\alpha_3]^{-1} = [\alpha_3]$.

2.2. The finite Cayley graph. We obtain our Ramanujan graph by taking a finite quotient of this infinite tree and this quotient will be realized as a Cayley graph by choosing a suitable normal subgroup H of G of finite index. One selects first a prime ρ of $\mathcal{A}$ which satisfies certain conditions (for more details see the following section). H is defined as the set of classes $[\alpha] = [a_0 + a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{ij}]$ for which a_1, a_2, a_3 are multiples of ρ. This set is clearly a normal subgroup for it can be seen as the kernel of the homomorphism ϕ

$$\begin{aligned} \phi : G &\to \mathbb{H}(\mathcal{A}/\rho\mathcal{A})^*/Z, \\ [\alpha] &\mapsto (\alpha \bmod \rho)Z, \end{aligned}$$

where $\mathbb{H}(\mathcal{A}/\rho\mathcal{A})$ denotes the ring of quaternions with entries in the field $\mathcal{A}/\rho\mathcal{A}$, $\mathbb{H}(\mathcal{A}/\rho\mathcal{A})^*$, the invertible elements of this ring, and Z its central subgroup, which is $\{a \in \mathcal{A}/\rho\mathcal{A} \; |a \neq 0\}$.

One of the attractive features of this way of constructing a Cayley graph is that the study of the number of cycles of a given length can now be expressed as a problem in number theory.

2.3. Counting cycles of a given length. In order to bound the number of cycles of a given length in the finite Cayley graph which has been constructed, we can observe the following facts.

FACT 1. *The number of elementary cycles of length l in a graph is less than the number of nonbacktracking closed walks of length l. In an undirected Cayley graph this is less than the number of vertices v of the graph times the number of nonbacktracking walks of length l which start at the identity of the group and which go back to this vertex.*

FACT 2. *A nonbacktracking walk of length l corresponds in the case described above to a sequence $\alpha_{i_1}\alpha_{i_2}\cdots\alpha_{i_l}$ such that the product of two consecutive terms never belongs to $\mathcal{A}$, and this nonbacktracking walk returns to its starting point (is closed) if and only if the product $[\alpha_{i_1}][\alpha_{i_2}]\cdots[\alpha_{i_l}]$ is an element of the normal subgroup H, or what amounts to the same thing, iff the product $\alpha_{i_1}\alpha_{i_2}\cdots\alpha_{i_l}$ is of the form $a_0+a_1\mathbf{i}+a_2\mathbf{j}+a_3\mathbf{ij}$, where a_1,a_2,a_3 are multiples of the prime ρ which defines H.*

Let us notice now that the norm is multiplicative and that this implies that the norm of a product $\alpha_{i_1}\alpha_{i_2}\cdots\alpha_{i_l}$ is π^l. Hence the following fact holds.

FACT 3. *The number of nonbacktracking closed walks of length l is less than*

$$v\,\#\left\{(a_0,a_1,a_2,a_3)\in\mathcal{A}^4|N(a_0+\rho a_1\mathbf{i}+\rho a_2\mathbf{j}+\rho a_3\mathbf{ij})=\pi^l\right\}.$$

The norm of a quaternion $a_0+a_1\mathbf{i}+a_2\mathbf{j}+a_3\mathbf{ij}$ is a quadratic form in (a_0,a_1,a_2,a_3), and all that we need now is a tool bounding the number of solutions in $\mathcal{A}$ of a certain quadratic equation. There are several methods which can be employed to estimate the number of solutions of the quadratic equation which arises in our case. The most precise one, which uses the work of Drinfeld, Eichler, and Igusa (see [4, 5, 12], respectively) does not give enough information on the number of "small" cycles. We use instead very simple (and classical) arguments (see [9], for example) to bound the number of solutions of such equations, and this is obtained by the following lemma.

LEMMA 2.1. *Let $\mathcal{A}$ be the ring $\mathbb{Z}$ or $\mathbf{F}_q[X]$, $R=\mathcal{A}+\mathcal{A}\mathbf{i}$, where $\mathbf{i}$ is an algebraic integer of degree 2 over $\mathcal{A}$ (i.e., $\mathbf{i}$ does not belong to $\mathcal{A}$ and satisfies an equation $\mathbf{i}^2+a\mathbf{i}+b=0$, with $a,b\in\mathcal{A}$). Let $\bar{\mathbf{i}}$ be the other solution of this equation and define the following automorphism of R, by $\overline{x+y\mathbf{i}}=x+y\bar{\mathbf{i}}$, and the multiplicative morphism N "the norm" from R to $\mathcal{A}$ by $N(x)=x\overline{x}$. If R is a unique factorization domain, then the number of solutions of the equation $N(x)=c$ (the unknown is x and c is a given element of $\mathcal{A}$) is $\mathcal{O}_\delta(c^\delta)$ if $\mathcal{A}=\mathbb{Z}$, and $\mathcal{O}_\delta(q^{\delta\deg c})$ if $\mathcal{A}=\mathbf{F}_q[X]$, and this for every $\delta>0$.*

See the appendix for a proof.

2.4. The bipartite cover. Actually, for the graphs we consider, we are able to give rather tight upper bounds on the cardinality of the set in Fact 3, i.e., on the number of nonbacktracking closed walks, only when their length l is even. This approach works well when the Cayley graph is bipartite because there are no odd cycles. When this graph is not bipartite, we shall move around this difficulty by considering its bipartite *double cover.*

DEFINITION 2.2. *Let $\Gamma(V,E)$ be a graph with set of vertices V and set of edges E. Its* double cover *$\widehat{\Gamma}$ is defined by the set of vertices $\widehat{V}=V\times\{0,1\}$ and the set of edges $\widehat{E}=\{\{(x,0),(y,1)\}$ for $\{x,y\}\in E\}$.*

An attractive feature of this double cover is as follows.

FACT 4. *The double cover of a graph Γ is*

(i) *connected iff Γ is connected and nonbipartite,*

TABLE 3.1

Constructions of Ramanujan graphs. There are some additional constraints on ρ which are not given here. We refer to [15], [17], [18] *for the missing details.*

	graphs constructed in [15], [17]	graphs constructed in [18]	graphs constructed in [18]
$\mathcal{A}$	$\mathbb{Z}$	$\mathbf{F}_q[X],\ q = p^n$, p odd prime	$\mathbf{F}_{2^n}[X]$
$\mathbb{H}(\mathcal{A})$ $=$ $\mathcal{A} + \mathcal{A}\mathbf{i} + \mathcal{A}\mathbf{j} + \mathcal{A}\mathbf{ij}$	$\mathbf{i}^2 = \mathbf{j}^2 = (\mathbf{ij})^2 = -1$ $\mathbf{ij} = -\mathbf{ji}$	$\mathbf{i}^2 = \eta$ η is not a square in $\mathbf{F}_q$ $\mathbf{j}^2 = X - 1, \mathbf{ij} = -\mathbf{ji}$	$\mathbf{i}^2 = \mathbf{i} + \eta$, η is such that $X^2 + X + \eta$ is irreducible over $\mathbf{F}_{2^n}$, $\mathbf{j}^2 = X, \mathbf{ij} = \mathbf{ji} + \mathbf{j}$
$\overline{\alpha}$, $\alpha = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{ij}$	$a - b\mathbf{i} - c\mathbf{j} - d\mathbf{ij}$	$a - b\mathbf{i} - c\mathbf{j} - d\mathbf{ij}$	$(a + b) + b\mathbf{i} + c\mathbf{j} + d\mathbf{ij}$
$N(\alpha)$	$a^2 + b^2 + c^2 + d^2$	$a^2 - \eta b^2$ $-(X - 1)(c^2 - \eta d^2)$	$a^2 + \eta b^2 + ab$ $+X(c^2 + \eta d^2 + cd)$
π	odd prime number q	X	$X + 1$
ρ	odd prime number p	irreducible polynomial $g(X) \in \mathbf{F}_q[X]$	irreducible polynomial $g(X) \in \mathbf{F}_{2^n}[X]$
degree of the graph	$q + 1$	$q + 1$	$2^n + 1$
Number of vertices of the Ramanujan graph	$p(p^2 - 1)$ if $(\frac{q}{p}) = -1$ $\frac{p(p^2-1)}{2}$ if $(\frac{q}{p}) = 1$	$q^{3d} - q^d$ if $(\frac{X}{g(X)}) = -1$ $\frac{q^{3d}-q^d}{2}$ if $(\frac{X}{g(X)}) = 1$ $d = \deg g(X)$	$2^{3nd} - 2^{nd}$ $d = \deg g(X)$
bipartite	yes if $(\frac{q}{p}) = -1$ no if $(\frac{q}{p}) = 1$	yes if $(\frac{X}{g(X)}) = -1$ no if $(\frac{X}{g(X)}) = 1$	never

(ii) *bipartite and has therefore only cycles of even length. Furthermore, the projection*

$$\begin{aligned} \widehat{V} &\longrightarrow V, \\ (x, i) &\mapsto x \end{aligned}$$

for $i = 0, 1$ induces a two-to-one correspondence between the nonbacktracking closed walks of $\widehat{\Gamma}$ and the nonbacktracking closed walks of even length *of* Γ.

By taking double covers if need be, we shall look therefore for graphs that satisfy the conditions of Proposition 1.2 among bipartite graphs.

3. Estimation of the number of cycles in some Ramanujan graphs. We are going to show in this section that some of the Ramanujan graphs constructed in [15, 17, 18] meet the hypotheses of Proposition 1.2, which implies that the associated families of cycle codes are optimal. We do not give all the steps involved in the construction of these graphs, and merely refer to [15, 17, 18, 19] for further details. A rough description in the spirit of the general presentation of section 2 will suffice for our needs. The parameters of these graphs which are relevant to counting cycles are gathered in Table 3.1.

3.1. The Ramanujan graphs constructed by Margulis and independently by Lubotzky, Philipps, and Sarnak. They correspond to the choice $\mathcal{A} = \mathbb{Z}$. We denote these graphs by $\mathcal{X}^{p,q}$, where q denotes the odd prime number chosen for π and p the odd prime number chosen for ρ. Now, upperbounding the number of vertices v by p^3, Fact 3 translates to the following fact.

FACT 3′. *The number of nonbacktracking closed walks of length l in $\mathcal{X}^{p,q}$ is less than*

$$p^3 \,\#\{(a_0, a_1, a_2, a_3) \in \mathbb{Z}^4 | a_0^2 + p^2 a_1^2 + p^2 a_2^2 + p^2 a_3^2 = q^l\}.$$

By using Lemma 2.1 it is straightforward to obtain a rather tight upper bound on the number of solutions of this diophantine equation when the length of the cycles is *even.*

LEMMA 3.1. *The number of nonbacktracking closed walks of length $2l$ in $\mathcal{X}^{p,q}$ is $\mathcal{O}_\varepsilon(q+\varepsilon)^{2l}$ for every $\varepsilon > 0$.*

Proof. Assume that $a_0^2 + p^2 a_1^2 + p^2 a_2^2 + p^2 a_3^2 = q^{2l}$; then $a_0^2 \equiv q^{2l} \bmod p^2$ and thus $a_0 \equiv \pm q^l \bmod p^2$. Therefore there are at most $\lceil 4q^l/p^2 \rceil$ choices for a_0. Since $p^2 a_1^2 < q^{2l}$, there are at most $\lceil 2q^l/p \rceil$ choices for a_1.

For fixed a_0, a_1, the number of choices we have for the couple (a_2, a_3) is not very large because $a_2^2 + a_3^2$ should be equal to $(q^{2l} - a_0^2 - p^2 a_1^2)/p^2$, which is a number smaller than q^{2l}/p^2, and from Lemma 2.1 the number of couples (a_2, a_3) which satisfy this inequality is $\mathcal{O}_\varepsilon(q^{2l}/p^2)^\varepsilon$.

Therefore, the total number of solutions is less than

$$\lceil 4q^l/p^2 \rceil \lceil 2q^l/p \rceil \mathcal{O}_\varepsilon(q^{2l}/p^2)^\varepsilon = \frac{1}{p^3}\mathcal{O}_\varepsilon(q^{2l(1+\varepsilon)}).$$

We conclude by applying Fact 3′. □

Those graphs $\mathcal{X}^{p,q}$ are bipartite if and only if q is not a quadratic residue modulo p and have in this case only cycles of even length whose numbers can be bounded with the previous lemma. Moreover, in this case the graphs $\mathcal{X}^{p,q}$ have a very large girth which is $\frac{4}{3}\log_q(p(p^2-1)) + \mathcal{O}(1)$. When q is a quadratic residue modulo p the graph is not bipartite, but its double cover $\widehat{\mathcal{X}}^{p,q}$ has still a large girth, namely, $\frac{4}{3}\log_q(p(p^2-1)) + \mathcal{O}(1)$.

Remarks.

1. The key fact in Lemma 3.1 has been observed in another setting by Davidoff and Sarnak (see [2]).

2. We wish to emphasize here that the results on the girth of $\mathcal{X}^{p,q}$ in the nonbipartite case, which can be found in the literature, give only the lower bound $\frac{2}{3}\log_q(p(p^2-1))$, so the result we invoke here shows that in some sense we can substantially "improve" these graphs by taking their double cover. We justify this by the fact that the double cover has only cycles of even length and that these project on $\mathcal{X}^{p,q}$ to either cycles of the same even length or to cycles of odd length half as long. The point is that the proof used in [15], for example, to show that the girth in the bipartite case is bigger than $\frac{4}{3}\log_q(p(p^2-1))$ depends only on the fact that a cycle of even length cannot be shorter than this quantity and therefore also gives a lower bound on the length of the shortest cycle of even length when the graph is not bipartite. That the girth is indeed $\frac{4}{3}\log_q(p(p^2-1)) + \mathcal{O}(1)$ follows from a straightforward generalization of results given in [17].

This leads to the following result by applying Fact 1 and using the discussion given in section 2.4 together with Proposition 1.2.

THEOREM 3.2. *Let q be a fixed odd prime. Let $\mathcal{X}_q = (\mathcal{X}^{p,q})$ be the family of those $\mathcal{X}^{p,q}$ for which $\left(\frac{q}{p}\right) = -1$. Let $\widehat{\mathcal{X}}_q = (\widehat{\mathcal{X}}^{p,q})$ be the family of those $\widehat{\mathcal{X}}^{p,q}$ for which*

$\left(\frac{q}{p}\right) = 1$. *Then*

$$\theta(\mathcal{X}_q) = \theta(\widehat{\mathcal{X}}_q) = \frac{1}{2}\left(1 - \sqrt{1 - \frac{1}{q^2}}\right).$$

3.2. The case $\boldsymbol{\mathcal{A} = \mathbf{F}_q[X]}$, $\boldsymbol{q}$ odd prime power. The corresponding Ramanujan graphs have been constructed by Morgenstern (see [18]) and are regular of degree $q+1$. From now on, we consider such a graph obtained by choosing $\rho = g(X)$ an irreducible polynomial of degree k.

By using Fact 3 given in section 2.3 and by using the fact that the groups over which these finite Cayley graphs are defined have less than q^{3k} elements, we obtain that the number $\mathcal{N}_{2l}$ of nonbacktracking closed walks of length $2l$ satisfies

$$\begin{aligned}\mathcal{N}_{2l} &\le q^{3k}\#\left\{(a,b,c,d) \in (\mathbf{F}_q[X])^4 | N(a + gb\mathbf{i} + gc\mathbf{j} + gd\mathbf{ij}) = X^{2l}\right\}\\ &\le q^{3k}\#\left\{(a,b,c,d) \in (\mathbf{F}_q[X])^4 | a^2 - \eta b^2 g^2 + (X-1)g^2(\eta d^2 - c^2) = X^{2l}\right\}.\end{aligned}$$

To obtain an estimation of the number of solutions of this equation, we use an upper bound on the number of solutions in $\mathbf{F}_q[X]$ of the equation $a^2 - \eta b^2 = P$, where the unknowns are a, b, and P is a given polynomial of degree l. For that purpose we use the classical method which consists of studying the ring $R = \mathbf{F}_q[X] + \mathbf{F}_q[X]\mathbf{i}$. The crucial property of this ring follows in Lemma 3.3.

LEMMA 3.3. $R = \mathbf{F}_q[X] + \mathbf{i}\mathbf{F}_q[X]$ *is a Euclidean domain.*

Proof. Let $\phi(a + b\mathbf{i}) = \deg\left((a + b\mathbf{i})\overline{(a + b\mathbf{i})}\right) = \deg(a^2 - \eta b^2)$. Since $a^2 - b^2\eta = 0$ implies $a = b = 0$, for $a, b \in \mathbf{F}_q[X]$, we deduce that $\phi(\alpha)$ is nonnegative for all $\alpha \neq 0$, and this combined with the relation $\phi(\alpha\beta) = \phi(\alpha) + \phi(\beta)$ shows that R is a domain. To show that R is Euclidean it remains to prove that for all α and β in R such that $\phi(\alpha) \ge \phi(\beta)$, there exists a γ in R such that $\phi(\alpha - \gamma\beta) < \phi(\beta)$ or $\alpha = \beta\gamma$.

Let $a + b\mathbf{i} = \alpha\overline{\beta}$ and $t = \beta\overline{\beta}$. Carry out the usual Euclidean division over $\mathbf{F}_q[X]$ of a and b by t : $a = q_1 t + r_1$, $b = q_2 t + r_2$, with $\deg(r_1), \deg(r_2) < \deg(t)$. We claim that we can choose $\gamma = q_1 + q_2\mathbf{i}$. This follows from

$$\begin{aligned}\phi(\alpha - \beta\gamma) + \phi(\overline{\beta}) &= \phi(\alpha\overline{\beta} - \beta\overline{\beta}\gamma)\\ &= \phi(a + b\mathbf{i} - t(q_1 + q_2\mathbf{i}))\\ &= \phi(r_1 + r_2\mathbf{i})\\ &< 2\deg(t) = \phi(\beta) + \phi(\overline{\beta}).\end{aligned}$$

Hence $\phi(\alpha - \beta\gamma) < \phi(\beta)$. This calculation is valid as long as either r_1 or r_2 is different from 0. We handle the case $r_1 = r_2 = 0$ by noticing that in such a case $\alpha\overline{\beta} = a + b\mathbf{i} = t\gamma = (\beta\overline{\beta})\gamma = (\beta\gamma)\overline{\beta}$. Therefore $\alpha = \beta\gamma$. □

The ring R is therefore a unique factorization domain. The units of R are exactly the invertible elements of R, which is the set $I = \mathbf{F}_q + \mathbf{F}_q\mathbf{i} - \{0\}$. By using Lemma 2.1 we obtain that, for a given polynomial P of $\mathbf{F}_q[X]$,

$$\#\{(x,y) \in \mathbf{F}_q[X] \times \mathbf{F}_q[X] \mid x^2 - \eta y^2 = P\} = \mathcal{O}_\varepsilon(q^{\varepsilon \deg P}). \tag{3.1}$$

From this we can give an upper bound on the number of solutions (a,b,c,d) of

$$a^2 - \eta b^2 g^2 + (X-1)g^2(\eta d^2 - c^2) = X^{2l}$$

by noticing that

1. $\deg\left(a^2 - \eta b^2 g^2 + (X-1)g^2(\eta d^2 - c^2)\right) = \max(2\deg a, 2(k+\deg b), 2(k+\deg c)+1, 2(k+\deg d)+1)$, and so $l - k \geq \deg(b)$, there are no more than q^{l-k+1} choices for b. The equality on the degree makes use of

$$x^2 - \eta y^2 = 0 \text{ iff } x = y = 0$$

for $x, y \in \mathbf{F}_q$ and therefore $\deg(a^2 - b^2\eta) = 2\max(\deg a, \deg b)$ for $a, b \in \mathbf{F}_q[X]$.

2. $a^2 \equiv x^{2l} \pmod{g^2}$ and therefore $a \equiv \pm X^l \pmod{g^2}$; thus a cannot take on more than $2q^{l-2k+1}$ different values (a is of degree l at most).

3. Once a and b are chosen, $\eta d^2 - c^2$ has to be equal to some polynomial of degree at most $2l - 2 - 2k$ and from (3.1) we deduce that the number of choices left for (c, d) is $\mathcal{O}_\varepsilon(q^{\varepsilon(2l-2-2k)})$.

This yields that the number of nonbacktracking closed walks of length $2l$ of our Ramanujan graph is $\mathcal{O}_\varepsilon(q+\varepsilon)^{2l}$ (for every $\varepsilon > 0$). We now have to treat two cases separately as follows:

(i) Either our graph is bipartite (this is if X is not a quadratic residue modulo $g(X)$). The girth of our graph is in this case larger than

$$\frac{4}{3}\log_q\left(\frac{q^{3\deg(g)} - q^{\deg(g)}}{2}\right) + 1$$

(see Theorem 4.13 of [18]).

(ii) or our graph is not bipartite (if X is a quadratic residue modulo $g(X)$). Then one can prove easily (by using the argument given in the proof of the lower bound on the girth of these graphs, in Theorem 4.13 in [18]) that the bipartite cover of our graphs has a girth which is greater than $4/3\log_q\left((q^{3k} - q^k)/2\right) + 1$ too.

We can now conclude by using Proposition 1.2.

THEOREM 3.4. *Let $\mathcal{X}^{g,q}$ be the Ramanujan graph of degree $q+1$ considered in this section obtained from the choice $\rho = g(X)$. Let $\mathcal{X}_q$ be the family of graphs $\mathcal{X}^{g,q}$ which are bipartite and $\mathcal{Y}_q$ be the family of double covers $\widehat{\mathcal{X}}^{g,q}$ of all graphs $\mathcal{X}^{g,q}$ which are not bipartite.*

$$\theta(\mathcal{X}_q) = \theta(\mathcal{Y}_q) = \frac{1}{2}\left(1 - \sqrt{1 - \frac{1}{q^2}}\right).$$

3.3. The case of $\boldsymbol{\mathcal{A} = \mathbf{F}_{2^n}[X]}$. The corresponding Ramanujan graphs have been constructed by Morgenstern (see [18]) and are regular of degree $2^n + 1$. From now on we consider such a graph obtained by choosing $\rho = g(X)$ an irreducible polynomial of degree k. We let $q = 2^n$ and we denote this graph by $\mathcal{X}^{g,q}$.

By using Fact 3 given in section 2 we see that the number $\mathcal{N}_{2l}$ of nonbacktracking closed walks of length $2l$ of these graphs $\mathcal{X}^{g,q}$ verifies

$$\begin{aligned}\mathcal{N}_{2l} &\leq q^{3k} \;\; \#\left\{(a,b,c,d) \in (\mathbf{F}_q[X])^4 | N(a + gb\mathbf{i} + gc\mathbf{j} + gd\mathbf{ij}) = (X+1)^{2l}\right\} \\ &\leq q^{3k} \;\; \#\left\{(a,b,c,d) \in (\mathbf{F}_q[X])^4 | a^2 + gab + \eta b^2 g^2 + Xg^2(c^2 + cd + \eta d^2) = (X+1)^{2l}\right\}.\end{aligned}$$

We proceed as for the graphs of odd degrees. We obtain first an estimation of the number of solutions in $\mathbf{F}_q[X]$ of the equation $a^2 + b^2\eta + ab = P$, where the unknowns are a, b, and P is a given polynomial. It can be shown in a similar way as Lemma 3.3 that $R = \mathbf{F}_q[X] + \mathbf{i}\mathbf{F}_q[X]$ is a Euclidean domain (the only difference being that this time we use the remark $a^2 + \eta b^2 + ab = 0$ iff $a = b = 0$ for $a, b \in \mathbf{F}_{2^n}[X]$). Hence by

Lemma 2.1 the number of solutions of the aforementioned equation is $\mathcal{O}_\varepsilon(q^{\varepsilon \deg(P)})$ for all $\varepsilon > 0$. This yields the upper bound which holds for every $\varepsilon > 0$,

$$\mathcal{N}_{2l} < \mathcal{O}_\varepsilon\left((q+\varepsilon)^{2l}\right).$$

It can be shown that the girth of the bipartite cover $\widehat{\mathcal{X}}^{g,q}$ is not less than $\frac{4}{3}\log_q(q^{3\deg(g)} - q^{\deg(g)})$ (by using the same proof technique as in Theorem 4.13 of [18] and by using the fact that there are only cycles of even length). We conclude as before.

THEOREM 3.5. *If $\widehat{\mathcal{Y}}_q$ is the family $\widehat{\mathcal{Y}}_q = (\widehat{\mathcal{X}}^{g,q})$, then*

$$\theta(\widehat{\mathcal{Y}}_q) = \frac{1}{2}\left(1 - \sqrt{1 - \frac{1}{q^2}}\right).$$

Appendix. Proof of the main lemma. In this section we prove Lemma 2.1.

Recall here a few facts about unique factorization domains.

—There exists a subset E of the unique factorization domain R, called the units, which is the set of elements of R which divide every other element of the domain. In our case this coincides with the set of elements of R of norm a unit of $\mathcal{A}$. These units define an equivalence relation over the domain. Two elements x and y are said to be *associated* if and only if there exists a unit u such that $x = yu$.

—There exists a subset Π of elements of the domain called the *primes*, i.e., the subset of elements not in E which are not a product of two nonunits elements. The set of associated elements to a prime is a set of prime elements and let us choose for each such class a representative element in an arbitrary way. In this case every element of the unique factorization domain can be written uniquely (up to reordering the factors) as

$$up_1^{\alpha_1}p_2^{\alpha_2}\cdots p_n^{\alpha_n},$$

where the p_is are representative elements of primes and u is a unit.

Furthermore, we will distinguish between sets of associated primes which contain conjugate pairs of primes and sets of associated primes which do not contain such conjugate pairs of primes. In what follows:

(i) u will always denote a unit,

(ii) q_i will always denote a representative of a set of associated primes which contains a conjugate pair of primes, and

(iii) p_i a representative of a set of associated primes which does not contain conjugate pairs of prime. We will choose the p_is such that every $\overline{p_i}$ is a representative prime of a set of associated primes too.

Let us factorize

$$c = up_1^{\alpha_1}p_2^{\alpha_2}\cdots p_m^{\alpha_m}\overline{p_1}^{\beta_1}\overline{p_2}^{\beta_2}\cdots\overline{p_m}^{\beta_m}q_1^{\gamma_1}q_2^{\gamma_2}\cdots q_n^{\gamma_n}$$

(where some of the powers can be 0). Since c is in $\mathcal{A}$, $\overline{c}$ is in $\mathcal{A}$, and by the unicity of factorization into primes we get that $\alpha_i = \beta_i$, for every i. If there exists $x \in R$ such that $x\overline{x} = c$, then we can factorize x and $\overline{x}$ by using the same primes p_is, the $\overline{p_i}$s and the q_is.

$$x = u'p_1^{\alpha'_1}p_2^{\alpha'_2}\cdots p_m^{\alpha'_m}\overline{p_1}^{\beta'_1}\overline{p_2}^{\beta'_2}\cdots\overline{p_m}^{\beta'_m}q_1^{\gamma'_1}q_2^{\gamma'_2}\cdots q_n^{\gamma'_n},$$

and therefore

$$\overline{x} = u''\overline{p_1}^{\alpha'_1}\overline{p_2}^{\alpha'_2}\cdots\overline{p_m}^{\alpha'_m} p_1^{\beta'_1} p_2^{\beta'_2}\cdots p_m^{\beta'_m} q_1^{\gamma'_1} q_2^{\gamma'_2}\cdots q_n^{\gamma'_n}.$$

Due to the unicity of factorization into primes, we obtain that for every i

$$\alpha'_i + \beta'_i = \alpha_i \quad \text{and} \quad \gamma_i = 2\gamma'_i. \tag{A.1}$$

Hence the number of solutions of the equation $x\overline{x} = c$ is equal to the number of ways of choosing an x whose factorization satisfies the conditions (A.1); the only choice is, in fact, the choice of u and the choice of the α'_i in $\{0, 1, \ldots, \alpha_i\}$. In order to get an upper bound on this number, let $c' = p_1^{\alpha_1} p_2^{\alpha_2}\cdots p_m^{\alpha_m}\overline{p_1}^{\beta_1}\overline{p_2}^{\beta_2}\cdots\overline{p_m}^{\beta_m} = (p_1\overline{p_1})^{\alpha_1}(p_2\overline{p_2})^{\alpha_2}\cdots(p_m\overline{p_m})^{\alpha_m}$ and let us notice that the number of choices for the α'_is is exactly the number of ways of choosing $y = (p_1\overline{p_1})^{\alpha'_1}(p_2\overline{p_2})^{\alpha'_2}\cdots(p_m\overline{p_m})^{\alpha_m}$ which divide c'. Since c' and y are in $\mathcal{A}$ this coincides with the number of divisors (in $\mathcal{A}$) of the element c'—where we do not distinguish between divisors which differ by a multiplication of an invertible element of $\mathcal{A}$. This number of divisors is $d(c')$ in the case $\mathcal{A} = \mathbb{Z}$, that is, the number of positive integers dividing c', and is equal to the number of polynomials whose leading coefficient is 1 which divide c' when $\mathcal{A} = \mathbf{F}_q[X]$. From Theorem 315 in chapter 18 of [11] we get an upper bound on $d(c')$ of the form $O_\delta(c'^\delta)$, for all $\delta > 0$, and we deduce from that the number of solutions s of the equation $x\overline{x}$ verifies (for every $\delta > 0$)

$$\begin{aligned} s &= \#E\ d(c') \\ &= 4d(c') \\ &= \mathcal{O}_\delta(c^\delta). \end{aligned}$$

We have similar results when $\mathcal{A} = \mathbf{F}_q[X]$. In this case, $E = \{u + \mathbf{i}v | u, v \in \mathbf{F}_q, (u, v) \neq (0, 0)\}$, therefore $\#E = q^2 - 1$ and the number of divisors of c' is $O_\delta(q^{\delta \deg c'})$. This is obtained by the following straightforward generalization of Theorem 316 in [11] to polynomials.

If a multiplicative function $f : \mathbf{F}_q[X] \mapsto \mathbb{R}$ satisfies $f(p^m) \to 0$ for every irreducible polynomial p when $m \deg(p) \to \infty$, then $f(a) \to 0$ when $\deg(a) \to \infty$.

We let $f(x) = q^{-\delta \deg(x)} d(x)$ which is clearly multiplicative and satisfies $f(p^m) = (m+1)q^{-\delta m \deg p} \to 0$ as $m \deg p \to \infty$, for an irreducible polynomial p. We can therefore apply the aforementioned generalization and deduce $f(a) = \mathcal{O}(1)$ and therefore $d(a) = \mathcal{O}_\delta(q^{\delta \deg(a)})$.

Acknowledgments. We wish to thank the referees for their help.

REFERENCES

[1] P. Chiu, *Cubic Ramanujan graphs*, Combinatorica, 12 (1992), pp. 275–285.

[2] G. Davidoff and P. Sarnak, *An Elementary Approach to Ramanujan Graphs*, preprint.

[3] L. Decreusefond and G. Zémor, *On the error-correcting capabilities of cycle codes of graphs*, Combin. Probab. Comput., 6 (1997), pp. 27–38.

[4] V. G. Drinfeld, *The proof of Peterson's conjecture for $GL(2)$ over global fields of characteristic p*, Functional Anal. Appl., 22 (1988), pp. 28–43.

[5] M. Eichler, *Quaternäre quadratische Formen und die Riemannsche Vermutung für die kongruent Zeta Funktion*, Arch. Math., 5 (1954), pp. 355–366 (in German).

[6] P. Elias, *Error-free coding*, IRE Trans. Inf. Theory, IT-4 (1954), pp. 29–37.

[7] G. D. Forney, *Concatenated Codes*, M.I.T. Press, Cambridge, MA, 1966.

[8] R. G. Gallager, *Low-density parity check codes*, IRE Trans. Inf. Theory, IT-8 (1962), pp. 21–28.

[9] E. Grosswald, *Representation of integers as sum of squares*, Springer-Verlag, New York, 1985.

[10] S. L. Hakimi and J. G. Bredeson, *Graph theoretic error-correcting codes*, IEEE Trans. Inf. Theory, IT-14 (1968), pp. 584–591.

[11] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers*, Oxford Press, Oxford, 1960.

[12] J. Igusa, *Fibre systems of Jacobian varieties* III, Amer. J., 81 (1959), pp. 453–476.

[13] J. Justesen, *A class of asymptotically good algebraic codes*, IEEE Trans. Inf. Theory, IT-18 (1972), pp. 652–656.

[14] A. Lubotsky, *Discrete Groups, Expanding Graphs, and Invariant Measures*, Lectures Notes, University of Oklahoma, Norman, OK, 1989.

[15] A. Lubotzky, R. Phillips, and P. Sarnak, *Ramanujan graphs*, Combinatorica, 8 (1988), pp. 261–277.

[16] G. A. Margulis, *Explicit constructions of graphs without short cycles and low density codes*, Combinatorica, 2 (1982), pp. 71–78.

[17] G. A. Margulis, *Explicit group theoretical constructions of combinatorial schemes and their application to the design of expanders and concentrators*, Problems Inform. Transmission, 1 (1988), pp. 51–60.

[18] M. Morgenstern, *Existence and explicit constructions of $q+1$-regular Ramanujan graphs for every prime power q*, J. Combin. Theory Ser. B, 62 (1994), pp. 44–62.

[19] P. Sarnak, *Some Applications of Modular Forms*, Cambridge University Press, Cambridge, 1990.

[20] M. Sipser and D. A. Spielman, *Expander codes*, in Proceedings of the 35th IEEE Symposium on Foundations of Computer Science, Santa Fe, NM, 1994, pp. 553–564.

SIAM J. Discrete Math.
Vol. 10, No. 3, pp. 460–468, August 1997

010

A CONSTRUCTION FOR (t, m, s)-NETS IN BASE q*

MICHAEL J. ADAMS† AND BRYAN L. SHADER†

Abstract. It has been shown that (t, m, s)-nets in base b can be characterized by combinatorial objects known as generalized orthogonal arrays. In this paper, a new construction for generalized orthogonal arrays leads to new (t, m, s)-nets in base q, q a prime power. The basic building block for the construction is an array of elements over F_q in which certain collections of rows are linearly independent. It is shown that if there exists an $[n, n-m, d]$ q-ary code with $d \geq 6+2p$, where $p \geq 0$ is an integer, then there exists a (t, m, s)-net in base q with $t = m - (4 + 2p)$ and $s = \lfloor \frac{n-1}{1+p} \rfloor$.

Key words. (t, m, s)-nets, generalized orthogonal arrays, linear codes

AMS subject classifications. 05B15, 05B30, 11K38, 11K45

PII. S0895480195295975

1. Introduction.

Finite multisets of points in the unit n-cube with low discrepancy have been used in developing pseudorandom number generators and quasi-Monte Carlo integration methods (see [4], [5], [6], [12]). In [13], Niederreiter introduces the notion of (t, m, s)-nets in base b as a class of low-discrepancy point sets, defined as follows. Fix integers $s \geq 1$ and $b \geq 2$. We denote by $I^s = [0, 1)^s$ the set of all points $(x_1, x_2, \ldots, x_s)$ in $\Re^s$ with $0 \leq x_i < 1$ for $i = 1, 2, \ldots, s$. An *elementary interval in base* b of I^s is a subset J of I^s of the form

$$J = \prod_{i=1}^{s} \left[\frac{a_i}{b^{d_i}}, \frac{a_i + 1}{b^{d_i}} \right),$$

where d_i and a_i are nonnegative integers with $a_i < b^{d_i}$ for $i = 1, 2, \ldots, s$. The *volume* of J, $vol(J)$, is $\prod_{i=1}^{s} \frac{1}{b^{d_i}}$. Let t and m be integers with $0 \leq t \leq m$. A (t, m, s)-*net in base* b is a multiset P of b^m points in I^s such that every elementary interval J in base b with $vol(J) = \frac{1}{b^{m-t}}$ contains exactly b^t points of P. A fundamental problem is to construct (t, m, s)-nets in base b with the largest possible value of s given t, m, and b. Additional work on (t, m, s)-nets is contained in [7], [8], [10], [11], [14], [15], [16], and [17].

In [13], Niederreiter recognizes necessary combinatorial constraints on the parameters of (t, m, s)-nets. This observation is generalized by Mullen and Whittle [11] in the language of orthogonal hypercubes. Niederreiter [16] then notes a connection between (t, m, s)-nets and combinatorial objects known as orthogonal arrays. An (N, s, b, t) *orthogonal array of index* λ is an $s \times N$ matrix A with entries from a b-element set such that each possible $t \times 1$ column occurs exactly λ times in each $t \times N$ submatrix of A. It follows from the definition that $N = b^t \lambda$. Basic properties and constructions of orthogonal arrays can be found in [19].

In [7], Lawrence defines generalized orthogonal arrays (GOAs) and uses these combinatorial objects to characterize (t, m, s)-nets. His construction of GOAs gives rise to new families of (t, m, s)-nets. We note that an equivalent characterization of (t, m, s)-nets appears in [20].

*Received by the editors December 11, 1995; accepted for publication (in revised form) June 19, 1996. This research was partially supported by NSA grant MDA904-94-H-2051.

http://www.siam.org/journals/sidma/10-3/29597.html

†Department of Mathematics, University of Wyoming, Laramie, WY 82071 (mjadams@uwyo.edu, bshader@uwyo.edu).

In section 2, we give a new construction for GOAs, obtaining new families of (t, m, s)-nets in base q, where q is a prime power. The basic building block for the construction is an array of elements over a finite field in which certain collections of rows are linearly independent. These arrays act as linear generators for GOAs. In section 3, we describe a construction of such arrays from the parity check matrices of linear codes. In section 4, we illustrate the techniques developed in sections 2 and 3 by building (t, m, s)-nets in base 2 with values of s larger than those currently appearing in the literature. We conclude by presenting an open problem related to our constructions.

2. A construction for GOAs. We begin this section with Lawrence's definition of a GOA. Let s, N, and l be positive integers. Consider the $s \times N \times l$ array $A = (a_{ijk})$. For $1 \leq i \leq s,\ 1 \leq k \leq l$ we define the (i, k)th *row* of A to be the $1 \times N$ vector

$$A_{i,k} = (a_{i1k}, a_{i2k}, a_{i3k}, \dots, a_{iNk}).$$

Consider a collection $\mathcal{C}$ of d rows from A. We call $\mathcal{C}$ a *qualifying collection of d rows* provided that whenever $A_{i,k}$ is in $\mathcal{C}$, we also have $A_{i,k'}$ in $\mathcal{C}$ for $1 \leq k' < k$. Let d be a positive integer with $d \leq sl$. The $s \times N \times l$ array $A = (a_{ijk})$ with entries from a b-element set $(b \geq 2)$ is an (N, s, l, b, d) *GOA* of size N, s constraints, height l, b levels, strength d, and index λ if every qualifying collection of d rows when viewed as a $d \times N$ matrix forms an orthogonal array with parameters (N, d, b, d) of index λ.

In [7, Theorem 5.4.1], Lawrence states and proves the following beautiful combinatorial characterization of (t, m, s)-nets.

THEOREM 2.1. *Let $s \geq 1,\ b \geq 2,\ t \geq 0$, and m be integers and assume that $m \geq t + 1$ to avoid degeneracy. Then there exists a (t, m, s)-net in base b if and only if there exists a $(b^m, s, m - t, b, m - t)$ GOA of index b^t.*

His proof is constructive; thus, construction of a (t, m, s)-net in base b is equivalent to constructing the corresponding GOA. In [7, Theorem 6.2.1], Lawrence provides a construction of GOAs using orthogonal arrays and is therefore able to construct new families of (t, m, s)-nets in base b. The construction requires that one have in hand an orthogonal array with parameters $(b^m, k, b, m - t)$ of index b^t, $m - t \geq 2$. The value of s is determined by k and $m - t$.

One method of constructing orthogonal arrays is given by a theorem of Bose and Bush, which first appears in [1]. For completeness we restate their theorem here with slight modifications in notation.

THEOREM 2.2. *Let q be a prime power and $t \geq 2$ an integer, and let C be a $k \times m$ matrix with entries in F_q. Assume that any t of the rows of C are linearly independent as elements of the F_q-vector space F_q^m or, equivalently, that every partial matrix obtained by taking any t rows of C is of rank t. Then there exists a (q^m, k, q, t) orthogonal array of index q^{m-t}.*

A similar linear construction exists for GOAs. Consider an $sl \times m$ array with entries from F_q. Partition this array into l *blocks*, $B^{(1)}, \dots,\ B^{(l)}$, a block consisting of s consecutive rows of the array. A set of t rows will be called a qualifying collection of rows if whenever the ith row of the block $B^{(k)}$ belongs to the collection, then so must the ith rows of the blocks $B^{(k')}$ for $1 \leq k' < k$. We now define an $M(s, l, m, q, t)$ *array* to be an $sl \times m$ array with entries from F_q such that every qualifying collection of t rows are linearly independent as elements of the vector space F_q^m.

An easy generalization of Theorem 2.2 yields a linear construction of GOAs over F_q.

Theorem 2.3. *Let s, m, l, and t be positive integers with $t \leq \min\{m, sl\}$. Assume there is an $M(s, l, m, q, t)$ array. Then there exists a (q^m, s, l, q, t) GOA of index q^{m-t}.*

Proof. Let C be an $M(s, l, m, q, t)$ array and let A be an $m \times q^m$ array over F_q in which each vector of F_q^m occurs exactly once as a column of A. Let $B = CA$. Partition B into l blocks, each of s consecutive rows, and let B' be the $s \times q^m \times l$ array whose (i, k)th row is the ith row of the kth block. We claim that B' is the desired GOA.

To see this, let $\mathcal{B}$ be any qualifying collection of t rows from B. There is a corresponding qualifying collection of rows $\mathcal{C}$ from C, where the rth row of B belongs to $\mathcal{B}$ if and only if the rth row of C belongs to $\mathcal{C}$. The rows of $\mathcal{B}$, written as a $t \times q^m$ matrix over F_q, can be expressed as the product $C'A$, where C' is the $t \times m$ matrix of the corresponding rows of $\mathcal{C}$. The matrix C' is of rank t, and thus the dimension of the column space of C' is t and the right null space is of dimension $m - t$. Hence, the matrix C' induces a linear transformation from F_q^m to F_q^t, which is a q^{m-t} to 1 and surjective map. It follows that each element of F_q^t occurs as a column of $C'A$ exactly q^{m-t} times. Since every qualifying collection of t rows of B' can be so described, it follows that B' is the desired GOA. □

We conclude this section with the following remarks.

An $n \times (n-k)$ matrix over F_q in which every $d-1$ row is linearly independent, but some d rows are linearly dependent, is a (transposed) parity check matrix of an $[n, k, d]$ q-ary linear code. In [2], linear codes are generalized to poset codes. In that setting, an $M(s, l, m, q, t)$ array is a parity check matrix of an $[sl, sl-m, d]$ q-ary poset code, $d \geq t+1$, for a poset consisting of s disjoint chains, each of length l.

Of special interest are those GOAs which characterize (t, m, s)-nets. The (t, m, s)-nets constructed from $M(s, l, m, q, t)$ arrays via Theorems 2.1 and 2.3 are equivalent to the nets constructed by Niederreiter in [13] taking the ring R in his construction to be the field F_q. These nets are now referred to as *digital nets.*

3. A new class of (t, m, s)-nets in base q. In this section we offer two constructions for (t, m, s)-nets in base q, q a prime power. Construction I yields (t, m, s)-nets with $m - t = 4$. Construction II is a generalization of Construction I and yields (t, m, s)-nets with $m - t \geq 4$ and even. Both constructions depend on the existence of linear codes over F_q with the appropriate parameters.

Construction I: We construct an $(m-4, m, s)$-net in base q using the parity check matrix of a linear code. The values of m and s will be determined by the code. By Theorems 2.1 and 2.3 it suffices to construct an $M(s, 4, m, q, 4)$ array C. Assume there is an $[n, n-m, d]$ q-ary linear code with $d \geq 6$ and let V be an $n \times m$ parity check matrix of the code. Let v_i denote the ith row of V. Set $s = n-1$ and let $C^{(1)}, \ldots, C^{(4)}$ denote the four blocks of C. Order the indices $1, 2, \ldots, n-1$ cyclically. Set

$$\begin{aligned} C_i^{(1)} &= v_i, \\ C_i^{(2)} &= v_{i+1} + v_n, \\ C_i^{(3)} &= v_{i+2} + v_n, \\ C_i^{(4)} &= v_{i+1}, \end{aligned}$$

where $C_i^{(k)}$ denotes the ith row of the block $C^{(k)}$ for $1 \leq i \leq s$. Following Theorem 3.1 we will establish that C is an $M(n-1, 4, m, q, 4)$ array. This finishes our description of Construction I.

For fixed values of $m \geq 5$ and prime power q, Construction I yields the largest values for s when we take V to be the parity check matrix of an $[n, n-m, 6]$ q-ary code with n as large as possible. In section 4 we illustrate both Constructions I and II by computing the values of s obtained when $q = 2$ and for selected values of t and m. To compute these values we used the tables in [3] of minimum-distance bounds for binary linear codes. The computed values of s exceed those reported in [7] and [9].

We motivate the next theorem and Construction II with the following observation. The $M(n-1, 4, m, q, 4)$ array C constructed above using rows of the $n \times m$ matrix V can be factored as $C = HV$, where H is the relatively sparse $M(n-1, 4, n, q, 4)$ array over F_q whose kth blocks, $1 \leq k \leq 4$, are given below:

$$H^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \tag{3.1}$$

$$H^{(2)} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix},$$

$$H^{(3)} = \begin{bmatrix} 0 & 0 & 1 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix},$$

$$H^{(4)} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \\ 1 & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

We claim that any qualifying collection of four rows of H forms a linearly independent set. Let $(e_i|x)$ denote a $1 \times n$ vector where in the first $n-1$ components there is a 1 in the ith position, 0's elsewhere, and in the nth position $x = 0$ or $x = 1$. Order the indices $1, 2, \ldots, n-1$ cyclically. Then a qualifying collection of four rows belongs to exactly one of the following cases.

Case 1: $(e_i|0)$, $(e_j|0)$, $(e_k|0)$, $(e_l|0)$, with i, j, k, l distinct.

Case 2: $(e_i|0)$, $(e_{i+1}|1)$, $(e_j|0)$, $(e_k|0)$, with i, j, k distinct.

Case 3: $(e_i|0)$, $(e_{i+1}|1)$, $(e_j|0)$, $(e_{j+1}|1)$, with i, j distinct.

Case 4: $(e_i|0)$, $(e_{i+1}|1)$, $(e_{i+2}|1)$, $(e_j|0)$, with i, j distinct.

Case 5: $(e_i|0),\ (e_{i+1}|1),\ (e_{i+2}|1),\ (e_{i+1}|0)$.

It is easy to verify that in each case the collections of vectors formed are linearly independent.

Consider an $M(s,l,m,q,t)$ array C. By the *support* of a row, we mean the set of coordinates for which the row has a nonzero entry. The *weight* of the row is the cardinality of its support. A *qualifying subspace* is a subspace of F_q^m generated by a qualifying collection of rows. The (generalized) *weight* of a qualifying subspace is the cardinality of the set of coordinates which occur in the support of at least one vector in the space. Note that the weight of any qualifying subspace generated by the rows of C is at least t.

The following notation will be used in the proof of the theorem below. Given an $m \times n$ matrix A and a set $\alpha \subseteq \{1, 2, \dots, m\}$, let $A[\alpha, :]$ denote the submatrix of A whose rows are the rows of A indexed by the set α. Similarly define $A[:, \beta]$ given $\beta \subseteq \{1, 2, \dots, n\}$, where β indexes a set of columns of A.

Theorem 3.1. *Assume there is an $M(s,l,n,q,t)$ array, all of whose qualifying subspaces have weight at most δ, and an $[n, n-m, d]$ q-ary linear code, with $d > \delta$. Then there is an $M(s,l,m,q,t)$ array.*

Proof. Let H be the $M(s,l,n,q,t)$ array and V an $n \times m$ parity check matrix of the code. We claim that $C = HV$ is the desired $M(s,l,m,q,t)$ array. To see this we must show that any qualifying collection of t rows of C is linearly independent. Such a set of rows may be written as a $t \times m$ matrix $C' = H'V$, where H' is the $t \times n$ matrix of the corresponding set of rows of H. Let α be the set of indices of the support of the qualifying subspace generated by the rows of H'. Then $C' = H'[:, \alpha]V[\alpha, :]$. Since $|\alpha| \leq \delta < d$, the rows of $V[\alpha, :]$ are linearly independent. It follows that the dimension of the row space of C' is equal to the dimension of the column space of H'. Thus, rank $C' =$ rank $H'[:, \alpha]$. Since the rows of H' are the rows of a qualifying collection of rows of H and α indexes the union of the supports of these rows, rank $H'[:, \alpha] = t$. It follows that the rows of C' are linearly independent. □

Now consider the factorization $C = HV$ of one of the arrays produced in Construction I. We have already seen that H is an $M(n-1, 4, n, q, 4)$ array. We claim that the maximum weight of any qualifying subspace of H is 5. Observe that each row of H has weight 1 or 2. Only those rows in $H^{(2)}$ and $H^{(3)}$ have weight 2, and any two such rows have common support in their nth component. Further, a qualifying collection of four rows of H can have at most two rows in $H^{(2)}$ and $H^{(3)}$. Thus, any qualifying subspace has weight at most 5. Since the qualifying collections in case 3 with $i,\ i+1,\ j$, and $j+1$ distinct have support size 5, we conclude that the maximum weight of any qualifying subspace of H is 5. By Theorem 3.1, H together with an $[n, n-m, d]$ q-ary linear code, $d > 5$, yields the desired $M(n-1, 4, m, q, 4)$ array. This establishes the validity of Construction I.

Families of (t, m, s)-nets in base q with "large" values of s may be constructed from the parity check matrices of good $[n, n-m, d]$ q-ary linear codes provided there exist $M(s, m-t, n, q, m-t)$ arrays, all of whose qualifying subspaces have weight at most $d-1$. We now describe the construction of such a family of arrays.

Construction II: Given an integer $s \geq 4$, let I_s represent the $s \times s$ identity matrix and W_s the $s \times s$ $(0,1)$ matrix with 1's in positions $(1,2),\ (2,3),\ \dots, (s-1, s)$ and $(s, 1)$. We define the sequence of arrays $\{H_{s,p}\}_{p=0}^{\infty}$ recursively. Let $H_{s,0}$ be an $M(s, 4, s+1, q, 4)$ array of the form described in (3.1). For $p \geq 1$, define $H_{s,p}$ to be the $(4+2p)s \times ((p+1)s+1)$ array whose kth blocks, $1 \leq k \leq 4+2p$, are given below:

$$H^{(1)}_{s,p} = \begin{bmatrix} & 0\cdots 0 \\ I_s & \vdots \ddots \vdots \\ & 0\cdots 0 \end{bmatrix},$$

$$H^{(k)}_{s,p} = \begin{bmatrix} 0\cdots 0 & \\ \vdots \ddots \vdots & H^{(k-1)}_{s,p-1} \\ 0\cdots 0 & \end{bmatrix},\ 2 \le k \le 3+2p, \tag{3.2}$$

$$H^{(4+2p)}_{s,p} = \begin{bmatrix} & 0\cdots 0 \\ W_s & \vdots \ddots \vdots \\ & 0\cdots 0 \end{bmatrix}.$$

It will be established below that $H_{s,p}$ is an $M(s, 4+2p, (p+1)s+1, q, 4+2p)$ array and that the maximum weight of any qualifying subspace is $5+2p$ for all $p \ge 0$. Assuming this to be the case, fix integers m, p, and t with $p \ge 0$, $m \ge 5+2p$, and $m-t = 4+2p$. Assume there exists an $[n, n-m, d]$ q-ary linear code, $d > 5+2p$. Set $s = \lfloor \frac{n-1}{1+p} \rfloor$ and let $H_{s,p}$ be an $M(s, 4+2p, (p+1)s+1, q, 4+2p)$ array of the form (3.2). Taking any $(p+1)s+1$ rows of a parity check matrix for the code, together with $H_{s,p}$, Theorem 3.1 implies that we may construct an $M(s, 4+2p, m, q, 4+2p)$ array. The existence of an $M(s, 4+2p, m, q, 4+2p)$ array and Theorem 2.3 allow us to construct a $(q^m, s, 4+2p, q, 4+2p)$ GOA of index $q^{m-(4+2p)}$, and by Theorem 2.1 this is equivalent to constructing an $(m-(4+2p), m, \lfloor \frac{n-1}{1+p} \rfloor)$-net in base q. This finishes our description of Construction II.

Note that in the initial case, $p = 0$, this is equivalent to Construction I. As in Construction I, the largest values for s are obtained when we construct V using a parity check matrix for an $[n, n-m, 6+2p]$ q-ary code with n as large as possible. Given an $[n, n-m, 6+2p]$ q-ary code, the Singleton bound (see, for example, [18]) implies that $m \ge 5+2p$, as required in the construction. We now show that the arrays $H_{s,p}$ are $M(s, 4+2p, (p+1)s+1, q, 4+2p)$ arrays, as claimed.

Let $\mathcal{H}'$ be a qualifying collection of $4+2p$ rows of $H_{s,p}$. Observe that the rows of $H_{s,p}$ each have weight 1 or 2. Only those rows in $H^{(2+p)}_{s,p}$ and $H^{(3+p)}_{s,p}$ have weight 2, and any two such rows have common support in their last component. Since $\mathcal{H}'$ can have at most two rows from blocks $H^{(2+p)}_{s,p}$ and $H^{(3+p)}_{s,p}$, the weight of the qualifying subspace generated by $\mathcal{H}'$ is at most $5+2p$. Since $\mathcal{H}'$={rows 1 and 3 from each of the blocks $H^{(k)}_{s,p},\ 1 \le k \le 2+p$} generates a subspace of weight $5+2p$, we conclude that the maximum weight of any qualifying subspace is $5+2p$.

The assertion about linear independence of any qualifying collection of rows follows from the proposition below.

Proposition 3.2. *Let H be any $M(s,l,n,q,l)$ array. Then the $(l+2)s \times (n+s)$ array H' over F_q whose rows are partitioned into the $l+2$ blocks below is an $M(s, l+2, n+s, q, l+2)$ array:*

$$H'^{(1)} = \begin{bmatrix} & 0\cdots 0 \\ I_s & \vdots \ddots \vdots \\ & 0\cdots 0 \end{bmatrix},$$

$$H'^{(k)} = \begin{bmatrix} 0\cdots 0 & \\ \vdots \ddots \vdots & H^{(k-1)} \\ 0\cdots 0 & \end{bmatrix},\ 2 \le k \le l+1,$$

$$H'^{(l+2)} = \begin{bmatrix} & 0\cdots 0 \\ W_s & \vdots \ddots \vdots \\ & 0\cdots 0 \end{bmatrix}.$$

Proof. Let $\mathcal{H}'$ be a qualifying collection of $l+2$ rows of H'. The rows of $\mathcal{H}'$ belonging to blocks 2 through $l+1$, when restricted to their last n components, correspond to a qualifying collection of at most l rows of H and hence form a linearly independent set. It is clear that any set of distinct rows of $H'^{(1)}$ forms a linearly independent set and that any nonzero linear combination of rows of $H'^{(1)}$ is independent of any nonzero linear combination of rows from blocks 2 through $l+1$. Thus, if $\mathcal{H}'$ contains no row from $H'^{(l+2)}$, then $\mathcal{H}'$ is a linearly independent set. If $\mathcal{H}'$ contains a row from $H'^{(l+2)}$, then $\mathcal{H}'$ necessarily consists of the ith row from each of blocks 1 through $l+2$ for some fixed index i. This is the same set which occurs in the qualifying collection consisting of the ith row from each of blocks 1 through $l+1$, together with the $(i+1)$st row of block 1 (assuming a cyclic ordering of the indices). As demonstrated above, this is a linearly independent set. Thus, $\mathcal{H}'$ is a linearly independent set. □

In summary, given an $[n, n-m, d]$ q-ary linear code with $d \geq 6+2p$, where $p \geq 0$ is an integer, we set $s = \lfloor \frac{n-1}{1+p} \rfloor$ and construct the $M(s, 4+2p, m, q, 4+2p)$ array $C = H_{s,p}V$, where V is a $((p+1)s+1) \times m$ array of rows from a parity check matrix for the code. Using Theorem 2.3, C yields a $(q^m, s, 4+2p, q, 4+2p)$ GOA which can be used to construct the corresponding (t, m, s)-net in base q, $m-t = 4+2p$. Using the results of this and the previous section, we have proved the following theorem.

THEOREM 3.3. *Suppose there exists an $[n, n-m, d]$ q-ary linear code with $d \geq 6+2p$, where $p \geq 0$ is an integer. Then there exists a (t, m, s)-net in base q with $t = m-(4+2p)$ and $s = \lfloor \frac{n-1}{1+p} \rfloor$.*

4. Examples and concluding remarks. To illustrate our results, we construct a $(3, 7, s)$-net in base 2. Since $m = 7$ and $t = 3$, we have $p = 0$. We seek an $[n, n-7, 6]$ binary linear code with n as large as possible. In [3] we see that there exists a $[9, 2, 6]$ binary linear code, so by Theorem 3.3 there exists a $(3, 7, 8)$-net in base 2. To construct such a net, we take $H_{8,0}$ to be the $M(8, 4, 9, 2, 4)$ array described in (3.2) and V to be a 9×7 parity check matrix of the code. By Theorem 3.1, $C = H_{8,0}V$ is an $M(8, 4, 7, 2, 4)$ array. By Theorem 2.3, C yields a $(2^7, 8, 4, 2, 4)$ GOA A of index 2^3 which determines a $(3, 7, 8)$-net in base 2: call this point set P. Following the proof of Theorem 2.1 [7, Theorem 5.4.1], we construct the set $P = \{x^j\}_{j=1}^{2^7} \subset I^8$, where $x^j = (x_1^j, x_2^j, \ldots, x_8^j)$, by setting $x_i^j = \sum_{k=1}^{4} A_{ijk}2^{-k}$ for $1 \leq j \leq 2^7$ and $1 \leq i \leq 8$.

In the following tables, for $q = 2$ and some small values of t and m, we compare the values of s achieved using Theorem 3.3 with those appearing in [7].

Note that as $m-t$ increases, the difference between the two values for s tends to decrease. In fact, for $m-t = 8$, using Lawrence's construction we can produce a $(20, 28, 32)$-net in base 2, whereas our construction yields a value of $s = 30$ for the same t and m. For $m-t \geq 4$ and even, our construction requires an $[n, n-m, d]$ q-ary linear code with $d \geq m-t+2$, yielding a value of $s = \lfloor \frac{2n-2}{m-t-2} \rfloor$. The construction described in [7, Theorem 6.2.1] requires that one have in hand an orthogonal array in order to build a GOA. In the event that a linear code is used to produce the requisite orthogonal array, then an $[n', n'-m, d']$ q-ary linear code with $d \geq m-t+1$ yields a value of $s = \lfloor \frac{2n'}{m-t} \rfloor$ when $m-t \geq 4$ and even. Thus, as $m-t$ increases, we expect our construction to yield values of s comparable to those produced in Lawrence's construction when linear codes are employed to construct the required orthogonal arrays.

Table 1

	t	m	Value of s from Thm. 3.3	Value of s from [7]
$m - t = 4$	3	7	8	7
	4	8	11	8
	5	9	17	11
	6	10	23	16
	7	11	33	31
	8	12	47	32
	9	13	65	40
	10	14	81	64

Table 2

	t	m	Value of s from Thm. 3.3	Value of s from [7]
$m - t = 6$	6	12	11	8
	7	13	12	9
	8	14	13	10
	9	15	15	12
	10	16	18	15
	11	17	23	21
	12	18	31	22
	13	19	34	29
	14	20	44	31
	15	21	47	42

Table 3

	t	m	Value of s from Thm. 3.3	Value of s from [7]
$m - t = 8$	16	24	18	16
	17	25	21	17
	18	26	22	19
	19	27	25	22

We pose the following open problem. The strength of Theorem 3.3 depends on the existence of an $M(s, m - t, n, q, m - t)$ array with s as large as possible and the maximum weight of any qualifying subspace as small as possible. With the goal of constructing (t, m, s)-nets in base q with large values of s, we propose as an open problem the construction of $M(s, m-t, n, q, m-t)$ arrays with n/s and the maximum weight of any qualifying subspace simultaneously as small as possible.

Since the submission of this paper, we have become aware of a manuscript submitted to Acta Arithmetica by W. Ch. Schmid and R. Wolf, in which they demonstrate the existence of $(t, t + 4, s)$-nets in base q with values of s which sometimes exceed those obtained by our constructions.

Acknowledgments. The authors wish to thank the anonymous referee for several insightful comments which helped improve the paper.

REFERENCES

[1] R. C. Bose and K. A. Bush, *Orthogonal arrays of strength two and three*, Ann. Math. Stat., 23 (1952), pp. 508–524.

[2] R. A. Brualdi, J. Graves, and K. M. Lawrence, *Codes with a poset metric*, Discrete Math., 147 (1995), pp. 57–72.

[3] A. E. Brouwer and T. Verhoeff, *An updated table of minimum-distance bounds for binary linear codes*, IEEE Trans. Inform. Theory, 39 (1993), pp. 662–677.

[4] E. Hlawka, *Uniform distribution modulo* 1 *and numerical analysis*, Compositio Math., 16 (1964), pp. 92–105.

[5] L. K. Hua and Y. Wang, *Applications of Number Theory to Numerical Analysis*, Springer-Verlag, Berlin, Heidelberg, New York, 1981.

[6] L. Kuipers and H. Niederreiter, *Uniform Distribution of Sequences*, Wiley-Interscience, New York, 1974.

[7] K. M. Lawrence, *Combinatorial Bounds and Constructions in the Theory of Uniform Point Distributions in Unit Cubes, Connections with Orthogonal Arrays and a Poset Generalization of a Related Problem in Coding Theory*, Ph.D. thesis, University of Wisconsin–Madison, Madison, WI, 1995.

[8] K. M. Lawrence, A. Mahalanabis, G. L. Mullen, and W. Ch. Schmid, *Construction of digital (t,m,s)-nets from linear codes*, Lecture Notes London Math. Soc., 233 (1996), pp. 189–208.

[9] G. L. Mullen, A. Mahalanabis, and H. Niederreiter, *Tables of* (t, m, s)*-net and* (t, s)*-sequence parameters*, in Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, H. Niederreiter and P.J.-S. Shiue, eds., Lecture Notes in Statistics, Vol. 106, Springer-Verlag, New York, 1995, pp. 58–86.

[10] G. L. Mullen and W. Ch. Schmid, *An equivalence between* (t, m, s)*-nets and strongly orthogonal hypercubes*, J. Combin. Theory Ser. A, 76 (1996), pp. 164–174.

[11] G. L. Mullen and G. Whittle, *Point sets with uniformity properties and orthogonal hypercubes*, Monatsh. Math., 113 (1992), pp. 265–273.

[12] H. Niederreiter, *Quasi-Monte Carlo methods and pseudo-random numbers*, Bull. Amer. Math. Soc., 84 (1978), pp. 957–1041.

[13] H. Niederreiter, *Point sets and sequences with small discrepancy*, Monatsh. Math., 104 (1987), pp. 273–337.

[14] H. Niederreiter, *Low-discrepancy and low-dispersion sequences*, J. Number Theory, 30 (1988), pp. 51–70.

[15] H. Niederreiter, *A combinatorial problem for vector spaces over finite fields*, Discrete Math., 96 (1991), pp. 221–228.

[16] H. Niederreiter, *Orthogonal arrays and other combinatorial aspects in the theory of uniform point distributions in unit cubes*, Discrete Math., 106/107 (1992), pp. 361–367.

[17] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, Number 63 in CBMS-NSF Series in Applied Mathematics, SIAM, Philadelphia, PA, 1992.

[18] V. Pless, *Introduction to the Theory of Error-Correction Codes*, 2nd ed., John Wiley, New York, 1989.

[19] D. Raghavarao, *Constructions and Combinatorial Problems in Design of Experiments*, John Wiley, New York, 1971.

[20] W. Ch. Schmid, *(T,M,S)-Nets: Digital Construction and Combinatorial Aspects*, Ph.D. thesis, Universität Salzburg, Salzburg, Austria, 1995.

SIAM J. Discrete Math.
Vol. 10, No. 3, pp. 469–481, August 1997

COMPUTING ALL SMALL CUTS IN AN UNDIRECTED NETWORK*

HIROSHI NAGAMOCHI†, KAZUHIRO NISHIMURA‡, and TOSHIHIDE IBARAKI†

Abstract. Let $\lambda(\mathcal{N})$ denote the weight of a minimum cut in an edge-weighted undirected network $\mathcal{N}$, and n and m denote the numbers of vertices and edges, respectively. It is known that $O(n^{2k})$ is an upper bound on the number of cuts with weights less than $k\lambda(\mathcal{N})$, where $k \geq 1$ is a given constant. This paper first shows that all cuts of weights less than $k\lambda(\mathcal{N})$ can be enumerated in $O(m^2n + n^{2k}m)$ time without using the maximum flow algorithm. The paper then proves for $k < \frac{4}{3}$ that $\binom{n}{2}$ is a tight upper bound on the number of cuts of weights less than $k\lambda(\mathcal{N})$, and that all those cuts can be enumerated in $O(m^2n + mn^2 \log n)$ time.

Key words. minimum cuts, graphs, edge-splitting, polynomial algorithm

AMS subject classifications. 05C35, 05C40

PII. S0895480194271323

1. Introduction.

Let $\mathcal{N}$ stand for an undirected network with its edges being weighted by nonnegative real numbers. Counting the number of cuts with small weights and deriving upper and lower bounds on their numbers play an important role in the reliability analysis of probabilistic networks whose edges are subject to failure [2], the graph augmentation problem, i.e., the problem of increasing the edge-connectivity by adding the smallest number of edges to a graph [15], and other problems.

Let $\lambda(\mathcal{N})$ denote the weight of a minimum cut in $\mathcal{N}$, and let n and m be the numbers of vertices and edges, respectively. It is known that an upper bound on the number of minimum cuts is $\binom{n}{2} = \frac{n}{2}(n-1)$, which is achievable when $\mathcal{N}$ is a ring consisting of n edges with weight $\lambda(\mathcal{N})/2$ [1], [3].

Recently, Vazirani and Yannakakis [17] showed that cuts of weights no more than the rth minimum weight can be enumerated by $O(r+n)$ maximum flow computations. Based on a probabilistic analysis, on the other hand, Karger [10] derived for arbitrary $k \geq 1$ an upper bound $O(n^{2k})$ on the number of cuts of weights no more than $k\lambda(\mathcal{N})$.

In this paper, for arbitrary $k > 1$, we enumerate all cuts with weights no more than $k\lambda(\mathcal{N})$ without relying on the maximum flow algorithm. Our enumeration algorithm makes use of the edge splitting operation (see section 4) to reduce the number of vertices by one while preserving the edge-connectivity. We repeatedly apply the edge-splitting operation until the network has only two vertices and obtain a sequence of such networks $\mathcal{N}_i$ with i vertices, $i = n, n-1, \ldots, 2$. After enumerating all small cuts (of weights no more than $k\lambda(N)$) in $\mathcal{N}_2$, the set of small cuts in $\mathcal{N}_{i+1}$ are then computed from the set of those cuts in $\mathcal{N}_i$ in the order of $i = 3, 4, \ldots, n-1$. We can show that the entire running time of this algorithm is $O(m^2n + n^{2k}m)$. Thus, if there are $\Theta(n^{2k})$ such cuts, each cut is found in linear time. We then prove that the number of cuts with weights less than $\frac{4}{3}\lambda(\mathcal{N})$ is at most $\binom{n}{2}$ (i.e., the upper bound on the number of minimum cuts), that this bound is tight for any number n of vertices,

*Received by the editors July 18, 1994; accepted for publication (in revised form) June 19, 1996. This research was partially supported by the Scientific Grant-in-Aid from the Ministry of Education, Science and Culture of Japan.

http://www.siam.org/journals/sidma/10-3/27132.html

†Kyoto University, Kyoto, Japan 606-01 (naga@kuamp.kyoto-u.ac.jp, ibaraki@kuamp.kyoto-u.ac.jp).

‡Toyota Motor Corporation, Sizuoka, Japan 410-11 (nisimura@fw23595.mj.toyota.co.jp).

and that $\frac{4}{3}$ is best possible for $\binom{n}{2}$ to be an upper bound. The time of our algorithm to enumerate all the cuts with weights less than $\frac{4}{3}\lambda(\mathcal{N})$ becomes $O(m^2n+mn^2\log n)$.

Recently, Henzinger and Williamson [9] extended the above argument to prove an $O(n^2)$ upper bound on the number of cuts with weights less than $\frac{3}{2}\lambda(\mathcal{N})$.

The remainder of this paper is organized as follows. Section 2 describes basic definitions and notations. Before presenting an algorithm to compute all small cuts, we review in sections 3–5 the concepts of s-connectivity, weighted edge-splitting, and vertex isolation, and discuss how to compute them. Based on these, section 6 gives the algorithm to compute all small cuts. Finally, section 7 derives an upper bound on the number of small cuts with weights less than $\frac{4}{3}\lambda(\mathcal{N})$ and applies this to evaluate the time complexity of the above algorithm.

2. Preliminaries.

Let $\mathcal{N}=(V,E,c)$ be an undirected network with a set V of vertices and a set E of edges weighted by $c: E \mapsto \boldsymbol{R}^+$, where $\boldsymbol{R}^+$ is the set of nonnegative real numbers. Throughout the paper, we assume, for notational convenience, that (V,E) forms a simple complete graph, and denote by $E_c \subseteq E$ the set of edges with *positive* weights (from the computational point of view, we only have to maintain graph (V,E_c)). An edge e with its end vertices u and v is denoted by (u,v) or (v,u), and its weight $c((u,v))$ $(=c((v,u)))$ is written by $c(u,v)$ $(=c(v,u))$, unless confusion arises. A vertex adjacent to a vertex $v \in V$ by an edge with positive weight is called a *neighbor* of v. Let $NB(v;\mathcal{N})=\{w \in V \mid (v,w) \in E_c\}$ denote the set of *neighbors* of a vertex v.

A singleton set $\{x\}$ may be simply written as x, and " $\subset$ " implies proper inclusion while " $\subseteq$ " means " $\subset$ " or " $=$ ".

For two disjoint subsets $X,Y \subset V$ of a network $\mathcal{N}$, $E(X,Y;\mathcal{N})$ denotes the set of edges one of whose end vertices is in X and the other is in Y and $c\,(X,Y;\mathcal{N})$ denotes the sum of edge weights in $E(X,Y;\mathcal{N})$. $E(X,Y;\mathcal{N})$ and $c(X,Y;\mathcal{N})$ may be written as $E(X,Y)$ and $c(X,Y)$, respectively, if $\mathcal{N}$ is clear from context.

A *cut* is defined as a subset X of V with $\emptyset \neq X \neq V$. We say that a cut X *separates* two disjoint subsets Y and Y' of V if $Y \subseteq X$ and $Y' \subseteq V-X$ (or $Y \subseteq V-X$ and $Y' \subseteq X$) hold. The *weight* of a cut X is defined by $c\,(X,V-X;\mathcal{N})$, which may be written as $c\,(X;\mathcal{N})$ or $c\,(X)$. A cut is called an *α-cut* if it has weight α. Clearly, a cut X and its complement $V-X$ (which is also a cut) have the same weight $c\,(X)=c\,(V-X)$. For this reason, we often do not distinguish two cuts X and $V-X$. In particular, in generating small cuts, we want to generate only one of X and $V-X$.

A cut X *crosses* another cut Y if $X\cap Y \neq \emptyset$, $X-Y\neq\emptyset$, $Y-X\neq\emptyset$ and $V-X-Y\neq\emptyset$. For two crossing cuts X,Y, we can easily see the following identity (see Fig. 1).

$$c(X)+c(Y)=c(X-Y)+c(Y-X)+2c(X\cap Y,V-(X\cup Y)). \tag{2.1}$$

The *local edge-connectivity* $\lambda(x,y;\mathcal{N})$ for two vertices $x,y\in V$ is defined to be the minimum weight of a cut that separates x and y (we define $\lambda(x,y;\mathcal{N})=+\infty$ if $x=y$). A cut X is a *minimum cut* if $c\,(X;\mathcal{N})$ is minimum among all cuts in $\mathcal{N}$. The weight of a minimum cut is called the *global edge-connectivity* of $\mathcal{N}$ and denoted by $\lambda(\mathcal{N})$ (we define $\lambda(\mathcal{N})=+\infty$ if $|V|=1$). In other words, $\lambda(\mathcal{N})=\min\{\lambda(x,y;\mathcal{N})\mid x,y\in V\}$. Throughout this paper, we assume $\lambda(\mathcal{N})>0$ (i.e., graph (V,E_c) is connected).

3. Computing s-connectivity $\lambda_s(\mathcal{N})$.

For a network $\mathcal{N}$, choose a vertex s as a designated vertex. A cut X is called *s-proper* if $\emptyset\neq X\subset V-s$ (recall that

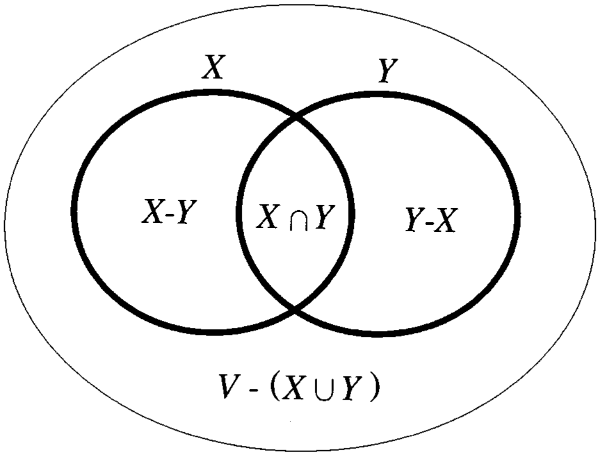


FIG. 1. *Illustration of two crossing cuts X and Y.*

$\subset$ denotes proper inclusion). The *s-connectivity* $\lambda_s(\mathcal{N})$ is the weight of a minimum s-proper cut (we define $\lambda_s(\mathcal{N}) = +\infty$ if $|V| \le 2$). In other words,

$$\lambda_s(\mathcal{N}) = \min\{\lambda(x, y; \mathcal{N}) \mid x, y \in V - s\}$$

(hence, $\lambda(\mathcal{N}) = \min\{\lambda_s(\mathcal{N}), c(s; \mathcal{N})\}$). An s-proper cut X is called *s-tight* if $c(X; \mathcal{N}) = \lambda_s(\mathcal{N})$.

LEMMA 3.1. *For a network $\mathcal{N} = (V, E, c)$ with a designated vertex $s \in V$, the s-connectivity $\lambda_s(\mathcal{N})$ and an s-tight cut T can be computed in $O(n(m+n\log n))$ time.*

Proof. Assume $n \ge 3$. We use the $O(m + n \log n)$ time graph traversal algorithm [13] that visits every vertex exactly once in the following max-adjacency order: (i) it first visits s and (ii) it chooses the ith vertex v_i from the unvisited vertices so that $c(\{v_1, v_2, \ldots, v_{i-1}\}, v_i; \mathcal{N})$ is maximized, where $v_1 = s, v_2, \ldots, v_{i-1}$ are the vertices visited so far. It is known [13] that the resulting order $v_1, \ldots, v_n$ of vertices satisfies

$$\lambda(v_{n-1}, v_n; \mathcal{N}) = c(v_n; \mathcal{N}) \tag{3.1}$$

(only positive capacities are handled in [13], but (3.1) follows from [13] by allowing capacities to take zero. See [6], [7], [14], [16] for simpler proofs of this property). Clearly, $s \ne v_{n-1}, v_n$. Let $\mathcal{N}_n := \mathcal{N}$ and $\mathcal{N}_{n-1}$ be the network obtained from $\mathcal{N}$ by contracting v_{n-1} and v_n into a single vertex. Any cut that separates v_{n-1} and v_n is s-proper, and hence the minimum weight of such a cut is $c(v_n; \mathcal{N}_n)$ by (3.1). Any s-proper cut that does not separates v_{n-1} and v_n remains in the contracted network $\mathcal{N}_{n-1}$. Thus, we have

$$\lambda_s(\mathcal{N}_n) = \min\{c(v_n; \mathcal{N}_n), \lambda_s(\mathcal{N}_{n-1})\}.$$

Therefore, by repeating this traversal and contraction procedure until the network has *three* vertices, $\lambda_s(\mathcal{N})$ is equal to

$$c(v'_p; \mathcal{N}_p) = \min\{c(v'_n; \mathcal{N}_n), c(v'_{n-1}; \mathcal{N}_{n-1}), \ldots, c(v'_3; \mathcal{N}_3)\},$$

where v'_i is the last vertex in $\mathcal{N}_i$ in the above traversal procedure. An s-tight cut in $\mathcal{N}$ can be obtained as the set of vertices in V that are contracted into the vertex v'_p. □

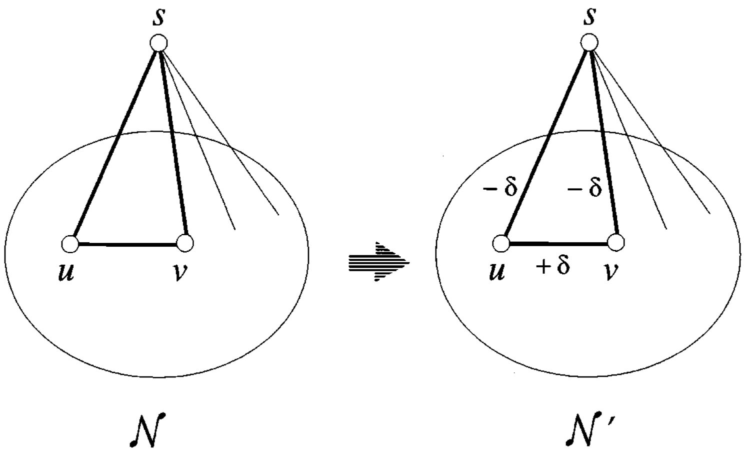


FIG. 2. *Illustration of edge-splitting (s,u) and (s,v) of weight δ.*

4. Weighted edge-splitting. Edge-splitting is one of the most useful operations, which reduces the size of a graph while preserving edge-connectivity [4], [5], [11], [12]. This section defines the operation of weighted edge-splitting and derives some key lemmas.

Given a network $\mathcal{N} = (V, E, c)$, a designated vertex $s \in V$, vertices $u, v \in NB(s;\mathcal{N})$ (possibly $u = v$), and a nonnegative real $\delta \leq \delta_{max}$, where

$$\delta_{max} = \min\{c(s,u), c(s,v)\},$$

we construct the following network $\mathcal{N}' = (V, E, c')$:

$c'(s,u) := c(s,u) - \delta, \quad c'(s,v) := c(s,v) - \delta, \quad c'(u,v) := c(u,v) + \delta,$
$c'(x,y) := c(x,y) \quad \text{for } (x,y) \in E - \{(s,u),(s,v),(u,v)\}$

(in case of $u = v$, we interpret $c'(s,u) := c(s,u) - 2\delta$, $c'(x,y) := c(x,y)$ for $(x,y) \in E - (s,u)$). We say that $\mathcal{N}'$ is obtained from $\mathcal{N}$ by *edge-splitting* (s,u) and (s,v) of *weight* δ and denote the resulting network $\mathcal{N}'$ by $\mathcal{N}/(u,v,\delta)$. (see Fig. 2.) Clearly, for any cut X,

$$(4.1) \quad c(X;\mathcal{N}/(u,v,\delta)) = \begin{cases} c(X;\mathcal{N}) - 2\delta & \text{if cut } X \text{ separates } \{s\} \text{ and } \{u,v\}, \\ c(X;\mathcal{N}) & \text{otherwise,} \end{cases}$$

and hence $\lambda_s(\mathcal{N}/(u,v,\delta)) \leq \lambda_s(\mathcal{N})$ holds for any $\delta \leq \delta_{max}$. Let $\delta_s(u,v;\mathcal{N})$ denote the maximum δ such that $\delta \leq \delta_{max}$ and $\lambda_s(\mathcal{N}/(u,v,\delta)) = \lambda_s(\mathcal{N})$; i.e.,

$$(4.2)\ \delta_s(u,v;\mathcal{N}) = \min\left\{\delta_{max}, \frac{1}{2}[\min\{c(X;\mathcal{N}) \mid \{u,v\} \subseteq X \subseteq V - s\} - \lambda_s(\mathcal{N})]\right\}.$$

Any s-tight cut in $\mathcal{N}$ remains s-tight in network $\mathcal{N}/(u,v,\varepsilon)$ for $0 \leq \varepsilon \leq \delta_s(u,v;\mathcal{N})$.

LEMMA 4.1. *For a network $\mathcal{N} = (V, E, c)$ with a designated $s \in V$ and two distinct $u, v \in NB(s;\mathcal{N})$, let $\mathcal{N}' = (V, E', c')$ denote the network $\mathcal{N}/(u,v,\delta)$ with $\delta = \delta_s(u,v;\mathcal{N})$. Then the following properties hold.*

(i) *$\delta_s(u,v;\mathcal{N})$ can be computed in $O(n(m + n\log n))$ time.*

(ii) *If $c'(s,u) > 0$ and $c'(s,v) > 0$ (i.e., $\{u,v\} \subseteq NB(s;\mathcal{N}')$), then $\mathcal{N}'$ has an s-tight cut T such that $\{u,v\} \subseteq T$. Furthermore, such T can be found in $O(n(m + n\log n))$ time.*

Proof. We show that $\delta_s(u,v;\mathcal{N})$ can be determined by computing how much $\mathcal{N}$ loses the s-connectivity by the edge splitting (s,u) and (s,v) of weight $\delta_{max} = \min\{c(s,u), c(s,v)\}$. Let $\mathcal{N}_{max} = \mathcal{N}/(u,v,\delta_{max})$. Clearly, $\lambda_s(\mathcal{N}_{max}) = \lambda_s(\mathcal{N})$ implies $\delta_s(u,v;\mathcal{N}) = \delta_{max}$. If $\lambda_s(\mathcal{N}_{max}) < \lambda_s(\mathcal{N})$, then we see from (4.1) that any s-tight cut T in $\mathcal{N}_{max}$ separates s and $\{u,v\}$. Then $\lambda_s(\mathcal{N}_{max}) = c(T;\mathcal{N}_{max}) = \min\{c(X;\mathcal{N}) - 2\delta_{max} \mid \{u,v\} \subseteq X \subseteq V - s\} = \min\{c(X;\mathcal{N}) \mid \{u,v\} \subseteq X \subseteq V - s\} - 2\delta_{max}$. Therefore, from (4.2), we have

$$\delta_s(u,v;\mathcal{N}) = \delta_{max} - \frac{1}{2}[\lambda_s(\mathcal{N}) - \lambda_s(\mathcal{N}_{max})].$$

By Lemma 3.1, $\lambda_s(\mathcal{N})$, $\lambda_s(\mathcal{N}_{max})$ and an s-tight cut T in $\mathcal{N}_{max}$ can be determined in $O(n(m + n\log n))$ time, respectively. This proves (i).

Let $\mathcal{N}' = \mathcal{N}/(u,v,\delta)$ for $\delta = \delta_s(u,v;\mathcal{N})$. If $\lambda_s(\mathcal{N}') > \lambda_s(\mathcal{N}_{max})$, then $c'(s,u) > 0$ and $c'(s,v) > 0$ hold in $\mathcal{N}'$. Also the above cut T in $\mathcal{N}_{max}$ satisfies $\{u,v\} \subseteq T$ by $\lambda_s(\mathcal{N}') > \lambda_s(\mathcal{N}_{max})$, and hence it is also s-tight in $\mathcal{N}'$, proving (ii). □

Call a weighted edge-splitting of (s,u) and (s,v) by δ to be *safe* if $\delta \le \delta_s(u,v;\mathcal{N})$. Notice that $(u,u,c(s,u)/2)$ is a safe edge-splitting if $|NB(s;\mathcal{N})| = 1$ because any s-proper cut X that separates s and u satisfies $c(X;\mathcal{N}) = c(\tilde{X};\mathcal{N}) + c(s,u) \ge \lambda_s(\mathcal{N}) + c(s,u)$, where $\tilde{X} = V - (X \cup \{s\})$ is an s-proper cut. We will show in the next section that safe weighted edge-splittings at s can be repeated for various u and v in $NB(s;\mathcal{N})$ until the resulting network $\mathcal{N}'$ has no neighbor of s (i.e., all edges (s,u), $u \in V$, have weight $c'(s,u) = 0$). We say that such $\mathcal{N}'$ is obtained by *isolating* s from $\mathcal{N}$. It is known in [4] that such $\mathcal{N}'$ always exists.

However, it is not trivial to show that any designated vertex s can be isolated after finite number of safe weighted edge-splittings. Frank [4] first proved that any vertex s can be isolated by repeating safe weighted edge-splittings at s at most $O(n)$ times. On the other hand, the new algorithm proposed in the next section executes safe weighted edge-splittings at most $|NB(s;\mathcal{N})|$ times, not just $O(|V|)$ times (this fact will be crucial to the time complexity of our final algorithm for enumerating small cuts in section 6).

The next two lemmas describe some properties of the network obtained by isolating s, and s-tight cuts, which will be used to validate the new algorithm in the next section.

LEMMA 4.2. *For a network $\mathcal{N} = (V,E,c)$ with a designated vertex $s \in V$, let $\mathcal{N}'$ be the network obtained from $\mathcal{N}$ by isolating s and let $\mathcal{N}_s = (V - s, E', c')$ be the network obtained from $\mathcal{N}'$ by eliminating s. Then the following properties hold.*

(a) *For every nonempty $X \subset V - s$, $c(X;\mathcal{N}_s) = c(X;\mathcal{N}') \le c(X;\mathcal{N})$.*

(b) $\lambda(\mathcal{N}_s) = \lambda_s(\mathcal{N}') = \lambda_s(\mathcal{N}) \ge \lambda(\mathcal{N})$.

Proof. The proof is immediate from the definition. □

LEMMA 4.3. *For a network $\mathcal{N} = (V,E,c)$ with a designated vertex $s \in V$, the following properties hold.*

(i) *$NB(s;\mathcal{N}) - T \ne \emptyset$ for any s-tight cut T in $\mathcal{N}$.*

(ii) *For two $u, v \in NB(s;\mathcal{N})$, let T' and T be two s-tight cuts in $\mathcal{N}$ such that $\{u,v\} \cap T' = \{u\}$ and $\{u,v\} \cap T = \{u,v\}$. Then $T' \cup \{v\} \subseteq T$ holds.*

Proof. (i) Assume that an s-tight cut T satisfies $NB(s;\mathcal{N}) \subseteq T$. Since T is s-proper, $V - s - T \ne \emptyset$ and hence $R = V - s - T$ is also an s-proper cut. Clearly, $c(T) = c(s,T) + c(R,T)$ and $c(s,T) > 0$, where $c(R,T) = c(R)$ by $NB(s;\mathcal{N}) \cap R = \emptyset$. Therefore, we have $c(T) > c(R)$, contradicting the s-tightness of $c(T)$.

(ii) Assume that $T' \cup \{v\} \not\subseteq T$, i.e., $T' - T \ne \emptyset$. We see that two cuts T' and T cross each other since $u \in T' \cap T \ne \emptyset$, $T' - T \ne \emptyset$, $v \in T - T' \ne \emptyset$ and $s \in V - T - T' \ne \emptyset$

hold. Now $c(T') = c(T) = \lambda_s(\mathcal{N})$, $c(T' - T) \geq \lambda_s(\mathcal{N})$, and $c(T - T') \geq \lambda_s(\mathcal{N})$ since cuts $T' - T$ and $T - T'$ are s-proper. Since $(s, u) \in E(T' \cap T, V - (T' \cup T))$ has a positive weight, $c(T' \cap T, V - (T' \cup T)) > 0$. From (2.1), however,

$$2\lambda_s(\mathcal{N}) = c(T') + c(T) = c(T' - T) + c(T - T') + 2c(T' \cap T, V - (T' \cup T)) > 2\lambda_s(\mathcal{N})$$

is a contradiction. □

5. Algorithm to isolate vertex s.

Based on the properties discussed so far, we will show that the next algorithm isolates s by repeating weighted edge-splitting at s $O(n)$ times.

Procedure ISOLATE

Input: a network $\mathcal{N} = (V, E, c)$ and a designated vertex $s \in V$;
Output: a network $\mathcal{N}_s = (V - s, E', c')$ satisfying Lemma 4.2 and a set Q_s of triplets (u, v, δ) that are used to isolate s;

```
 1 begin
 2   $\mathcal{N}^* := \mathcal{N}$; $T^* := Q_s := \emptyset$;
 3   while $|NB(s;\mathcal{N}^*)| \geq 2$ do
 4     begin
 5       if $T^* \cap NB(s;\mathcal{N}^*) = \emptyset$ then $T^* := \{u\}$ for a $u \in NB(s;\mathcal{N}^*)$ endif;
 6       Choose a $u \in NB(s;\mathcal{N}^*) \cap T^*$ and a $v \in NB(s;\mathcal{N}^*) - T^*$;
 7       Compute $\delta = \delta_s(u, v; \mathcal{N}^*)$;
 8       $\mathcal{N}^* := \mathcal{N}^*/(u, v, \delta)$ (edge-splitting);
 9       $Q_s := Q_s \cup \{(u, v, \delta)\}$;
10       if $\{u, v\} \subseteq NB(s;\mathcal{N}^*)$ then
11         Find an $s$-tight cut $T$ with $\{u, v\} \subseteq T$ in $\mathcal{N}^*$;
12         $T^* := T$
13       endif
14     end;
15   if $|NB(s;\mathcal{N}^*)| = 1$ then
16   $\mathcal{N}^* := \mathcal{N}^*/(u, u, c^*(s, u)/2)$ and $Q_s := Q_s \cup \{(u, u, c^*(s, u)/2)\}$ for the
     $u \in NB(s;\mathcal{N}^*)$
17   endif;
18   Let $\mathcal{N}_s$ be the $\mathcal{N}^*$ from which $s$ is removed
19 end.
```

Theorem 5.1. *Algorithm* ISOLATE *correctly isolates $s \in V$ of a network $\mathcal{N} = (V, E, c)$ after repeating edge-splitting at most $|NB(s;\mathcal{N})|$ times, and runs in $O(|NB(s;\mathcal{N})|n(m + n \log n))$ time, where $n = |V|$ and $m = |E_c|$. Moreover, the resulting network $\mathcal{N}_s$ has no more edges with positive weights than $\mathcal{N}$.*

Proof. To prove the correctness of ISOLATE, we first note that lines 5, 6, and 11 of ISOLATE are always possible to perform (clearly any other lines can be carried out). Since $|NB(s;\mathcal{N}^*)| \geq 2$ holds in the while loop, we can choose a $u \in NB(s;\mathcal{N})$ in line 5. During the while loop, T^* is set to be either a single vertex $u \in NB(s;\mathcal{N}^*)$ or an s-tight cut T in the current network $\mathcal{N}'$. If T^* is a single vertex, line 6 can be clearly performed. Furthermore, Lemma 4.3(i) guarantees that line 6 can be performed if T^* is an s-tight cut in $\mathcal{N}^*$. As to the s-tight cut T in line 11, Lemma 4.1(ii) guarantees that it always exists and can be found in $O(n(m + n \log n))$ time.

Next we show that the while loop terminates after a finite number of iterations. For this, we prove that $|NB(s;\mathcal{N}^*) - T^*|$ decreases at least by 1 after each execution of the while loop, which implies that the while loop of lines 3–14 is repeated at most $|NB(s;\mathcal{N})| - 1$ times (until $|NB(s;\mathcal{N}^*)| < 2$ holds). First note that $|NB(s;\mathcal{N}^*) - T^*|$

decreases by 1 if line 5 is executed. If line 10 holds, we see that the s-tight cut T in line 11 contains the previous T^* and vertex $v \in NB(s; \mathcal{N}^*)$, because Lemma 4.3(ii) applies if T^* is s-tight in the current network and $T^* = \{u\} \subset \{u, v\} \subseteq T$ if T^* consists of a single vertex u. Thus, $|NB(s; \mathcal{N}^*) - T^*|$ decreases at least by 1. If line 10 does not hold, $|NB(s; \mathcal{N}^*)|$ again decreases at least by 1 since one of u and v is no longer a neighbor of s, while T^* remains unchanged. Therefore, $|NB(s; \mathcal{N}^*) - T^*|$ decreases at least by 1 after each execution of the while loop. This proves the correctness.

As shown in the above, the while loop is repeated at most n time. Since lines 7 and 11 can be carried out in $O(n(m+n\log n))$ time by Lemma 4.1(i) and (ii), respectively, and the time for other lines in the while loop is minor, the entire running time is $O(n(m + n\log n))$.

Finally note that $|NB(s; \mathcal{N})|$ edges with positive weights incident to s are removed in the resulting network $\mathcal{N}_s$. Furthermore, since at most $|NB(s; \mathcal{N})|$ edge-splittings, including the one in line 16, are applied in ISOLATE, at most $|NB(s; \mathcal{N})| - 1$ new edges with positive weights are created in $\mathcal{N}_s$. This shows the last statement of the lemma. □

Recently, Gabow [8] developed an $O(n^2 m \log(n^2/m))$ time algorithm for isolating a vertex s independently of us. Our algorithm ISOLATE repeats a modification of the $O(mn + n^2 \log n)$ time minimum cut algorithm of [13] $O(n)$ times, while Gabow's algorithm applies Hao and Orlin's $O(nm\log(n^2/m))$ time minimum cut algorithm $O(n)$ times. Our algorithm provides a slightly better bound, although Gabow's algorithm is also valid for a directed multigraph.

6. Enumerating all small cuts. For a given $\alpha > 0$, let $\mathcal{C}^{<\alpha}(\mathcal{N})$ denote the set of all β-cuts in $\mathcal{N}$ satisfying $\beta < \alpha$. In this case, we do not distinguish cut X from its complement $V - X$. To avoid the duplication of X and $V - X$, therefore, we choose an arbitrary vertex $r \in V$ as a *reference vertex*, and denote by $\mathcal{C}_r^{<\alpha}(\mathcal{N})$ the set of all β-cuts $X \in \mathcal{C}^{<\alpha}(\mathcal{N})$ with $r \notin X$. Note that $|\mathcal{C}^{<\alpha}(\mathcal{N})| = |\mathcal{C}_r^{<\alpha}(\mathcal{N})|$ by definition, and in what follows, we compute $\mathcal{C}_r^{<\alpha}(\mathcal{N})$ to avoid confusion.

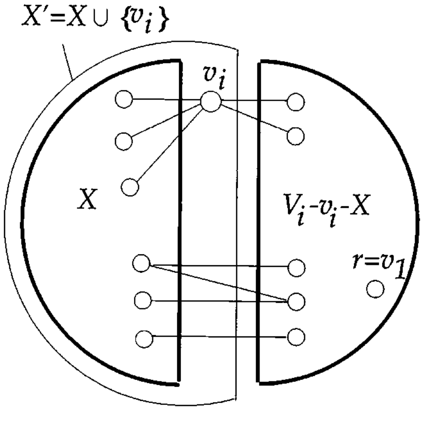


FIG. 3. *Illustration of three cuts $X, Y,$ and Z.*

We first give outline of our algorithm for enumerating small cuts. Now, given a network $\mathcal{N} = (V, E, c)$ and ordered set $V = \{v_1, v_2, \ldots, v_n\}$, define a sequence of networks $\mathcal{N}_i$, $i = n, n-1, \ldots, 2$ as follows. Let $\mathcal{N}_n = \mathcal{N}$, and let $\mathcal{N}_{i-1}$, $i = n, \ldots, 3$, be the network obtained from $\mathcal{N}_i$ by isolating vertex v_i. Set $V_i = \{v_1, v_2, \ldots, v_i\}$ denotes the vertices in $\mathcal{N}_i$. In what follows, we explain a relation between networks

$\mathcal{N}_i$ and $\mathcal{N}_{i-1}$. Any cut X with $\{v_i\} \neq X \neq V_i - v_i$ in $\mathcal{N}_i$ is also a cut in $\mathcal{N}_{i-1}$, and $c(X;\mathcal{N}_i) \geq c(X;\mathcal{N}_{i-1})$ holds by Lemma 4.2(a). Also, note that two cuts X and X' in $\mathcal{N}_i$ such that $v_i \notin X$ and $X' = X \cup \{v_i\}$ becomes the same cut in $\mathcal{N}_{i-1}$ (see Fig. 3).

Choose v_1 as the reference vertex r. From the above observation, we see that any cut $X \in \mathcal{C}_r^{<\alpha}(\mathcal{N}_i)$ appears in exactly one of the three sets: $\mathcal{C}_r^{<\alpha}(\mathcal{N}_{i-1})$, $\{X \cup \{v_i\} \mid X \in \mathcal{C}_r^{<\alpha}(\mathcal{N}_{i-1})\}$, or $\{\{v_i\}\}$.

In other words,

$$\mathcal{C}_{+v_i}[\mathcal{C}_r^{<\alpha}(\mathcal{N}_{i-1})] = \mathcal{C}_r^{<\alpha}(\mathcal{N}_{i-1}) \cup \{X \cup \{v_i\} \mid X \in \mathcal{C}_r^{<\alpha}(\mathcal{N}_{i-1})\} \cup \{\{v_i\}\} \tag{6.1}$$

contains $\mathcal{C}_r^{<\alpha}(\mathcal{N}_i)$ and hence

$$\mathcal{C}_r^{<\alpha}(\mathcal{N}_i) = \{X \in \mathcal{C}_{+v_i}[\mathcal{C}_r^{<\alpha}(\mathcal{N}_{i-1})] \mid c(X;\mathcal{N}_i) < \alpha\}. \tag{6.2}$$

Suppose that we have weights of cuts $\{c(X;\mathcal{N}_{i-1}) \mid X \in \mathcal{C}_r^{<\alpha}(\mathcal{N}_{i-1})\}$ and set Q_{v_i} of triplets (u, v, δ) that are used to isolate v_i in $\mathcal{N}_i$. These are obtained by ISOLATE. For each $Y \in \mathcal{C}_{+v_i}[\mathcal{C}_r^{<\alpha}(\mathcal{N}_{i-1})]$, its weight $c(Y;\mathcal{N}_i)$ can be easily computed from $\mathcal{C}_r^{<\alpha}(\mathcal{N}_{i-1})$ and Q_{v_i} as follows. If $Y = \{v_i\}$, then clearly

$$c(Y;\mathcal{N}_i) = 2 \sum_{(u,v,\delta) \in Q_{v_i}} \delta.$$

If $Y \in \mathcal{C}_r^{<\alpha}(\mathcal{N}_{i-1})$, then we have

$$c(Y;\mathcal{N}_i) = c(Y;\mathcal{N}_{i-1}) + 2 \sum \{\delta \mid (u, v, \delta) \in Q_{v_i} \text{ such that } \{u, v\} \subseteq Y\},$$

since $c(Y;\mathcal{N}_{i-1})$ decreases by 2δ at each edge-splitting $(u, v, \delta) \in Q_{v_i}$ such that Y separates $\{v_i\}$ and $\{u, v\}$, by (4.1). Analogously, if $Y = Y' \cup \{v_i\}$ for a $Y' \in \mathcal{C}_r^{<\alpha}(\mathcal{N}_{i-1})$, then

$$c(Y;\mathcal{N}_i) = c(Y';\mathcal{N}_{i-1}) + 2 \sum \{\delta \mid (u, v, \delta) \in Q_{v_i} \text{ such that } \{u, v\} \cap Y' = \emptyset\}.$$

To compute $c(Y;\mathcal{N}_i)$ efficiently, we use a data structure that enables us to check if $w \in Y$ in $O(1)$ time, e.g., by preparing a membership mapping $f_Y : V \mapsto \{0, 1\}$ with $f_Y(v) = 1$ $(v \in Y)$ and $f_Y(v) = 0$ $(v \in V - Y)$. Then from $|Q_{v_i}| \leq |NB(v_i;\mathcal{N}_i)|$ by Lemma 5.1, each $c(Y;\mathcal{N}_i)$ can be computed in $|Q_{v_i}| = O(|NB(v_i;\mathcal{N}_i)|)$ time by using $\{c(X;\mathcal{N}_{i-1}) \mid X \in \mathcal{C}_r^{<\alpha}(\mathcal{N}_{i-1})\}$, Q_{v_i} and a membership mapping f_Y.

Consequently, all cuts in $\mathcal{C}^{<k\lambda(\mathcal{N})}(\mathcal{N})$ can be enumerated in the following manner.

```
Procedure ENUMERATE
Input: a network $\mathcal{N} = (V, E, c)$ with $V = \{v_1, \ldots, v_n\}$ and a positive real $k > 1$;
Output: $\mathcal{C}^{<k\lambda(\mathcal{N})}(\mathcal{N})$;
begin
  Compute $\lambda(\mathcal{N})$;  $\alpha := k\lambda(\mathcal{N})$;  $\mathcal{N}_n := \mathcal{N}$;
  for $i = n, n-1, \ldots, 2$ do
    begin
      Let $v_i$ be the vertex $v$ in $\mathcal{N}_i$ that minimizes $|NB(v;\mathcal{N}_i)|$;
      Isolate $v_i$ in $\mathcal{N}_i$ by applying procedure ISOLATE;
      Let $Q_{v_i}$ be the set of triplets $(u, v, \delta)$ obtained by ISOLATE;
      Denote the resulting network by $\mathcal{N}_{i-1}$
    end;
  Let $r := v_1$ (reference vertex);
  $\mathcal{C}_r^{<\alpha}(\mathcal{N}_1) := \emptyset$;
```

```
12    for $j = 2, 3, \ldots, n$ do
13      begin
14        $c(\{v_j\}; \mathcal{N}_j) := 2\sum\{\delta \mid (u, v, \delta) \in Q_{v_j}\}$;
15        for each $X \in \mathcal{C}_r^{<\alpha}(\mathcal{N}_{j-1})$ begin
16          $c(X; \mathcal{N}_j) := c(X; \mathcal{N}_{j-1}) + 2\sum\{\delta \mid (u, v, \delta) \in Q_{v_j}$ such that $\{u, v\} \subseteq X\}$;
17          $c(X \cup \{v_j\}; \mathcal{N}_j) := c(X; \mathcal{N}_{j-1}) + 2\sum\{\delta \mid (u, v, \delta) \in Q_{v_j}$ such that $\{u, v\}$
            $\cap X = \emptyset\}$;
18        endfor;
19        $\mathcal{C}_{+v_j}[\mathcal{C}_r^{<\alpha}(\mathcal{N}_{j-1})] := \mathcal{C}_r^{<\alpha}(\mathcal{N}_{j-1}) \cup \{X \cup \{v_j\} \mid X \in \mathcal{C}_r^{<\alpha}(\mathcal{N}_{j-1})\} \cup \{\{v_j\}\}$;
20        $\mathcal{C}_r^{<\alpha}(\mathcal{N}_j) := \{X \in \mathcal{C}_{+v_j}[\mathcal{C}_r^{<\alpha}(\mathcal{N}_{j-1})] \mid c(X; \mathcal{N}_j) < \alpha\}$
21      end;
22    output $\mathcal{C}^{<k\lambda(\mathcal{N})}(\mathcal{N}) := \mathcal{C}_r^{<\alpha}(\mathcal{N}_n)$
23 end.
```

THEOREM 6.1. *For a network $\mathcal{N} = (V, E, c)$ and a real number $k > 1$,* ENUMERATE *computes $\mathcal{C}^{<k\lambda(\mathcal{N})}(\mathcal{N})$ correctly and runs in $O(m^2 n + n^{2k} m)$ time, where $n = |V|$ and $m = |E_c|$.*

Proof. The correctness of ENUMERATE follows from the discussion so far, in particular from (6.1) and (6.2). In line 2, $\lambda(\mathcal{N})$ can be computed in $O(nm + n^2 \log n)$ time [13]. Let m_i denote the number of edges with positive weights in $\mathcal{N}_i$, $i = 2, \ldots, n$. By Theorem 5.1, $m_i \leq m$ holds for all i. First consider how many times the operation of weighted edge-splitting is executed in ISOLATE throughout ENUMERATE. Since each vertex v_i to be isolated minimizes $|NB(v; \mathcal{N}_i)|$ in $\mathcal{N}_i$, we have $|Q_{v_i}| \leq |NB(v_i; \mathcal{N}_i)| \leq \frac{2m_i}{i}$ by Theorem 5.1. Therefore, all $\mathcal{N}_i$s in the first loop of lines 3–9 can be constructed in

$$O\left(\frac{2m}{n} n(m + n\log n) + \frac{2m}{n-1}(n-1)(m + n\log n) + \ldots + \frac{2m}{1}(1)(m + n\log n)\right)$$

$$= O(mn(m + n\log n))$$

time. Now we consider the time required for the second for loop of lines 12–21. By Lemma 4.2(b)

$$\lambda(\mathcal{N}_n) \leq \lambda(\mathcal{N}_{n-1}) \leq \cdots \leq \lambda(\mathcal{N}_2)$$

holds for the networks $\mathcal{N}_i$ obtained by ISOLATE. Updating the membership mapping f_X for a cut X requires $O(|X|) = O(|V|)$ time. It is known [10] that for any network $\mathcal{N}^*$ with i vertices and a real $k \geq 1$ the number of cuts with weights less than $k\lambda(\mathcal{N}^*)$ is $O(i^{2k})$. Hence, $|\mathcal{C}_r^{<\alpha}(\mathcal{N}_i)| \leq |\mathcal{C}_r^{<k\lambda(\mathcal{N}_i)}(\mathcal{N}_i)| = O(i^{2k})$, $i = 2, 3, \ldots, n$. As discussed before the description of ENUMERATE, each $c(X; \mathcal{N}_i)$ for a cut $X \in \mathcal{C}_{+v_i}[\mathcal{C}_r^{<\alpha}(\mathcal{N}_{i-1})]$ can be updated from $c(X; \mathcal{N}_{i-1})$ in $O(|Q_{v_i}|)$ time. Then, updating $\mathcal{C}_r^{<\alpha}(\mathcal{N}_i)$ for all i requires

$$\sum_{i=2}^{n}(|\mathcal{C}_r^{<k\lambda(\mathcal{N}_i)}(\mathcal{N}_i)||Q_{v_i}|) = O\left(\sum_{i=2}^{n}(i^{2k}|Q_{v_i}|)\right) = O\left(\sum_{i=2}^{n}(i^{2k-1}m)\right) = O(n^{2k}m)$$

time. Therefore, the entire running time of ENUMERATE is $O(mn(m + n\log n) + n^{2k}m) = O(m^2 n + n^{2k} m)$. □

7. New bound on the number of small cuts. The objective of this section is to improve the upper bound $O(n^{2k})$ on $|\mathcal{C}^{<k\lambda(\mathcal{N})}(\mathcal{N})|$ obtained by Karger

[10] to $\binom{n}{2}$ for $k \le \frac{4}{3}$. This also improves the time complexity of ENUMERATE to $O(m^2n + mn^2 \log n)$ for such k.

THEOREM 7.1. *For any network* $\mathcal{N} = (V, E, c)$, *it holds* $|\mathcal{C}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})| \le \binom{n}{2}$.

The proof of this theorem will be given in the latter half of this section. Here we note that this bound $\binom{n}{2}$ is also known as a tight upper bound on the number of minimum cuts in $\mathcal{N}$ [1], [3]. In fact, $|\mathcal{C}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})| = \binom{n}{2}$ is actually attained by a ring network consisting of $n \ge 2$ vertices in which $\lambda(\mathcal{N}) = 2$ holds. In this network, there are $\binom{n}{2}$ minimum cuts and all other cuts X satisfy $c(X) \ge 4 > \frac{4}{3}\lambda(\mathcal{N})$. We also see that coefficient $k = \frac{4}{3}$ of $\alpha = \frac{4}{3}\lambda(\mathcal{N})$ is the largest possible for $\binom{n}{2}$ to be an upper bound. For this, consider a complete network K_4 with four vertices, where each edge is weighted 1. Clearly, $\lambda(K_n) = 3$, and K_4 has four 3-cuts and three 4-cuts, indicating $|\mathcal{C}^{<(\frac{4}{3}+\varepsilon)\lambda(K_n)}(K_n)| \ge 7 > \binom{4}{2}(= 6)$ for any $\varepsilon > 0$.

COROLLARY 7.2. *For a network* $\mathcal{N} = (V, E, c)$ *and* $1 < k \le \frac{4}{3}$, $\mathcal{C}^{<k\lambda(\mathcal{N})}(\mathcal{N})$ *can be computed in* $O(m^2n + mn^2 \log n)$ *time.* □

Proof. From the proof of Theorem 6.1, we see that the running time of ENUMERATE is $O(m^2n+mn^2 \log n+\sum_{i=2}^{n}(|\mathcal{C}_r^{<k\lambda(\mathcal{N}_i)}(\mathcal{N}_i)||Q_{v_i}|))$. Therefore, by Theorem 7.1, ENUMERATE runs in $O(m^2n + mn^2 \log n + \sum_{i=2}^{n}(\binom{i}{2}m/i)) = O(m^2n + mn^2 \log n)$ time for $1 < k \le \frac{4}{3}$. □

Now we prove Theorem 7.1 via several lemmas.

For a network $\mathcal{N} = (V, E, c)$ with a designated vertex $s \in V$ and an $\alpha > 0$, if two cuts X and $X' = X \cup \{s\}$ both belong to $\mathcal{C}^{<\alpha}(\mathcal{N})$, then we call $\{X, X'\}$ a pair of *twin cuts* with respect to (s, α). Let $r \in V - s$ be a reference vertex, and define

$$\mathcal{TC}_{r,s}^{<\alpha}(\mathcal{N}) \equiv \Big\{X \;\Big|\; X \subseteq V - \{s, r\}, \text{ and } X, X \cup \{s\} \in \mathcal{C}_r^{<\alpha}(\mathcal{N})\Big\}.$$

From (6.1), we have

$$|\mathcal{C}_r^{<\alpha}(\mathcal{N})| \le |\mathcal{C}_r^{<\alpha}(\mathcal{N}_s)| + |\mathcal{TC}_{r,s}^{<\alpha}(\mathcal{N})| + 1. \tag{7.1}$$

Based on this inequality, we prove Theorem 7.1 by induction on $n = |V|$. The theorem clearly holds for $n = 2$. Assume here that Theorem 7.1 holds for any network with less than n vertices. If we can show

$$|\mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})| \le n - 2, \tag{7.2}$$

then we have from (7.1) that

$$\begin{aligned}
\left|\mathcal{C}_r^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})\right| &\le |\mathcal{C}_r^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N}_s)| + |\mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})| + 1 \\
&\le |\mathcal{C}_r^{<\frac{4}{3}\lambda(\mathcal{N}_s)}(\mathcal{N}_s)| + |\mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})| + 1 \\
&\qquad (\text{from } \lambda(\mathcal{N}_s) \ge \lambda(\mathcal{N}) \text{ (Lemma 4.2(b)))} \\
&\le \binom{n-1}{2} + (n-2) + 1 \quad (\text{induction hypothesis and (7.2)}) \\
&= \binom{n}{2}. \qquad (7.3)
\end{aligned}$$

Therefore, property (7.2) proves Theorem 7.1. The proof of (7.2) will be given in Lemmas 7.3–7.7.

LEMMA 7.3. *For a network* $\mathcal{N} = (V, E, c)$ *with a reference vertex* $r \in V$, *let* X, Y, *and* Z *be three distinct cuts in* $\mathcal{C}_r^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})$ *and define a partition* W_i, W_i' $(i =$

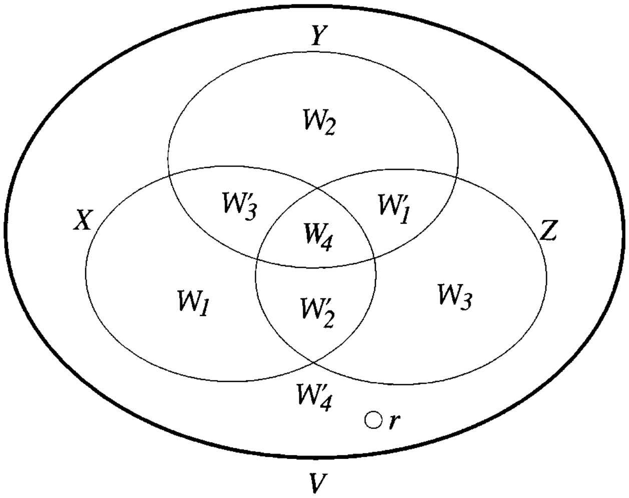


FIG. 4. *Illustration of three cuts X, Y, and Z.*

$1, 2, 3, 4)$ *of V as follows (see Fig.* 4).

$$W_1 = X \cap \overline{Y} \cap \overline{Z},\quad W_2 = \overline{X} \cap Y \cap \overline{Z},\quad W_3 = \overline{X} \cap \overline{Y} \cap Z,\quad W_4 = X \cap Y \cap Z,$$
$$W_1' = \overline{X} \cap Y \cap Z,\quad W_2' = X \cap \overline{Y} \cap Z,\quad W_3' = X \cap Y \cap \overline{Z},\quad W_4' = \overline{X} \cap \overline{Y} \cap \overline{Z}. \tag{7.4}$$

Then at least one of W_1, W_2, W_3, and W_4 is empty, and at least one of W_1', W_2', and W_3' is also empty.

Proof. If none of W_1, W_2, W_3, and W_4 is empty, then we would have

$$\begin{aligned} 3 \times \frac{4}{3}\lambda(\mathcal{N}) &> c(X) + c(Y) + c(Z) \\ &\geq c(W_1) + c(W_2) + c(W_3) + c(W_4) \geq 4\lambda(\mathcal{N}), \end{aligned}$$

which is a contradiction. Analogously, since $r \in W_4' \neq \emptyset$, one of W_1', W_2', and W_3' must be empty. □

LEMMA 7.4. *Let $\mathcal{N} = (V, E, c)$ be a network, and let $s \in V$ and $r \in V - s$. If no two cuts in $\mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})$ cross each other, then there are two disjoint nonempty subsets $X_A, X_B \subset V - \{s, r\}$ such that every cut in $\mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})$ separates X_A and X_B.*

Proof. For a cut $X \in \mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})$, s-proper cut $\tilde{X} = V - s - X$ satisfies

$$\begin{aligned} \frac{4}{3}\lambda(\mathcal{N}) > c(\tilde{X}) &= c(X) - c(\{s\}, X) + c(\{s\}, \tilde{X}) \\ &\leq \lambda(\mathcal{N}) - c(\{s\}, X) + c(\{s\}, \tilde{X},), \end{aligned}$$

from which

$$c(\{s\}, X) - c(\{s\}, \tilde{X}) > -\frac{1}{3}\lambda(\mathcal{N}).$$

From this and $c(\{s\}, X) + c(\{s\}, \tilde{X}) = c(\{s\}) \geq \lambda(\mathcal{N})$, we obtain

$$c(\{s\}, X) > \frac{1}{3}c(\{s\}). \tag{7.5}$$

Take $X_A \in \mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})$ with the smallest cardinality, i.e.,

$$|X_A| = \min\left\{|X| \mid X \in \mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})\right\}.$$

For any other cut $X \in \mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})$, X and X_A do not cross each other by the assumption. Then, from the minimality of $|X_A|$, we see

$$X \supseteq X_A \text{ or } X \cap X_A = \emptyset \quad \text{for any } X \in \mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N}).$$

Define $\mathcal{TC}_A = \{X \in \mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N}) \mid X \supseteq X_A\}$ and $\mathcal{TC}_{\overline{A}} = \{X \in \mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N}) \mid X \cap X_A = \emptyset\}$. We next choose $X_B \in \mathcal{TC}_{\overline{A}}$ such that

$$|X_B| = \min\{|X| \mid X \in \mathcal{TC}_{\overline{A}}\}.$$

Then, analogously to the above, we can show that

$$X \supseteq X_B \text{ or } X \cap X_B = \emptyset \quad \text{for any } X \in \mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N}).$$

In particular, this implies $X_A \cap X_B = \emptyset$.

Finally, we see that there is no cut $Y \in \mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})$ such that $Y \subseteq V - X_A - X_B$, because, otherwise, $c(\{s\}, Y) + c(\{s\}, X_A) + c(\{s\}, X_B) > c(\{s\})$ would follow from (7.5), which is a contradiction. Therefore, any cut $X \in \mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})$ must separate X_A and X_B. It is also clear that $X_A \cup X_B \subseteq V - \{s, r\}$. □

If two cuts X_i and X_j do not cross each other and both separate X_A and X_B, then (i) $X_i \subseteq X_j$ or $X_i \supseteq X_j$ or (ii) $X_i \cap X_j = \emptyset$, $X_A \subseteq X_i$, and $X_B \subseteq X_j$ (or $X_i \cap X_j = \emptyset$, $X_B \subseteq X_i$, and $X_A \subseteq X_j$).

LEMMA 7.5. *If no two cuts in* $\mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})$ *cross each other, then* $|\mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})| \leq n - 2$.

Proof. Recall that $X \subseteq V - \{s, r\}$ for $X \in \mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})$ from the definition of $\mathcal{TC}_r$. In this case, by Lemma 7.4, any cut X in $\mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})$ satisfies $X_A \subseteq X \subseteq V - s - X_B$ or $X_B \subseteq X \subseteq V - s - X_A$. Therefore, all cuts in $\mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}\mathcal{N})$ can be numbered as $X_1, X_2, \ldots, X_p$ so that

$$X_A \subseteq X_1 \subset X_2 \subset \cdots \subset X_k, \quad X_B \subseteq X_p \subset X_{p-1} \subset \cdots \subset X_{k+1},$$

where $X_k \cap X_{k+1} = \emptyset$ and $\{s, r\} \subseteq V - (X_k \cup X_{k+1})$ hold. From this, we see that $p \leq n - 2$. □

LEMMA 7.6. *For a network* $\mathcal{N} = (V, E, c)$, *let* $s \in V$ *and* $r \in V - s$ *and let* X, Y, *and* Z *be three cuts in* $\mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})$. *Define* W_i *and* W'_i $(i = 1, 2, 3, 4)$ *as in* (7.4). *Then at least two of* W_1, W_2, W_3, *and* W_4 *are empty, and at least two of* W'_1, W'_2 *and* W'_3, *are also empty.*

Proof. Since no cut in $\mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})$ contains r, it holds $r \in W'_4 - s$. By Lemma 7.3, one of $W_i (i = 1, 2, 3, 4)$, say W_1, is empty (other cases can be treated analogously). Then consider $X' = X \cup \{s\} \in \mathcal{C}_r^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})$, where X and X' are a pair of twin cuts. Again by applying Lemma 7.3 to three cuts X', Y and Z, we see that one of $W_1 \cup \{s\}, W_2, W_3$, and W_4 must be empty. Since $W_1 \cup \{s\} \neq \emptyset$, one of W_2, W_3, and W_4 is also empty. Similar argument also proves the result for subsets W'_i $(i = 1, 2, 3)$. □

LEMMA 7.7. *If there are two cuts in* $\mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})$ *which cross each other, then* $|\mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})| = 2$.

Proof. Suppose that two cuts X and Y in $\mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})$ cross each other. In this case, if $\mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})$ contains other cut (say Z), then Lemma 7.6 applies to W_i and W_i' $(i = 1, 2, 3, 4)$. However, this is impossible since two crossing cuts X and Y have already produced four nonempty subsets among W_i's and W_i''s. Hence, the third cut Z cannot exist. Thus, $|\mathcal{TC}_{r,s}^{<\frac{4}{3}\lambda(\mathcal{N})}(\mathcal{N})| = 2$ follows. □

Lemmas 7.5 and 7.7 prove property (7.2), which also completes the proof of Theorem 7.1.

Acknowledgments. The authors are grateful to Professor A. Frank of Eötvös University for his valuable comments, based on which Lemma 3.1 is established.

REFERENCES

[1] R. BIXBY, *The minimum number of edges and vertices in a graph with edge-connectivity n and m n-bonds*, Networks, 5 (1975), pp. 235–298.

[2] C. J. COLBOURN, *The Combinatorics of Network Reliability*, Oxford University Press, New York Oxford, 1987.

[3] E. DINITS, A. V. KARZANOV, AND M. V. LOMONOSOV, *On the structure of a family of minimal weighed cuts in a graph*, in Studies in Discrete Optimization, A. A. Fridman, ed., Nauka, Moscow, 1976, pp. 290–306 (in Russian).

[4] A. FRANK, *Augmenting graphs to meet edge-connectivity requirements*, SIAM J. Discrete Math., 5 (1992), pp. 25–53.

[5] A. FRANK, *On a theorem of Mader*, Discrete Math., 101 (1992), pp. 49–57.

[6] A. FRANK, *On the Edge-Connectivity Algorithm of Nagamochi and Ibaraki*, Laboratoire Artemis, IMAG, Université J. Fourier, Grenoble, March, 1994.

[7] S. FUJISHIGE, *A note on Nagamochi and Ibaraki's min-cut algorithm and its simple proofs by Stoer, Wagner and Frank*, manuscript, Forschungsinstitut für Diskrete Mathematik, Universität Bonn, June, 1994.

[8] H. N. GABOW, *Efficient splitting off algorithms for graphs*, in Proceedings of the 26th ACM Symposium on Theory of Computing, Montreal, Quebec, 1994, pp. 696–705.

[9] M. R. HENZINGER AND D. WILLIAMSON, *On the number of small cuts*, Inform. Process. Lett., 59 (1996), pp. 41–44.

[10] D. R. KARGER, *Global min-cuts in RNC, and other ramifications of a simple min-cut algorithm*, in Proceedings of the 4th ACM-SIAM Symposium on Discrete Algorithms, Austin, TX, 1993, pp. 21–30.

[11] L. LOVÁSZ, *Combinatorial Problems and Exercises*, North-Holland, Amsterdam, 1979.

[12] W. MADER, *A reduction method for edge-connectivity in graphs*, Ann. Discrete Math., 3 (1978), pp. 145–164.

[13] H. NAGAMOCHI AND T. IBARAKI, *Computing the edge-connectivity of multigraphs and capacitated graphs*, SIAM J. Discrete Math., 5 (1992), pp. 54–66.

[14] H. NAGAMOCHI, T. ISHII, AND T. IBARAKI, *A Simple and Constructive Proof of a Minimum Cut Algorithm*, Technical Report 96001, Department of Applied Mathematics and Physics, Kyoto University, 1996.

[15] D. NAOR, D. GUSFIELD, AND C. MARTEL, *A fast algorithm for optimally increasing the edge connectivity*, in Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science, St. Louis, MO, 1990, pp. 698–707.

[16] M. STOER AND F. WAGNER, *A simple min cut algorithm*, Lecture Notes in Comput. Sci., 855 (1994), pp. 141–147.

[17] V. V. VAZIRANI AND M. YANNAKAKIS, *Suboptimal cuts: Their enumeration, weight, and number*, Lecture Notes in Comput. Sci., 623 (1992), pp. 366–377.

SIAM J. Discrete Math.
Vol. 10, No. 3, pp. 482–498, August 1997

A THRESHOLD FUNCTION FOR HARMONIC UPDATE[*]

SHAO C. FANG[†] AND SANTOSH S. VENKATESH[†]

Abstract. Harmonic update is a randomized on-line algorithm which, given a random m-set of vertices $U(m) \subseteq \{-1,1\}^n$ in the n-dimensional cube, generates a random vertex $\mathbf{w} \in \{-1,1\}^n$ as a putative solution to the system of linear inequalities: $\sum_{i=1}^n w_i u_i \geq 0$ for each $\mathbf{u} \in U(m)$. Using tools from large deviation multivariate normal approximation and Poisson approximation, we show that $\sqrt{n}\big/\sqrt{\log n}$ is a threshold function for the property that the vertex $\mathbf{w}$ generated by harmonic update has positive inner product with each vertex in $U(m)$. More explicitly, let $P(n,m)$ denote the probability that $\sum_{i=1}^n w_i u_i \geq 0$ for each $\mathbf{u} \in U(m)$. Then, as $n \to \infty$, $P(n,m) \to 0$ or 1 according to whether $m = m_n$ varies with n such that $m \gg \sqrt{n}\big/\sqrt{\log n}$ or $m \ll \sqrt{n}\big/\sqrt{\log n}$, respectively. The analysis also exposes the fine structure of the threshold function.

Key words. polytopes, threshold function, randomized algorithm, harmonic update, binary integer programming, neural networks, large deviations, normal approximation, Poisson approximation

AMS subject classifications. 05C80, 60C05, 52B11, 60G85, 68R05

PII. S0895480195283701

1. Information and finite memory.

How much information can a single bit of memory updated on-line retain about a Bernoulli sequence? More specifically, the following problem was posed by J. Komlós. Write $\mathbb{B} \triangleq \{-1,1\}$ and let $\{\, u^{(t)}, t \geq 1 \,\}$ be a sequence of symmetric Bernoulli trials, where

$$u^{(t)} = \begin{cases} -1 & \text{with probability } 1/2, \\ +1 & \text{with probability } 1/2. \end{cases}$$

Suppose a single bit of memory $w \in \mathbb{B}$ is available to record this sequence; write $w^{(t)} \in \mathbb{B}$ for the state of the one-bit memory at epoch t (with $w^{(1)} \in \mathbb{B}$ being an arbitrary initial state of memory). We suppose that input bits $u^{(t)}$ arrive sequentially in time and memory updates $\big(w^{(t)}, u^{(t)}\big) \mapsto w^{(t+1)}$ proceed on-line governed by a sequence $\{\, f^{(t)} \colon \mathbb{B} \times \mathbb{B} \to \mathbb{B} \;\big|\; t \geq 1 \,\}$ of (possibly random) Boolean functions of two Boolean variables: $w^{(t+1)} = f^{(t)}\big(w^{(t)}, u^{(t)}\big)$. After m epochs, input bits $u^{(1)}, \ldots, u^{(m)}$ have been presented sequentially in time leading to the current state of memory $w^{(m+1)} \in \mathbb{B}$, which now constitutes the sole "record" (insofar as a single bit may be said to constitute a record) of the entire past, i.e., the sequence of bits $u^{(1)}, \ldots, u^{(m)}$. One measure of the efficacy of the update sequence $\{f^{(t)}\}$ in storing information up to this moment in the one-bit memory is the minimum covariance $\min_{1 \leq t \leq m} \mathbb{E}\big(w^{(m+1)} u^{(t)}\big)$: a minimum covariance of zero implies that there is at least one bit in the past about which the one-bit memory carries no information; a positive minimum covariance, on the other hand, indicates that the single bit of memory carries information about every one of the inputs in the past. In this context, Komlós posed the following

[*] Received by the editors March 24, 1995; accepted for publication (in revised form) June 25, 1996. This research was supported in part by Air Force Office of Scientific Research grants F49620-93-1-0120 and F49620-92-J-0344. A preliminary version of this paper was presented in part at the *Fourth ACM Conference on Computational Learning Theory*, Santa Cruz, CA, 1991.

http://www.siam.org/journals/sidma/10-3/28370.html

[†] Department of Electrical Engineering, University of Pennsylvania, Philadelphia, PA 19104 (fang@ee.upenn.edu, venkatesh@ee.upenn.edu).

problem: what is $I_m \triangleq \max_{f^{(1)},\dots,f^{(m)}} \min_{1\le t\le m} \mathbb{E}\bigl(w^{(m+1)}u^{(t)}\bigr)$? The quantity I_m may be taken as an intrinsic measure of the amount of information that a single bit of memory updated on-line can retain about *each* past input.

In [12], Venkatesh and Franklin provide comprehensive answers to this and related questions. In particular, they show that

$$\tfrac{1}{m} \le I_m < \tfrac{2}{m},$$

whence $I_m = \Theta\bigl(m^{-1}\bigr)$. Their results also directly imply that any sequence of deterministic update rules $\bigl\{f^{(t)}\bigr\}$ yields exponentially small (in m) minimum covariances at best and hence that the optimal sequence of update rules $\bigl\{f^{(t)}_{\mathrm{opt}}\bigr\}$ is necessarily randomized.

Consider an application of this notion of on-line information storage to the following classical problem in mathematical programming. Let $\mathbb{B}^n = \{-1,1\}^n$ denote the vertices of a cube in n dimensions and let $U(m) = \bigl\{\, \mathbf{u}^{(t)}, 1 \le t \le m \bigr\}$ be a random m-set of vertices in $\mathbb{B}^n$ obtained by independent sampling from the uniform distribution on $\mathbb{B}^n$. Write $\mathbf{u}^{(t)} = \bigl(u_1^{(t)},\dots,u_n^{(t)}\bigr)$. Does there exist a vertex $\mathbf{w} = (w_1,\dots,w_n) \in \mathbb{B}^n$ for which the inequalities

$$\begin{array}{c} w_1u_1^{(1)} + w_2u_2^{(1)} + \cdots + w_nu_n^{(1)} \ge 0 \\ w_1u_1^{(2)} + w_2u_2^{(2)} + \cdots + w_nu_n^{(2)} \ge 0 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ w_1u_1^{(m)} + w_2u_2^{(m)} + \cdots + w_nu_n^{(m)} \ge 0 \end{array} \tag{1.1}$$

are simultaneously satisfied? Let $\langle\cdot,\cdot\rangle$ denote the usual inner product in Euclidean n-space $\mathbb{R}^n$. Then each point $\mathbf{w}$ in $\mathbb{R}^n$ determines a *positive half-space* defined by $H^+(\mathbf{w}) = \{\, \mathbf{u} \in \mathbb{R}^n : \langle \mathbf{w},\mathbf{u}\rangle \ge 0 \,\}$. Geometrically speaking, our question is equivalent to asking whether there exists a vertex $\mathbf{w} \in \mathbb{B}^n$ such that the convex hull of $U(m)$ is contained in the positive half-space determined by $\mathbf{w}$.[1]

If $\mathbf{w}$ is allowed to range over $\mathbb{R}^n \setminus \{\mathbf{0}\}$, the problem is just an instance of linear programming and a solution to the system of inequalities (1.1), if one exists, can be found in polynomial time using interior point methods (cf. Karmarkar [6]). When, however, $\mathbf{w}$ is restricted to the vertices of the n-dimensional cube, the decision problem becomes NP-complete as an instance of binary integer programming (cf. Garey and Johnson [5], Pitt and Valiant [7]).

Consider an on-line programming scenario in which the random examples $\mathbf{u}^{(t)}$ comprising $U(m)$ arrive in sequence at epochs $t = 1,\dots,m$. In the information storage analogy, we are provided with n bits of memory and access to a sequence of memory update rules $\bigl\{\, f^{(t)} \colon \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}^n \bigm| t \ge 1 \,\bigr\}$. Starting from an arbitrary initial memory state $\mathbf{w}^{(1)} \in \mathbb{B}^n$, we then recursively generate memory states $\mathbf{w}^{(t+1)} = f^{(t)}\bigl(\mathbf{w}^{(t)},\mathbf{u}^{(t)}\bigr)$ for $t \ge 1$. The on-line procedure is successful if, after presentation of the mth example $\mathbf{u}^{(m)}$, the vertex ("memory state") $\mathbf{w} \triangleq \mathbf{w}^{(m+1)}$ generated by the algorithm satisfies the system of linear inequalities (1.1).

In [11], a sequence of randomized update rules $\bigl\{f^{(t)}\bigr\}$, dubbed *harmonic update*, is constructed starting from the following trivial observations: (i) for each t, the sum $\sum_{i=1}^n w_iu_i^{(t)}$ is more likely to be positive if the individual summands $w_iu_i^{(t)}$ are likely

[1] The above mathematical programming problem can also be formulated as a learning problem in a formal model of a neuron or perceptron (cf. Fang and Venkatesh [1]).

to be positive, i.e., if the random element $w_i \triangleq w_i^{(m+1)}$ is positively correlated with $u_i^{(t)}$, and (ii) the *smallest* of the inner products $\langle \mathbf{w}, \mathbf{u}^{(t)} \rangle$ must be nonnegative if (1.1) is to hold. Looking at the ith column ($1 \le i \le n$) in (1.1), these considerations suggest that the update rules be chosen so that $\min_{1\le t\le m} \mathbb{E}\bigl(w_i u_i^{(t)}\bigr)$ is maximized. The optimal update rules for Komlós's problem would do very nicely here if they could be determined explicitly; applying the optimal one-bit memory update rules n times, once for every component, will maximize the smallest expectation of the summands in each column, as desired. Harmonic update uses auxiliary randomization in an attempt to achieve this desideratum.

Harmonic update. Given examples $\mathbf{u}^{(t)} = \bigl(u_1^{(t)}, \dots, u_n^{(t)}\bigr) \in \mathbb{B}^n$ presented sequentially at epochs $t = 1, \dots, m$, the algorithm recursively generates a sequence of n-bit memory states $\mathbf{w}^{(t+1)} = \bigl(w_1^{(t+1)}, \dots, w_n^{(t+1)}\bigr) \in \mathbb{B}^n$, where $\mathbf{w}^{(t+1)}$ is a random function of $\mathbf{w}^{(t)}$ and $\mathbf{u}^{(t)}$ only. After m epochs, the algorithm returns the final n-bit memory state $\mathbf{w} \triangleq \mathbf{w}^{(m+1)}$ as a putative vertex solution to the system of inequalities (1.1).

H1. [Initialize.] Set $\mathbf{w}^{(1)} = \bigl(w_1^{(1)}, \dots, w_n^{(1)}\bigr)$ to be an arbitrary vertex in $\mathbb{B}^n$. Set $t \leftarrow 1$.

H2. [New example.] Obtain example $\mathbf{u}^{(t)}$.

H3. [Reinitialize component index.] Set $i \leftarrow 1$.

H4. [Update memory components.] If $w_i^{(t)} = u_i^{(t)}$, set $w_i^{(t+1)} = w_i^{(t)}$; else if $w_i^{(t)} = -u_i^{(t)}$, set

$$w_i^{(t+1)} = \begin{cases} -w_i^{(t)} & \text{with probability } 1/t, \\ +w_i^{(t)} & \text{with probability } 1 - 1/t. \end{cases}$$

H5. [Iterate.] Set $i \leftarrow i + 1$. If $i \le n$, go back to step H4; otherwise set $t \leftarrow t + 1$. If $t \le m$ go back to step H2; otherwise set $\mathbf{w} = \mathbf{w}^{(m+1)}$ and terminate the algorithm.

Remarks.

Computation. While the actual memory updates in step H4 are done in place, it is convenient to keep the notation $\mathbf{w}^{(t)}$ to identify the state of the memory at epoch t for purposes of later analysis.

Probability space. It is assumed implicitly that the auxiliary randomization in step H4 is independent across i and t; in particular, we can assume that a biased coin with the appropriate success probability is tossed independently each time step H4 is encountered.

The intuition behind the algorithm is as follows: at epoch t, the current state $w_i^{(t)}$ of the ith memory component presumably contains information about the ith components of the first $t-1$ examples. No problem arises in updating the state of the ith bit of memory if the ith component $u_i^{(t)}$ of the current example has the same sign as the current state $w_i^{(t)}$ of the ith memory component; setting $w_i^{(t+1)} = w_i^{(t)}$ adds $u_i^{(t)}$ to the knowledge base at no cost to the previously stored components. Complications arise, however, if $w_i^{(t)}$ and $u_i^{(t)}$ have opposite signs. In this case, retaining the sign of $w_i^{(t)}$ results in all information about $u_i^{(t)}$ being irrevocably lost; conversely, changing the sign of $w_i^{(t)}$ results in a loss of information about $u_i^{(1)}, \dots, u_i^{(t-1)}$. The solution in this case is to change the sign of the ith bit of memory probabilistically—and with increasing reluctance as time passes (when there is presumably considerable past history stored in the bit of memory). The exact measure of this reluctance to change the sign of the memory bit with increasing time is given probabilistically by the harmonic sequence[2] $1/t$. The effect of this randomized update rule is to ensure that each component of memory retains an equal amount of information about the corresponding component of every example.

As m increases, the probability that there exists any solution for the system of inequalities (1.1) decreases monotonically, and for large enough m the random vertex set $U(m)$ will fail to be linearly separable (in the sense that there is no solution for (1.1)) with high probability. In what follows we allow $m = m_n$ to depend implicitly on the dimensionality n. Our goal is to determine the "largest" rate of increase of m with n for which the system of inequalities (1.1) is satisfied with asymptotically high probability as $n \to \infty$. In the language of random graphs (cf. Spencer [10]), we wish to determine a threshold function for the property that (1.1) is satisfied.

Indeed, Füredi [4] showed that $2n$ is a threshold function for the property that there exists a *real* vector $\mathbf{w} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ satisfying (1.1). (Equivalently, $2n$ is a threshold function for the property that the convex hull of the random vertex set $U(m)$ contains the origin.) When restricted to vertex solutions for (1.1), we may expect the probability that there exists a solution $\mathbf{w} \in \mathbb{B}^n$ satisfying (1.1) to decay rather faster with m. In fact, a trite application of Boole's inequality readily establishes n as an upper bound for the rate of growth of m for *any* algorithm if we are to hold out hopes for a solution in $\mathbb{B}^n$ for (1.1). Informally, if m grows faster than n, then (1.1) admits no solution in $\mathbb{B}^n$ with probability $1 - \mathfrak{o}(1)$. Much sharper results can be shown for harmonic update, and the following theorem, which is our main result, exposes the fine structure of a threshold function for the algorithm.

Let $\mathbf{w} \in \mathbb{B}^n$ be the vertex generated by harmonic update and write $H^-(\mathbf{w}) = \{\, \mathbf{u} \in \mathbb{R}^n : \langle \mathbf{w}, \mathbf{u} \rangle < 0 \,\}$ for the negative half-space determined by $\mathbf{w}$. Let $Z = Z_{n,m} \triangleq \bigl|U(m) \cap H^-(\mathbf{w})\bigr|$ denote the number of $\mathbf{u}^{(t)}$ $(1 \le t \le m)$ that fall into the negative half-space determined by the vector $\mathbf{w}$ generated by harmonic update. Our main theorem shows that for a suitable rate of growth of $m = m_n$ with n, the random variable Z_{n,m_n} has a limiting Poisson distribution as $n \to \infty$. A sharp threshold for the event of interest $\{Z_{n,m} = 0\}$ that $\mathbf{w}$ has positive inner product with each vertex in the random m-set $U(m)$ follows immediately.

Main Theorem. *Let $\lambda > 0$ be any fixed positive number and suppose that $m = m_n$ grows with n such that*

$$m_n = \sqrt{\frac{n}{\log n}} \left\{ 1 + \frac{\log\log n + \log\bigl(\lambda\sqrt{2\pi}\,\bigr)}{\log n} + \mathcal{O}\biggl(\frac{\log\log n}{(\log n)^2}\biggr) \right\}. \tag{1.2}$$

Then Z_{n,m_n} tends in distribution to $\mathrm{Po}(\lambda)$*, the Poisson distribution with parameter λ, as $n \to \infty$. In particular, for each fixed k,*

$$\mathbb{P}\{Z_{n,m_n} = k\} \to \frac{\lambda^k}{k!}\, e^{-\lambda}$$

as $n \to \infty$.

Corollary. *Write $P(n,m) = \mathbb{P}\{Z_{n,m} = 0\}$ for the probability that the vertex $\mathbf{w} \in \mathbb{B}^n$ generated by harmonic update satisfies the system of inequalities* (1.1). *Then, with $m = m_n$ as in* (1.2), *$P(n,m) \to e^{-\lambda}$ as $n \to \infty$. In particular, for every fixed $\epsilon > 0$, the following assertions hold:*

[2] Hence we have the name harmonic update.

(a) *if m varies with n such that $m \le (1-\epsilon)\sqrt{\frac{n}{\log n}}$ then $P(n,m) \to 1$ as $n \to \infty$,*

(b) *if m varies with n such that $m \ge (1+\epsilon)\sqrt{\frac{n}{\log n}}$ then $P(n,m) \to 0$ as $n \to \infty$.*

In other words, $\sqrt{n}\big/\sqrt{\log n}$ is a threshold function for the attribute that the binary vector $\mathbf{w}$ generated by harmonic update satisfies the system of linear inequalities (1.1).

Remark. The Main Theorem provides a lower bound for the rate of growth of m with n for which a vertex solution exists for the system of inequalities (1.1). At this point, it is natural to wonder if the gap between the lower bound $\sqrt{n}\big/\sqrt{\log n}$ and the upper bound n can be reduced by increasing the computational complexity of the memory update rule. Indeed, substantial improvements in information storage can accrue if off-line procedures are permitted or the examples are recycled infinitely often in an on-line scenario [11, 12]. For instance, the majority rule algorithm introduced in [11] is an off-line procedure which, given the random m-set of examples $U(m)$, selects a vertex closest to the centroid of $U(m)$ as a putative solution to (1.1). In a companion exposition [2], we established a threshold function for majority rule at $\frac{n}{\pi \log n}$.

The rest of the paper is devoted to a proof of the Main Theorem. The main technical tools used in the proof are multivariate normal approximation and Poisson approximation, the former via a multivariate integral limit theorem for large deviations and the latter in the form of a probabilistic sieve. These technical results are collected in the next section for ease of later reference. The proof follows.

2. Preliminaries.

Notation. As already indicated, we use $\mathbb{B}$ to denote the set $\{-1, 1\}$, with $\mathbb{B}^n = \{-1,1\}^n$ the vertices of the cube in n dimensions. The set of all integers is denoted by $\mathbb{Z}$, with $\mathbb{Z}^n$ denoting the corresponding set of lattice points in n dimensions. Also, we denote the real line by $\mathbb{R}$, with $\mathbb{R}^n$ denoting n-dimensional Euclidean space equipped with the usual inner product $\langle \mathbf{x}, \mathbf{y}\rangle = \sum_{i=1}^n x_i y_i$ and the induced norm $\|\mathbf{x}\| = \langle \mathbf{x}, \mathbf{x}\rangle^{1/2}$. If $\mathbf{x} = (x_1, \dots, x_n)$ and $\mathbf{y} = (y_1, \dots, y_n)$ are points in $\mathbb{R}^n$, the vector inequality $\mathbf{x} \ge \mathbf{y}$ means that the scalar inequalities $x_i \ge y_i$ hold for each $i = 1, \dots, n$; likewise, the vector inequality $\mathbf{x} > \mathbf{y}$ means that the corresponding componentwise inequalities are strict: $x_i > y_i$ $(1 \le i \le n)$. We write $\mathbf{0}$ for the vector $(0, \dots, 0)$ with all components identically 0 and $\mathbf{1}$ for the vector $(1, \dots, 1)$ with all components identically 1. Superscripts r, s, and t and subscripts i and j are employed exclusively to index vectors and their components, respectively.

For purposes of definiteness in vector–matrix operations we assume that all vectors are *row* vectors; a prime ($'$) denotes vector and matrix transpose. We also reuse the notation $|V|$ to denote the determinant of a square matrix V as well as the cardinality of a set V. The usage will be clear from the context.

Throughout, $\mathbb{P}$ stands for probability measure on the underlying probability space, $\mathbb{E}$ denotes expectation, Var denotes variance, and Cov denotes covariance. For any integer $k \ge 1$, if $\mathbf{X} = (X_1, \dots, X_k)$ is a Gaussian (row) vector with zero mean, $\mathbb{E}\,\mathbf{X} = \mathbf{0}$, and nondegenerate covariance matrix $K = \operatorname{Cov}(\mathbf{X}) = \mathbb{E}\big(\mathbf{X}'\mathbf{X}\big)$, $|K| > 0$, we write

$$\phi_K(\mathbf{x}) \triangleq \frac{1}{(2\pi)^{k/2}|K|^{1/2}}\, e^{-\frac{1}{2}\mathbf{x}\,K^{-1}\mathbf{x}'}$$

for the multivariate Gaussian density, and likewise

$$\Phi_K(\mathbf{x}) \triangleq \int_{\mathbf{u} \le \mathbf{x}} \phi_K(\mathbf{u})\, d\mathbf{u}$$

for the multivariate Gaussian distribution function. For the univariate case we simply write $\phi(x)$ and $\Phi(x)$ for the standard $\mathcal{N}(0,1)$ Gaussian density and distribution, respectively.

All logarithms are to base e.

We use standard asymptotic order notation with the following caveats: if $\{h_n\}$ and $\{g_n\}$ denote real sequences, by $h_n = \mathcal{O}(g_n)$ we mean that $|h_n|/|g_n|$ is bounded above; in particular, sign information is explicitly jettisoned in our use of the "big-oh" notation. In addition, we will find it expedient to occasionally use the more graphic $h_n \ll g_n$ and $h_n \gg g_n$ to mean $h_n = \mathfrak{o}(g_n)$ and $h_n = \omega(g_n)$, respectively.

In what follows we will be concerned with asymptotics as $n \to \infty$, and we will allow the number of elements m in the random vertex set $U(m)$ to depend on n. As a notational convention, however, we shall frequently write simply m instead of the more explicit m_n, while keeping in mind that m is to be thought of as a function of n.

Technical lemmas. The method of proof of the Main Theorem is to reduce the problem to the study of a random walk $\mathbf{S}_n$ where, for each n, $\mathbf{S}_n$ is the row sum of a triangular array of lattice variables. The principal result that will be needed is a sharp estimate for the probability of large deviations of the walk $\mathbf{S}_n$. The setup is as follows.

Let k be a fixed positive integer and consider a triangular array of k-dimensional lattice random vectors

$$\mathbf{X}_{ni} = \bigl(X_{ni}^{(1)}, \dots, X_{ni}^{(k)}\bigr) \qquad (i = 1, \dots, n;\ n = 1, 2, \dots),$$

where, for each n, the random vectors $\mathbf{X}_{n1}, \dots, \mathbf{X}_{nn}$ comprising the nth row of the array are independent, identically distributed lattice random vectors with probability one support in $\{0,1\}^k$ and with common distribution $p_n(\mathbf{x}) = \mathbb{P}\bigl\{\mathbf{X}_{ni} = \mathbf{x}\bigr\}$, where $p_n(\mathbf{0}) > 0$ and $p_n(\mathbf{e}_\nu) > 0$ for each of the canonical unit vectors $\mathbf{e}_\nu \in \{0,1\}^k$ $(1 \leq \nu \leq k)$.[3] Observe that the distribution of $\mathbf{X}_{ni}$ has minimal lattice $\mathbb{Z}^k$. Write $\boldsymbol{\mu}_n = \bigl(\mu_n^{(1)}, \dots, \mu_n^{(k)}\bigr) \triangleq \mathbb{E}\bigl(\mathbf{X}_{ni}\bigr)$ for the mean vector and $V_n \triangleq \operatorname{Cov}\bigl(\mathbf{X}_{ni}\bigr) = \mathbb{E}\bigl(\mathbf{X}_{ni}'\mathbf{X}_{ni}\bigr) - \boldsymbol{\mu}_n'\boldsymbol{\mu}_n$ for the covariance matrix.

Specializing to the case of interest, we assume that there exists a discrete probability distribution $p(\mathbf{x})$ with probability one support in the vertices of the cube $\mathbf{x} \in \{0,1\}^k$ such that $p_n(\mathbf{x}) \to p(\mathbf{x})$ as $n \to \infty$ for each $\mathbf{x} \in \{0,1\}^k$. We suppose that, for each n,

$$\operatorname{Cov}\bigl(X_{ni}^{(t)}, X_{ni}^{(s)}\bigr) = \mathbb{E}\bigl\{\bigl(X_{ni}^{(t)} - \mu_n^{(t)}\bigr)\bigl(X_{ni}^{(s)} - \mu_n^{(s)}\bigr)\bigr\} = \begin{cases} \sigma_n^2 & \text{if } t = s, \\ \psi_n & \text{if } t \neq s, \end{cases}$$

whence the covariance matrix of $\mathbf{X}_{ni}$ is of the form

$$V_n = \operatorname{Cov}\bigl(\mathbf{X}_{ni}\bigr) = \begin{bmatrix} \sigma_n^2 & \psi_n & \psi_n & \dots & \psi_n \\ \psi_n & \sigma_n^2 & \psi_n & \dots & \psi_n \\ \multicolumn{5}{c}{\dotfill} \\ \psi_n & \psi_n & \psi_n & \dots & \sigma_n^2 \end{bmatrix}.$$

We will suppose that, as $n \to \infty$, $\psi_n \to 0$ and $\sigma_n^2 \to \sigma^2$ for some positive constant σ^2. In particular, observe that V_n is nonsingular for sufficiently large n and $|V_n| \to \sigma^{2k}$ as $n \to \infty$.

[3] For $k > 1$ we can dispense with the condition $p_n(\mathbf{0}) > 0$.

For each n, form the lattice random vector $\mathbf{S}_n$ as the row sum

$$\mathbf{S}_n = \sum_{i=1}^{n} \mathbf{X}_{ni}.$$

We are interested in the tails of the random walk $\mathbf{S}_n$ in $\mathbb{Z}^k$. Write

$$\mathbf{S}_n^* = \frac{\mathbf{S}_n - n\boldsymbol{\mu}_n}{\sqrt{n}}$$

for the normalized row sum. The following result is a global version of an extension of a large deviation local limit theorem for sums of independent random vectors due to Richter [8, Theorem 2] to row sums of triangular arrays. The proof of the result follows easily along the lines of Richter's proof and is sketched in [2]. We omit the derivation here.

Lemma 2.1 (large deviation global limit theorem). *Suppose* $\boldsymbol{\xi}_n = \bigl(\xi_n^{(1)}, \dots, \xi_n^{(k)}\bigr)$ *is any sequence of positive real vectors whose components satisfy* $1 \ll \xi_n^{(t)} \ll n^{1/6}$ $(1 \le t \le k)$ *as* $n \to \infty$*. Then under the previous assumptions*

$$\mathbb{P}\{\mathbf{S}_n^* > \boldsymbol{\xi}_n\} \sim \Phi_{V_n}(-\boldsymbol{\xi}_n) \quad \textit{and likewise} \quad \mathbb{P}\{\mathbf{S}_n^* < -\boldsymbol{\xi}_n\} \sim \Phi_{V_n}(-\boldsymbol{\xi}_n)$$

as $n \to \infty$.

Remark. Observe that asymptotic normality persists for deviations of $\mathbf{S}_n$ from the mean as large as $\mathfrak{o}\bigl(n^{2/3}\bigr)$, which admits of deviations much larger than the $\mathcal{O}\bigl(\sqrt{n}\,\bigr)$ deviations, which are the province of the standard central limit theorem.

It will be convenient in the analysis to find an elementary estimate of the multivariate Gaussian tail in Lemma 2.1. For the univariate case, the classical estimate of the tail $\Phi(-x)$ of the Gaussian can be expressed in terms of Mill's ratio in the form

$$\frac{\Phi(-x)}{\phi(x)} \sim x^{-1} \qquad (x \to \infty).$$

(See Feller [3, Lemma VII.1.2], for example.) The following specialization of a result of Ruben yields an analogous result for the multivariate case. We refer the reader to Ruben's paper [9] for the proof.

Lemma 2.2 (multivariate Mill's ratio). *Let* $\{A(\rho_n)\}$ *be a sequence of* $k \times k$ *covariance matrices, where* $A(\rho_n)$ *has unity as its diagonal elements and* ρ_n *as its off-diagonal elements, and suppose* $\rho_n \to 0$ *as* $n \to \infty$*. Let* $\{x_n\}$ *be any positive sequence satisfying* $x_n \to \infty$ *as* $n \to \infty$*. Then, writing* $\mathbf{1} \in \mathbb{R}^k$ *for the vector with all components identically* 1*, we have the asymptotic estimate*

$$\Phi_{A(\rho_n)}(-x_n\mathbf{1}) \sim x_n^{-k}\phi_{A(\rho_n)}(x_n\mathbf{1})$$

as $n \to \infty$.

Observe that the classical estimate for the tail of the univariate Gaussian follows directly with $k = 1$ and $K = [1]$.

The final technical result that will be needed is a probabilistic sieve. Suppose $\bigl\{B_n^{(t)}\bigr\}$ is a triangular array of events in a probability space with m_n events $(1 \le t \le m_n)$ in the nth row, and let $\bigl\{\mathfrak{z}_n^{(t)}\bigr\}$ denote the corresponding triangular array of indicator random variables for these events. Let $Z_{n,m_n} = \sum_{t=1}^{m_n} \mathfrak{z}_n^{(t)}$ denote the number of events that occur simultaneously in the nth row. We will be interested in the limiting distribution of the row sums Z_{n,m_n}.

Additionally, for each $k = 1, \dots, m_n$ and each $n = 1, 2, \dots$, define

$$S^{(k)}_{n,m_n} = \sum \mathbb{P}\bigl\{B^{(t_1)}_n \cap \cdots \cap B^{(t_k)}_n\bigr\} = \sum \mathbb{E}\bigl(\mathfrak{z}^{(t_1)}_n \cdots \mathfrak{z}^{(t_k)}_n\bigr),$$

where the sum is over all subsets $\{t_1, \dots, t_k\}$ of cardinality k drawn from $\{1, \dots, m_n\}$. Observe that $S^{(1)}_{n,m_n} = \mathbb{E}\, Z_{n,m_n}$, while, in general, $S^{(k)}_{n,m_n} = \mathbb{E}\bigl(Z^{[k]}_{n,m_n}/k!\bigr)$, where $Z^{[k]}_{n,m_n}$ denotes the number of ordered k-sets of events (no repetition) in the nth row for which all k events occur simultaneously.

Lemma 2.3 (Poisson tendency). *Suppose there is a constant λ such that, for every fixed k, $S^{(k)}_{n,m_n} \to \lambda^k/k!$ as $n \to \infty$. Then Z_{n,m_n} converges in distribution to* Po(λ)*, the Poisson distribution with parameter λ. In particular, for every fixed k,*

$$\mathbb{P}\{Z_{n,m_n} = k\} \to \frac{\lambda^k}{k!}\, e^{-\lambda}$$

as $n \to \infty$.

The proof follows directly from inclusion and exclusion by use of Bonferroni's inequalities to bound on both sides the probability $\mathbb{P}\{Z_{n,m_n} = k\}$ that exactly k of the events in the nth row occur simultaneously (cf. Feller [3, Theorem IV.3.1], for instance). We omit the standard proof.

3. Proof of Main Theorem.

We are interested in the probability

$$P(n,m) = \mathbb{P}\Biggl\{\bigcap_{t=1}^{m} \bigl\{\langle \mathbf{w}, \mathbf{u}^{(t)}\rangle \ge 0\bigr\}\Biggr\} = 1 - \mathbb{P}\Biggl\{\bigcup_{t=1}^{m} \bigl\{\langle \mathbf{w}, \mathbf{u}^{(t)}\rangle < 0\bigr\}\Biggr\}$$

of the event that $\mathbf{w} = \mathbf{w}^{(m+1)}$ has positive inner product with each vertex in $U(m)$. The following gives a thumbnail sketch of the principal ideas involved in the estimation of $P(n,m)$. We begin by showing via elementary arguments that the random summands $Y^{(t)}_i \triangleq w_i u^{(t)}_i$ $(1 \le t \le m)$ are exchangeable and then evaluate the first two mixed moments. Invoking Lemma 2.1, we then proceed to show that the events $\bigl\{\langle \mathbf{w}, \mathbf{u}^{(t)}\rangle < 0\bigr\}$ are governed asymptotically by a normal law even though the tail probabilities of interest correspond to deviations from the mean rather larger than the $\mathcal{O}\bigl(\sqrt{n}\,\bigr)$ deviations that fall under the usual province of the central limit theorem. Direct calculations of the relevant probabilities are still difficult, however, because of insidious statistical dependencies, albeit somewhat weak, evinced in the events $\bigl\{\langle \mathbf{w}, \mathbf{u}^{(t)}\rangle < 0\bigr\}$. The next stage in the proof involves quelling these dependencies with a firm hand using Lemma 2.2 to conclude that the events of interest are "asymptotically independent." The method of inclusion and exclusion embodied in Lemma 2.3 then allows us to conclude that, in the range of interest, the distribution of errors $\bigl\{\langle \mathbf{w}, \mathbf{u}^{(t)}\rangle < 0\bigr\}$ approaches a Poisson distribution asymptotically. The final stage of the calculation is a relatively straightforward bootstrap which rapidly produces an estimate of the critical sample size m by successive approximation.

A. Exchangeable random variables.

Since the m-set of examples $U(m) = \bigl\{\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(m)}\bigr\}$ is generated by independent sampling from the uniform distribution on the vertices $\mathbb{B}^n$, it follows that the example components $\bigl\{u^{(t)}_i, 1 \le t \le m, 1 \le i \le n\bigr\}$ are independent, identically distributed random variables taking values -1 and $+1$ only, each with probability $1/2$. Now, for each i, the ith memory component

$w_i = w_i^{(m+1)}$ is a (random) function solely of $u_i^{(1)}, \dots, u_i^{(m)}$. It is clear by symmetry that for every sample path which results in $w_i = +1$ there exists a sample path of equal probability (its reflection) which results in $w_i = -1$, and vice versa. Consequently, w_i is a symmetric Bernoulli random variable taking values -1 and $+1$ only, each with probability 1/2. Furthermore, as i runs through 1 to n, the sets $\{u_i^{(1)}, \dots, u_i^{(m)}\}$ partition the set of mn example components $\{u_i^{(t)}\}$ into n disjoint, identically distributed subsets. It follows that the memory components $\{\, w_i, 1 \le i \le n \,\}$ are independent, identically distributed symmetric binary random variables.

We begin with a preliminary result. Fix the index i and recall that $w_i^{(r)} \in \mathbb{B}$ represents the state of the ith bit of memory following the presentation of the $(r-1)$th example $\mathbf{u}^{(r-1)}$. Write

$$p_{r;t_1,\dots,t_k} \triangleq \mathbb{P}\{w_i^{(r)} = u_i^{(t_1)} = \cdots = u_i^{(t_k)}\}$$

for every choice of epochs $t_1, \dots, t_k$, r.

Assertion 1. *Let k and r be any fixed integers in the range $1 \le k < r \le m+1$. Then*

$$p_{r;t_1,\dots,t_k} = 2^{-k}\bigl(1 + \tfrac{k}{r-1}\bigr)$$

for every selection of k distinct epochs $t_1, \dots, t_k$ satisfying $1 \le t_j \le r-1$ $(1 \le j \le k)$.

Proof. To keep the notation unencumbered, suppress the subscript i and simply write $w^{(r)}$ and $u^{(t)}$ instead of the explicit $w_i^{(r)}$ and $u_i^{(t)}$. We may also assume without loss of generality that the indices are so ordered that $1 \le t_1 < t_2 < \cdots < t_k \le r-1$ as $p_{r;t_1,\dots,t_k}$ is invariant with respect to the permutation of the indices $t_1, \dots, t_k$. The proof of the assertion is by induction over r, k, and $t_1, \dots, t_k$.

`Induction base:` With $k = 1$, $t_1 = 1$, and $r = 2$, we have

$$p_{2;1} = \mathbb{P}\{w^{(2)} = u^{(1)}\} = 1 = \tfrac{1}{2}(1+1).$$

`Induction hypothesis:` For some $2 \le r \le m$, suppose that

$$p_{r;t_1,\dots,t_k} = 2^{-k}\bigl(1 + \tfrac{k}{r-1}\bigr) \qquad (1 \le k \le r-1;\ 1 \le t_1 < \cdots < t_k \le r-1).$$

Now consider $p_{r+1;t_1,\dots,t_k}$. We break the induction into two cases.

Case 1: $1 \le k \le r-1$, $1 \le t_1 < \cdots < t_k \le r-1$. Conditioning on $w^{(r)}$, we obtain

$$\begin{aligned}\mathbb{P}\{&w^{(r+1)} = u^{(t_1)} = \cdots = u^{(t_k)} = 1\}\\ &= \mathbb{P}\{w^{(r+1)} = 1 \mid w^{(r)} = 1\}\,\mathbb{P}\{w^{(r)} = u^{(t_1)} = \cdots = u^{(t_k)} = 1\}\\ &\quad + \mathbb{P}\{w^{(r+1)} = 1 \mid w^{(r)} = -1\}\,\mathbb{P}\{w^{(r)} = -1, u^{(t_1)} = \cdots = u^{(t_k)} = 1\}\end{aligned}$$

as $w^{(r+1)}$ is conditionally independent of $u^{(t_1)}, \dots, u^{(t_k)}$ given $w^{(r)}$ for $1 \le t_1 < \cdots < t_k \le r-1$. The conditional probabilities are readily evaluated: condition on $u^{(r)}$ and exploit the independence of $u^{(r)}$ and $w^{(r)}$ to obtain

$$\begin{aligned}\mathbb{P}\{w^{(r+1)} = 1 \mid w^{(r)} = 1\} &= \tfrac{1}{2}\,\mathbb{P}\{w^{(r+1)} = 1 \mid w^{(r)} = 1, u^{(r)} = 1\}\\ &\quad + \tfrac{1}{2}\,\mathbb{P}\{w^{(r+1)} = 1 \mid w^{(r)} = 1, u^{(r)} = -1\}\\ &= \tfrac{1}{2} + \tfrac{1}{2}\bigl(1 - \tfrac{1}{r}\bigr) = 1 - \tfrac{1}{2r}.\end{aligned} \tag{3.1}$$

The reflection principle shows that the random variables $\{ w^{(r)}, 2 \le r \le m+1 \}$ have symmetric marginal distributions

$$\mathbb{P}\{w^{(r)} = -1\} = \mathbb{P}\{w^{(r)} = +1\} = \tfrac{1}{2} \qquad (2 \le r \le m+1).$$

A simple application of Bayes's rule hence yields

$$\begin{aligned}\mathbb{P}\{w^{(r+1)} = 1 \mid w^{(r)} = -1\} &= \mathbb{P}\{w^{(r)} = -1 \mid w^{(r+1)} = 1\} = 1 - \mathbb{P}\{w^{(r)} = 1 \mid w^{(r+1)} = 1\} \\ &= 1 - \mathbb{P}\{w^{(r+1)} = 1 \mid w^{(r)} = 1\} = \tfrac{1}{2r},\end{aligned}$$

the last step following from (3.1). It follows that

$$\begin{aligned}\mathbb{P}\{w^{(r+1)} = u^{(t_1)} = \cdots = u^{(t_k)} = 1\} &= \left(1 - \tfrac{1}{2r}\right) \mathbb{P}\{w^{(r)} = u^{(t_1)} = \cdots = u^{(t_k)} = 1\} \\ &\quad + \tfrac{1}{2r}\, \mathbb{P}\{w^{(r)} = -1, u^{(t_1)} = \cdots = u^{(t_k)} = 1\}.\end{aligned}$$

Now observe that

$$\mathbb{P}\{w^{(r)} = -1, u^{(t_1)} = \cdots = u^{(t_k)} = 1\} = 2^{-k} - \mathbb{P}\{w^{(r)} = u^{(t_1)} = \cdots = u^{(t_k)} = 1\},$$

whence

$$\mathbb{P}\{w^{(r+1)} = u^{(t_1)} = \cdots = u^{(t_k)} = 1\} = \tfrac{2^{-k}}{2r} + \left(1 - \tfrac{1}{r}\right) \mathbb{P}\{w^{(r)} = u^{(t_1)} = \cdots = u^{(t_k)} = 1\}.$$

Likewise,

$$\mathbb{P}\{w^{(r+1)} = u^{(t_1)} = \cdots = u^{(t_k)} = -1\} = \tfrac{2^{-k}}{2r} + \left(1 - \tfrac{1}{r}\right) \mathbb{P}\{w^{(r)} = u^{(t_1)} = \cdots = u^{(t_k)} = -1\}.$$

It follows that for $1 \le k \le r-1$ and $1 \le t_1 < \cdots < t_k \le r-1$

$$p_{r+1;t_1,\ldots,t_k} = \tfrac{2^{-k}}{r} + \left(1 - \tfrac{1}{r}\right) p_{r;t_1,\ldots,t_k} = 2^{-k}\left(1 + \tfrac{k}{r}\right)$$

by the induction hypothesis.

Case 2: $1 \le k \le r$, $1 \le t_1 < \cdots < t_k = r$. Conditioning on $w^{(r)}$ again, we obtain

$$\begin{aligned}\mathbb{P}\{w^{(r+1)} = u^{(t_1)} = \cdots = u^{(t_k)} = 1\} &= \mathbb{P}\{w^{(r+1)} = u^{(t_1)} = \cdots = u^{(t_{k-1})} = u^{(r)} = 1\} \\ = \tfrac{1}{2}\, \mathbb{P}\{w^{(r+1)} = 1 \mid w^{(r)} = -1, u^{(r)} = 1\}&\, \mathbb{P}\{w^{(r)} = -1, u^{(t_1)} = \cdots = u^{(t_{k-1})} = 1\} \\ + \tfrac{1}{2}\, \mathbb{P}\{w^{(r+1)} = 1 \mid w^{(r)} = u^{(r)} = 1\}&\, \mathbb{P}\{w^{(r)} = u^{(t_1)} = \cdots = u^{(t_{k-1})} = 1\}\end{aligned}$$

as $w^{(r+1)}$ is conditionally independent of $u^{(t_1)}, \ldots, u^{(t_{k-1})}$ given $w^{(r)}$ and $u^{(r)}$, and $u^{(r)}$ is independent of $w^{(r)}$ and $u^{(t_1)}, \ldots, u^{(t_{k-1})}$.[4] The conditional probabilities above are completely determined by the auxiliary randomization in the algorithm. Hence

$$\begin{aligned}\mathbb{P}\{w^{(r+1)} = u^{(t_1)} = \cdots = u^{(t_k)} = 1\} &= \tfrac{1}{2}\, \mathbb{P}\{w^{(r)} = u^{(t_1)} = \cdots = u^{(t_{k-1})} = 1\} \\ &\quad + \tfrac{1}{2r}\, \mathbb{P}\{w^{(r)} = -1, u^{(t_1)} = \cdots = u^{(t_{k-1})} = 1\}.\end{aligned}$$

Now observe that

$$\mathbb{P}\{w^{(r)} = -1, u^{(t_1)} = \cdots = u^{(t_{k-1})} = 1\} = 2^{-(k-1)} - \mathbb{P}\{w^{(r)} = u^{(t_1)} = \cdots = u^{(t_{k-1})} = 1\},$$

[4] As usual, we identify the joint event $\bigcap_{j=1}^{k-1} \{u^{(t_j)} = 1\}$ with the certain event if $k = 1$.

whence

$$\mathbb{P}\bigl\{w^{(r+1)} = u^{(t_1)} = \cdots = u^{(t_k)} = 1\bigr\} = \tfrac{2^{-k}}{r} + \bigl(\tfrac{1}{2} - \tfrac{1}{2r}\bigr)\,\mathbb{P}\bigl\{w^{(r)} = u^{(t_1)} = \cdots = u^{(t_{k-1})} = 1\bigr\}.$$

A completely analogous argument gives

$$\begin{aligned}\mathbb{P}\bigl\{w^{(r+1)} = u^{(t_1)} = \ldots = u^{(t_{k-1})} = u^{(r)} = -1\bigr\}&\\ = \tfrac{2^{-k}}{r} + \bigl(\tfrac{1}{2} - \tfrac{1}{2r}\bigr)\,\mathbb{P}\bigl\{w^{(r)} = u^{(t_1)} = \cdots = u^{(t_{k-1})} = -1\bigr\}.&\end{aligned}$$

It follows that

$$p_{r+1;t_1,\ldots,t_{k-1},r} = \tfrac{2^{-k+1}}{r} + \tfrac{1}{2}\bigl(1 - \tfrac{1}{r}\bigr)p_{r;t_1,\ldots,t_{k-1}}.$$

Applying the induction hypothesis, it follows that

$$p_{r+1;t_1,\ldots,t_k} = 2^{-k}\bigl(1 + \tfrac{k}{r}\bigr) \qquad (1 \le k \le r;\ 1 \le t_1 < \cdots < t_{k-1} < t_k = r),$$

as was to be shown. The two cases taken together completes the induction. □

Now define the random variables

$$Y_i^{(t)} \triangleq w_i u_i^{(t)} \qquad (1 \le t \le m;\ 1 \le i \le n).$$

Observe that, for each t, $\bigl\langle \mathbf{w}, \mathbf{u}^{(t)} \bigr\rangle = \sum_{i=1}^n Y_i^{(t)}$ is a sum of independent, identically distributed ± 1 random variables, i.e., a random walk on the line. The walk is asymmetric with a positive drift, as we shall see shortly.

In what follows it will be slightly more convenient to consider the related $(0,1)$ random variables

$$X_i^{(t)} = \tfrac{1}{2}\bigl(1 + Y_i^{(t)}\bigr) \qquad (1 \le t \le m;\ 1 \le i \le n),$$

whence $\bigl\langle \mathbf{w}, \mathbf{u}^{(t)} \bigr\rangle = 2\sum_{i=1}^n X_i^{(t)} - n$. Consider the random set $\mathcal{X}_i \triangleq \bigl\{X_i^{(1)}, \ldots, X_i^{(m)}\bigr\}$. Since w_i is a function only of $u_i^{(1)}, \ldots, u_i^{(m)}$, it follows that $\mathcal{X}_i$ is also determined only by $u_i^{(1)}, \ldots, u_i^{(m)}$. Consequently, the random sets $\mathcal{X}_1, \ldots, \mathcal{X}_n$ are statistically independent and have identical (joint) distributions. Assertion 1 now allows us to explicitly characterize the joint distribution of the random set $\mathcal{X}_i$.

For every choice of epochs $t_1, \ldots, t_{\mathfrak{h}}$, $t_{\mathfrak{h}+1}, \ldots, t_{\mathfrak{h}+\mathfrak{k}}$, write

$$q_{m;t_1,\ldots,t_{\mathfrak{h}};t_{\mathfrak{h}+1},\ldots,t_{\mathfrak{h}+\mathfrak{k}}} \triangleq \mathbb{P}\bigl\{X_i^{(t_1)} = 1, \ldots, X_i^{(t_{\mathfrak{h}})} = 1, X_i^{(t_{\mathfrak{h}+1})} = 0, \ldots, X_i^{(t_{\mathfrak{h}+\mathfrak{k}})} = 0\bigr\}.$$

It will also be convenient to define

$$f_m(\mathfrak{h}, \mathfrak{k}) \triangleq 2^{-(\mathfrak{h}+\mathfrak{k})}\bigl(1 + \tfrac{\mathfrak{h}-\mathfrak{k}}{m}\bigr).$$

Observe that $p_{m+1;t_1,\ldots,t_k} = f_m(k, 0)$ by Assertion 1. As an immediate consequence, we have the following.

Assertion 2. *The $(0,1)$ random variables $X_i^{(1)}, \ldots, X_i^{(m)}$ are exchangeable. In particular,*

$$q_{m;t_1,\ldots,t_{\mathfrak{h}};t_{\mathfrak{h}+1},\ldots,t_{\mathfrak{h}+\mathfrak{k}}} = f_m(\mathfrak{h}, \mathfrak{k}) \tag{3.2}$$

for every pair of nonnegative integers $\mathfrak{h}$ and $\mathfrak{k}$ with $\mathfrak{h} + \mathfrak{k} \le m$ and $\mathfrak{h} + \mathfrak{k}$ distinct indices $t_1, \ldots, t_{\mathfrak{h}+\mathfrak{k}}$ in $\{1, \ldots, m\}$.

Remark. In accordance with usual convention, we identify the joint event

$$\bigcap_{j=1}^{\mathfrak{h}}\bigl\{X_i^{(t_j)}=1\bigr\}\cap\bigcap_{j=\mathfrak{h}+1}^{\mathfrak{h}+\mathfrak{k}}\bigl\{X_i^{(t_j)}=0\bigr\}$$

with the certain event if $\mathfrak{h}=\mathfrak{k}=0$. The assertion then holds when $\mathfrak{h}=\mathfrak{k}=0$ as well when the desired probability is identically $f_m(0,0)=1$.

Proof. The result follows quickly by induction on $\mathfrak{k}$.

`Induction base:` When $\mathfrak{k}=0$, it follows immediately that for every $0\le\mathfrak{h}\le m$,

$$q_{m;t_1,\dots,t_{\mathfrak{h}}}=\mathbb{P}\bigl\{X_i^{(t_1)}=1,\dots,X_i^{(t_{\mathfrak{h}})}=1\bigr\}=p_{m+1;t_1,\dots,t_{\mathfrak{h}}}=2^{-\mathfrak{h}}\bigl(1+\tfrac{\mathfrak{h}}{m}\bigr)=f_m(\mathfrak{h},0)$$

depends only on $\mathfrak{h}$ and m and is independent of the choice of (distinct) indices $t_1,\dots,t_{\mathfrak{h}}$. The proof is completed by induction over $\mathfrak{k}$.

`Induction hypothesis:` Suppose that for some choice of $\mathfrak{k}\ge 0$, (3.2) holds for every $\mathfrak{h}\ge 0$ with $\mathfrak{h}+\mathfrak{k}\le m$ and every choice of $\mathfrak{h}+\mathfrak{k}$ distinct indices $t_1,\dots,t_{\mathfrak{h}}$, $t_{\mathfrak{h}+1},\dots,t_{\mathfrak{h}+\mathfrak{k}}$ in $\{1,\dots,m\}$. It follows that, for every $\mathfrak{h}\ge 0$ with $\mathfrak{h}+\mathfrak{k}+1\le m$ and every distinct collection of $\mathfrak{h}+\mathfrak{k}+1$ indices $t_1,\dots,t_{\mathfrak{h}}$, $t_{\mathfrak{h}+1}$, $t_{\mathfrak{h}+2},\dots,t_{\mathfrak{h}+\mathfrak{k}+1}$ in $\{1,\dots,m\}$,

$$\begin{aligned}q_{m;t_1,\dots,t_{\mathfrak{h}};t_{\mathfrak{h}+1},t_{\mathfrak{h}+2},\dots,t_{\mathfrak{h}+\mathfrak{k}+1}}&=q_{m;t_1,\dots,t_{\mathfrak{h}};t_{\mathfrak{h}+2},\dots,t_{\mathfrak{h}+\mathfrak{k}+1}}-q_{m;t_1,\dots,t_{\mathfrak{h}},t_{\mathfrak{h}+1};t_{\mathfrak{h}+2},\dots,t_{\mathfrak{h}+\mathfrak{k}+1}}\\&=f_m(\mathfrak{h},\mathfrak{k})-f_m(\mathfrak{h}+1,\mathfrak{k})\\&=f_m(\mathfrak{h},\mathfrak{k}+1),\end{aligned}$$

the penultimate step following from the induction hypothesis and the last step following by the definition of f_m. This completes the induction. □

The moments of the random variables $X_i^{(1)},\dots,X_i^{(m)}$ are now readily determined. In particular,

$$\begin{aligned}\mu&\triangleq\mathbb{E}\bigl(X_i^{(t)}\bigr)=f_m(1,0)=\tfrac{1}{2}+\tfrac{1}{2m},\\\sigma^2&\triangleq\operatorname{Var}\bigl(X_i^{(t)}\bigr)=f_m(1,0)-f_m(1,0)^2=\tfrac{1}{4}-\tfrac{1}{4m^2},\\\psi&\triangleq\operatorname{Cov}\bigl(X_i^{(s)},X_i^{(t)}\bigr)=f_m(2,0)-f_m(1,0)^2=-\tfrac{1}{4m^2}\qquad(s\ne t),\\\rho&\triangleq\tfrac{\psi}{\sigma^2}=-\tfrac{1}{m^2-1}=\mathcal{O}\bigl(m^{-2}\bigr).\end{aligned}$$

B. Normal tendency. Now consider the random walks

$$\bigl\langle\mathbf{w},\mathbf{u}^{(t)}\bigr\rangle=\sum_{i=1}^{n}w_iu_i^{(t)}=2\sum_{i=1}^{n}X_i^{(t)}-n\qquad(1\le t\le m).$$

Recall that the n random sets $\mathcal{X}_i=\bigl\{X_i^{(1)},\dots,X_i^{(m)}\bigr\}$ $(1\le i\le n)$ are mutually independent with identical joint distributions. It now follows as a consequence of Assertion 2 that the random variables $\bigl\langle\mathbf{w},\mathbf{u}^{(1)}\bigr\rangle,\dots,\bigl\langle\mathbf{w},\mathbf{u}^{(m)}\bigr\rangle$ are exchangeable.

Let k be any fixed positive integer and consider any distinct set of k indices $t_1,\dots,t_k$ in $\{1,\dots,m\}$. Write

$$\mathfrak{P}_{n,m}(k)\triangleq\mathbb{P}\bigl\{\bigl\langle\mathbf{w},\mathbf{u}^{(t_1)}\bigr\rangle<0,\dots,\bigl\langle\mathbf{w},\mathbf{u}^{(t_k)}\bigr\rangle<0\bigr\}.$$

Note that $\mathfrak{P}_{n,m}(k)$ is just the probability that any given k inequalities in (1.1) are violated, where the random $\mathbf{w}$ is determined by harmonic update. Since the random walks $\langle \mathbf{w}, \mathbf{u}^{(t)} \rangle$ $(1 \le t \le m)$ are exchangeable random variables, $\mathfrak{P}_{n,m}(k)$ does not depend on the specific choice of indices t_j and we may suppose without any loss of generality that $t_j = j$ $(1 \le j \le k)$. Thus,

$$\mathfrak{P}_{n,m}(k) = \mathbb{P}\bigl\{\langle \mathbf{w}, \mathbf{u}^{(1)} \rangle < 0, \dots, \langle \mathbf{w}, \mathbf{u}^{(k)} \rangle < 0\bigr\} = \mathbb{P}\Biggl\{\sum_{i=1}^{n} X_i^{(1)} < \tfrac{n}{2}, \dots, \sum_{i=1}^{n} X_i^{(k)} < \tfrac{n}{2}\Biggr\}.$$

Let us now explicitly allow $m = m_n$ to vary with n and acknowledge this dependence on n by writing,

$$X_i^{(t)} = X_{ni}^{(t)} = \tfrac{1}{2}\bigl(1 + w_i u_i^{(t)}\bigr) \qquad (1 \le t \le m_n;\ 1 \le i \le n),$$

where $u_i^{(t)} = u_{ni}^{(t)}$ $(1 \le t \le m_n; 1 \le i \le n)$ are the components of the random examples and $w_i = w_{ni}$ $(1 \le i \le n)$ are the components of the memory state determined by harmonic update. Now consider the triangular array of k-dimensional lattice random vectors

$$\mathbf{X}_{ni} = \bigl(X_{ni}^{(1)}, \dots, X_{ni}^{(k)}\bigr) \qquad (i = 1, \dots, n;\ n = 1, 2, \dots).$$

For each n, the random vectors $\mathbf{X}_{n1}, \dots, \mathbf{X}_{nn}$ comprising the nth row of the array are independent, identically distributed lattice random vectors with probability one support in $\{0,1\}^k$. Let $p_n(\mathbf{x}) = \mathbb{P}\bigl\{\mathbf{X}_{ni} = \mathbf{x}\bigr\}$ $(\mathbf{x} \in \{0,1\}^k)$ denote the distribution of $\mathbf{X}_{ni}$. Write $|\mathbf{x}|$ for the number of components of $\mathbf{x} \in \{0,1\}^k$ that take value one (this is just the L^1 vector norm in this case). Observe then, as a consequence of Assertion 2, that

$$p_n(\mathbf{x}) = f_{m_n}\bigl(|\mathbf{x}|, k - |\mathbf{x}|\bigr) = \frac{1}{2^k}\left(1 + \frac{2|\mathbf{x}| - k}{m_n}\right) \qquad \bigl(\mathbf{x} \in \{0,1\}^k\bigr).$$

Suppose $m_n \to \infty$ as $n \to \infty$. Then, for sufficiently large n, $p_n(\mathbf{x}) > 0$ for every $\mathbf{x} \in \{0,1\}^k$; in particular, $\mathbf{X}_{ni}$ takes values $\mathbf{0}$, $\mathbf{e}_1, \dots, \mathbf{e}_k$ (where $\mathbf{e}_\nu$ $(1 \le \nu \le k)$ denotes the canonical unit vectors in $\{0,1\}^k$) with positive probability. Furthermore, $p_n(\mathbf{x}) \to 2^{-k}$ (uniformly) for all $\mathbf{x} \in \{0,1\}^k$, whence the distribution of $\mathbf{X}_{ni}$ becomes uniform over $\{0,1\}^k$ in the limit.

Let us also explicitly write

$$\begin{aligned} \mu = \mu_n &= \tfrac{1}{2} + \tfrac{1}{2m_n}, \\ \sigma^2 = \sigma_n^2 &= \tfrac{1}{4} - \tfrac{1}{4m_n^2}, \\ \rho = \rho_n &= -\tfrac{1}{m_n^2 - 1} = \mathcal{O}\bigl(m_n^{-2}\bigr) \end{aligned}$$

for the mean, variance, and correlation coefficient, respectively, of the $(0,1)$ random variables $X_{ni}^{(t)}$ $(1 \le t \le m_n)$. We then have

$$\begin{aligned} \mathbb{E}\bigl(\mathbf{X}_{ni}\bigr) &= \boldsymbol{\mu}_n = \mu_n \mathbf{1}, \\ \operatorname{Cov}\bigl(\mathbf{X}_{ni}\bigr) &= V_n = \sigma_n^2 A(\rho_n), \end{aligned}$$

where, as before, $A(\rho_n)$ denotes a $k \times k$ covariance matrix which has unity as its diagonal elements and ρ_n as its off-diagonal elements. Observe that $\mu_n \to \frac{1}{2}$, $\sigma_n^2 \to \frac{1}{4}$, and $\rho_n \to 0$ if $m_n \gg 1$ as $n \to \infty$.

Everything is now set for an application of Lemma 2.1.

Assertion 3. *If $m = m_n$ increases with n in such a way that $n^{1/3} \ll m_n \ll n^{1/2}$ then, for every fixed positive integer k,*

$$\mathfrak{P}_{n,m_n}(k) \sim \Phi_{A(\rho_n)}\Big(-\tfrac{\sqrt{n}}{m_n}\,\mathbf{1}\Big)$$

as $n \to \infty$.

Proof. Form the row sums and the corresponding normalized row sums

$$\mathbf{S}_n = \sum_{i=1}^{n} \mathbf{X}_{ni}, \qquad \mathbf{S}_n^* = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \big(\mathbf{X}_{ni} - \boldsymbol{\mu}_n\big),$$

and define the sequence

$$\xi_n \triangleq \big(\mu_n - \tfrac{1}{2}\big)\sqrt{n} = \frac{\sqrt{n}}{2m_n}.$$

We then obtain

$$\mathfrak{P}_{n,m_n}(k) = \mathbb{P}\big\{\mathbf{S}_n < \tfrac{n}{2}\,\mathbf{1}\big\} = \mathbb{P}\big\{\mathbf{S}_n^* < -\xi_n \mathbf{1}\big\}.$$

Observe that in the range $n^{1/3} \ll m_n \ll n^{1/2}$ we have $1 \ll \xi_n \ll n^{1/6}$ as $n \to \infty$. Applying Lemma 2.1, we hence obtain

$$\mathfrak{P}_{n,m_n}(k) \sim \Phi_{V_n}\Big(\tfrac{-\sqrt{n}}{2m_n}\,\mathbf{1}\Big) = \Phi_{A(\rho_n)}\Big(\tfrac{-\sqrt{n}}{2\sigma_n m_n}\,\mathbf{1}\Big) \qquad (n \to \infty).$$

Now observe that

$$\tfrac{\sqrt{n}}{2\sigma_n m_n} = \tfrac{\sqrt{n}}{m_n}\big(1 - m_n^{-2}\big)^{-1/2} = \tfrac{\sqrt{n}}{m_n}\big[1 + \mathcal{O}\big(m_n^{-2}\big)\big] = \tfrac{\sqrt{n}}{m_n} + \mathcal{O}\big(\tfrac{\sqrt{n}}{m_n^3}\big) = \tfrac{\sqrt{n}}{m_n} + \mathfrak{o}(1)$$

when $m_n \gg n^{1/3}$. Furthermore, $\rho_n \to 0$ as $n \to \infty$, whence the covariance matrix $A(\rho_n)$ and its inverse converge componentwise to the identity matrix. Consequently, $\mathfrak{P}_{n,m_n}(k) \sim \Phi_{A(\rho_n)}\big(\tfrac{-\sqrt{n}}{m_n}\,\mathbf{1}\big)$ as $n \to \infty$. □

We can parlay the asymptotic normality of finite sets of the random variables $\big\langle \mathbf{w}, \mathbf{u}^{(t)} \big\rangle$ into a statement about the distribution of errors. This follows next.

C. Poisson tendency. We begin by exploiting the fact that the covariances between the random variables $X_i^{(t)}$ $(1 \le t \le m_n)$ vanish asymptotically. Coupling this with the exchangeability of the events $\big\{\big\langle \mathbf{w}, \mathbf{u}^{(t)} \big\rangle < 0\big\}$, we will indeed be able to show that, for a suitable rate of increase of $m = m_n$ with n, the distribution of errors $\big\langle \mathbf{w}, \mathbf{u}^{(t)} \big\rangle < 0$ is asymptotically Poisson. This will suffice to complete the proof of the Main Theorem.

It will be convenient to first define the double sequence

$$\mathfrak{Q}_{n,m} \triangleq \frac{m}{\sqrt{2\pi n}} \exp\Big\{-\frac{n}{2m^2}\Big\}$$

before launching into the result.

Assertion 4. *Suppose $m = m_n$ increases with n such that $n^{1/3} \ll m_n \ll n^{1/2}$. Then, for every fixed positive integer k, $\mathfrak{P}_{n,m_n}(k) \sim \mathfrak{P}_{n,m_n}(1)^k \sim \mathfrak{Q}_{n,m_n}^k$ as $n \to \infty$.*

Proof. Write $x_n \triangleq m_n^{-1}\sqrt{n}$. Note that $x_n \to \infty$ as $n \to \infty$ for the given rate of growth of m_n with n. Applying Assertion 3 to the case $k = 1$ yields $\mathfrak{P}_{n,m_n}(1) \sim \Phi(-x_n)$, while Lemma 2.2 shows that the right-hand side is asymptotic to $\mathfrak{Q}_{n,m_n}$ as

$n \to \infty$. Now suppose k is any fixed positive integer. Recall that $\rho_n = \mathcal{O}\bigl(m_n^{-2}\bigr) = \mathfrak{o}(1)$ as $m_n \gg 1$. We may hence apply Lemma 2.2 again to obtain

$$\begin{aligned}\mathfrak{P}_{n,m_n}(k) \sim \Phi_{A(\rho_n)}(-x_n\mathbf{1}) \sim x_n^{-k}\phi_{A(\rho_n)}(x_n\mathbf{1})&\\ = (2\pi)^{-k/2}x_n^{-k}|A(\rho_n)|^{-1/2}e^{-\frac{1}{2}x_n^2\mathbf{1}A(\rho_n)^{-1}\mathbf{1}'}&.\end{aligned}$$

Induction on k readily yields explicit expressions for the determinant and inverse of the covariance matrix $A(\rho_n)$,

$$\begin{aligned}|A(\rho_n)| &= \bigl(1+(k-1)\rho_n\bigr)(1-\rho_n)^{k-1},\\ A(\rho_n)^{-1} &= a_nA(b_n),\end{aligned}$$

where a_n and b_n are specified by

$$a_n = \frac{1+(k-2)\rho_n}{(1-\rho_n)[1+(k-1)\rho_n]} \quad \text{and} \quad b_n = \frac{-\rho_n}{1+(k-2)\rho_n}.$$

We hence have

$$\begin{aligned}\mathfrak{P}_{n,m_n}(k) &\sim (2\pi)^{-k/2}(1-\rho_n)^{-(k-1)/2}\bigl(1+(k-1)\rho_n\bigr)^{-1/2}x_n^{-k}e^{-kx_n^2/2\{1+(k-1)\rho_n\}}\\ &= (2\pi)^{-k/2}(1-\rho_n)^{-(k-1)/2}\bigl(1+(k-1)\rho_n\bigr)^{-1/2}x_n^{-k}e^{-\frac{1}{2}kx_n^2+\mathcal{O}(x_n^2\rho_n)}\\ &= (2\pi)^{-k/2}x_n^{-k}e^{-\frac{1}{2}kx_n^2}\bigl\{1+\mathcal{O}(\rho_n)+\mathcal{O}(x_n^2\rho_n)\bigr\}\\ &= \mathfrak{Q}_{n,m_n}^k\bigl\{1+\mathcal{O}\bigl(m_n^{-2}\bigr)+\mathcal{O}\bigl(nm_n^{-4}\bigr)\bigr\}.\end{aligned}$$

All order terms on the right-hand side approach zero asymptotically for the given rate of growth of m_n, whence $\mathfrak{P}_{n,m_n}(k) \sim \mathfrak{Q}_{n,m_n}^k$ as $n \to \infty$. □

In slightly imprecise language—the events $\bigl\{\bigl\langle \mathbf{w}, \mathbf{u}^{(t)}\bigr\rangle < 0\bigr\}$ are asymptotically independent.

All the pieces are now in place. We complete the proof of the Main Theorem by invoking Lemma 2.3.

For each n, suppose $m = m_n$ random vertices $\mathbf{u}^{(t)} = \mathbf{u}_n^{(t)}$ $(1 \le t \le m_n)$ are generated by independent sampling from the uniform distribution on $\mathbb{B}^n$ and let $\mathbf{w} = \mathbf{w}_n \in \mathbb{B}^n$ be the corresponding vertex generated by harmonic update. Consider the triangular array of "error" events

$$B_n^{(t)} = \bigl\{\bigl\langle \mathbf{w}_n, \mathbf{u}_n^{(t)}\bigr\rangle < 0\bigr\} \qquad (t = 1, \dots, m_n;\ n = 1, 2, \dots),$$

where the nth row has $m = m_n$ elements, and let $\bigl\{\mathfrak{z}_n^{(t)}\bigr\}$ be the corresponding triangular array of indicator random variables for these events. Thus,

$$\mathfrak{z}_n^{(t)} = \begin{cases} 0 & \text{if } \bigl\langle \mathbf{w}_n, \mathbf{u}_n^{(t)}\bigr\rangle \ge 0,\\ 1 & \text{if } \bigl\langle \mathbf{w}_n, \mathbf{u}_n^{(t)}\bigr\rangle < 0.\end{cases}$$

Form the row sums $Z_{n,m_n} = \sum_{t=1}^{m_n}\mathfrak{z}_n^{(t)}$. For each n, Z_{n,m_n} is just the number of examples $\mathbf{u}^{(t)} = \mathbf{u}_n^{(t)}$ $(1 \le t \le m_n)$ which fall into the negative half-space determined by the vector $\mathbf{w} = \mathbf{w}_n$ generated by harmonic update. Alternatively, for given n, Z_{n,m_n} is the number of examples $\mathbf{u}^{(t)} = \mathbf{u}_n^{(t)}$ for which the corresponding inequality in (1.1) is violated. For each $k = 1, \dots, m_n$ and each $n = 1, 2, \dots$, define

$$S_{n,m_n}^{(k)} = \sum \mathbb{E}\bigl(\mathfrak{z}_n^{(t_1)} \cdots \mathfrak{z}_n^{(t_k)}\bigr) = \sum \mathbb{P}\bigl\{B_n^{(t_1)} \cap \cdots \cap B_n^{(t_k)}\bigr\},$$

where the sum is over all subsets $\{t_1, \dots, t_k\}$ of cardinality k drawn from $\{1, \dots, m_n\}$. For each n, the events $B_n^{(t)}$ $(1 \le t \le m_n)$ are exchangeable, so that each of the summands above is just $\mathfrak{P}_{n,m_n}(k)$. We hence obtain

$$S_{n,m_n}^{(k)} = \binom{m_n}{k} \mathfrak{P}_{n,m_n}(k)$$

for every choice of k and n.

Let us now fix the rate of growth of $m = m_n$ with n. Let λ denote any fixed positive quantity and suppose

$$m_n = \sqrt{\frac{n}{\log n}} \left\{ 1 + \frac{\log\log n + \log\bigl(\lambda\sqrt{2\pi}\,\bigr)}{\log n} + \mathcal{O}\biggl(\frac{\log\log n}{(\log n)^2}\biggr) \right\}. \tag{3.3}$$

Clearly, m_n satisfies the conditions $n^{1/3} \ll m_n \ll n^{1/2}$ as $n \to \infty$. Invoking Assertion 4, it is now simple to verify that

$$S_{n,m_n}^{(1)} = m_n \mathfrak{P}_{n,m_n}(1) \sim m_n \mathfrak{Q}_{n,m_n} = \frac{m_n^2}{\sqrt{2\pi n}} \exp\left\{ -\frac{n}{2m_n^2} \right\} \to \lambda$$

as $n \to \infty$. Now, fix any value of k and allow m_n to grow as in (3.3). Observe that $\binom{m_n}{k} \sim \frac{m_n^k}{k!}$ as $n \to \infty$. Invoking Assertion 4 again, we hence obtain the asymptotic estimate

$$S_{n,m_n}^{(k)} \sim \binom{m_n}{k} \mathfrak{Q}_{n,m_n}^k \sim \frac{(m_n \mathfrak{Q}_{n,m_n})^k}{k!} \to \frac{\lambda^k}{k!} \qquad (n \to \infty).$$

We can now directly apply Lemma 2.3 to conclude that Z_{n,m_n} converges in distribution to the Poisson distribution with parameter λ. This completes the proof of the Main Theorem.

Finally, for the rate of growth given in (3.3), $P(n, m_n) = \mathbb{P}\{Z_{n,m_n} = 0\} \to e^{-\lambda}$ as $n \to \infty$. Now, for any choice of $\lambda > 0$, however small, and any choice of $1 > \epsilon > 0$, a sample size of $m \le (1 - \epsilon)\sqrt{n}\big/\sqrt{\log n}$ will be eventually dominated by the right-hand side of (3.3), so that $P(n, m)$ will approach one as $n \to \infty$ by monotonicity. Conversely, for any choice of $\lambda < \infty$, however large, and any choice of $1 > \epsilon > 0$, a sample size of $m \ge (1 + \epsilon)\sqrt{n}\big/\sqrt{\log n}$ will eventually dominate the right-hand side of (3.3), so that, by analogous reasoning, $P(n, m)$ will approach zero as $n \to \infty$. This establishes the corollary to the Main Theorem.

REFERENCES

[1] S. C. Fang and S. S. Venkatesh, *Learning binary perceptrons perfectly efficiently*, J. Comput. System Sci., 52 (1996), pp. 374–389.

[2] S. C. Fang and S. S. Venkatesh, *The capacity of Majority Rule*, Random Structures Algorithms, to appear.

[3] W. Feller, *An Introduction to Probability Theory and Its Applications*, Volume I, 3rd ed., John Wiley, New York, 1968.

[4] Z. Füredi, *Random polytopes in the d-dimensional cube*, Discrete Comput. Geom., 1 (1986), pp. 315–319.

[5] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, CA, 1979.

[6] N. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[7] L. Pitt and L. G. Valiant, *Computational limitations on learning from examples*, J. Assoc. Comput. Mach., 35 (1988), pp. 965–984.

[8] W. Richter, *Multidimensional local limit theorems for large deviations*, Theory Probab. Appl., 3 (1958), pp. 100–106.

[9] H. Ruben, *An asymptotic expansion for a class of multivariate normal integrals*, J. Austral. Math. Soc., 2 (1962), pp. 253–264.

[10] J. Spencer, *Ten Lectures on the Probabilistic Method*, CBMS-NSF Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1987.

[11] S. S. Venkatesh, *On learning binary weights for majority functions*, in Proc. of the Fourth Annual Workshop on Computational Learning Theory, San Mateo, CA, Morgan Kaufmann, 1991.

[12] S. S. Venkatesh and J. Franklin, *How much information can one bit of memory retain about a Bernoulli sequence?*, IEEE Trans. Inform. Theory, 37 (1991), pp. 1595–1604.

SIAM J. Discrete Math.
Vol. 10, No. 3, pp. 499–504, August 1997

A COMBINATORIAL PROOF FOR STOCKHAUSEN'S PROBLEM[*]

LILY YEN[†]

Abstract. We consider problems in the enumeration of sequences suggested by the problem of determining the number of ways of performing a piano composition (*Klavierstück* XI) by Karlheinz Stockhausen. An explicit formula and a combinatorial proof for the general problem are given.

Key words. Stockhausen, sequences, bijections, sign-reversing involution, Gray codes

AMS subject classification. 05A19

PII. S0895480195288741

The score of the piano work *nr.* 7 *Klavierstück* XI by Karlheinz Stockhausen (1957) [S] consists of 19 fragments of music. The performer is instructed to choose at random one of these fragments and play it, then choose another, different, fragment and play that, and so on. If a fragment is chosen that has already been played twice, the performance ends. It is natural to ask how many different ways there are of performing this piece and what the expected length of a performance is. The enumerative results for this problem and some generalizations are given in [RY]; we show here combinatorial proofs for the generalized Stockhausen numbers.

DEFINITION. *An r-Stockhausen sequence is a sequence of symbols from $\mathcal{N}_n = \{1, 2, \dots, n\}$ such that*

(1) *adjacent symbols are distinct,*

(2) *the terminal symbol occurs exactly r times,*

(3) *all other symbols occur at most $r-1$ times.*

Let $s_r(n)$ be the number of such sequences.

The generating series for $c_r(n,k)$, the number of r-Stockhausen sequences of length k on n symbols, is (*see* [Y])

$$\Phi_r(z,u) = uz^r L_w^{(r-1)} \Theta_w \exp\left(u[t^{r-1}]\frac{1}{1-t}e^{wt/(1+t)}\right),$$

where u is the exponential marker for the number of available symbols, z is the ordinary marker for the length of the string, and

$$L_w^{(r)} f := \frac{1}{r!}\,\frac{\partial^r f}{\partial w^r}\bigg|_{w=1}, \qquad \Theta_w\, w^k \mapsto k!\, w^k.$$

Therefore,

$$(1) \qquad s_r(n) = \left[\frac{u^n}{n!}\right]\Phi_r(1,u) = nL_w^{(r-1)}\Theta_w\left([t^{r-1}]\frac{1}{1-t}e^{wt/(1+t)}\right)^{n-1}.$$

Let the coefficients a_j be given by

$$(2) \qquad [t^{r-1}]\frac{1}{1-t}e^{wt/(1+t)} = a_0 + a_1 w + a_2\frac{w^2}{2!} + \cdots + a_{r-1}\frac{w^{r-1}}{(r-1)!} =: f(w).$$

[*] Received by the editors July 7, 1995; accepted for publication (in revised form) June 25, 1996. This research was supported by a postdoctoral fellowship from the Natural Sciences and Engineering Research Council of Canada.

http://www.siam.org/journals/sidma/10-3/28874.html

[†] Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada (lyen@cecm.sfu.ca).

Then

$$s_r(n) = n \sum_{b_1,b_2,\ldots,b_{r-1}} \binom{n-1}{b_1,b_2,\ldots,b_{r-1},n-1-\sum b_i} \times \left(\sum_{j=1}^{r-1} jb_j\right)! \prod_{j=1}^{r-1} (a_j j!)^{b_j} \binom{\sum_{j=1}^{r-1} jb_j}{r-1} \tag{3}$$

by direct computation. In the case $a_1 = 0$, the multinomial expression does not contain b_1, and the product $\prod(\frac{a_j}{j!})^{b_j}$ is over $j = 2, 3, \ldots, r-1$.

Remarks.

(i) The classical generating function of the Laguerre polynomial sequence is

$$\sum_{n\geq 0} L_n(x)t^n = \frac{1}{1-t}\exp\left(\frac{-xt}{1-t}\right).$$

Thus, (1) can be expressed in terms of Laguerre polynomials to yield an explicit expression for a_j's defined in (2). However, we will derive the explicit expression for a_j's from first principles that shed light on the steps leading to a combinatorial proof for r-Stockhausen numbers.

(ii) About r^n terms are in (3).

(iii) Though for each fixed j, a_j depends on r, we use only one subscript because r is clear from the context.

The idea of the proof is first to identify a (multi-) set of sequences on $\{1, 2, \ldots, n\}$ enumerated by

$$s_r^+(n) = n \sum_{b_1,b_2,\ldots,b_{r-1}} \binom{n-1}{b_1,b_2,\ldots,b_{r-1},n-1-\sum b_i} \left(\sum_{j=1}^{r-1} jb_j\right)! \times \prod_{j=1}^{r-1} (|a_j| j!)^{b_j} \binom{\sum_{j=1}^{r-1} jb_j}{r-1},$$

the expression obtained from $s_r(n)$ by taking $|a_j|$ instead of a_j. To account for the negative signs in $s_r(n)$, we establish a sign-reversing involution by way of assigning signs to the objects enumerated by $s_r^+(n)$, then establish a bijection between some of the + objects and all of the − objects. The last step is to identify the + objects that are fixed by the involution as r-Stockhausen sequences via a simple mapping.

We illustrate the ideas of the proof in the first two Stockhausen numbers.

Proposition 0.1. *The number of* 3*-Stockhausen sequences is*

$$s_3(n) = n\sum_{j=0}^{n-1} \binom{n-1}{j} \frac{(2j)!}{2^j} \binom{2j}{2}. \tag{4}$$

Proof. Given n symbols, there are n ways of choosing a symbol to be the terminal symbol that occurs three times. For the remaining $n-1$ symbols, there are $\binom{n-1}{j}$ ways of choosing j symbols to use in making $(2j)!/2^j$ strings such that each of the j symbols appears twice with no adjacency restrictions, and $\binom{2j}{2}$ ways of marking two positions on each such string. The terminal symbol is placed immediately before the two marked positions and of course at the terminal position of the string, thus making $n\sum_j \binom{n-1}{j}\binom{2j}{2}\frac{(2j)!}{2^j}$. By replacing any repeated symbols yy by a single symbol y, we obtain strings whose adjacent symbols are all distinct, and the correspondence is one-to-one. □

PROPOSITION 0.2. *The number of* 4*-Stockhausen sequences is*

$$(5)\quad s_4(n) = n\sum_{i,j,k=0}^{i+j+k=n-1}\binom{n-1}{i,j,k,n-1-i-j-k}\binom{3i+2j+k}{3}\frac{(3i+2j+k)!}{(3!)^i(-2)^j}.$$

Proof. The number of sequences on n symbols such that the terminal symbol occurs four times nonadjacently and all the other symbols occur at most thrice with no adjacency restrictions is

$$n\sum_{i,j,k}\binom{n-1}{i,j,k,n-1-i-j-k}\frac{(3i+2j+k)!}{(3!)^i(2!)^j}\binom{3i+2j+k}{3}.$$

Let $\mathcal{A}$ denote the set of such sequences. We partition $\mathcal{A}$ into $\mathcal{A}^+$ and $\mathcal{A}^-$ and define a sign-reversing involution ϕ on the elements of $\mathcal{A}$. We call a sequence σ *negative* if

$$(-1)^{\text{number of twice repeated symbols in }\sigma}$$

is negative, and $\mathcal{A}^-(\subset\mathcal{A})$ contains only the negative sequences. We show that $|\mathcal{A}^+| - |\mathcal{A}^-| = \{\sigma|\phi(\sigma)=\sigma\} = s_4(n)$.

Let ϕ be a map from $\mathcal{A}$ onto $\mathcal{A}$. For sequences σ such that nonterminal symbols of σ occur once or thrice in the form $X\dots X\dots X$ or $X\dots XX$, $\phi(\sigma)=\sigma$. Otherwise σ has at least one nonterminal symbol that occurs twice or thrice as XXX or $XX\dots X$. In order of appearance of such symbols, assign 1 and 0 for twice- and thrice-repeated symbols, respectively, thus obtaining a k-digit binary number where k is the number of symbols that are twice repeated or thrice repeated with the patterns XXX and $XX\dots X$. According to the ordering of reflected Gray codes, ϕ takes the binary number associated with a given sequence σ and maps the $2n$th (respectively, $(2n+1)$th) binary number to the $(2n+1)$th (respectively, $2n$th) binary number. For the entry in the binary number where a change takes place, use the scheme for $0\leftrightarrow 1$

$$XXX\leftrightarrow XX,\qquad\qquad XX\dots X\leftrightarrow X\dots X$$

to replace the occurrence of the symbol corresponding to the change of the digit in the binary number. Clearly ϕ is a sign-reversing involution. Hence $|\mathcal{A}^+| - |\mathcal{A}^-| = \{\sigma|\phi(\sigma)=\sigma\}$. Since the fixed points of ϕ are the sequences where nonterminal symbols occur once or thrice in $X\dots X\dots X$ or $X\dots XX$, the identification of XX to X yields all 4-Stockhausen sequences. □

The ideas common to both proofs are first the grouping of ordered partitions of $1,2,\dots,r$ with the same number of parts, and the identification of blocks of the same symbol to one occurrence of the symbol after cancellation of positive and negative objects. The properties of $a_1,a_2,\dots,a_r$ guarantee that after grouping the partitions of $1,2,\dots,r$ which have the same number of parts, exactly one partition of every number of parts with multiplicity 1 remains. We put the properties for the a_is together in the lemma following the definition.

DEFINITION. *Let M_r (respectively, M_r^-) be the $r\times r$ upper triangular matrix such that the (i,j)th entry is $\binom{j}{i}$ (respectively, $(-1)^{i+j}\binom{j}{i}$).*

Note that $M_r M_r^- = M_r^- M_r = I_{(r\times r)}$ [R, Chapter 1].

LEMMA 0.3. *Let* $a_0 + a_1 w + a_2 \frac{w^2}{2!} + \cdots + a_r \frac{w^r}{r!} = [t^r]\frac{1}{1-t}e^{wt/(1+t)}$. *Then* $a_0 = 1$,

$$M_r^- \begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix} = \begin{bmatrix}a_1\\a_2\\\vdots\\a_r\end{bmatrix}, \quad \textit{and} \quad M_r \begin{bmatrix}a_1\\a_2\\\vdots\\a_r\end{bmatrix} = \begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}. \tag{6}$$

The first part of (6) yields

$$a_i = \sum_{j=0}^{r} (-1)^{i+j}\binom{j}{i}, \quad (i = 1, 2, \ldots, r),$$

a more explicit expression than the original definition of a_i.

Proof. Since $[t^r]\frac{1}{1-t}e^{wt/(1+t)}$ is the sum of the coefficients of $t^0, t^1, \ldots, t^r$ in the expression $e^{wt/(1+t)}$, and

$$\begin{aligned} e^{wt/(1+t)} &= 1 + w\left(\frac{t}{1+t}\right) + \frac{w^2}{2!}\left(\frac{t}{1+t}\right)^2 + \cdots \\ &= 1 + w(t - t^2 + t^3 - \cdots) + \frac{w^2}{2!}(t^2 - 2t^3 + 3t^4 - \cdots) + \cdots, \end{aligned}$$

the first result follows upon expanding $(\frac{t}{1+t})^k$. The second equation follows from the fact that M_r and M_r^- are inverses of each other. □

Hence, after grouping ordered partitions according to the number of parts and taking signed multiplicity $(a_1, a_2, \ldots, a_r)$ into consideration, we get exactly one representative of every number of parts. Thus the number $s_r(n)$ of r-Stockhausen sequences is the number of sequences such that the final symbol occurs r times in nonadjacent places, and all other symbols occur in the fashion prescribed by what remains after grouping. Replacing maximal blocks of repeated symbols by one symbol gives the desired Stockhausen sequences.

In the proof for $s_4(n)$, reflected Gray code listing is used to match + and − sequences. In the next example, a proof of $s_5(n)$ uses Gray codes with subscripts to account for the multiplicity. We show the grouping for proving $s_6(n)$ and sketch the proof.

PROPOSITION 0.4. *The number of* 5*-Stockhausen sequences is*

$$s_5(n) = n \sum_{i,j,k=0}^{i+j+k=n-1} \binom{n-1}{i, j, k, n-1-i-j-k} \times \frac{(4i+3j+2k)!}{4!^i}\left(\frac{-2}{3!}\right)^j \left(\frac{2}{2!}\right)^k \binom{4i+3j+2k}{4}. \tag{7}$$

Proof. Let $\mathcal{M}$ be a multiset of sequences on n symbols such that

(1) the terminal symbol occurs 5 times nonadjacently,

(2) all other symbols occur $0, 2, 3$, or 4 times without adjacency restrictions,

(3) the multiplicity of a sequence is

$$2^{\text{number of twice and thrice repeated symbols}}.$$

Then the cardinality of $\mathcal{M}$ is $s_5^+(n)$. Define $\mathcal{M}_{ab} \subset \mathcal{M}$ to be the multiset of sequences such that at least one symbol occurs with a pattern in

$$\Pi = \{(2), (1,1), (3), (2,1), (1,2), (1,1,1), (2,2), (1,3), (1,2,1), (1,1,2)\}.$$

For each sequence in $\mathcal{M}_{ab}$ of multiplicity 2^k where $l (\geq k)$ symbols occur with patterns in Π, associate a subscripted binary number for each copy of the sequence as follows: assign from left to right according to the occurrence of a symbol 1 to thrice occurring symbols and 0 to the others with patterns in Π, and subscripts a and b to 1 and 0 according to the scheme

$$\mu := \left\{ \begin{array}{rrr} & (3) \mapsto 1_a \text{ and } 1_b, & (2,2) \mapsto 0_a, \\ (2) \mapsto 0_a \text{ and } 0_b, & (2,1) \mapsto 1_a \text{ and } 1_b, & (1,3) \mapsto 0_b, \\ (1,1) \mapsto 0_a \text{ and } 0_b, & (1,2) \mapsto 1_a \text{ and } 1_b, & (1,2,1) \mapsto 0_a, \\ & (1,1,1) \mapsto 1_a \text{ and } 1_b, & (1,1,2) \mapsto 0_b. \end{array} \right\}$$

Thus we have distinguished each copy of the same sequence with multiplicity 2^k by associating to the copies 2^k l-digit subscripted binary numbers. Call the set of subscripted binary numbers for a particular sequence σ, $\mathcal{M}_{ab}^{\sigma}$.

We assign $+$ and $-$ signs to sequences in $\mathcal{M}$ and define a sign-reversing involution $\phi : \mathcal{M} \mapsto \mathcal{M}$. The sequences in $\mathcal{M} \setminus \mathcal{M}_{ab}$ are $+$ objects. For $\sigma \in \mathcal{M}_{ab}$, the sign of σ is

$$(-1)^{\text{number of thrice repeated symbols}}.$$

Let $\phi : \mathcal{M} \mapsto \mathcal{M}$ be as follows: $\phi(\sigma) = \sigma$ if $\sigma \in \mathcal{M} \setminus \mathcal{M}_{ab}$; otherwise consider $\mathcal{M}_{ab}^{\sigma}$, the set of binary numbers with subscripts for some sequence $\sigma \in \mathcal{M}_{ab}$. According to the reflected Gray code order, and for all elements of $\mathcal{M}_{ab}^{\sigma}$, ϕ maps the $2n$th (respectively, $(2n+1)$th) binary number to the $(2n+1)$th (respectively, $2n$th) binary number keeping the subscripts unchanged. Then for each binary number with subscripts, use the map

$$\left\{ \begin{array}{l} \quad\ (2) \xleftrightarrow[b]{a} (3) \\ (1,1) \xleftrightarrow[b]{a} (2,1) \\ (1,3) \xleftrightarrow{b} (1,2) \xleftrightarrow{a} (2,2) \\ (1,1,2) \xleftrightarrow{b} (1,1,1) \xleftrightarrow{a} (1,2,1) \end{array} \right\}$$

to obtain a sequence τ with a prescribed subscripted binary number in $\mathcal{M}_{ab}^{\tau}$. The map ϕ is a sign-reversing involution because the sign of a sequence is the number of 1's in the binary number, and Gray code listing is used.

Therefore $s_5(n)$ is the cardinality of the sequences fixed by ϕ, namely, $\mathcal{M} \setminus \mathcal{M}_{ab}$. But such sequences have multiplicity 1, and all nonterminal symbols occur not at all or four times in the fashion {(4), (3,1), (2,1,1), (1,1,1,1)}. Replacing maximal blocks of symbols $y \dots y$ by y, we get all 5-Stockhausen sequences. $\square$

The following is a grouping of ordered partitions for $s_6(n)$. Note that signed ordered partitions are cancelled in each group, and exactly one multiplicity-one partition of every number of parts remains. In each group, partitions with a negative multiplicity are assigned 1, and the others 0. The number of subscripts for a group is the number of negative (or positive) partitions. Finally, replacing maximal blocks of symbols $y \dots y$ by y, we get all 6-Stockhausen sequences.

Multiplicity	Ordered partitions						
$a_1 = 1$	1						
$a_2 = -2$	2	1,1					
$a_3 = 4$	3	2,1		1,1,1			
		1,2					
$a_4 = -3$	4	3,1		2,1,1		1,1,1,1	
		2,2		1,2,1			
		1,3		1,1,2			
$a_5 = 1$	5	4,1	1,4	3,1,1	1,2,2	2,1,1,1	1,1,1,1,1
		3,2		1,3,1		1,2,1,1	
		2,3		1,1,3		1,1,2,1	
				2,2,1		1,1,1,2	
				2,1,2			

Acknowledgments. The author would like to thank D. M. Jackson for many helpful discussions, C. D. Godsil for directing her to [R], the referees' many valuable comments, and M. Lemvig Hansen for TEXing complicated tables.

REFERENCES

[RY] R. C. Read and L. Yen, *A note on the Stockhausen problem*, J. Combin. Theory Ser. A., 76 (1996), pp. 1–10.

[R] J. Riordan, *An Introduction to Combinatorial Analysis*, John Wiley & Sons, New York, 1958.

[SW] D. Stanton and D. White, *Constructive Combinatorics*, Undergraduate Texts in Mathematics, Springer-Verlag, New York, 1986.

[S] K. Stockhausen, *nr.* 7 *Klavierstück* XI, 12654 LW, Universal Edition, London, 1979.

[W] H. S. Wilf, *Combinatorial Algorithms: An Update*, SIAM, Philadelphia, 1989.

[Y] L. Yen, *A Symmetric Functions Approach to Stockhausen's Problem*, Electron. J. Combin., http://www.combinatorics.org (3 (1996)).

SIAM J. DISCRETE MATH.
Vol. 10, No. 3, pp. 505–514, August 1997

014

STEINER 2-EDGE CONNECTED SUBGRAPH POLYTOPES ON SERIES-PARALLEL GRAPHS[*]

MOURAD BAÏOU[†] AND ALI RIDHA MAHJOUB[†]

Abstract. Given a graph $G = (V, E)$ with weights on its edges and a set of specified nodes $S \subseteq V$, the Steiner 2-edge survivable network problem is to find a minimum weight subgraph of G such that between every two nodes of S there are at least two edge-disjoint paths. This problem has applications to the design of reliable communication and transportation networks. In this paper, we give a complete linear description of the polytope associated with the solutions to this problem when the underlying graph is series-parallel. We also discuss related polyhedra.

Key words. Steiner 2-edge connected subgraphs, polytopes, series-parallel graphs

AMS subject classifications. 05C85, 90C27

PII. S0895480193259813

1. Introduction.

A graph $G = (V, E)$ is said to be *k-edge* (*respectively, k-node*) *connected* ($1 \le k \le |V| - 1$) if for any pair of nodes $i,\ j \in V$ there are at least k edge-disjoint (respectively, node-disjoint) paths from i to j. Let $G = (V, E)$ be a graph and $w \in R^E$ a weight vector associated with the edges of G. The weight of a subgraph of G is the sum of the weights of its edges. Given a subset of distinguished nodes $S \subseteq V$, called *terminals*, *the Steiner 2-edge survivable network problem* (STESNP) is the problem of finding a minimum weight subgraph of G spanning S such that between every two nodes i, j of S there are at least two edge-disjoint paths between i and j. The STESNP has applications to the design of reliable communication and transportation networks [5], [25], [26].

In this paper, we discuss the polytope associated with the solutions to this problem. We give a complete linear description of this polytope when the graph is series-parallel.

The STESNP is NP-hard in general. It has been shown to be polynomially solvable in some special cases of graphs. In [28], [29], Winter devised linear time algorithms to solve the STESNP in Halin graphs [28] and series-parallel graphs [29]. Actually, Winter considers the following more general problem called the *general Steiner problem:* Given a set $S \subseteq V$ and an integer $(|S|, |S|)$-matrix $R = (R_{ij})$ (defining certain pairwise connectivity requirements), find a minimum weight subgraph spanning S such that between every pair (i, j) of nodes in S there are at least R_{ij} edge (node)-disjoint paths. He showed that this problem can be solved in linear time if the graph is series-parallel or a Halin graph. This problem has been considered later by Grötschel and Monma [18] and Grötschel, Monma, and Stoer [19], [20], [21] in the framework of a more general model. In particular, Grötschel, Monma, and Stoer studied polyhedral aspects of that model and devised cutting plane algorithms.

Given a graph $G = (V, E)$ and a node subset $W \subseteq V$ of G, the set of edges having one endnode in W and the other in $V \setminus W$ is called a *cut* of G and denoted by $\delta(W)$.

[*] Received by the editors November 10, 1993; accepted for publication (in revised form) July 19, 1996.

http://www.siam.org/journals/sidma/10-3/25981.html

[†] Laboratoire SPO, Département d'Informatique, Université de Bretagne Occidentale, 6 Avenue Victor Le Gorgeu, B. P. 809, 29285 Brest Cedex, France (baiou@univ-ubs.fr, Ridha.Mahjoub@univ-brest.fr).

If $v \in V$ is a node of G, then we write $\delta(v)$ for the cut $\delta(\{v\})$ and we say that $\delta(v)$ is defined by v. If a cut contains k edges, it is also called a *k-edge cut set*.

Let $G = (V, E)$ be a graph. Let $x(e)$ be a variable associated with each edge e and for an edge subset $F \subseteq E$, the 0-1 vector $x^F \in R^E$ with $x^F(e) = 1$ if $e \in F$ and $x^F(e) = 0$ otherwise is called the *incidence vector* of F. For any subset of edges $F \subseteq E$, we define $x(F) = \sum_{e \in F} x(e)$. If $W \subseteq V$, then we denote by $E(W)$ the set of edges having both endnodes in W. The STESNP can be formulated as the following integer linear program.

$$\begin{array}{lll} & \text{Min } wx & \\ & \text{subject to} & \\ (1.1) & 0 \le x(e) \le 1 & \text{for all } e \in E, \\ (1.2) & x(\delta(W)) \ge 2 & \text{for all } W \subseteq V,\ S \ne W \bigcap S \ne \emptyset, \\ (1.3) & x(e) \in \{0, 1\} & \text{for all } e \in E. \end{array}$$

Let

$$\text{STESNP}(G, S) = \text{conv}\{x \in R^E \ \mid x \text{ satisfies (1.1), (1.2), and (1.3)}\}$$

be the polytope associated with the STESNP.

Using a polynomial time algorithm for the maximum flow problem [10], [12] and the famous maximum flow-minimum cut theorem (cf. Ford and Fulkerson [14]), one can solve in polynomial time the separation problem for inequalities (1.2) (the problem that consists to determine whether a given solution x satisfies the inequalities (1.2), and if not, to find an inequality among (1.2) which is violated by x). This implies, from the ellipsoid method [17], that there is a polynomial time algorithm for solving STESNP whenever STESNP(G, S) is completely described by the inequalities (1.1) and (1.2). Also one can obtain an equivalent extended compact formulation for the system given by (1.1) and (1.2) using the max flow-min cut theorem. This yields a further polynomial time algorithm for solving the STESNP when STESNP(G, S) is described by these inequalities.

In this paper, we will show that if the graph is series-parallel, then the polytope STESNP(G, S) is given by inequalities (1.1) and (1.2).

To the best of our knowledge, the STESNP(G, S) has not been considered in the literature. However some special cases received much attention. In particular, the case where $S = V$ has been extensively investigated.

For $S = V$, Mahjoub [22] gave a complete description of STESNP(G, S) in the case where the graph is series-parallel and he introduced a large class of facet defining inequalities for the polytope STESNP(G, S) called the odd-wheel inequalities. This class of facet defining inequalities has been generalized by Grötschel, Monma, and Stoer [19] for more general polyhedra. In [2] Barahona and Mahjoub characterized the polytope STESNP(G, S) for Halin graphs. In [18] Grötschel and Monma discuss a general model related to the design of minimum-cost survivable networks. They discuss polyhedral aspects of this model and identify basic facets of the associated polyhedra. Grötschel, Monma, and Stoer [19], [20], [21] describe further classes of facets of these polyhedra, develop a cutting plane algorithm for the associated problem and present computational results. A complete survey of that model and related work is given in Stoer [26].

Coullard, Rais, Rardin, and Wagner [7], [8], [9] consider the Steiner 2-node connected subgraph polytope, that is the polytope, the extreme points of which are the incidence vectors of the edge sets of the 2-node connected subgraphs of G, spanning S. They give a complete description of that polytope when the graph is series-parallel [7]. In [9] they characterize the dominant of that polytope for the graphs noncontractible to W_4 (the wheel on five nodes). In [8] they devise linear time algorithms for the Steiner 2-node connected subgraph problem in the graphs noncontractible to W_4 and Halin graphs.

Related work can also be found in [4], [6], [13]. In [6] Cornuéjols, Fonlupt, and Naddef studied some related polyhedra to STESNP(G, S). They showed that when $S = V$ and G is series-parallel, the polyhedron given by the nonnegativity inequalities and the cut-inequalities is integral. Fonlupt and Naddef [13] characterized the class of graphs for which the system given by these inequalities defines the convex hull of the incidence vectors of the tours of G (a tour is a cycle going at least once through each node). Chopra [4] considers the polyhedron, the extreme points of which are the incidence vectors of the edge sets of the k-edge connected spanning subgraphs of G, when multiple copies of an edge may be considered. He characterized this polyhedron for the class of outerplanar graphs when k is odd.

In the next section, we describe the polytope STESNP(G, S) for series-parallel graphs, and we give some structural properties for the system of inequalities defining that polytope. In section 3 we prove our main result. Concluding remarks are given in section 4. The remainder of this section is devoted to more definitions and notations.

The graphs we consider are finite, undirected, connected, and may have multiple edges and loops. We denote a graph by $G = (V, E)$ where V is the *node set* and E is the *edge set* of G. If e is an edge with endnodes u and v, then we write $e = uv$.

A graph G is said to be *contractible* to a graph H if H may be obtained from G by a sequence of elementary removal and contractions of edges. A contraction consists of identifying a pair of adjacent vertices and of preserving all other vertices and of preserving all other adjacencies between vertices. A graph is called *series-parallel* [11] if it is not contractible to K_4 (the complete graph on four nodes). Note that if G is a series-parallel graph and G is contractible to a graph H, then H is series-parallel. It is easily seen that series-parallel graphs have the following property.

LEMMA 1. *Any connected series-parallel graph with more than two nodes and without nodes defining* 2*-edge cut sets contains multiple edges.*

If $G = (V, E)$ is a graph and $W \subseteq V$ is a subset of nodes, we denote by $G(W)$ the subgraph of G induced by W. For $W,\ W' \subseteq V,\ (W,\ W')$ denotes the set of edges having one endnode in W and the other in W'. If $F \subseteq E$, $V(F)$ will denote the set of the nodes of the edges of F. If $W \subseteq V$, we let $\overline{W} = V \setminus W$. Given a constraint $ax \geq \alpha,\ a \in R^E$ and a solution x^*, we will say that $ax \geq \alpha$ is *tight* for x^* if $ax^* = \alpha$. If $G = (V, E)$ is a graph and $e \in E$, $G - e$ will denote the graph obtained from G by removing e.

2. The polytope STESNP(G,S) of a series-parallel graph.

Let $G = (V, E)$ be a graph and $S \subseteq V$ a set of terminals. We will suppose $|S| \geq 2$, (if $|S| = 1$, then an optimal solution to the problem STESNP would consist of the edges of negative weights). Let $P(G, S)$ denote the polytope given by inequalities (1.1) and (1.2). These inequalities will be called, respectively, *trivial* and *Steiner-cut inequalities.* A cut corresponding to a Steiner-cut inequality will be called *Steiner-cut.* Given a Steiner-cut $\delta(W)$ and a solution x for which the corresponding Steiner-cut inequality is tight, we will also say that $\delta(W)$ is a Steiner-cut tight for x.

Our main result is the following.

THEOREM 2. *If $G = (V, E)$ is a series-parallel graph and $S \subseteq V$ a set of terminals,* *then STESNP$(G, S) = P(G, S)$.*

The proof of this theorem will be given in the following section. In what follows, we are going to discuss some properties of the extreme points of the polytope $P(G, S)$. These properties will be useful in the sequel. First we give a technical lemma.

LEMMA 3. *Let x be a solution of $P(G, S)$. If $\delta(W_1)$ and $\delta(W_2)$ are two Steiner-cuts tight for x and $(W_1 \cap W_2) \cap S \neq \emptyset$ and $(\overline{W_1 \cup W_2}) \cap S \neq \emptyset$ (respectively, $(W_1 \setminus W_2) \cap S \neq \emptyset$ and $(W_2 \setminus W_1) \cap S \neq \emptyset$), then $\delta(W_1 \cap W_2)$ and $\delta(\overline{W_1 \cup W_2})$ (respectively, $\delta(W_1 \setminus W_2)$ and $\delta(W_2 \setminus W_1)$) are two Steiner-cuts tight for x, and $x(W_1 \setminus W_2,\ W_2 \setminus W_1) = 0$ (respectively, $x(W_1 \cap W_2, \overline{W_1 \cup W_2}) = 0$.*

Proof. Since $\delta(W_1)$ and $\delta(W_2)$ are tight for x we have

$$\begin{aligned} 4 &= x(\delta(W_1)) + x(\delta(W_2)) \\ &= x(\delta(W_1 \cap W_2)) + x(\delta(W_1 \cup W_2)) + 2x(W_1 \setminus W_2, W_2 \setminus W_1) \\ &\geq x(\delta(W_1 \cap W_2)) + x(\delta(W_1 \cup W_2)) \\ &\geq 4. \end{aligned}$$

The two last inequalities follow from the fact that $x(e) \geq 0$ for all $e \in E$ and $\delta(W_1 \cap W_2)$ and $\delta(W_1 \cup W_2)$ are both Steiner-cuts. This implies that all the above inequalities are satisfied at equality. Consequently, $x(\delta(W_1 \cap W_2)) = x(\delta(W_1 \cup W_2)) = 2$ and $x(W_1 \setminus W_2, W_2 \setminus W_1) = 0$.

If $(W_1 \setminus W_2) \cap S \neq \emptyset$ and $(W_2 \setminus W_1) \cap S \neq \emptyset$, then the cuts $\delta(W_1 \setminus W_2)$ and $\delta(W_2 \setminus W_1)$ are Steiner-cuts and in a similar way, we obtain that these cuts are tight for x and $x(W_1 \cap W_2, \overline{W_1 \cup W_2}) = 0$. □

If x is an extreme point of $P(G, S)$, then there exist two edge subsets, $E^0,\ E^1 \subseteq E$ of G and a family of Steiner-cuts $\{\delta(W_i),\ i = 1, ..., l\}$ such that x is the unique solution of the system

$$(2.1) \qquad \begin{cases} x(e) = 0 & \text{for all } e \in E^0, \\ x(e) = 1 & \text{for all } e \in E^1, \\ x(\delta(W_i)) = 2 & \text{for } i = 1, \ldots, l, \end{cases}$$

where $|E^0| + |E^1| + l = |E|$.

LEMMA 4. *Let $x \in R^E$ be a solution of $P(G, S)$ such that $x(e) > 0$ for all $e \in E$. If $\delta(W)$ is a Steiner-cut tight for x, then $G(W)$ and $G(\overline{W})$ are both connected.*

Proof. Suppose, for instance, that $G(\overline{W})$ is not connected. Let $\overline{W}^1$, $\overline{W}^2$ be a partition of $\overline{W}$ such that $(\overline{W}^1, \overline{W}^2) = \emptyset$. Since G is connected, it follows that $(W, \overline{W}^1) \neq \emptyset \neq (W, \overline{W}^2)$. From the hypothesis we then have

$$(2.2) \qquad x(W, \overline{W}^1) > 0, x(W, \overline{W}^2) > 0.$$

In addition, since $\overline{W} \cap S \neq \emptyset$, we may, without loss of generality (w.l.o.g.), assume that $\overline{W}^1 \cap S \neq \emptyset$. Hence $\delta(\overline{W}^1)$ is a Steiner-cut of G. However, as

$$x(\delta(W)) = x(W, \overline{W}^1) + x(W, \overline{W}^2) = 2,$$

it follows by (2.2) that $x(\delta(\overline{W}^1)) = x(W, \overline{W}^1) < 2$, a contradiction. □

The following remark will be used frequently in the next section.

Remark 5. Let $G' = (V', E')$ be a graph obtained from G by contracting a connected edge subset $F \subseteq E$. Let $S' = (S \setminus V(F)) \cup \{s'\}$ if $S \cap V(F) \neq \emptyset$ and

$S' = S$ if not, where s' is the node that arises in the contraction of F. Let x' be the restriction of x on G'. Then x' is a solution of $P(G', S')$.

Proof. Obviously, x' satisfies the inequalities (1.1). Furthermore, since any Steiner-cut $\delta(W)$ of G' with respect to S' is a Steiner-cut of G with respect to S, it follows that inequalities (1.2) are also satisfied by x'. □

3. Proof of Theorem 2. The proof is by induction on the number of edges. The theorem is trivially true for a graph with no more than two edges. Suppose it is true for any series-parallel graph with no more than m edges and suppose G contains exactly $m+1$ edges. Let us assume that, on the contrary, $\text{STESNP}(G, S) \neq P(G, S)$. And let x be a fractional extreme point of $P(G, S)$. Also let us assume that, under the induction hypothesis, $|S|$ is maximum. That is, for any series-parallel graph $G' = (V', E')$ with $|E'| = m + 1$ and a set of terminals S' such that $|S'| > |S|$, we have $\text{STESNP}(G', S') = P(G', S')$. We have the following lemmas:

LEMMA 6. *$x(e) > 0$ for all $e \in E$.*

Proof. If e_0 is an edge such that $x(e_0) = 0$, then let x' be the point given by $x'(e) = x(e)$ for all $e \in E \setminus \{e_0\}$. Clearly, x' belongs to $P(G - e_0)$. Moreover x' is an extreme point of $P(G - e_0)$. Since x' is fractional, we have a contradiction. □

LEMMA 7. *Let x be an extreme point of $P(G, S)$ and $g = uv$ an edge of G such that $x(g) > 0$. Then there exist at least two constraints containing g in the system* (2.1) *defining x.*

Proof. System (2.1) must be of full rank and therefore there must exist at least one constraint containing g in the system (2.1). So, let us assume that there exists exactly one constraint of system (2.1) that contains g. Let (2.1)$'$ be the system obtained from (2.1) by deleting this constraint. Let $x' \in R^m$ be the solution given by $x'(e) = x(e)$ for all $e \in E \setminus \{g\}$. We claim that x' is fractional. In fact, this is clear if $x(g) = 1$. If not, then g belongs to a tight Steiner-cut and thus there must exist at least one more edge in G with a fractional value, which implies that x' is fractional. Moreover, x' is the unique solution of the system (2.1)$'$. Now let G' be the graph obtained from G by contracting g. Let $S' = (S - \{u, v\}) \bigcup \{w\}$ if $g \in E(S)$ and $S' = S$ if not, where w is the node arising from the contraction of g. By Remark 5 we have $x' \in P(G', S')$. Furthermore, note that the system (2.1)$'$ is included in $P(G', S')$. This implies that x' is an extreme point of $P(G', S')$. Since G' is series-parallel and has less edges than G this contradicts the induction hypothesis and thus our lemma is proved. □

LEMMA 8. *G does not contain a node defining a 2-edge cut set.*

Proof. Suppose that G contains a node v such that $\delta(v) = \{e_1, e_2\}$ where $e_1 = vw_1$ and $e_2 = vw_2$. We will distinguish two cases.

Case 1. $x(e_1) = x(e_2)$.

Let G' be the graph obtained from G by contracting e_1. Clearly, G' is series-parallel. Let x' be the restriction of x on G' and let $S' = (S \setminus \{v, w_1\}) \cup \{v'\}$ if $\{v, w_1\} \cap S \neq \emptyset$ and $S' = S$ if not, where v' is the node that arises in the contraction of e_1. By Remark 5, we have that x' belongs to $P(G', S')$. We claim that x' is an extreme point of $P(G', S')$. In fact, if this is not the case, then there are two solutions y^1 and y^2 of $P(G', S')$, $y^1 \neq y^2$ such that $x' = \frac{1}{2}(y^1 + y^2)$. Let x^1 and x^2 be the solutions given by

$$x^1(e) = \begin{cases} y^1(e) & \text{if } e \in E \setminus \{e_1\}, \\ y^1(e_2) & \text{if } e = e_1, \end{cases}$$

and

$$x^2(e) = \begin{cases} y^2(e) & \text{if } e \in \ E \setminus \{e_1\}, \\ y^2(e_2) & \text{if } e = e_1. \end{cases}$$

We claim that x^1 and x^2 both belong to $P(G, S)$. Clearly, both x^1 and x^2 satisfy the trivial inequalities. In what follows we show that they also satisfy inequalities (1.2). We show this for x^1, the proof for x^2 is identical.

Let $\delta(W)$ be a Steiner-cut of G. If $e_1 \notin \delta(W)$, then $\delta(W)$ is a Steiner-cut of G' with respect to S' and then $x^1(\delta(W)) = y^1(\delta(W)) \geq 2$. So suppose that $e_1 \in \delta(W)$. Also, suppose, w.l.o.g, that $v \in \overline{W}$. Hence $w_1 \in W$. We consider two cases.

Case 1.1. $v \in S$.

Since $\delta(v)$ is a Steiner-cut, it follows from inequalities (1.1) and (1.2) that $x(e_1) = x(e_2) = 1$. Hence

$$x^1(e_1) = x^1(e_2) = 1. \tag{3.1}$$

We claim that $w_2 \in S$. In fact, first remark that $x(\delta(Z)) \geq 2$ holds for every cut $\delta(Z)$ such that $S \subseteq Z$ and $w_2 \in \overline{Z}$. This is clear if w_1 (and w_2) belong to $\overline{Z}$. Now suppose that $w_1 \in Z$. Let $Z' = Z \setminus \{v\}$. Since $|S| \geq 2$ and $v \in S$, we have that $Z' \cap S \neq \emptyset$ and $\overline{Z'} \cap S \neq \emptyset$. Thus $\delta(Z')$ is a Steiner-cut of G. Moreover, we have $\delta(Z') = (\delta(Z) \setminus \{e_2\}) \cup \{e_1\}$. Since $x(e_1) = x(e_2)$ it then follows that $x(\delta(Z)) = x(\delta(Z')) \geq 2$.

Now if $w_2 \notin S$, then let $\overline{S} = S \cup \{w_2\}$. From the above remark, we have that x is an extreme point of $P(G, \overline{S})$. Since x is fractional and $|S| < |\overline{S}|$, this contradicts the maximality of S.

Thus $w_2 \in S$. Now if $e_1, e_2 \in \delta(W)$ then by (3.1), we have $x^1(\delta(W)) \geq 2$. If not, since $e_1 \in \delta(W)$, we have $\{e_1, e_2\} \cap \delta(W) = \{e_1\}$, and thus $w_2 \in \overline{W}$. Let $W' = (W \setminus \{w_1\}) \cup \{v'\}$. As v' and w_2 belong to S, $\delta(W')$ is a Steiner-cut of G'. Since $\delta(W') = (\delta(W) \setminus \{e_1\}) \cup \{e_2\}$, it follows by (3.1) that

$$x^1(\delta(W)) = x^1(\delta(W')) = y^1(\delta(W')) \geq 2. \tag{3.2}$$

Case 1.2. $v \notin S$.

First of all note that every constraint of type (1.2), with $e_1, e_2 \in \delta(W)$ is redundant in $P(G, S)$. Since $w_1 \in W$ and $v \in \overline{W}$ we may then suppose that $\{e_1, e_2\} \cap \delta(W) = \{e_1\}$. By setting $W' = (W \setminus \{w_1\}) \cup \{v'\}$, we obtain that $\delta(W')$ is a Steiner-cut in G' and that (3.2) holds.

In both cases, x^1 satisfies the inequality associated with $\delta(W)$, and thus $x^1 \in P(G, S)$. Consequently, $x^1, x^2 \in P(G, S)$. But $x = \frac{1}{2}(x^1 + x^2)$. Since $x^1 \neq x^2$, this contradicts the extremality of x.

Case 2. $x(e_1) \neq x(e_2)$.

Without loss of generality, we may suppose that $x(e_1) > x(e_2)$. Thus e_1 cannot belong to any Steiner-cut tight for x. In fact, first note that v cannot be in S. Otherwise $\delta(v)$ would be a Steiner-cut not satisfied by x which is impossible. Now suppose that there is a Steiner-cut $\delta(W)$ tight for x with $e_1 \in \delta(W)$. W.L.O.G., we may suppose $v \in W$. Then $\delta(W')$ where $W' = W \setminus \{v\}$ is a Steiner-cut of G. Moreover, $x(\delta(W')) = x((\delta(W) \setminus \{e_1\}) \cup \{e_2\}) = 2 - x(e_1) + x(e_2) < 2$, a contradiction. As a consequence, e_1 belongs to only one constraint of system (2.1), namely $x(e_1) = 1$. But this contradicts Lemma 7 and our lemma is proved. □

LEMMA 9. *G cannot contain two multiple edges f and g such that $x(f) = x(g) = 1$.*

Proof. Suppose the contrary. Let $G' = (V', E')$ be the graph obtained from G by contracting the edges f and g. Clearly, G' is series-parallel. Let $S' = (S \setminus \{u, v\}) \cup \{w\}$, if $S \cap \{u, v\} \neq \emptyset$ and $S' = S$ if not, where u and v are the endnodes of f and g and w is the node arising from the contraction of f and g. Let $x' \in R^{m-1}$ be the solution given by $x' = x(e)$ for all $e \in E \setminus \{f, g\}$. By Remark 5, x' is a solution of $P(G', S')$. Moreover x' is an extreme point of $P(G', S')$. Indeed, if this is not the case, then there must exist two solutions y^1 and y^2, $y^1 \neq y^2$, of $P(G', S')$ such that $x' = \frac{1}{2}(y^1 + y^2)$. Now consider the solutions $y^{1'}$, $y^{2'} \in R^{m+1}$ given by

$$y^{1'}(e) = \begin{cases} y^1(e) & \text{if } e \in E \setminus \{f, g\}, \\ 1 & \text{if } e \in \{f, g\}, \end{cases}$$

and

$$y^{2'}(e) = \begin{cases} y^2(e) & \text{if } e \in E \setminus \{f, g\}, \\ 1 & \text{if } e \in \{f, g\}. \end{cases}$$

It is clear that $y^{1'}$ and $y^{2'}$ both belong to $P(G, S)$. Also we have that $x = \frac{1}{2}(y^{1'} + y^{2'})$, a contradiction. Consequently, x' is an extreme point of $P(G', S')$. Since x' is fractional and $|E'| < |E|$, this contradicts the induction hypothesis. □

LEMMA 10. *G does not contain two multiple edges f and g such that $x(f) = 1$ and* $0 < x(g) < 1$.

Proof. Let us suppose the contrary. Let u and v be the endnodes of f and g. Since $x(g)$ is fractional, there must exist a Steiner-cut $\delta(W_1)$, $W_1 \subset V$, tight for x, and containing g (and f). From Lemma 4, it follows that $G(W_1)$ and $G(\overline{W}_1)$ are both connected. We consider two cases.

Case 1. $|W_1| \geq 2$, $|\overline{W}_1| \geq 2$.

Let G^1 and G^2 be the graphs obtained from G by contracting W_1 and $\overline{W}_1$, respectively. Since $G(W_1)$ and $G(\overline{W}_1)$ are connected, both graphs G^1 and G^2 are series-parallel. Let $S^1 = (S \cap \overline{W}_1) \cup \{s_1\}$ and $S^2 = (S \cap W_1) \cup \{s_2\}$ where s_1 and s_2 are the nodes arising from the contractions of W_1 and $\overline{W}_1$, respectively. Since G^1 and G^2 contain less edges than G, by the induction hypothesis, $P(G^1, S^1)$ and $P(G^2, S^2)$ are both integral. Let x^1 and x^2 be the restrictions of x on G^1 and G^2, respectively. By Remark 5, x^1 and x^2 are solutions of $P(G^1, S^1)$ and $P(G^2, S^2)$, respectively. Hence there must exist two integral solutions y^1 and y^2 of $P(G^1, S^1)$ and $P(G^2, S^2)$ such that every constraint of $P(G^1, S^1)$ (respectively, $P(G^2, S^2)$) that is tight for x^1 (respectively, x^2) is also tight for y^1 (respectively, y^2). In particular we have $y^1(\delta(W_1)) = y^2(\delta(W_1)) = 2$ and $y^1(f) = y^2(f) = 1$. Moreover, since $0 < x^1(g) = x^2(g) = x(g) < 1$, y^1 and y^2 can be chosen so that $y^1(g) = y^2(g) = 1$. Consequently, $y^1(\delta(W_1) \setminus \{f, g\}) = y^2(\delta(W_1) \setminus \{f, g\}) = 0$. Now consider the solution $x^* \in R^{m+1}$ given by

$$x^*(e) = \begin{cases} y^1(e) & \text{if } e \in E(\overline{W}_1), \\ y^2(e) & \text{if } e \in E(W_1), \\ 1 & \text{if } e \in \{f, g\}, \\ 0 & \text{otherwise.} \end{cases}$$

We claim that every constraint of $P(G, S)$ that is tight for x is also tight for x^*.

Let $e \in E$ such that $x(e) = 1$. Then e belongs either to $E(W_1)$ or $E(\overline{W}_1)$ or $e = f$. If $e \in E(W_1)$ (respectively, $e \in E(\overline{W}_1)$) then, $x^*(e) = y^2(e) = x^2(e) = 1$ (respectively, $x^*(e) = y^1(e) = x^1(e) = 1$). From Lemma 6 it then follows that every inequality of type (1.1) that is tight for x is also tight for x^*.

Consider now a Steiner-cut $\delta(W)$ tight for x.

(a) If $W \subseteq W_1$, then $x(\delta(W)) = x^2(\delta(W)) = y^2(\delta(W)) = 2$. Since $x^*(\delta(W)) = y^2(\delta(W))$, we obtain that $\delta(W)$ is tight for x^*.

(b) If $W \subseteq \overline{W}_1$, we obtain similarly that $\delta(W)$ is tight for x^*.

(c) Suppose that $W \not\subset W_1$, $W_1 \not\subset W$ and $W \cap W_1 \neq \emptyset$.

(c.1) Consider first the case where at least one of the sets $(W_1 \setminus W) \cap S$ and $(W \setminus W_1) \cap S$ is empty. Since both $\delta(W)$ and $\delta(W_1)$ are Steiner-cuts, it follows that $(W_1 \cap W) \cap S \neq \emptyset$ and $(\overline{W_1 \cup W}) \cap S \neq \emptyset$. Hence by Lemma 3, $\delta(W_1 \cap W)$ and $\delta(\overline{W_1 \cup W})$ are both Steiner-cuts tight for x and $x(W_1 \setminus W, W \setminus W_1) = 0$. By Lemma 6, this implies that $(W_1 \setminus W, W \setminus W_1) = \emptyset$. Furthermore, since $(W_1 \cap W) \subset W_1$, and $(\overline{W_1 \cup W}) \subset \overline{W}_1$, from Cases (a) and (b) above, it follows that $\delta(W_1 \cap W)$ and $\delta(\overline{W_1 \cup W})$ are both tight for x^*. Thus we have

$$x^*(\delta(W)) = x^*(\delta(W_1 \cap W)) + x^*(\delta(\overline{W_1 \cup W})) - x^*(\delta(W_1)) = 2 + 2 - 2 = 2.$$

And the constraint $x(\delta(W)) \geq 2$ is tight for x^*.

(c.2) If $(W_1 \setminus W) \cap S \neq \emptyset$ and $(W \setminus W_1) \cap S \neq \emptyset$, then by Lemma 3 we have that $\delta(W_1 \setminus W)$ and $\delta(W \setminus W_1)$ are both Steiner-cuts tight for x and $x(W_1 \cap W, \overline{W_1 \cup W}) = 0$. Using this, we obtain in a similar way as in c.1) that $x(\delta(W) \geq 2$ is also tight for x^*.

Consequently, every constraint of $P(G, S)$ that is tight for x is also tight for x^*. Since $x \neq x^*$, this contradicts the fact that x is an extreme point of $P(G, S)$.

Case 2. $|W_1| = 1$.

By Lemma 7, there must exist a further Steiner-cut $\delta(W_2)$ tight for x and containing g (and f). If $|W_2| \geq 2$, $|\overline{W}_2| \geq 2$ then Case 1 applies. Thus let us assume, for instance, that $|W_2| = 1$. Hence we may suppose that $W_1 = \{u\}$ and $W_2 = \{v\}$. This implies that $(V \setminus \{u, v\}) \cap S = \emptyset$. Otherwise $\delta(V \setminus \{u, v\})$ would be a Steiner-cut not satisfied by x, a contradiction. Hence any Steiner-cut of G contains both edges f and g. Furthermore, note that every Steiner-cut tight for x contains only one edge with integer value, namely f. Now consider the solution $\bar{x} \in R^E$ defined as

$$\bar{x}(e) = \begin{cases} 1 & \text{if } x(e) = 1 \text{ or } e = g, \\ 0 & \text{if not.} \end{cases}$$

We have that $\bar{x} \in P(G, S)$. Moreover any inequality of $P(G, S)$ which is tight for x is also tight for $\bar{x}$. Since $x \neq \bar{x}$, this contradicts the fact that x is an extreme point of $P(G, S)$, which achieves the proof of our lemma. □

From Lemmas 1, 8, 9, and 10 it follows that G contains two multiple edges f and g such that $0 < x(f) < 1$ and $0 < x(g) < 1$. Let x' be the solution such that

$$x'(e) = \begin{cases} x(e) + \epsilon & \text{if } e = g, \\ x(e) - \epsilon & \text{if } e = f, \\ x(e) & \text{otherwise,} \end{cases}$$

where ϵ is a scalar sufficiently small. Since any cut of G either contains both edges f and g or none of them, it follows that x' is also a solution of system (2.1). Since $x \neq x'$, we have a contradiction, and the proof of our theorem is complete. □

4. Concluding remarks. We have studied the Steiner 2-edge survivable network problem and have given a complete linear description of the associated polytope when the underlying graph is series-parallel. We have shown that in this case, the polytope is given by the trivial inequalities and the Steiner cut inequalities.

The following related problem, called the *Steiner 2-edge connected subgraph problem* (STECSP) has also been studied. Given a graph $G = (V, E)$ with weights on its edges and a set of terminals $S \subseteq V$, find a minimum 2-edge connected subgraph of G, spanning S. Note that any solution of STECSP is also a solution of STESNP. Moreover, if the weights are positive, then an optimal solution of STESNP is also an optimal solution of STECSP. And if $S = V$, then both problems coincide.

As the STESNP, the STECSP is NP-hard in general. Wald and Colbourn [27] showed that the STECSP can be solved in polynomial time in outerplanar graphs. Also from [24], [29] it can be shown that this problem is polynomially solvable in the more general class of series-parallel graphs.

The STECSP has also been studied by Monma, Munson, and Pulleyblank [23] in the metric case, that is when the underlying graph $G = (V, E)$ is complete and the weight function satisfies the triangle inequality (i.e., $w(e_1) \leq w(e_2) + w(e_3)$ for every three edges e_1, e_2, e_3 defining a triangle in G). In particular, Monma, Munson, and Pulleyblank showed that in this case the weight of a minimum 2-edge connected spanning subgraph in $(S, E(S))$ is at most $\frac{4}{3}$ times the weight of a minimum 2-edge connected subgraph of G, spanning S. Further structural properties and worst case analysis are given in Frederickson and Ja'Ja' [15], Bienstock, Brickell, and Monma [3] and Goemans and Bertsimas [16].

If (W, F), $W \subseteq V$, is a 2-edge connected subgraph of G, spanning S, then x^F, the incidence vector of F satisfies the following inequalities:

$$x(\delta(W)) - 2x(e) \geq 0 \qquad \text{for all } W \subseteq V, S \subseteq W, e \notin E(W). \tag{4.1}$$

Inequalities (4.1) express the fact that for a cut $\delta(W)$ that leaves S on one side, any 2-edges connected subgraph spanning S and containing an edge from $E \setminus E(W)$ must contain at least two edges from $\delta(W)$.

Let STECSP(G, S) be the polytope associated with the STECSP, that is, the convex hull of the incidence vectors of the edge sets of all the 2-edge connected subgraphs of G spanning S. Let Q(G, S) be the system given by inequalities (1.1), (1.2), and (4.1). We have the following result; its proof uses similar techniques as that of Theorem 2.

THEOREM 11. *If $G = (V, E)$ is a series-parallel graph and $S \subset V$ a set of terminals, then STECSP(G,S)=Q(G,S).*

Proof. For the proof, see [1].

Acknowledgments. We are grateful to the referees for their helpful comments. We thank M. Didi Biha for very stimulating suggestions.

REFERENCES

[1] M. Baïou, *Le problème du sous graphe Steiner* 2*-arête connexe: Approche polyédrale*, Ph.D. dissertation, N 1639, Université de Rennes 1, Rennes, France, 1993.

[2] F. Barahona and A. R. Mahjoub, *On two-connected subgraph polytopes*, Discrete Math., 147 (1995), pp. 19–34.

[3] D. Bienstock, E. F. Brickell, and C. L. Monma, *On the structure of minimum weight k-connected spanning networks*, SIAM J. Discrete Math., 3 (1990), pp. 320–329.

[4] S. Chopra, *The k-edge connected spanning subgraph polyhedron*, SIAM J. Discrete Math., 7 (1994), pp. 245–259.

[5] N. Christofides and C. A. Whitlock, *Network synthesis with connectivity constraints-A survey*, in Oper. Res., 81, J. P. Brans, ed., North-Holland, Amsterdam, 1981, pp. 705–723.

[6] G. Cornuéjols, J. Fonlupt, and D. Naddef, *The traveling salesman problem on a graph and some related integer polyhedra*, Math. Programming, 33 (1985), pp. 1–27.

[7] R. Coullard, A. Rais, R. L. Rardin, and D. K. Wagner. *The* 2*-Connected-Steiner Subgraph Polytope for Series-Parallel Graphs*, Report CC-91-23, School of Industrial Engineering, Purdue University, West Lafayette, IN, 1991.

[8] R. Coullard, A. Rais, R. L. Rardin, and D. K. Wagner, *Linear-Time Algorithm for the* 2*-Connected Steiner Subgraph Problem on Special Classes of Graphs*, Report 91-25, School of Industrial Engineering, Purdue University, West Lafayette, IN, 1991.

[9] R. Coullard, A. Rais, R. L. Rardin, and D. K. Wagner, *The Dominant of the* 2*-Connected-Steiner Subgraph Polytope for* W_4*-Free Graphs*, Report 91-28, School of Industrial Engineering, Purdue University, West Lafayette, IN, 1991.

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

[11] R. J. Duffin, *Topology of series-parallel networks*, J. Math. Anal. Appl., 10 (1965), pp. 303–318.

[12] J. Edmonds and R. M. Karp, *Theoretical improvement in algorithm efficiency for network flow problems*, J. Assoc. Comput. Mach., 19 (1972), pp. 248–264.

[13] J. Fonlupt and D. Naddef, *The traveling salesman problem in graphs with some excluded minors*, Math. Programming, 53 (1992), pp. 147–172.

[14] L. R. Ford and D. R. Fulkerson, *Maximal flow through a network*, Can. J. Math., 8 (1956), pp. 399–404.

[15] G. N. Frederickson and J. Ja'Ja', *On the relationship between the biconnectivity augmentations and traveling salesman problem*, Theoret. Comput. Sci., 13 (1982), pp. 189–201.

[16] M. X. Goemans and D. J. Bertsimas, *Survivable networks, linear programming and the parsimonious property*, Math. Programming, 60 (1993), pp. 145–166.

[17] M. Grötschel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica, 1 (1981), pp. 70–89.

[18] M. Grötschel and C. Monma, *Integer polyhedra arising from certain network design problems with connectivity constraints*, SIAM J. Discrete Math., 3 (1990), pp. 502–523.

[19] M. Grötschel, C. Monma, and M. Stoer, *Facets for polyhedra arising in the design of communication networks with low-connectivity constraints*, SIAM J. Optim., 2 (1992), pp. 474–504.

[20] M. Grötschel, C. Monma, and M. Stoer, *Polyhedral approaches to network survivability*, in Reliability of Computer and Communication Networks, F. Roberts, F. Hwang, and C. L. Monma, eds., Discrete Mathematics and Computer Science, Vol. 5, AMS/ACM, Providence, RI, 1991, pp. 121–141.

[21] M. Grötschel, C. Monma, and M. Stoer, *Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints*, Oper. Res., 40 (1992), pp. 309–330.

[22] A. R. Mahjoub, *Two-edge connected spanning subgraphs and polyhedra*, Math. Programming, 64 (1994), pp. 199–208.

[23] C. L. Monma, B. S. Munson, and W. R. Pulleyblank, *Minimum-weight two connected spanning networks*, Math. Programming, 46 (1990), pp. 153–171.

[24] C. L. Monma and J. B. Sidney, *Sequencing with series-parallel precedence constraints*, Math. Oper. Res., 4 (1979), pp. 215–224.

[25] K. Steiglitz, P. Weiner, and D. J. Kleitman, *The design of minimum cost survivable networks*, IEEE Transactions and Circuit Theory, 16 (1969), pp. 455–460.

[26] M. Stoer, *Design of survivable networks*, Lecture Notes in Mathematics 1531, Springer-Verlag, New York, 1992.

[27] J. A. Wald and C. J. Colbourn, *Steiner trees in outerplanar graphs*, Congressus Numeratum, 36 (1982), pp. 15–22.

[28] P. Winter, *Generalized Steiner problem in Halin networks*, in Proc. 12th International Symposium on Mathematical Programming, MIT, Cambridge, MA, 1985.

[29] P. Winter, *Generalized Steiner problem in series-parallel networks*, J. Algorithms, 7 (1986), pp. 549–566.

SIAM J. DISCRETE MATH.
Vol. 10, No. 4, pp. 515–528, November 1997

001

THE ORDER DIMENSION OF PLANAR MAPS[*]

GRAHAM R. BRIGHTWELL[†] AND WILLIAM T. TROTTER[‡]

Abstract. This is a sequel to a previous paper entitled *The Order Dimension of Convex Polytopes*, by the same authors [*SIAM J. Discrete Math.*, 6 (1993), pp. 230–245]. In that paper, we considered the poset $\mathbf{P_M}$ formed by taking the vertices, edges, and faces of a 3-connected planar map $\mathbf{M}$, ordered by inclusion, and showed that the order dimension of $\mathbf{P_M}$ is always equal to 4. In this paper, we show that if $\mathbf{M}$ is any planar map, then the order dimension of $\mathbf{P_M}$ is still at most 4.

Key words. partially ordered sets, dimension, planar maps, planar graphs, convex polytopes

AMS subject classifications. 06A07, 05C35

PII. S0895480192238561

1. Introduction.

In this paper, we are concerned with planar maps. We shall allow loops and multiple edges, and we always consider a fixed representation of a graph in the plane. More formally, given a multigraph $G = (V, E)$, a *plane drawing* D of G is a representation of G by points and arcs in $\mathbf{R}^2$ in which two edges meet only at common vertices. A *planar map* $\mathbf{M}$ is a pair (G, D) consisting of a multigraph and a plane drawing thereof. In what follows, we do not distinguish between a vertex (edge) of G and the corresponding point (arc) of $\mathbf{R}^2$.

Deleting the vertices and edges of a planar map $\mathbf{M}$ from the plane leaves several connected components whose closures are the *faces* of $\mathbf{M}$. The unique unbounded face is called the *exterior face.* For the purposes of this paper, it is not treated in any special way.

Given a planar map $\mathbf{M}$, the planar dual $\mathbf{M}^*$ is defined in the usual way, taking a vertex F^* for each face F of $\mathbf{M}$, and, for each edge e of $\mathbf{M}$, an edge e^* in $\mathbf{M}^*$ joining the vertices of $\mathbf{M}^*$ corresponding to the two faces separated by e in $\mathbf{M}$. (In the special case where the edge e is a bridge, the dual edge e^* is a loop on the dual of the unique face containing e.) Then each vertex v of $\mathbf{M}$ corresponds to a face v^* in $\mathbf{M}^*$. If $\mathbf{M}$ is connected, then $\mathbf{M}^{**}$ is isomorphic to $\mathbf{M}$.

For a planar map $\mathbf{M}$, we form a poset $\mathbf{P_M}$ by taking the vertices, edges, and faces of $\mathbf{M}$ (including the exterior face), ordered by inclusion. See Figure 1.1 for an example of a planar map $\mathbf{M}$ and its associated poset $\mathbf{P_M}$. Let us note immediately that, if $\mathbf{M}$ is connected, the poset $\mathbf{P_{M^*}}$ associated with the dual map is just the dual poset $(\mathbf{P_M})^*$ (i.e., the set of vertices, edges, and faces ordered by reverse inclusion).

The *order dimension* $\dim(\mathbf{P})$ of a partial order $\mathbf{P}$ is the smallest number t such that $\mathbf{P}$ is the intersection of t linear orders on the same vertex set. The following result was proved in [1], answering a question of Reuter [3].

THEOREM 1.1. *For every* 3*-connected planar map* $\mathbf{M}$*,* $\dim(\mathbf{P_M}) = 4$. □

This result is to be compared with one due to Schnyder [4]: if G is any graph and $\mathbf{P}(G)$ is the poset formed from the vertices and edges of G, ordered by inclusion, then $\dim(\mathbf{P}(G)) \leq 3$ iff G is planar. If G is planar, and $\mathbf{M}$ is a map with underlying

[*] Received by the editors October 9, 1992; accepted for publication July 19, 1996.
http://www.siam.org/journals/sidma/10-4/23856.html

[†] Department of Mathematics, London School of Economics, Houghton Street, London WC2A 2AE, U. K. (g.r.brightwell@lse.ac.uk).

[‡] Department of Mathematics, Arizona State University, Tempe, AZ 82587-1804 (trotter@asu.edu). The research of this author was supported in part by the Office of Naval Research.

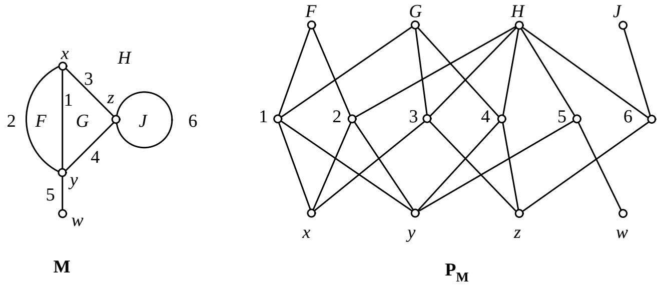


FIG. 1.1. *A planar map* $\mathbf{M}$ *and the poset* $\mathbf{P}_{\boldsymbol{M}}$

graph G, then $\mathbf{P}(G)$ is an induced subposet of our poset $\mathbf{P_M}$. Thus, although we do not refer to it again explicitly, Schnyder's work underpins much of what we do in this paper.

One reason for restricting attention to 3-connected planar maps in [1] was the connection with convex polytopes in $\mathbf{R}^3$: each convex polytope gives rise to a 3-connected planar map $\mathbf{M}$ and the poset $\mathbf{P_M}$ corresponds to the set of vertices, edges, and faces of the polytope, ordered by inclusion.

The main purpose of this paper is to prove the following result, extending Theorem 1.1 to general planar maps.

THEOREM 1.2. *Let* $\mathbf{M}$ *be a planar map, and let* $\mathbf{P_M}$ *be the poset of all vertices, edges, and faces of* $\mathbf{M}$ *ordered by inclusion. Then* $\dim(\mathbf{P_M}) \leq 4$.

For more information as to the origin of the problem, see [1], Reuter [3], or Schnyder [4].

In the course of proving Theorem 1.2, we shall use a result (Theorem 3.2) that is slightly stronger than Theorem 1.2 itself as the base case for an induction argument. However the machinery developed in [1] is used only in the proof of Theorem 3.2.

Before we begin, we need a few concepts from the theory of order dimension. For a comprehensive treatment of dimension theory for finite posets, we refer the reader to the monograph [6]. Other sources include the survey articles [2] and [5] and our previous paper [1]. Given a partial order $\mathbf{P}$, a set $\mathcal{R} = \{L_1, \ldots, L_t\}$ of linear extensions of $\mathbf{P}$ is called a *realizer* of $\mathbf{P}$ if the intersection of the L_i is exactly $\mathbf{P}$. Thus the order dimension of $\mathbf{P}$ is the minimum cardinality of a realizer.

An ordered pair (a, b) of elements of a partial order $\mathbf{P}$ is called a *critical pair* if the following three conditions hold:

(i) a and b are incomparable;
(ii) if $c < a$ in $\mathbf{P}$, then $c < b$; and
(iii) if $b < d$ in $\mathbf{P}$, then $a < d$.

An ordered pair (a, b) of elements of $\mathbf{P}$ is said to be *reversed* by a linear extension L if $b < a$ in L. It is fairly easy to see that a set $\{L_1, \ldots, L_t\}$ of linear extensions of $\mathbf{P}$ is a realizer if and only if every critical pair is reversed by some L_i.

If F is a face of $\mathbf{M}$ and x is a vertex not on F, then the pair (x, F) is a critical pair. We call this a *vertex-face critical pair* and extend the terminology in the obvious way. If all critical pairs of $\mathbf{P_M}$ are of this vertex-face type, we say that $\mathbf{M}$ is *well formed*. It is easy to see that every 3-connected planar map (with no loops or multiple edges) is well formed.

For a planar map $\mathbf{M}$, we define another partial order $\mathbf{Q_M}$ by taking just the vertices and faces of $\mathbf{M}$, ordered by inclusion. (Figure 1.2 shows the poset $\mathbf{Q_M}$ for the map $\mathbf{M}$ in Figure 1.1.) Evidently $\mathbf{Q_M}$ is an induced subposet of $\mathbf{P_M}$, and so $\dim(\mathbf{Q_M}) \leq \dim(\mathbf{P_M})$. The reverse inequality is not true in general, but it does hold whenever $\mathbf{M}$ is well formed.

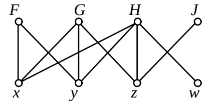
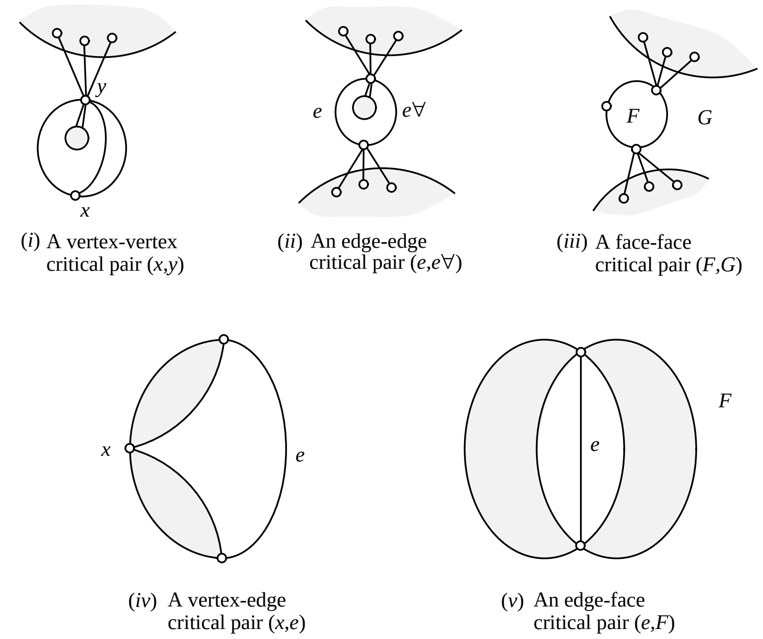


FIG. 1.2. *The poset* $\mathbf{Q_{\mathit{M}}}$

LEMMA 1.3. *Suppose that* $\mathbf{M}$ *is well formed. Then* $\dim(\mathbf{P_M}) = \dim(\mathbf{Q_M})$.

Proof. We have seen that $\dim(\mathbf{P_M}) \geq \dim(\mathbf{Q_M})$. Conversely, given a realizer $\{L_1, \ldots, L_t\}$ of $\mathbf{Q_M}$, we can insert the edges of $\mathbf{M}$ into each linear extension L_i in a way consistent with $\mathbf{P_M}$: this then gives a realizer of $\mathbf{P_M}$, since the critical pairs of $\mathbf{P_M}$ are all of vertex-face type and so are reversed by some L_i. □



(*i*) A vertex-vertex critical pair (*x*,*y*)

(*ii*) An edge-edge critical pair (*e*,*e*∀)

(*iii*) A face-face critical pair (*F*,*G*)

(*iv*) A vertex-edge critical pair (*x*,*e*)

(*v*) An edge-face critical pair (*e*,*F*)

FIG. 1.3. *Examples of critical pairs.*

For a general planar map $\mathbf{M}$, the poset $\mathbf{P_M}$ may have vertex-vertex, edge-edge, and face-face critical pairs, but only if $\mathbf{M}$ is not 2-connected: see Figure 1.3(i)–(iii). However, $\mathbf{Q_M}$ can have vertex-vertex or face-face critical pairs even if $\mathbf{M}$ is 2-connected; for instance, if x is a vertex of degree 2 with distinct neighbors y and z, then (x, y) and (x, z) are critical pairs in $\mathbf{Q_M}$.

If (e, e') is an edge-edge critical pair in $\mathbf{P_M}$, the two edges must share the same endpoints and separate the same faces, as in Figure 1.3(ii). This makes edge-edge critical pairs very easy to deal with: given a set $\{L_1, L_2, \ldots, L_k\}$ of linear extensions reversing all other critical pairs, we move e' to the place immediately above e in L_1, and to the place immediately below e in all the other L_i. This yields a realizer. Thus we may effectively ignore edge-edge critical pairs.

Even if $\mathbf{M}$ is 2-connected, $\mathbf{P_M}$ may have vertex-edge or edge-face critical pairs. See Figure 1.3(iv) and (v) for examples. If e is an edge in such a critical pair, we call e a *critical edge.* The following trivial observation will be useful later.

LEMMA 1.4. *Let e be an edge of a planar map $\mathbf{M}$. Then e cannot be in both a vertex-edge and an edge-face critical pair.* □

Before we begin the proof of Theorem 3.2, we must clarify what we mean by k-connectivity for planar maps. The definition we use is not quite the usual one, since it is appropriate for the concept to be invariant under duality. For instance, the map in Figure 1.4 should not be 3-connected, since its dual isn't 3-connected.

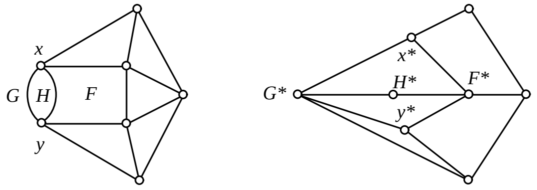


FIG. 1.4. *A map that is not 3-connected and its dual.*

The approach we adopt here is to define the connectivity of a planar map $\mathbf{M}$ to be the minimum of the connectivities of the underlying graphs of $\mathbf{M}$ and $\mathbf{M}^*$. Note that at least one of these graphs always contains a vertex of degree at most 3, so the only 4-connected maps are those with underlying graph K_4.

With the exception of a few graphs with at most three vertices, we have the following alternative characterizations. A map has connectivity 0 iff it is disconnected, connectivity 1 iff it is connected and has a cutvertex, and connectivity 2 iff either its underlying graph has connectivity 2 or it has a double edge, as in Figure 1.4.

If a map $\mathbf{M}$ with at least four vertices has connectivity exactly 2, then it has a pair $\{x, y\}$ of vertices and a pair $\{F, G\}$ of faces such that $\mathbf{R}^2 - (F \cup G \cup \{x, y\})$ falls into two components, neither of which is a single edge. We call $\{x, y, F, G\}$ a *separating system.* For instance, in Figure 1.4, $\{x, y, F, G\}$ is a separating system.

We shall approach Theorem 3.2 via the following intermediate result.

LEMMA 1.5. *Let $\mathbf{M}$ be a 2-connected planar map. Then* $\dim(\mathbf{Q_M}) \leq 4$.

The next section is devoted to the deduction of Theorem 3.2 from Lemma 1.5. Then in section 3 we prove Lemma 1.5. The basic idea involves modifying and combining families of linear extensions given to us from Theorem 1.1. However, the following observation gives some indication of the fundamental difference between the 3-connected case and the general case we are considering here.

For a 3-connected map $\mathbf{M}$, the poset $\mathbf{Q_M}$ is 4-irreducible, as shown in section 6 of [1]. Indeed, the proof of Theorem 1.1 was very much geared to proving that $\mathbf{Q_M}$ is "almost 3-dimensional": producing three linear extensions that are almost a realizer. But the poset $\mathbf{Q_M}$ for the map $\mathbf{M}$ in Figure 1.5 is not 4-irreducible; each critical pair (x_i, F_i) must be reversed by a different linear extension, so $\mathbf{Q_M}$ minus the outside face still has dimension 4. Thus, to prove Lemma 1.5 we shall have to make full use of the fact that we have four linear extentions to work with.

2. Reduction to the 2-connected case. We shall prove the following result, which clearly combines with Lemma 1.5 to give Theorem 3.2.

LEMMA 2.1. *If $\mathbf{M}$ is a planar map, then there exists a 2-connected planar map $\mathbf{M}_0$ such that* $\dim(\mathbf{P_M}) \leq \dim(\mathbf{Q_{M_0}})$.

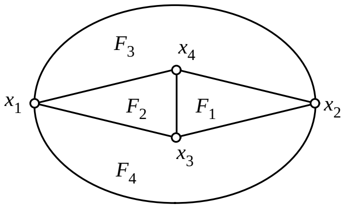


FIG. 1.5. *A map* $\mathbf{M}$ *for which* $\mathbf{Q}_{\mathbf{M}}$*is not* 4*-irreducible.*

Proof. If the map $\mathbf{M}$ is well formed and 2-connected, the result is immediate by Lemma 1.3. Thus, we shall consider in turn each of the ways in which $\mathbf{M}$ may fail to be 2-connected and well formed.

Our approach will be to construct a sequence of intermediate maps $\mathbf{M}_i$ from $\mathbf{M}$ such that a realizer of $\mathbf{P}_{\mathbf{M}_i}$ or of $\mathbf{Q}_{\mathbf{M}_i}$ can be converted into a realizer of $\mathbf{P}_{\mathbf{M}}$.

We illustrate the process by showing in Figure 2.1 the sequence $\mathbf{M}_i$ of maps generated by starting from the map $\mathbf{M}$ with two vertices and one loop.

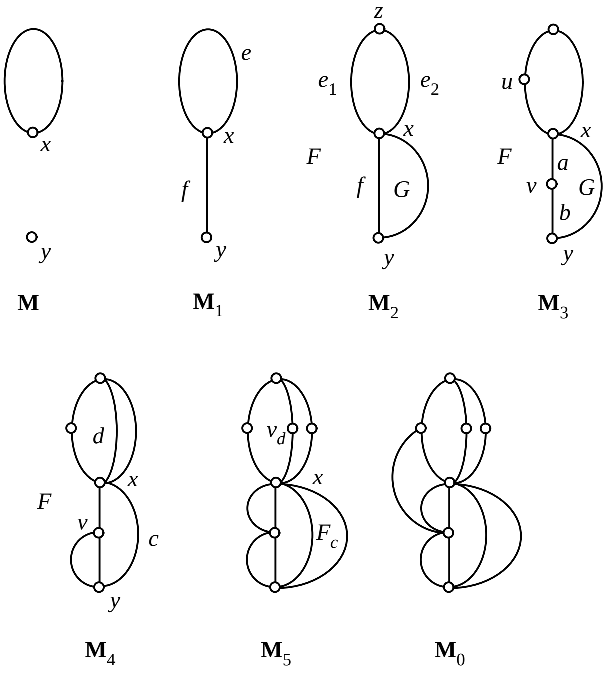


FIG. 2.1. *The proof of Lemma* 2.1.

(1) Making M connected. Given a planar map $\mathbf{M}$, we construct a connected map $\mathbf{M}_1$ from $\mathbf{M}$ by adding bridges between components as necessary. Clearly $\mathbf{P}_{\mathbf{M}}$ is an induced subposet of $\mathbf{P}_{\mathbf{M}_1}$, so $\dim(\mathbf{P}_{\mathbf{M}}) \leq \dim(\mathbf{P}_{\mathbf{M}_1})$.

(2) Destroying loops and vertices of degree 1. Suppose that, as in Figure 2.1, there is a loop e on a vertex x in $\mathbf{M}_1$. In this case, we form $\mathbf{M}_1'$ by subdividing e; i.e., we replace e by a vertex z and a pair of edges e_1 and e_2 joining x to z. Identifying e with e_1, we see that $\mathbf{P}_{\mathbf{M}_1}$ is an induced subposet of $\mathbf{P}_{\mathbf{M}'_1}$, and thus $\dim(\mathbf{P}_{\mathbf{M}_1}) \leq \dim(\mathbf{P}_{\mathbf{M}'_1})$.

By duality, we can also deal with the case where $\mathbf{M}_1$ has a vertex of degree 1. Note that the dual operation to subdividing an edge is that of duplicating an edge: replacing an edge f from x to y by two such edges surrounding a new face.

By repeating the process as often as necessary, we obtain a connected map $\mathbf{M}_2$ with no loops or vertices of degree 1 such that $\dim(\mathbf{P}_{\mathbf{M}_1}) \leq \dim(\mathbf{P}_{\mathbf{M}_2})$.

(3) Destroying vertex-vertex and face-face critical pairs. Suppose that, again as in Figure 2.1, there is a vertex-vertex critical pair (y, x) in $\mathbf{M}_2$. Then all the edges including y have x as their other endpoint. Choose one such edge e' and subdivide it with a vertex v, introducing new edges a, between x and v, and b, between v and y, in place of e'. This operation decreases the number of vertex-vertex critical pairs without introducing any extra face-face critical pairs. Let F and G be the two faces separated by e'. Call the new map $\mathbf{M}_2'$.

Suppose $\{L_1, \ldots, L_t\}$ is a realizer of $\mathbf{P}_{\mathbf{M'}_2}$. For each $i = 1, \ldots, t$, we construct a linear extension L_i' of $\mathbf{P}_{\mathbf{M}_2}$ from L_i as follows. We insert e' immediately above the highest of a, b, v in L_i; then we delete a, b, and v from the ordering. This is certainly a linear extension, since e' is placed above x, y and below F, G.

We claim that $\{L_1', \ldots, L_t'\}$ is a realizer of $\mathbf{P}_{\mathbf{M}_2}$. When restricted to $\mathbf{P}_{\mathbf{M}_2} - e'$, the intersection of the L_i' is the same as the intersection of the L_i, so it remains to check that all critical pairs involving e' are reversed. Clearly e' is not in any vertex-edge critical pairs and, as mentioned in section 1, edge-edge critical pairs can be ignored. If (e, H) is an edge-face critical pair, then (v, H) is reversed in some L_k, and hence (e, H) is reversed in L_k'.

Thus $\dim(\mathbf{P}_{\mathbf{M}_2}) \leq \dim(\mathbf{P}_{\mathbf{M'}_2})$. Proceeding in this manner we can remove all the vertex-vertex critical pairs. Thus we construct a map $\mathbf{M}_3$ with no critical pairs of this type such that $\dim(\mathbf{P}_{\mathbf{M}_2}) \leq \dim(\mathbf{P}_{\mathbf{M}_3})$.

Using the dual case of the above argument, we can next find a map $\mathbf{M}_4$ with no critical pairs of either vertex-vertex or face-face type such that $\dim(\mathbf{P}_{\mathbf{M}_3}) \leq \dim(\mathbf{P}_{\mathbf{M}_4})$. For instance, in the map $\mathbf{M}_3$ of Figure 2.1, (G, F) is a critical pair, which is destroyed by duplicating the edge b.

(4) Destroying vertex-edge and edge-face critical pairs. Our approach to critical pairs of these types will be slightly different. We shall deal with all the vertex-edge and edge-face critical pairs in one step, forming an auxiliary map $\mathbf{M}_5$ such that $\dim(\mathbf{P}_{\mathbf{M}_4}) \leq \dim(\mathbf{Q}_{\mathbf{M}_5})$.

Recall from Lemma 1.4 that no edge is in both a vertex-edge and an edge-face critical pair. We form $\mathbf{M}_5$ as follows. For every edge e, say between x and y, of $\mathbf{M}_4$ which is in a vertex-edge critical pair, replace e by a double edge from x to y, and call the face between the two edges F_e. For every edge e of $\mathbf{M}_4$ in an edge-face critical pair, subdivide e with a vertex v_e. (The idea is that the new element F_e or v_e will represent the critical edge e in $\mathbf{M}_5$.) For instance, in the map $\mathbf{M}_4$ of Figure 2.1, (v, c) and (d, F) are critical pairs of $\mathbf{P}_{\mathbf{M}_4}$, so c is duplicated to produce a face F_c, and d is subdivided by a vertex v_d.

Let $\{L_1, \ldots, L_t\}$ be a realizer of $\mathbf{Q}_{\mathbf{M}_5}$. From each L_i, we construct a linear extension L_i' of $\mathbf{P}_{\mathbf{M}_4}$ as follows. We start from L_i, which includes all vertices and faces of $\mathbf{P}_{\mathbf{M}_4}$, and insert the edges according to the following rules. First, noncritical edges of $\mathbf{M}_4$ are inserted anywhere consistent with the order $\mathbf{P}_{\mathbf{M}_4}$. Next, if e is a critical edge in a vertex-edge critical pair, with e separating faces F and G in $\mathbf{M}_4$, say, then e is inserted just below the lowest of F, G, and F_e in L_i. Similarly, if e is an edge in an edge-face critical pair, with e joining x and y, then e is inserted into L_i just above the highest of x, y, and v_e. Finally the auxiliary vertices and faces v_e and

F_e are deleted from the linear extension.

The L_i thus constructed are clearly linear extensions of $\mathbf{P}_{\mathbf{M}_4}$. It may be that some edge-edge critical pairs are not reversed: if this is the case, we alter the L_i so that they are, as in section 1. Certainly all vertex-face critical pairs in $\mathbf{P}_{\mathbf{M}_4}$ are reversed by some L_i. It remains to be shown that all vertex-edge and edge-face critical pairs are reversed. The two cases are dual, so we need only consider a vertex-edge critical pair (v, e) of $\mathbf{P}_{\mathbf{M}_4}$. For such a pair, we have an auxiliary face F_e, and the pair (v, F_e) is reversed in some L_k. Hence (v, e) is reversed in L'_k.

Thus all critical pairs of $\mathbf{P}_{\mathbf{M}_4}$ are reversed by some L'_i, and so $\{L'_1, \ldots, L'_t\}$ is a realizer of $\mathbf{P}_{\mathbf{M}_4}$, as required.

(5) Making the map 2-connected. We proceed by reducing the number of blocks of the underlying graph of $\mathbf{M}_5$ to 1, noting that no endblock is a single edge or a loop. If $\mathbf{M}_5$ is not 2-connected, let x be any cutvertex of the underlying graph, and let F be a face with x occurring at least twice on its boundary, as in Figure 2.1. The sequence of vertices encountered by travelling around the boundary of F thus includes x (indeed, more than once): let u and v be the vertices just before and after x in one such encounter. Form $\mathbf{M}'_5$ by joining y and z by an edge, thus decreasing the number of blocks. Clearly $\mathbf{Q}_{\mathbf{M'}_5} = \mathbf{Q}_{\mathbf{M}_5}$. Repeating as necessary, we end with a 2-connected map $\mathbf{M}_0$ such that $\mathbf{Q}_{\mathbf{M}_0} = \mathbf{Q}_{\mathbf{M}_5}$.

Combining all the steps, we see that $\dim(\mathbf{P}_\mathbf{M}) \leq \dim(\mathbf{Q}_{\mathbf{M}_0})$, as desired. □

3. Proof of Lemma 1.5. Throughout this section, e will be a distinguished edge in a 2-connected planar map $\mathbf{M}$. The endpoints of e will always be denoted x and y, and the faces separated by e by F and G.

For a planar map $\mathbf{M}$ with distinguished edge e, we say that a realizer $\mathcal{R}$ of $\mathbf{Q}_\mathbf{M}$ is an *e-realizer* if it has order 4, and the four linear extensions in $\mathcal{R}$ can be labelled L_1, L_2, L_3, L_4 so as to satisfy the following conditions:

(a) x is the highest vertex in L_1,
(b) y is the highest vertex in L_2,
(c) F is the lowest face in L_3, and
(d) G is the lowest face in L_4.

We shall prove the following result, which is stronger than Lemma 1.5.

THEOREM 3.1. *Let* $\mathbf{M}$ *be a 2-connected planar map, and let* e *be an edge of* $\mathbf{M}$. *Then there is an e-realizer of* $\mathbf{Q}_\mathbf{M}$.

One technical problem we have to deal with is that $\mathbf{Q}_\mathbf{M}$ will in general have vertex-vertex and face-face critical pairs. In fact, a glance at the proof of Lemma 2.1 shows that we can ignore these: to prove Theorem 1.2 it is enough to show that, for every 2-connected map $\mathbf{M}$, there is a set of four linear extensions of $\mathbf{Q}_\mathbf{M}$ reversing every vertex-face critical pair of $\mathbf{Q}_\mathbf{M}$. However, it involves essentially no extra work to prove Theorem 3.1 as it stands, since the constructions we shall give do yield realizers of $\mathbf{Q}_\mathbf{M}$.

Let us first see that Theorem 3.1 holds if $\mathbf{M}$ is 3-connected. We use the notation and techniques of [1]. The reader who does not have that paper at hand may rest assured that the proof is a straightforward application of the methods developed there.

THEOREM 3.2. *Let* $\mathbf{M}$ *be a 3-connected planar map, and let* e *be an edge of* $\mathbf{M}$. *Then there is an e-realizer of* $\mathbf{Q}_\mathbf{M}$.

Proof. Arrange for G to be the outside face, with x and y two vertices of a triad $(v_1 = x, v_2 = y, v_3)$, and apply the construction of [1] with this triad to obtain a realizer consisting of four linear extensions L_1, L_2, L_3, and L_4, as in [1]. Certainly G is the lowest face in the fourth linear extension L_4. Also, x is the highest vertex in L_1, since it is the only vertex w with $S(w, 1)$ equal to the whole of $\mathbf{R}^2 - \text{int}(G)$. Similarly, y is the highest vertex in L_2.

The face F is contained in $S(w, 3)$ for every vertex w except for x and y. Thus if z is any vertex on F and u is any vertex not on F, we have $S(z, 3) \subseteq S(u, 3)$. If $S(z, 3) = S(u, 3)$, then either (F, y) witnesses $(z, u) \in \mathcal{R}'_3$ or (F, x) witnesses $(z, u) \in \mathcal{L}'_3$. In any case, (z, u) in Q'_3. Thus in fact F lies below all vertices not on F in L_3 and is certainly the lowest face in that order. Therefore the set $\{L_1, L_2, L_3, L_4\}$ is an e-realizer. □

We make one more observation before the proof of Theorem 3.1. Let $\mathcal{R}$ be an e-realizer of a planar map $\mathbf{M}$. We call $\mathcal{R}$ a *strong e-realizer* if its four linear extensions can be labelled L_1, L_2, L_3, L_4 so that, in addition to properties (a) to (d) above, we have that

(e) y is the lowest element of L_1, and F and G the two highest elements;
(f) x is the lowest element of L_2, and F and G the two highest elements;
(g) x and y are the two lowest elements of L_3, and G the highest element; and
(h) x and y are the two lowest elements of L_4, and F the highest element.

LEMMA 3.3. *Let e be a distinguished edge in a 2-connected planar map* $\mathbf{M}$. *If* $\mathbf{Q_M}$ *has an e-realizer, then it has a strong e-realizer.*

Proof. Let (L_1, L_2, L_3, L_4) be a realizer satisfying (a) through (d). If there are any faces above x in L_1 which do not contain x, they can be moved to a position in L_1 below x but above all other vertices. The altered set of linear extensions is clearly still an e-realizer of $\mathbf{Q_M}$. Thus we may assume that all critical pairs involving x are reversed in L_1.

Having made this assumption, we may then also suppose that x is the lowest element in all of the other three linear extensions: if not, it can be moved to the bottom, since the only critical pairs this affects are those involving x.

Proceeding in a similar way, we can alter the linear extensions so as to move y, F, and G to the positions required by (e) through (h). □

Proof of Theorem 3.1. We proceed by induction on the number of edges of $\mathbf{M}$. It is easily checked that the result is true for all 2-connected planar maps with at most, say, 4 edges.

Let $\mathbf{M}$ be a 2-connected planar map with $m \geq 5$ edges, and suppose that the result is true for all 2-connected maps with fewer than m edges. Let e be an edge of $\mathbf{M}$.

If $\mathbf{M}$ is 3-connected, then $\dim(\mathbf{Q_M}) \leq 4$ by Theorem 3.2. Suppose then that $\mathbf{M}$ is not 3-connected.

The dual map $\mathbf{M}^*$ of $\mathbf{M}$ is also 2-connected. Let e^* be the edge of $\mathbf{M}^*$ corresponding to e, and suppose that there is an e^*-realizer $\{L_1, \ldots, L_4\}$ of $\mathbf{Q_{M^*}}$. Then the set $\{L_1^*, \ldots, L_4^*\}$ of reverse linear orders provides an e-realizer of $\mathbf{Q_M}$. In other words it would suffice to prove the result for $\mathbf{M}^*$ and e^* instead of for $\mathbf{M}$ and e.

We split the argument into two cases, according to whether or not e is a critical edge in $\mathbf{M}$. In both cases, our task is to construct either an e-realizer of $\mathbf{Q_M}$ or an e^*-realizer of $\mathbf{Q_{M^*}}$.

(A) *e* is a critical edge. Suppose that (e, H) is an edge-face critical pair: if instead e is in a vertex-edge critical pair, then we work instead in the dual.

Removal of x, y, e, and H from the plane leaves two components, one containing F and the other G. Let $\mathbf{M}_1$ be the submap of $\mathbf{M}$ specified by the edges in the F-component together with e; and let $\mathbf{M}_2$ be the submap specified by e and the edges in the G-component. In both cases, let H stand for the exterior face. See Figure 3.1.

Thus the elements in common between $\mathbf{Q}_{\mathbf{M}_1}$ and $\mathbf{Q}_{\mathbf{M}_2}$ are just x, y, and H; and there are no relations in $\mathbf{Q}_{\mathbf{M}}$ between an element of $\mathbf{Q}_{\mathbf{M}_1}$ and an element of $\mathbf{Q}_{\mathbf{M}_2}$ except those involving x, y, or H. Also, if (α, β) is a vertex-vertex or face-face critical pair, then α and β must either both be in $\mathbf{Q}_{\mathbf{M}_1}$ or both be in $\mathbf{Q}_{\mathbf{M}_2}$, except that (F, G) or (G, F) could be a critical pair.

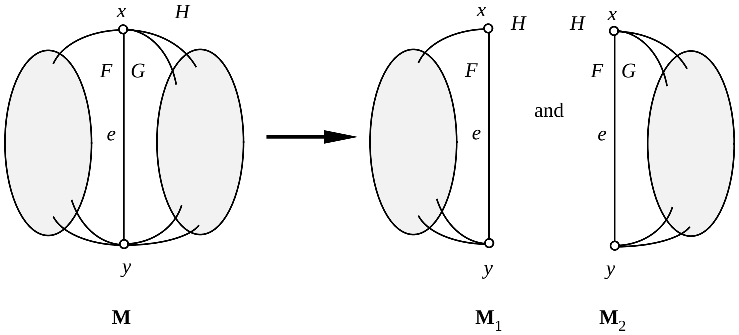


FIG. 3.1. *Splitting* $\mathbf{M}$ *into* $\mathbf{M_1}$ *and* $\mathbf{M_2}$.

Now $\mathbf{M}_1$ and $\mathbf{M}_2$ both have fewer edges than $\mathbf{M}$, so we can find an e-realizer for each map. To be more specific, we can find a realizer $(L_1^1, L_2^1, L_3^1, L_4^1)$ of $\mathbf{Q}_{\mathbf{M}_1}$ and a realizer $(L_1^2, L_2^2, L_3^2, L_4^2)$ of $\mathbf{Q}_{\mathbf{M}_2}$ satisfying the following:

(i) x is the highest vertex in both L_1^1 and L_1^2,
(ii) y is the highest vertex in both L_2^1 and L_2^2,
(iii) F is the lowest face in L_3^1,
(iv) G is the lowest face in L_4^2, and
(v) H is the lowest face in both L_3^2 and L_4^1.

By Lemma 3.3, we may also take these two realizers to be strong e-realizers, so in particular we may assume that H is the highest element in both L_3^1 and L_4^2 and that x and y are the lowest elements in L_3^2 and L_4^1.

Now, for $j = 1, \ldots, 4$, we combine the linear extensions L_j^1 and L_j^2 to form a linear extension L_j of $\mathbf{Q}_{\mathbf{M}}$ as follows. For $j = 1, 2$, we form L_j in any way such that the restriction of L_j to the elements of $\mathbf{Q}_{\mathbf{M}_i}$ is L_j^i, for $i = 1, 2$. Hence x is the highest vertex in L_1, and y the highest in L_2.

For L_3, we essentially put L_3^2 above L_3^1. To be more precise, we put every element of $\mathbf{Q}_{\mathbf{M}_2}$ other than x and y at the top, in the order given by L_3^2, then below them the elements of $\mathbf{Q}_{\mathbf{M}_1}$ other than H, in the order given by L_3^1. Again, the restriction of L_3 to the elements of $\mathbf{Q}_{\mathbf{M}_i}$ is L_3^i, for $i = 1, 2$. Clearly F is the lowest face in L_3.

The fourth extension L_4 is constructed in an analogous manner, putting L_4^1 on top of L_4^2. We claim that the four orders L_j, shown in Figure 3.2, constitute a realizer of $\mathbf{Q}_{\mathbf{M}}$. Clearly they are linear extensions of $\mathbf{Q}_{\mathbf{M}}$: it remains to be shown that every critical pair is reversed.

If (α, β) is a critical pair with α and β both in $\mathbf{Q}_{\mathbf{M}_i}$, for $i = 1$ or 2, then (α, β) is reversed in some L_j^i and so also in L_j.

If v is a vertex in M_1 other than x or y, and J is a face in M_2 other than H, then (v, J) is reversed in L_4. Similarly every critical pair (w, E), where w is a vertex of $\mathbf{M}_2$ and E is a face of $\mathbf{M}_1$, is reversed in L_3.

The only other possible critical pairs are (F, G) and (G, F), and these are reversed in L_4 and L_3, respectively. Therefore $L_1, \ldots, L_4$ is an e-realizer of $\mathbf{Q}_{\mathbf{M}}$.

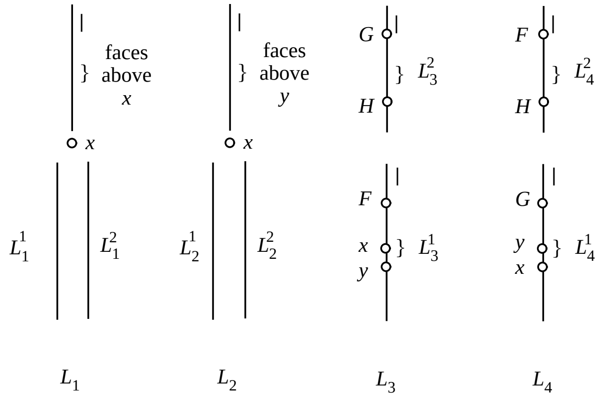


FIG. 3.2. *The new linear extensions L_j.*

(B) *e* is not a critical edge. Let $\{u, v, D, E\}$ be a separating system such that the component $C(e)$ of $\mathbf{R}^2 - \{u, v\} - \mathrm{int}(D) - \mathrm{int}(E)$ containing e is minimal. Let $\mathbf{M}_1$ be the submap determined by the edges in this component together with an edge between u and v separating D and E.

We also form another map $\mathbf{M}_2$ by removing all the edges of $\mathbf{M}_1$ from $\mathbf{M}$ and replacing them with a single edge f between u and v separating D and E. Both $\mathbf{M}_1$ and $\mathbf{M}_2$ have fewer edges than $\mathbf{M}$. See Figure 3.3.

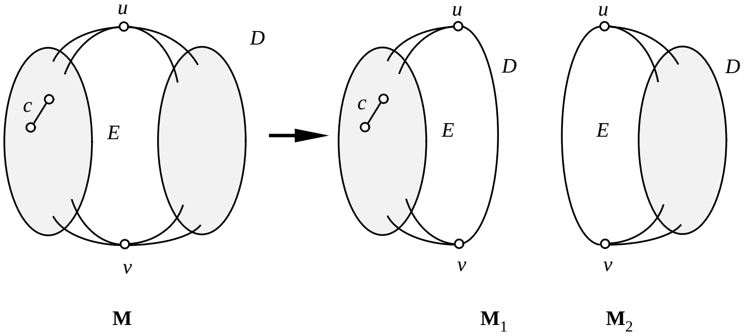


FIG. 3.3. *The maps $\mathbf{M_1}$ and $\mathbf{M_2}$.*

Suppose that there is a face H of $\mathbf{M}_1$ other than D and E containing both u and v. Then $\{u, v, D, H\}$ and $\{u, v, E, H\}$ are separating systems in $\mathbf{M}$, and for one of them, say $\{u, v, D, H\}$, the component of e in the complement is a strict subset of $C(e)$. Therefore $uDvH$ is not separating, and so the component of e is the single edge e itself, between u and v. In that case, (e, E) is a critical pair, contradicting the assumption that e is not critical.

Thus D and E are the only faces of $\mathbf{M}_1$ containing both u and v. By duality, we also have that u and v are the only vertices of $\mathbf{M}_1$ on both D and E. In particular, v has a neighbor z on D distinct from u, and there is a face C of $\mathbf{M}_1$ distinct from D and E containing the edge vz. See Figure 3.4.

By a similar argument, we see that neither u nor v is involved in a vertex-vertex critical pair in $\mathbf{Q}_{\mathbf{M}_1}$, and neither D nor E is in a face-face critical pair in $\mathbf{Q}_{\mathbf{M}_1}$.

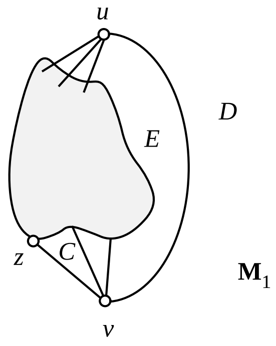


FIG. 3.4. *The vertex z and face C.*

The map $\mathbf{M}_1$ has fewer edges than $\mathbf{M}$, so there is an e-realizer of $\mathbf{Q}_{\mathbf{M}_1}$. In fact, we would like this realizer to have certain extra properties as specified below.

We call a linear extension L of $\mathbf{Q}_{\mathbf{M}_1}$ *u-good* if u is above v and also some face in L. Similarly we call L *v-good* if v is above u and some face in L. The extension L is *D-good* if D is below E and some vertex in L, and L is *E-good* if E is below D and a vertex in L. Note that if $\{L_1, \ldots, L_4\}$ is a realizer of $\mathbf{Q}_{\mathbf{M}_1}$ and $\alpha \in \{u, v, D, E\}$, then one of the L_i is α-good. The next lemma states that rather more is true.

LEMMA 3.4. *There is an e-realizer (K_u, K_v, K_D, K_E) of $\mathbf{Q}_{\mathbf{M}_1}$ such that, for $\alpha = u, v, D, E$, the linear extension K_α is α-good.*

Note that some of u, v, D, E might coincide with some of x, y, F, G, so the conditions above might preclude (K_u, K_v, K_D, K_E) from being a strong e-realizer.

Proof. Take $\mathcal{R}$ to be an e-realizer of $\mathbf{Q}_{\mathbf{M}_1}$ maximizing the number N of α in the set $\{u, v, D, E\}$ such that there are two α-good linear extensions amongst the linear extensions in $\mathcal{R}$. If $N = 4$, then it is a simple matter to label these linear extensions as K_u, K_v, K_D, K_E in an appropriate manner.

Thus we may assume without loss of generality that only one of the linear extensions is u-good: say L^1 is the only linear extension in $\mathcal{R}$ with u above v and also above some face. In particular, u is above the face C in L^1. Thus the critical pair (z, E) is reversed in some other linear extension, say L^2, of $\mathcal{R}$. Thus L^2 is E-good. A symmetrical argument shows that another linear extension L^3 in $\mathcal{R}$ is D-good. If the last linear extension L^4 of $\mathcal{R}$ is v-good, then we can immediately label the L^i's as (K_u, K_E, K_D, K_v) in that order.

If this is not the case, then L^4 is neither u-good nor v-good, so u and v are both below the lowest face H in L^4. If H does not contain u, then u can be moved to the position immediately above H in L^4: the new set of linear extensions is still an e-realizer, but now both L^1 and L^4 are u-good, and so the value of N is higher for this new set, a contradiction. Similarly if H does not contain v, then v can be moved to the position just above H: this makes L^4 v-good, and so we can label the L^i's as before. Hence we may assume that H contains both u and v and therefore is either D or E—without loss of generality D.

This certainly implies that L^4 is D-good. Now we can apply the same argument as above to L^3 and conclude that D is the lowest face in that order as well. Note that L^2 is necessarily v-good.

It may well be that D is one of F or G, so is forced to be the lowest face in, say, L^3 by the condition that the L^i's form an e-realizer. However, this cannot also be the case in L^4. Also, as in Lemma 3.3, we may assume that all critical pairs involving D are reversed in L^3. Thus D can be moved upward in L^4, and the system is still an e-realizer.

If E is below some vertex in L^4, then putting D at the top of L^4 makes the linear extension E-good, enabling us to label the linear extensions as (K_u, K_v, K_D, K_E). So suppose that E is above all vertices in L^4.

Now put D directly above the second lowest face J in L^4: this keeps L^4 D-good. One of u or v is not on J: place this vertex between D and J. As before, this either increases N or allows a labelling as desired. □

We take an e-realizer (K_u, K_v, K_D, K_E) of $\mathbf{Q}_{\mathbf{M}_1}$ satisfying the conclusions of Lemma 3.4, and a strong f-realizer $\mathcal{S}$ of $\mathbf{Q}_{\mathbf{M}_2}$, and combine them to make an e-realizer of $\mathbf{Q}_{\mathbf{M}}$ as follows.

Consider first the linear extension K_u of $\mathbf{Q}_{\mathbf{M}_1}$, in which u is above v and some face of $\mathbf{M}_1$. We take also that linear extension L_u of $\mathbf{Q}_{\mathbf{M}_2}$ in $\mathcal{S}$, in which u is the top vertex, v the bottom element, and D and E are the top two elements. We combine these to make a linear extension L^u of $\mathbf{Q}_{\mathbf{M}}$ by replacing u in K_u by all of $\mathbf{Q}_{\mathbf{M}_2}$ except for v, D, E, in the order given by L_u. This does indeed give a linear extension of $\mathbf{Q}_{\mathbf{M}}$, and we note also that the top vertex and bottom face in L^u are the same as in K_u. See Figure 3.5.

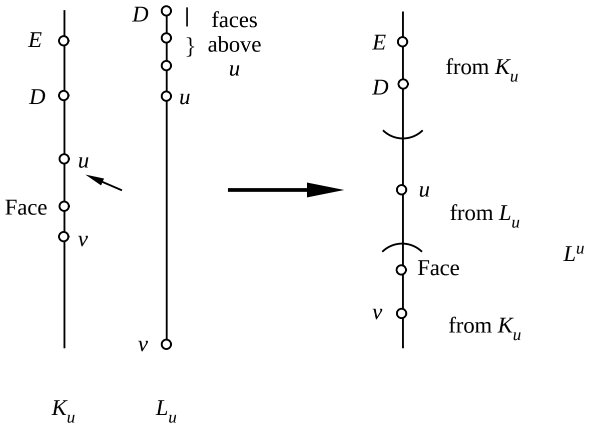


FIG. 3.5. *The new linear extensions.*

We repeat with the other linear extensions to obtain four linear extensions L^u, L^v, L^D, L^E of $\mathbf{Q}_{\mathbf{M}}$. It remains to be shown that these form a realizer. Notice that if, for instance, u is above a face H in K_α, then every vertex in $\mathbf{M}_2$, other than perhaps v, comes above H in L^α.

We consider each possible type of critical pair in turn, checking it is reversed by one of the four linear extensions.

We start with critical pairs involving u. For H a face in $\mathbf{M}_1$ not including u, (u, H) is reversed in L^α whenever it is reversed in K_α. For β an element of $\mathbf{M}_2$ with (u, β) a critical pair, (u, β) is reversed in L_u, and hence also in L^u. Similarly all critical pairs involving v, D, or E are catered to.

Let z be a vertex of $\mathbf{M}_1$ other than u and v. Without loss of generality z is not on the face E, so the pair (z, E) is reversed in some K_α. Hence all the faces of $\mathbf{M}_2$ come below z in L^α, so all critical pairs of the form (z, H) for H a face of $\mathbf{M}_2$ are reversed in L^α. By duality, all pairs of the form (w, J) for w a vertex of $\mathbf{M}_2$ and J a face in $\mathbf{M}_1$ are also reversed.

Finally, if β and γ are elements of the same $\mathbf{M}_i$, then if (β, γ) is a critical pair then it is reversed in some K_α or L_α, and hence is reversed in the corresponding L^α.

Thus every critical pair is reversed by some L^α, and so the family (L^u, L^v, L^D, L^E) constitutes a realizer. Since the top and bottom elements are the same in L^α as in K_α, this is an e-realizer.

In both cases, we have constructed an e-realizer for our poset $\mathbf{Q_M}$. Thus, by induction, $\mathbf{Q_M}$ has an e-realizer for every 2-connected map $\mathbf{M}$ and edge e. □

4. Concluding remarks. It is proved in Reuter [3], and in [1], that, for every 3-connected map $\mathbf{M}$, $\dim(\mathbf{Q_M}) \geq 4$, and therefore $\dim(\mathbf{P_M}) = \dim(\mathbf{Q_M}) = 4$. Obviously this is not true if the 3-connectedness condition is removed, and we are left with the questions of characterizing the planar maps $\mathbf{M}$ with $\dim(\mathbf{P_M})$ or $\dim(\mathbf{Q_M})$ equal to 3 (or 2). We offer a few remarks on some of these problems.

Let us first ask which maps $\mathbf{M}$ have $\dim(\mathbf{P_M})$ equal to 2. Note that, if $\mathbf{M}$ contains any cycle with at least 3 vertices, then $\dim(\mathbf{P_M}) \geq 3$, since the subposet of $\mathbf{P_M}$ induced by the vertices and edges of the cycle is a crown. If $\mathbf{M}$ contains any edges with multiplicity at least 3, they give rise to a cycle in the dual, so again $\mathbf{P_M}$ has dimension at least 3. Similarly, if any vertex (face) of $\mathbf{M}$ has three distinct neighbors, then $\dim(\mathbf{P_M}) \geq 3$. Hence, if $\dim(\mathbf{P_M}) = 2$, then each component of the underlying graph of M is a path, possibly with loops and/or double edges. Similar considerations lead to the conclusions that only the final edges of paths can be double edges, that all loops separate one endvertex of the path from the other, and that, if X and Y are two components of the graph, then an endvertex of X must share a face with an endvertex of Y. These restrictions give us a complete characterization of maps $\mathbf{M}$ with $\dim(\mathbf{P_M}) = 2$: a typical such map is shown in Figure 4.1.

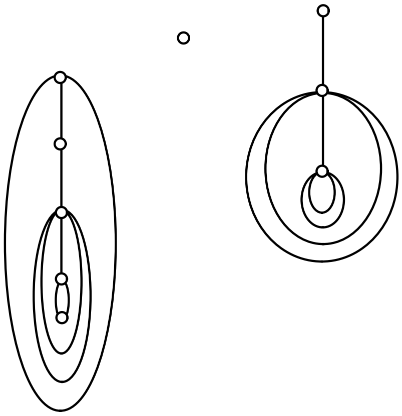


FIG. 4.1. *A map* $\mathbf{M}$ *with* $\mathbf{P}_{\boldsymbol{M}}$= 2.

As far as we can tell, none of the other three problems suggested at the beginning of this section has as neat a solution. Maybe the right question is, are there polynomial algorithms to determine whether $\dim(\mathbf{P_M})$ or $\dim(\mathbf{Q_M})$ is equal to 3? It is known that this problem for a general partial order is NP-complete, but there is a polynomial algorithm to determine whether a partial order has dimension 2.

Another related line of inquiry is to ask which maps $\mathbf{M}$ have $\mathbf{Q_M}$ 4-irreducible. We know from [1] that all 3-connected maps have this property, and it is tempting to conjecture the converse: if $\mathbf{Q_M}$ is 4-irreducible, then $\mathbf{M}$ is 3-connected. However, the example in Figure 4.2 shows that this is false.

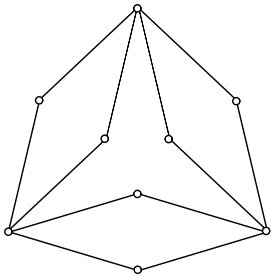


FIG. 4.2. *A non-3-connected map* $\mathbf{M}$ *with* $\mathbf{Q}_{M}$*4-irreducible.*

Again, we suspect that there is no particularly neat characterization, and the complexity version of the problem may be more fruitful.

Finally, it is natural to ask how the results of [1] and this paper extend to other surfaces. If $\mathbf{M}$ is a map drawn on a surface of genus k, then there are some bounds $f(k), g(k)$ for $\dim(\mathbf{P_M})$ and $\dim(\mathbf{Q_M})$. What are the best possible bounds? Are they the same in both cases? We tentatively venture the suggestion that $\dim(\mathbf{P_M})$ and $\dim(\mathbf{Q_M})$ are still bounded above by 4 when $\mathbf{M}$ is a map drawn on the torus.

REFERENCES

[1] G. R. BRIGHTWELL AND W. T. TROTTER, *The order dimension of convex polytopes*, SIAM J. Discrete Math., 6 (1993), pp. 230–245.

[2] D. KELLY AND W. T. TROTTER, *Dimension theory for ordered sets*, in Proceedings of the Symposium on Ordered Sets, I. Rival et al., eds., Reidel, Boston, MA, 1982, pp. 171–212.

[3] K. REUTER, *On the Order Dimension of Convex Polytopes*, preprint.

[4] W. SCHNYDER, *Planar graphs and poset dimension*, Order, 15 (1989), pp. 323–343.

[5] W. T. TROTTER, *Progress and new directions in dimension theory for finite partially ordered sets*, in Extremal Problems for Finite Sets, P. Frankl et al., eds., Bolyai Soc. Math. Studies 3, 1991, pp. 457–477.

[6] W. T. TROTTER, *Combinatorics and Partially Ordered Sets: Dimension Theory*, The Johns Hopkins University Press, Baltimore MD, 1992.

SIAM J. DISCRETE MATH. © 1997 Society for Industrial and Applied Mathematics
Vol. 10, No. 4, pp. 529–550, November 1997 002

ALGORITHMS FOR VERTEX PARTITIONING PROBLEMS ON PARTIAL k-TREES [*]

JAN ARNE TELLE[†] AND ANDRZEJ PROSKUROWSKI[‡]

Abstract. In this paper, we consider a large class of vertex partitioning problems and apply to them the theory of algorithm design for problems restricted to partial k-trees. We carefully describe the details of algorithms and analyze their complexity in an attempt to make the algorithms feasible as solutions for practical applications.

We give a precise characterization of vertex partitioning problems, which include domination, coloring and packing problems, and their variants. Several new graph parameters are introduced as generalizations of classical parameters. This characterization provides a basis for a taxonomy of a large class of problems, facilitating their common algorithmic treatment and allowing their uniform complexity classification.

We present a design methodology of practical solution algorithms for generally $\mathcal{NP}$-hard problems when restricted to partial k-trees (graphs with treewidth bounded by k). This "practicality" accounts for dependency on the parameter k of the computational complexity of the resulting algorithms.

By adapting the algorithm design methodology on partial k-trees to vertex partitioning problems, we obtain the first algorithms for these problems with reasonable time complexity as a function of treewidth. As an application of the methodology, we give the first polynomial-time algorithm on partial k-trees for computation of the Grundy number.

Key words. treewidth, algorithms, implementations, graph partitioning

AMS subject classifications. 68R10, 68Q25

PII. S0895480194275825

1. Introduction.

Many inherently difficult ($\mathcal{NP}$-hard) optimization problems on graphs become tractable when restricted to trees or to graphs with some kind of tree-like structure. A large class of such graphs is the class of partial k-trees (equivalently, graphs with treewidth bounded by k). Although tractability requires fixed k, this class contains all graphs with n vertices when the parameter k is allowed to vary through positive integers up to $n - 1$. Many natural classes of graphs have bounded treewidth [21]. There are many approaches to finding a template for the design of algorithms on partial k-trees with time complexity polynomial, or even linear, in the number of vertices [1, 2, 3, 4, 5, 7, 11, 23, 24, 30]. Proponents of these approaches attempt to encompass as wide a class of problems as possible, often at the expense of simplicity of the resulting algorithms, and also at the expense of increased algorithm time complexity as a function of k. In contrast, results giving explicit practical algorithms in this setting are usually limited to a few selected problems on either (full) k-trees [9], partial 1-trees, or partial 2-trees [25]. We intend to cover the middle ground between these two extremes by investigating time complexity as a function of both input size and treewidth k.

We assume that the input graph is given with a width k tree-decomposition, computable in linear time for fixed k [6]. Our algorithms employ a *binary parse tree* of the input partial k-tree, easily derived from a tree-decomposition of the graph.

[*] Received by the editors October 17, 1994; accepted for publication (in revised form) August 21, 1996. This research was supported in part by NSF grant CCR9213439 and forms part of the first author's Ph.D. thesis at the University of Oregon.

http://www.siam.org/journals/sidma/10-4/27582.html

[†] Department of Informatics, University of Bergen, 5020 Bergen, Norway (telle@ii.uib.no). The research of this author was supported by a grant from the Norwegian Research Council.

[‡] Department of Computer and Information Science, University of Oregon, Eugene, OR 97403 (andrzej@cs.uoregon.edu).

This parse tree is based on very simple graph operations that mimic the construction process of an embedding k-tree. We propose a design methodology that for many $\mathcal{NP}$-hard problems results in algorithms with time complexity linear in the size of the input graph and only exponential in its treewidth, lowering the exponent of previously known solutions. We give a careful description of the algorithm design details with the aim of easing the task of implementation for practical applications. We include a brief report on an ongoing implementation project.

A large class of inherently difficult discrete optimization problems can be expressed in the *vertex partitioning* formalism. This formalism involves neighborhood constraints on vertices in different classes (blocks) of a partition and provides a basis for a taxonomy of vertex partitioning problems. We define this formalism and then use it to provide a uniform algorithmic treatment on partial k-trees of vertex partitioning problems. As an example of application of our paradigm, we give the first polynomial-time algorithms on partial k-trees for the Grundy number. The efficiency of our algorithm follows from (i) the description of the Grundy number problem as a vertex partitioning problem, (ii) a careful investigation of time complexity of vertex partitioning problems on partial k-trees, and (iii) a new logarithmic bound on the Grundy number of a partial k-tree.

We present these ideas as follows: in section 3, we describe the binary parse tree of partial k-trees and the general algorithm design method, in section 4 we define vertex partitioning problems, in section 5 we apply the partial k-tree algorithm design method to vertex partitioning problems, and in section 6 we give the efficient solution algorithm for the Grundy number on partial k-trees. We conclude the paper with a brief report on experiences with implementations.

2. Definitions. We denote the nonnegative integers by $\mathbb{N}$ and the positive integers by $\mathbb{P}$. The graph $G = (V(G), E(G))$ has vertex set $V(G)$ and edge set $E(G)$. We consider simple, undirected graphs, unless otherwise specified. For $S \subseteq V(G)$, let $G[S] = (S, \{(u, v) : u, v \in S \wedge (u, v) \in E(G)\})$ denote the subgraph *induced* in G by S. For $S \subseteq V(G)$, let $G \setminus S = G[V(G) \setminus S]$. A *component* in a graph is a maximal connected subgraph. A *separator* of a graph G is a subset of vertices $S \subseteq V(G)$ such that $G \setminus S$ has more components than G. In a *complete* graph there is an edge for every two-element subset of vertices.

A graph G is a k-tree if it is a complete graph on k vertices (a k-clique), or if it has a vertex $v \in V(G)$ whose neighbors induce a k-clique of size k such that $G \setminus \{v\}$ is again a k-tree. Such a *reduction process* of G (or the corresponding *construction* process) determines its *parse tree.* A partial k-tree H is a subgraph of a k-tree and a construction process of this embedding k-tree defines a parse tree of H. A *tree-decomposition* of a graph G is a tree T whose nodes are subsets of vertices of G such that for every edge (u, v) of G, there is a node containing both u and v, and for every vertex u of G, the set of nodes of T that contain u induces a (nonempty, connected) subtree of T. The nodes of T are often called *bags.* The *width* of a tree-decomposition T is defined as one less than the maximum size of a bag. The *treewidth* of G is the minimum width of a tree-decomposition of G. It is fairly easy to see that a parse tree of a partial k-tree G defines (through maximal cliques of G) a width k tree-decomposition of G. Similarly, based on such a decomposition one can find a k-tree embedding G. For any partial k-tree G with at least k vertices, there is a k-tree H with the same number of vertices for which G is a subgraph. The fact that we can assume vertex sets equality follows from the treewidth formulation.

A linear ordering $\pi = v_1, \ldots, v_n$ of the vertices of a graph is a *perfect elimination*

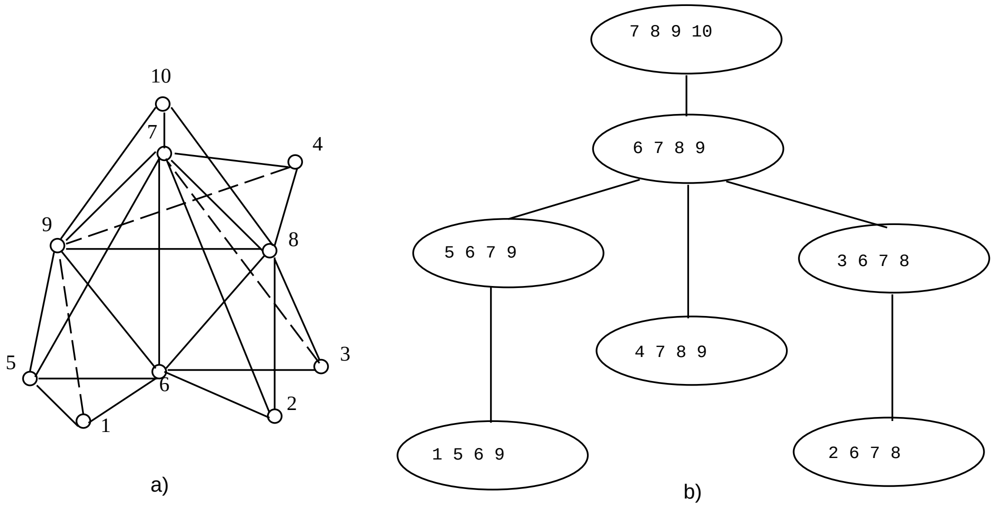


FIG. 2.1. *a) A partial* 3*-tree* G*, embedded in a* 3*-tree* H*, dashed edges in* $E(H) - E(G)$*. b) Its peo-tree* P *with respect to peo*=1,2,3,4,5,6,7,8,9,10.

ordering (*peo*) if, for each i, $1 \leq i \leq n$, the higher-numbered neighbors of v_i induce a clique. A k-tree H has a peo $\pi = v_1, \dots, v_n$ such that for each i, $1 \leq i \leq n-k$, the vertex set $B_i = \{v_i\} \cup (N_H(v_i) \cap \{v_{i+1}, \dots, v_n\})$ induces a $k+1$-clique in H. The set $B_i \setminus \{v_i\}$ is a minimal separator of the graph H. See Figure 2.1 for an example of a partial 3-tree embedded in a 3-tree. Analogous to the role $(k+1)$-cliques of H play in a width k tree-decomposition of H, we call $B_i, 1 \leq i \leq n-k$, a $(k+1)$-bag in G under π, and each of its k-vertex subsets is similarly called a k-bag of G under π. The remaining definitions in this and the following sections are all for given graphs G, H, a peo $\pi = v_1, \dots, v_n$, and bags B_i as above. We first define a *peo-tree* P of G. The peo-tree P of G based on π is a rooted tree with nodes $V(P) = \{B_1, \dots, B_{n-k}\}$. The node B_{n-k} is the root of P; a node B_i, $1 \leq i < n-k$, has as its parent in P the node B_j, $i < j \leq n-k$, such that j is the minimum bag index with $|B_i \cap B_j| = k$ (note that this intersection does not contain v_i). The peo-tree P is a clique tree of H and also a width k tree-decomposition of both G and H (since $B_i \cap B_j$ is a separator of G). See Figure 2.1 for an example of a peo-tree.

3. Practical algorithms on partial *k*-trees. Many $\mathcal{NP}$-hard problems on graphs, when restricted to partial k-trees, for fixed values of k, have solution algorithms that execute in polynomial, or even linear time as a function of input graph size. In this section, we improve on the practicality of such algorithms, both in terms of their complexity and their derivation, by accounting for dependency on the treewidth k. Since each such algorithm is designed for fixed k, we consider a class of algorithms parameterized by k. We first define a binary parse tree of partial k-trees that is based on very simple graph operations. Then we discuss the derivation and complexity analysis of dynamic programming solution algorithms which follow this parse tree.

3.1. Binary parse tree. Based on the peo-tree of a partial k-tree as defined above, we construct a *binary parse tree.* We first introduce an algebra of i-sourced graphs. Terms in this algebra will evaluate to partial k-trees and their expression trees will be the binary parse trees of the resulting graphs.

Let a graph with i distinguished vertices (also called sources) have type G_i. We define the following graph operations:

- *Primitive*: $\rightarrow G_{k+1}$. This 0-ary operation introduces the graph $G[B]$, for some $(k+1)$-bag B.
- *Reduce*: $G_{k+1} \rightarrow G_k$. This unary operation eliminates a source designation of the $(k+1)$-st source vertex, leaving the graph otherwise unchanged.
- *Join*: $G_{k+1} \times G_k \rightarrow G_{k+1}$. This binary operation takes the union of its two argument graphs (say, A and B), where the sources of the second graph (a k-bag S_B) are a subset of the sources of the first graph (a $(k+1)$-bag S_A); these are the only shared vertices, and adjacencies for shared vertices are the same in both graphs. In other words, $V(A) \cap V(B) = S_B \subseteq S_A$ and $E(A[S_B]) = E(B[S_B])$, giving the resulting graph $Join(A, B) = (V(A) \cup V(B), E(A) \cup E(B))$ with sources S_A.
- *Forget*: $G_{k+1} \rightarrow G_0$. This operation eliminates the source designation of all source vertices.

The above definitions imply that in a term of the sourced graphs algebra that evaluates to a graph G, the source sets are $(k+1)$-bags and k-bags in a width k tree-decomposition of G. A binary parse tree of a graph G is the expression tree of such a term.

We show how to construct a binary parse tree from a peo-tree. Intuitively, each node of the peo-tree is "stretched" into a leaf-towards-root path of the binary parse tree. Let P be a peo-tree of a partial k-tree G under a peo π. For a node B_i of P, $1 \leq i \leq n-k$, with c children, define a path starting in a Primitive node evaluating to $G[B_i]$, with c Join nodes as interior vertices (one for each child of B_i), and ending in a Reduce node which drops the source designation of v_i. From the resulting collection of $|V(P)|$ Primitive-Join*-Reduce paths (note the total number of Join nodes is $|E(P)| = |V(P)| - 1$) we construct the binary parse tree by assigning Reduce nodes as children of the appropriate Join nodes. The only exception is the Reduce node associated with the root of P, which becomes the child of a new Forget node, the root of the resulting binary parse tree. The Reduce node associated with a node B_i of P with $parent(B_i)$ becomes the child of a Join node on the path associated with $parent(B_i)$. These assignments are easily done so that each Join node has a unique Reduce node as a child. Note that we have the freedom of choosing the order in which the children of a given node in P are Joined. This freedom, and also a possible choice of π, can be exploited to keep the resulting parse tree shallow, an important attribute in the design of parallel algorithms for partial k-trees. See Figure 3.1 for an example of a binary parse tree; note the $|V(P)|$ paths from leaves to their Reduce ancestors.

THEOREM 3.1. *Given a peo-tree P of a partial k-tree G, the graph algebra term that corresponds to the constructed binary parse tree T evaluates to G.*

Proof. The constructed tree T is the expression tree of a well-formed term in the given algebra, since Primitive nodes are exactly its leaves, and children of other nodes have the right types. Primitive nodes contain all edges of G, as they represent all subgraphs induced by $(k+1)$-bags of G. For each node B_i of P, the Reduce operation associated with it merely drops the source designation of v_i. Thus, we need only show that the Join operations act correctly on their argument graphs by identifying their sources. The Join operations are in a natural one-to-one correspondence with the edges of the peo-tree P, a tree-decomposition of G, where identification of vertices is done simply by taking the union of the two bags at endpoints of the edge. Let a Join

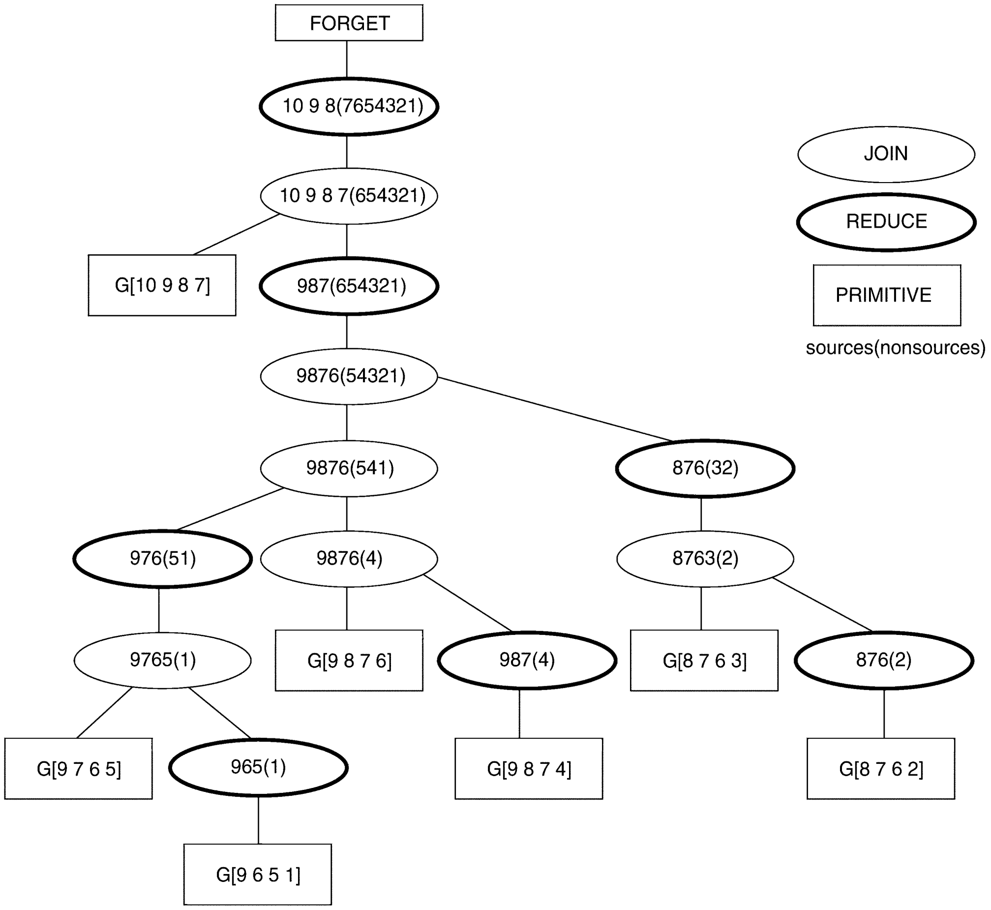


FIG. 3.1. *The binary parse tree T of the partial* 3*-tree G based on the peo-tree P (see Figure* 1*). Nodes $u \in V(T)$ labeled by $V(G_u)$ with nonsources in parenthesis.*

operation $Join(X, Y)$ correspond in this way to the edge between a node B_i of P and its parent B_j. We have $|B_i \cap B_j| = k$ with $B_i \setminus B_j = \{v_i\}$. By structural induction on T, we assume that subtrees representing X (of type G_{k+1}) and Y (of type G_k) have correctly identified vertices of G, so that the sources of X and Y are B_j and $B_i \setminus \{v_i\}$, respectively. The operation $Join(X, Y)$ identifies exactly the vertices $B_i \cap B_j$, and the resulting subtree rooted at this node has sources B_j. The Forget node at the root of T drops all source designations, so the graph algebra term that corresponds to the constructed binary parse tree T evaluates to G. □

We say that T *represents* G. Since P is a peo-tree with $n - k$ nodes, the binary parse tree T of G derived from P has $n - k$ Primitive leaves and $n - k$ Reduce nodes, one for each node of P, it has $n - k - 1$ Join nodes, one for each edge of P, and a single Forget node at the root.

3.2. Complexity analysis accounting for treewidth. The following algorithm design methodology is an adaptation to the binary parse tree of the earlier paradigm of [3]. A dynamic programming solution algorithm for a problem on a partial k-tree G will follow a bottom-up traversal of the binary parse tree T. As usual, with each node u of T we associate a data structure *table*. Each index of these tables represents a different constrained version of the problem. The corresponing entry of a table associated with a node u of T characterizes the optimal solutions to the constrained subproblem restricted to G_u, the sourced subgraph of G represented by the subtree of T rooted at u. The table of a leaf is initialized according to the base case, usually by a brute-force strategy. The table of an interior node is computed in a bottom-up traversal of T according to the tables of its children. The overall solution is obtained from the table at the root of T.

The paradigm for designing such algorithms is especially attractive for the class of vertex state problems. For a vertex state problem, we define a set of *vertex states*, that represent the different ways that a solution to a subproblem can affect a single source vertex.

We illustrate these concepts by an example. Suppose we want to solve the minimum dominating set problem on a partial k-tree G: minimize $|S|$ over all $S \subseteq V(G)$ such that every vertex not in S has at least one neighbor in S. Relative to some partial dominating set $S \subseteq V(G_u)$, a source vertex $v \in V(G_u)$ of a node u of the parse tree could be in one of three states: [dominator] $v \in S$; [nondominator, nondominated] $v \notin S \wedge |N_{G_u}(v) \cap S| = 0$; [nondominator, dominated] $v \notin S \wedge |N_{G_u}(v) \cap S| \geq 1$ (we call S a partial dominating set since at nonroot nodes of the parse tree source vertices can be in the state [nondominator, nondominated].) A table entry at node u gives the minimum number of dominator *nonsources* in G_u necessary to ensure that all nonsources are either dominators or dominated and that the vertex states for source vertices of G_u correspond to the table index.

Consider the binary parse tree in Figure 3.1. The table of the lower left Join node, labelled 9765(1), would have 3^4 entries, one entry for each assignment of one of the three vertex states to the four sources. In the subgraph associated with this Join node (see Figure 2.1), the sources 9,7,6, and 5 form a clique and vertices 5 and 6 share the neighboring nonsource vertex 1. We first describe the vertex state assignments that indicate an illegal configuration. Since we are solving a minimization problem, the corresponding table entries will have value $+\infty$:

- two sources have the pair of states [dominator] and [nondominator, nondominated];
- 7 or 9 have state [nondominator, dominated] but no source has state [dominator]; and
- 5 or 6 have state [nondominator, nondominated].

The latter case is illegal since then nonsource vertex 1 can neither be dominator nor dominated. For the remaining possibilities we have two cases:

- 5 or 6 have state [dominator] and
- 5 and 6 both have state [nondominator, dominated].

In the first case, table entries have value zero, since then no dominator nonsources are needed to dominate the nonsources and ensure these vertex states for sources. In the latter case, table entries have value one, since nonsource vertex 1 will then have to be a dominator itself (1 has neighbors 5 and 6 only and must be either dominated or dominator).

As mentioned earlier, the sources of G_u constitute a k or $(k + 1)$-bag and form a separator of G, which renders possible the table update for all the operations, and in particular $Join(A, B)$, based on the tables of A and B. An algorithm for a given problem must describe the tables involved and also describe how tables are computed during traversal of the parse tree. A candidate table is verified by the correctness proof of table update procedures for all operations involved. The introduction of Reduce and Join greatly simplifies this verification process, since these operations make only minimal changes to their argument graphs. In general, the algorithm computing a parameter $R(G)$ for a partial k-tree G given with a tree-decomposition has the following structure:

```
Algorithm-R, where $R$ is a graph parameter
Input: $G, k$, width $k$ tree-decomposition of $G$
Output: $R(G)$
(1) Based on tree-decomposition find a binary parse tree $T$ of $G$.
(2) Initialize Primitive-Tables at leaves of $T$.
(3) Traverse $T$ bottom-up using Join-Tables and Reduce-Table.
(4) Optimize Root-Table at root of $T$ gives $R(G)$.
```

Note that a tree-decomposition of width k is given as part of the input. For a given graph G on n vertices and any fixed k, Bodlaender [6] gives an $\mathrm{O}(n)$ algorithm for deciding whether the treewidth of G is at most k and, in the affirmative case, finding a width k tree-decomposition of G. The time complexity of his algorithm has a coefficient that is exponential in a polynomial in k, a polynomial which is not given explicitly in his paper. Improving on his algorithm to decrease this polynomial is an important problem that we do not address here. A construction of a k-tree embedding, given a tree-decomposition, is described in [21]. From a k-tree embedding it is straightforward to find a peo and the corresponding peo-tree and to construct the binary parse tree as described in the previous subsection. The time for step (1) becomes $\mathrm{O}(nk^2)$.

For a vertex state problem R with vertex state set A, the most expensive operation in the partial k-tree algorithm outlined above is the computation of the table associated with the Join operation. The complexity of this computation at a node of the parse tree is proportional to the number of pairs of indices, one index from the table of each of its two children. The table index sets associated with the children of a Join node for the problem R have size $|A|^k$ and $|A|^{k+1}$, and there are fewer than n Join nodes in the parse tree. The overall complexity of the algorithm, given a tree-decomposition, is dominated by the total of Join-Tables computation and is equal to $T(n, k, A) = \mathcal{O}(n|A|^{2k+1})$. When $|A|$ does not depend on n we have a finite-state problem and a linear-time algorithm on partial k-trees, for fixed k. Note that a vertex state problem can be solved in polynomial time whenever $|A|$ is polynomial in n.

In section 4, we define a class of vertex partitioning problems, and then in section 5 we give a procedure to produce a set of vertex states and table update procedures for each such problem definition.

4. Vertex partitioning problems.

In this section, we define a class of discrete optimization problems in which each vertex has a *state*, an attribute that is verifiable by a local neighborhood check.

Our motivation for defining these vertex partitioning problems is twofold. On the one hand, this formalism provides a general and uniform description of many existing problems in which a solution consists of a selection of vertex subsets. On the other, being vertex state problems, their restriction to partial k-trees have efficient solution algorithms that can be designed according to a general paradigm that follows their vertex partitioning description.

Considering partitions of the vertex set of a given graph is an attempt to unify graph properties expressible by either vertex subsets, such as independent dominating set, or by vertex coloring of graphs. Both these constructs are constrained by the structure of neighborhoods of vertices in different subsets. We define this formally.

Definition 4.1. *A* degree constraint *matrix D_q is a $q \times q$ matrix with entries being subsets of natural numbers $\{0, 1, \ldots\}$. A D_q*-partition *in a graph G is a partition $V_1, V_2, \ldots, V_q$ of $V(G)$ such that, for $1 \leq i, j \leq q$, we have $\forall v \in V_i : |N_G(v) \cap V_j| \in D_q[i, j]$.*

For technical reasons, we will allow a partition $V_1, \ldots, V_q$ of $V(G)$ to possibly have some empty partition classes; i.e., if the degree constraints on a partition class V_i are satisfied by $V_i = \emptyset$, then we allow this possibility. Given a degree constraint matrix D_q, it is natural to ask about the existence of a D_q-partition in an input graph. We call this the $\exists D_q$ problem. We might also ask for an extremal value of the cardinality of a vertex partition class over all D_q-partitions. Additionally, given a sequence of degree constraint matrices, $D_1, D_2, \ldots$, we might want to find an extremal value of q for which a D_q-partition exists in the input graph. We call these partition minimization and partition maximization problems.

To illustrate and give weight to this formalism, we express some well-known problems[1] in the terminology of vertex partitioning and also define new vertex partitioning problems as generalizations of old problems. In each case, correctness of the vertex partitioning formulation follows immediately from Definition 4.1.

4.1. Vertex subset problems. Many domination-type problems can be called *vertex subset* problems, as they ask for existence or optimization of a vertex subset with certain neighborhood properties. For example,

INDEPENDENT DOMINATING SET (IDS)
INSTANCE: Graph G.
QUESTION: Does G have an independent dominating set, i.e., is there a subset $S \subseteq V(G)$ such that S is independent (no two vertices in S are neighbors) and dominating (each vertex not in S has a neighbor in S)?

Equivalently, the IDS problem is defined with $\sigma = \{0\}$, $\rho = \{1, 2, \ldots\}$ and asks, Does G have a

$$D_2 = \begin{pmatrix} \sigma & \mathbb{N} \\ \rho & \mathbb{N} \end{pmatrix}$$

partition? Such a description defines a $[\rho, \sigma]$-*property.* Table 4.1 shows some classical vertex subset properties expressed using this notation [14, 8]. The complexity of optimization and existence problems defined over $[\rho, \sigma]$-properties for general graphs was studied in [26]; the existence problem is $\mathcal{NP}$-complete whenever both ρ and σ are finite nonempty sets and $0 \notin \rho$ (note that the IDS problem is trivial; every graph has such a set).

4.2. Uniform vertex partitioning problems. For a $[\rho, \sigma]$-property, we can also define partition maximization, partition minimization, and q-partition existence problems by taking the degree constraint matrix D_q with diagonal entries σ and off-diagonal entries ρ. We call these problems $[\rho, \sigma]$-*Partition* problems. For example,

GRAPH K-COLORABILITY[GT4]
INSTANCE: Graph G, positive integer k.
QUESTION: Is G k-colorable, i.e., is there a partition of $V(G)$ into k independent sets?

[1] [GTx] as a citation refers to the Graph Theory problem number x in Garey and Johnson [12].

TABLE 4.1
Some vertex subset properties.

ρ	σ	Standard terminology
$\{0, 1, ...\}$	$\{0\}$	Independent set
$\{1, 2, ...\}$	$\{0, 1, ...\}$	Dominating set
$\{0, 1\}$	$\{0\}$	Strong Stable set or 2-Packing
$\{1\}$	$\{0\}$	Perfect Code or Efficient Dominating set
$\{1, 2, ...\}$	$\{0\}$	Independent Dominating set
$\{1\}$	$\{0, 1, ...\}$	Perfect Dominating set
$\{1, 2, ...\}$	$\{1, 2, ...\}$	Total Dominating set
$\{1\}$	$\{1\}$	Total Perfect Dominating set
$\{0, 1\}$	$\{0, 1, ...\}$	Nearly Perfect set
$\{0, 1\}$	$\{0, 1\}$	Total Nearly Perfect set
$\{1\}$	$\{0, 1\}$	Weakly Perfect Dominating set
$\{0, 1, ...\}$	$\{0, 1, ..., p\}$	Induced Bounded-Degree subgraph
$\{p, p{+}1, ...\}$	$\{0, 1, ...\}$	p-Dominating set
$\{0, 1, ...\}$	$\{p\}$	Induced p-Regular subgraph

The graph k-colorability problem is defined with $\sigma = \{0\}$, $\rho = \{0, 1, \ldots\}$, D_k a $k \times k$ degree constraint matrix with diagonal elements σ and off-diagonal elements ρ, and asks: Does G have a D_k-partition?

Chromatic number is the partition minimization problem over degree constraint matrices $D_1, D_2, \ldots$, each one defined as D_k above. Similarly, *Domatic Number* [GT3] asks for a partition into maximum number of dominating sets ($\sigma = \mathbb{N}$, $\rho = \{1, 2, \ldots\}$) and *Partition into Perfect Matchings* [GT16] asks for a partition into minimum number of induced 1-regular subgraphs ($\sigma = \{1\}, \rho = \mathbb{N}$).

As an example of a generalization, consider the degree constraint matrix defining a partition into two Perfect Dominating Sets

$$D_2 = \begin{pmatrix} \mathbb{N} & \{1\} \\ \{1\} & \mathbb{N} \end{pmatrix}$$

and the question, Does a given graph G have a D_2-partition? This problem, which asks for a special cut of the graph, can also be posed as a vertex labelling question.

PERFECT MATCHING CUT
INSTANCE: Graph G.
QUESTION: Does G have a perfect matching cut, *i.e.* can the vertices of G be labelled with two labels such that each vertex has exactly one neighbor labelled differently from itself?

As an example, binomial trees and hypercubes have perfect matching cuts. This follows immediately from their iterative definition, i.e., the binomial tree B_0 is a single vertex and, for $i > 0$, the binomial tree B_i is constructed by adding a new leaf to every vertex in B_{i-1}. In [15], the complexity of uniform vertex partitioning problems is studied; Perfect Matching Cut is $\mathcal{NP}$-complete even when restricted to 3-regular graphs.

We can also consider vertex partitions into subsets with different properties. In general, take vertex subset properties $[\rho_1, \sigma_1], [\rho_2, \sigma_2], \ldots, [\rho_q, \sigma_q]$, and construct a degree constraint matrix D_q with column i having entry σ_i in position i and ρ_i elsewhere. The $\exists D_q$-problem asks if a graph G has a partition $V_1, V_2, \ldots, V_q$ of $V(G)$ where V_i is a $[\rho_i, \sigma_i]$-set in G.

4.3. Iterated removal problems. A variation of these problems arises by asking if a graph G has a partition $V_1, V_2, \ldots, V_q$, where V_i is a $[\rho, \sigma]$-set in $G \setminus (V_1 \cup V_2 \cup$

$\cdots \cup V_{i-1})$. To define this we use the degree constraint matrix D_q with diagonal entries σ, above-diagonal entries $\mathbb{N}$, and below-diagonal entries ρ. We call the resulting problems $[\rho, \sigma]$-*Iterated Removal* problems, since V_1 is a $[\rho, \sigma]$-set in $G_1 = G$, while V_2 is a $[\rho, \sigma]$-set in $G_2 = G_1 \setminus V_1$, and, in general, V_i is a $[\rho, \sigma]$-set in $G_i = G_{i-1} \setminus V_{i-1}$ $(1 < i \leq q)$. Here we may have to add the requirement that all partition classes be nonempty. For example,

GRAPH GRUNDY NUMBER [GT56, undirected version]
INSTANCE: Graph G, positive integer k.
QUESTION: Is the Grundy number of G at least k; i.e., is there a function $f : V(G) \rightarrow \{1, 2, \ldots, k'\}$ for some $k' \geq k$ such that, for each v, $f(v)$ is the least positive integer not contained in the set $\{f(u) : u \in N_G(v)\}$?

Note that if such a function f exists, then the color classes $V_i = \{v : f(v) = i\}, 1 \leq i \leq k'$, form a partition of $V(G)$, and each V_i is an independent dominating set in the graph $G \setminus (V_1 \cup V_2 \cup \cdots \cup V_{i-1})$. We can therefore define the Graph Grundy number problem as an *Iterated Removal* partition maximization problem. Let $\sigma = \{0\}$, $\rho = \{1, 2, \ldots\}$, and let $D_{k'}$ be a k' by k' degree constraint matrix with diagonal entries σ, above-diagonal entries $\mathbb{N}$, and below-diagonal entries ρ. The Graph Grundy number problem is: does G have a $D_{k'}$-partition, with nonempty partition classes, for some $k' \geq k$?

4.4. *H*-coloring and *H*-covering problems. For some vertex partitioning problems the degree constraint matrix is constructed using the adjacency matrix of an arbitrary graph H. For example,

H-COLORING (GRAPH HOMOMORPHISM)[GT52, fixed H version]
INSTANCE: Graph G.
QUESTION: Is there a homomorphism from G to H; i.e., is there a function f: $V(G) \rightarrow V(H)$ such that $uv \in E(G) \Rightarrow f(u)f(v) \in E(H)$?

We frame H-coloring as a vertex partitioning problem using the degree constraint matrix $D_{|V(H)|}$, obtained from the adjacency matrix of H by replacing 1-entries with $\mathbb{N}$ and 0-entries with $\{0\}$. The question to be asked is: Does G have a $D_{V(H)}$-partition? H-coloring is $\mathcal{NP}$-complete if H is not bipartite and polynomial-time solvable otherwise [16].

H-COVERING
INSTANCE: Graph G.
QUESTION: Does G cover H; i.e., is there a degree-preserving function $f : V(G) \rightarrow V(H)$ such that for all $v \in V(G)$ we have $\{f(u) : u \in N_G(v)\} = N_H(f(v))$?

Similarly, the H-cover problem, whose complexity was studied in [19], is formulated as an $\exists D_q$ problem using the adjacency matrix of H with singleton entries $\{1\}$ and $\{0\}$.

5. Algorithms for vertex partitioning problems on partial *k*-trees. We give algorithms for solving vertex partitioning problems on partial k-trees. These algorithms take a graph G and a width k tree-decomposition of G as input. Earlier work by Arnborg, Lagergren, and Seese [2] establishes the existence of pseudo-efficient algorithms for most, but not all, of these problems. They are pseudo-efficient in the sense that their time complexity is polynomial in the size of the input for fixed k, but with horrendous multiplicative constants ("towers" of powers of k). In contrast to this behavior, the algorithms presented here have running times with more reasonable bounds as a function of both input size and treewidth, e.g., $\mathcal{O}(n2^{4k})$ for well-known vertex subset problems. Since these problems are $\mathcal{NP}$-hard in general, and a tree-decomposition of width $n-1$ is easily found for any graph on n vertices, we should not expect polynomial dependence on k.

We devote most of this section to describe algorithms that solve $\exists D_q$-problems, for any degree constraint matrix D_q (as defined in the preceding section). In section 5.4 we describe extensions to partition minimization and maximization problems, and problems asking for an extremal value of the cardinality of a vertex partition class.

The algorithms will follow the general outline given in section 3.2, giving an answer YES if the input graph has a D_q-partition and NO otherwise. We first discuss the pertinent vertex and separator states and give a description of the tables involved in the algorithm. We then fill in details of table operations, prove their correctness, and give their time complexities.

5.1. Vertex and separator states. To define the set of vertex states A for an $\exists D_q$ problem, we start with the definition of the problem as captured by the degree constraint matrix D_q. To check whether a given partition $V_1, \dots, V_q$ of $V(G)$ is a D_q-partition we first assign to each vertex $v \in V(G)$ with $v \in V_i$ and $|N(v) \cap V_j| = d_j, j = 1, \dots, q$ the state $(i)(d_1, d_2, \dots, d_q)$ and then check if this state satisfies the constraints imposed by row i of D_q. The states allowed by D_q are called the *final* vertex states. In our partial k-tree algorithms we must consider a refined version of the original problem. For a given partition on a subgraph, a vertex may start out in a state not allowed by D_q and then acquire neighbors through Join operations so that the augmented partition indeed becomes a D_q-partition. To define this larger set of vertex states that are either final or can become final by adding new neighbors we need to define the *augmented* degree constraint matrix AD_q.

For $t \in \mathbb{N}$, we view $\geq t$ as a single element, and define the sets $Y_t \overset{df}{=} \{0, 1, \dots, t\}$, $W_0 = \{\geq 0\}$, $W_t \overset{df}{=} Y_{t-1} \cup \{\geq t\}$ if $t > 0$, and let $R \overset{df}{=} \{Y_t : t \in \mathbb{N}\} \cup \{W_t : t \in \mathbb{N}\} \cup \{\mathbb{N}\}$. Note that $|Y_t| = |W_t| = t + 1$. We now define a function $\beta : 2^{\mathbb{N}} \to R$ such that $AD_q[i,j] = \beta(D_q[i,j])$.

DEFINITION 5.1. $AD_q[i,j] = \beta(D_q[i,j])$, *where*

$$\beta(D_q[i,j]) = \begin{cases} Y_t & \text{if } \exists t \in D_q[i,j] \text{ such that } t = \max\{D_q[i,j]\}, \\ W_t & \text{if } \exists t \in D_q[i,j] \text{ with } t \text{ minimum s.t. } \{t, t+1, \dots\} \subseteq D_q[i,j], \\ \mathbb{N} & \text{otherwise.} \end{cases}$$

The set of vertex states A for an $\exists D_q$ problem is defined according to the rows of matrix AD_q. A vertex state consists of a pair $(i)(M)$, where $1 \leq i \leq q$ indexes a row of AD_q and M is an element of the Cartesian product $AD_q[i,1] \times AD_q[i,2] \times \cdots \times AD_q[i,q]$. We assume that $AD_q[i,j] \neq \mathbb{N}$ for any entry of AD_q, as otherwise we would have an infinite vertex state set and our algorithmic template would not work. Equivalently, we assume that every entry of the degree constraint matrix D_q is cofinite.

DEFINITION 5.2. *For an $\exists D_q$ problem, with cofinite entries of D_q, we define the vertex state set A and a subset, the final vertex state set $F \subseteq A$:*

$$A = \{(i)(M_{i1}M_{i2}\dots M_{iq}) \;\; : i \in \{1, \dots, q\} \wedge \forall j (j \in \{1, \dots, q\} \Rightarrow M_{ij} \in AD_q[i,j])\},$$

$$F = \{(i)(M_{i1}M_{i2}\dots M_{iq}) \in A : i \in \{1, \dots, q\} \wedge \forall j (j \in \{1, \dots, q\} \Rightarrow$$
$$(M_{i,j} \in D_q) \vee (AD_q[i,j] = W_t \wedge M_{i,j} = \geq t))\}.$$

$$D_3 = \begin{bmatrix} \{0\} & \{1,2,...\} & \{1,2,...\} \\ \{1,2,...\} & \{0\} & \{1,2,...\} \\ \{1,2,...\} & \{1,2,...\} & \{0\} \end{bmatrix}$$

$$AD_3 = \begin{bmatrix} \{0\} & \{0, \geq 1\} & \{0, \geq 1\} \\ \{0, \geq 1\} & \{0\} & \{0, \geq 1\} \\ \{0, \geq 1\} & \{0, \geq 1\} & \{0\} \end{bmatrix}$$

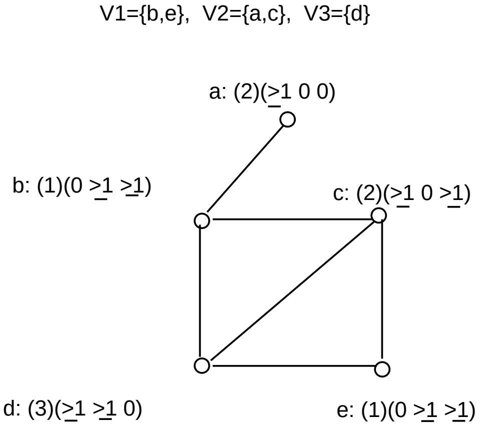


FIG. 5.1. *The degree constraint matrix D_3 and the augmented degree constraint matrix AD_3 for deciding whether there exists a partition into 3 independent dominating sets. To the right an example with the resulting vertex states for a given partition.*

Before continuing, let us first consider an example. Figure 5.1 shows the matrix D_3 such that the $\exists D_3$ problem decides whether vertices of a graph can be partitioned into 3 independent dominating sets. Note that the partition given in the example is not a D_3-partition, as can be seen from vertex a which needs a new neighbor in V_3 if this partition is to be augmented to a D_3-partition of some supergraph. By applying Definition 5.1 we get, for $i = 1, 2, 3$, $AD_3[i,i] = \beta(D_3[i,i]) = \beta(\{0\}) = Y_0 = \{0\}$ and, for $i \neq j$, $AD_3[i,j] = \beta(D_3[i,j]) = \beta(\{1,2,\ldots\}) = W_1 = \{0, \geq 1\}$. Applying Definition 5.2 we then get the 12 vertex states in the vertex state set A:

$$\begin{aligned} \{&(1)(0\ \ 0\ \ 0),\ \ (1)(0\ \ 0\ \ \geq 1),\ \ (1)(0\ \ \geq 1\ \ 0),\ \ (1)(0\ \ \geq 1\ \ \geq 1),\\ &(2)(0\ \ 0\ \ 0),\ \ (2)(0\ \ 0\ \ \geq 1),\ \ (2)(\geq 1\ \ 0\ \ 0),\ \ (2)(\geq 1\ \ 0\ \ \geq 1),\\ &(3)(0\ \ 0\ \ 0),\ \ (3)(0\ \ \geq 1\ \ 0),\ \ (3)(\geq 1\ \ 0\ \ 0),\ \ (3)(\geq 1\ \ \geq 1\ \ 0)\}. \end{aligned}$$

The three states at the rightmost column above (the 4th, 8th, and 12th) constitute the final state set F, corresponding to the three rows of the degree constraint matrix D_3. For any partition V_1, V_2, V_3 of $V(G)$ and a vertex $v \in V(G)$ this algorithm uses the following natural definition of $state_{V_1,V_2,V_3}(v)$:

$$state_{V_1,V_2,V_3}(v) = \begin{cases} (1)(0\quad 0\quad 0) & \text{if } v \in V_1 \text{ and } |N_G(v) \cap V_1| = 0 \wedge \\ & \quad \wedge |N_G(v) \cap V_2| = 0 \wedge |N_G(v) \cap V_3| = 0, \\ \ldots & \\ (3)(\geq 1\ \ \geq 1\ 0) & \text{if } v \in V_3 \text{ and } |N_G(v) \cap V_1| \geq 1 \wedge \\ & \quad \wedge |N_G(v) \cap V_2| \geq 1 \wedge |N_G(v) \cap V_3| = 0, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Note that this state function is total (defined everywhere) for the set of partitions that could possibly be augmented to a D_3-partition by addition of neighbors to the graph, i.e., all vertices of the graph are assigned a state if (and only if) V_1, V_2, V_3 are independent sets. For a general $\exists D_q$-problem the state function is total for all partitions $V_1, \ldots, V_q$ that could possibly be augmented to D_q-partitions.

We return to the discussion of a general $\exists D_q$-algorithm and examine first the size of the vertex state set A. Assume for simplicity that the matrix D_q has all diagonal entries equal and all off-diagonal entries equal, with $A_\sigma = AD_q[i,i]$ and $A_\rho = AD_q[i,j]$

for $i \neq j$. With A the set of vertex states for the $\exists D_q$-problem, we thus have $|A| = q|A_\sigma||A_\rho|^{q-1}$ vertex states, since vertex states are of the form $(i)(M_{i1}M_{i2}\dots M_{iq})$ with $i \in \{1,2,\dots,q\}$, $M_{ii} \in A_\sigma$, and $M_{ij} \in A_\rho$ for $i \neq j$. We now examine the index set I_k of the table at a node u of the parse tree representing a subgraph with k sources. The table at node u will have $|I_k|$ entries. Let the bag of sources (the separator) at node u be $B_u = \{w_1, w_2, \dots, w_k\}$. Each of the sources can take on a vertex state in $|A|$ and the table thus has index set $I_k = \{\mathbf{s} = s_1, \dots, s_k\}$ where $s_i \in A$. Thus the size of the table is $|I_k| = |A|^k = q^k|A_\sigma|^k|A_\rho|^{k(q-1)}$. For the earlier example, partition into three independent dominating sets, we get $|I_k| = 12^k = 3^k1^k2^{k(3-1)}$.

Next, we discuss the values of table entries. For D_q, a subgraph G_u with sources $B_u = \{w_1, \dots, w_k\}$, and a k-vector of vertex states $\mathbf{s} = \{s_1, \dots, s_k\}$, $\forall i\ s_i \in A$, we define a family Ψ of equivalent partitions $V_1, V_2, \dots, V_q$ of $V(G_u)$ such that in G_u, a source w_i has state s_i and a nonsource vertex has a final state in F. Note that for a nonsource vertex $v \in V_i$ we thus have $|N_G(v) \cap V_j| \in D_q[i,j], j = 1, \dots, q$, as dictated by the degree constraint matrix.

DEFINITION 5.3. *For problem $\exists D_q$, with vertex states A and final states F, a graph G_u with sources $B_u = \{w_1, w_2, \dots, w_k\}$ and a k-vector $\mathbf{s} = s_1, \dots, s_k : \forall i$ $s_i \in A$, we define*

$$\Psi \overset{df}{=} \{V_1, \dots, V_q \textit{ a q-partition of } V(G_u) : \forall w_i \in B_u \quad \forall v \in V(G_u) \setminus B_u$$
$$(\mathit{state}_{V_1,\dots,V_q}(w_i) = s_i \textit{ and } \mathit{state}_{V_1,\dots,V_q}(v) \in F)\}.$$

Ψ forms an equivalence class of solutions to the subproblem on G_u, and its elements are called Ψ-partitions respecting G_u and $\mathbf{s}$. The binary contents of $\mathit{Table}_u[\mathbf{s}]$ records whether any solution respecting G_u and $\mathbf{s}$ exists.

DEFINITION 5.4.

$$\mathit{Table}_u[\mathbf{s}] = \begin{cases} 1 & \textit{if } \Psi \neq \emptyset, \\ 0 & \textit{if } \Psi = \emptyset. \end{cases}$$

5.2. Table operations. We now elaborate on the operations of Initialize-Table, Reduce-Table, Join-Tables, and Optimize-Root-Table in the context of an $\exists D_q$-problem with vertex states A and final states F. Each of the following subsections defines the appropriate procedure, gives the proof of its correctness, and analyzes its complexity.

5.2.1. Initialize-Tables. A leaf u of T is a Primitive node, and G_u is the graph $G[B_u]$, where $B_u = \{w_1, \dots, w_{k+1}\}$. Let $\mathit{Partitions}(B_u)$ be the family of all q^{k+1} partitions of B_u into partition classes $V_1, \dots, V_q$. Following Definition 5.4, we initialize Table_u by a brute-force method in two steps:

(1) $\forall \mathbf{s} \in I_{k+1} : \mathit{Table}_u[\mathbf{s}] := 0$,

(2) $\forall V_1, V_2, \dots, V_q \in \mathit{Partitions}(B_u)$: if in $G[B_u]$ for $i = 1, \dots, k+1$ we have $\mathit{state}_{V_1,\dots,V_q}(w_i) = s_i \in A$, then for $\mathbf{s} = s_1, \dots, s_{k+1}$, $\mathit{Table}_u[\mathbf{s}] := 1$.

We need only consider partitions that assign a state in A to all vertices, since any other partition is in violation of D_q and cannot be augmented to a D_q-partition of the input graph. The complexity of this initialization for each leaf of T is $\mathcal{O}(|I_{k+1}| + (k+1)q^{k+2})$, since for each partition we must check the q neighborhood constraints of $k+1$ vertices.

5.2.2. Reduce-Table. A Reduce node u of T has a single child a such that $B_u = \{w_1, \dots, w_k\}$ and $B_a = \{w_1, \dots, w_{k+1}\}$. We compute Table_u based on Table_a as follows:

$$\forall \mathbf{s} \in I_k : \mathit{Table}_u[\mathbf{s}] := \bigvee\nolimits_{\mathbf{p}} \quad \{\mathit{Table}_a[\mathbf{p}]\},$$

where the disjunction is over all $\mathbf{p} = \{p_1, \ldots, p_{k+1}\} \in I_{k+1}$ with $\forall l : 1 \leq l \leq k, p_l = s_l$ and $p_{k+1} \in F$. Correctness of the operation follows by noting that G_a and G_u designate the same subgraph of G and differ only by w_{k+1} not being a source in G_u. By definition, $Table_u[\mathbf{s}]$ should store a 1 if and only if there is some Ψ-set respecting G_a and $\mathbf{s}$, where the state of nonsources, and thus also w_{k+1}, is constrained by D_q, and thus assigned a final state. The complexity of this operation for each Reduce node of T is $\mathcal{O}(|I_{k+1}|)$, assuming that in constant time we can both (i) decide whether an index of $Table_a$ represents a final state for w_{k+1} and (ii) access the corresponding entry of $Table_u$.

5.2.3. Join-Tables. A Join node u of T has children a and b such that $B_u = B_a = \{w_1, \ldots, w_{k+1}\}$ and $B_b = \{w_1, \ldots, w_k\}$ is a k-subset of B_a. Moreover, G_a and G_b share exactly the subgraph induced by B_b, $G[B_b]$. We compute $Table_u$ by considering all pairs of table entries of the form $Table_a[\mathbf{p}], Table_b[\mathbf{r}]$. Recall that the separator state $\mathbf{p}$ consists of $k+1$ vertex states $p_1, p_2, \ldots, p_{k+1}$, where the state p_i is associated with vertex w_i. For a vertex state $p_i = (j)(M_{j1}, \ldots, M_{jq})$ we call j the partition class index, $class(p_i)$, and the cardinality M_{jl}, $size(p_i, l)$. In the algorithm for the Join operation, we first check that $\mathbf{p}, \mathbf{r}$ is a *compatible* separator state pair, meaning the partition class assigned to vertex $w_i, i \in \{1, \ldots, k\}$, is identical in both $\mathbf{p}$ and $\mathbf{r}$.

$$compatible(\mathbf{p}, \mathbf{r}) := \begin{cases} 1 & \text{if } class(p_i) = class(r_i)\ \forall i \in \{1, \ldots, k\}, \\ 0 & \text{otherwise.} \end{cases}$$

We then combine, in a manner described below, for each $w_i, i \in \{1, \ldots, k+1\}$, the contributions from $\mathbf{p}$ and $\mathbf{r}$ to give the resulting separator state $\mathbf{s} = combine(\mathbf{p}, \mathbf{r})$, and update $Table_u[\mathbf{s}]$ based on $Table_a[\mathbf{p}]$ and $Table_b[\mathbf{r}]$. For a vertex w_i under $\mathbf{s}$, the resulting q-vector of neighborhood sizes is computed by (componentwise) addition of its q-vectors under $\mathbf{p}$ and $\mathbf{r}$. Moreover, since the neighbors w_i has in $B_b = \{w_1, \ldots, w_k\}$ are the same in both G_a and G_b, we must subtract the shared V_j neighbors w_i has in B_b under $\mathbf{p}$ and $\mathbf{r}$. This addition at the jth component is performed using the following definition of $a \oplus b \ominus c$ which adds two size values a, b and subtracts $c \in \mathbb{N}$. The definition of $a \oplus b \ominus c$ depends on whether a, b are of type Y_t or W_t, and returns a value of the same type, unless undefined.

DEFINITION 5.5. *For $a, b \in Y_t$ and $c \in \mathbb{N}$*

$$a \oplus b \ominus c = \begin{cases} a + b - c & \textit{if } a + b - c \in Y_t, \\ \uparrow & \textit{otherwise.} \end{cases}$$

For $a, b \in W_t$ and $c \in \mathbb{N}$,

$$a \oplus b \ominus c = \begin{cases} \geq t & \textit{if either } a \textit{ or } b \textit{ is the element } \geq t, \\ \geq t & \textit{if } a + b - c \in \{t, t+1, \ldots\}, \\ a + b - c & \textit{if } a + b - c \in \{0, 1, \ldots, t-1\}, \\ \uparrow & \textit{otherwise.} \end{cases}$$

We thus use

$$\begin{aligned} combine(\mathbf{p}, \mathbf{r}) := s_1, s_2, \ldots, s_{k+1},\ & \text{where } \forall i \in \{1, \ldots, k\},\ \ \forall j \in \{1, \ldots, q\}, \\ & class(s_i) = class(r_i) = class(p_i),\ \text{and } s_{k+1} = p_{k+1}, \text{ and} \\ & size(s_i, j) = size(p_i, j) \oplus size(r_i, j) \ominus \end{aligned}$$

$$|\{w_l \in B_b : (w_i, w_l) \in E(G) \wedge class(p_l) = j\}|.$$

We can now state the two-step procedure for the Join operation:

(1) $\forall \mathbf{s} \in I_{k+1} : Table_u[\mathbf{s}] := 0$,

(2) $\forall (\mathbf{p} \in I_{k+1}, \mathbf{r} \in I_k) :$ if $compatible(\mathbf{p}, \mathbf{r})$ and $Table_a[\mathbf{p}] = Table_b[\mathbf{r}] = 1$,
then $Table_u[combine(\mathbf{p}, \mathbf{r})] := 1$

In step (2) we assume that $Table_u$ is accessed only if $combine(\mathbf{p}, \mathbf{r})$ designates a vertex state, i.e., only if each of its *size* components is defined.

THEOREM 5.6. *The procedure given for the Join operation at a node u with children a and b correctly updates $Table_u$ based on $Table_a$ and $Table_b$.*

Proof. We argue the correctness of the Join operation at a node u with sources $B_u = \{w_1, \ldots, w_{k+1}\}$, based on correct table entries at its children a and b, with notation as before. Consider any $\mathbf{s} = s_1, \ldots, s_{k+1}$ such that there exists a partition $V_1, \ldots, V_q$ of $V(G_u)$ respecting D_q with $state_{V_1,\ldots,V_q}(w_i) = s_i$ for $i = 1$ to $k+1$ in the graph G_u. We will show that $Table_u[\mathbf{s}]$ is then correctly set to the value 1. Let $A_1, \ldots, A_q$ and $B_1, \ldots, B_q$ be the induced partitions on $V(G_a)$ and $V(G_b)$, respectively, i.e., $V_i \cap V(G_a) = A_i$ and $V_i \cap V(G_b) = B_i$. Let $\mathbf{p} = p_1, \ldots, p_{k+1}$ and $\mathbf{r} = r_1, \ldots, r_k$ be defined by $p_i = state_{A_1,\ldots,A_q}(w_i)$ in G_a and $r_i = state_{B_1,\ldots,B_q}(w_i)$ in G_b, respectively. By the assumption that $Table_a$ and $Table_b$ are correct we must have $Table_a[\mathbf{p}] = Table_b[\mathbf{r}] = 1$. This follows since any vertex in $V(G_a) \setminus B_u$ has the exact same state in G_a under $A_1, \ldots, A_q$ as it has in G_u under $V_1, \ldots, V_q$, by the fact that there are no adjacencies between a vertex in $V(G_a) \setminus B_u$ and a vertex in $V(G_b) \setminus B_u$. Similarly for G_b. We can check that from the definitions we have $compatible(\mathbf{p}, \mathbf{r}) = 1$ and $combine(\mathbf{p}, \mathbf{r}) = \mathbf{s}$, so indeed $Table_u[\mathbf{s}]$ is set to 1 when the pair $\mathbf{p}, \mathbf{r}$ is considered by the algorithm.

Now consider an $\mathbf{s}$ such that there is no q-partition of $V(G_u)$ respecting D_q and resulting in $\mathbf{s}$ as the state for the separator. We will show, by contradiction, that in this case $Table_u[\mathbf{s}]$ is set to 0 initially and then never altered. If $Table_u[\mathbf{s}] = 1$, there must be a compatible pair $\mathbf{p}, \mathbf{r}$ such that $combine(\mathbf{p}, \mathbf{r}) = \mathbf{s}$ and $Table_a[\mathbf{p}] = Table_b[\mathbf{r}] = 1$. Let $A_1, \ldots, A_q$ and $B_1, \ldots, B_q$ be partitions of $V(G_a)$ and $V(G_b)$, respectively, that cause these table entries to be set to 1. Then $V_1, \ldots, V_q$ defined by $V_i = A_i \cup B_i$ is a q-partition of $V(G_u)$ respecting D_q such that the resulting state for the separator is $\mathbf{s}$, because $B_u = \{w_1, \ldots, w_{k+1}\}$ separates G_u into $G_a \setminus B_u$ and $G_b \setminus B_u$. This contradicts our assumption that such a q-partition does not exist. We conclude that the Join-Tables operation is correct. □

For each Join node of T, the complexity of Join-Tables is $\mathcal{O}(kq|I_k||I_{k+1}|)$ since any pair of entries from tables of children is considered at most once, and for each compatible pair the combine operation considers kq size pairs.

5.2.4. Optimize-Root-Table. Let the root of T have child r with $B_r = \{w_1, \ldots, w_k\}$. We decide whether G has a D_q-partition based on $Table_r$ as follows:

YES if $\exists \mathbf{s} = s_1, \ldots, s_k \in I_k$ with $Table_r[\mathbf{s}] = 1$ and $s_i \in F$ for $1 \leq i \leq k$,
NO otherwise.

Correctness of this optimization follows from the definition of table entries and final states and the fact that G_r is the graph G with sources B_r. The complexity of Optimize-Root-Table at the root of T is $\mathcal{O}(|I_{k+1}|)$, assuming that in constant time we can decide whether an index of $Table_r$ represents a final state for each vertex in B_r.

5.3. Overall correctness and complexity. Correctness of an algorithm based on this algoritmic template follows by induction on the binary parse tree T. As noted

Table 5.1
Time complexity for specific problems on partial k-trees of n vertices.

Problem	q	$\lvert A_\sigma\rvert$	$\lvert A_\rho\rvert$	Time complexity
CHROMATIC NUMBER	$1 \leq q \leq k+1$	1	1	$\mathcal{O}(nk^{2(k+1)})$
q-COLORING	q	1	1	$\mathcal{O}(nq^{2(k+1)})$
H-COVER	$q = \lvert V(H)\rvert$	1	2	$\mathcal{O}(n2^{3k\lvert V(H)\rvert})$
H-COLOR	$q = \lvert V(H)\rvert$	1	1	$\mathcal{O}(n\lvert V(H)\rvert^{2(k+1)})$
DOMATIC NUMBER	$1 \leq q \leq k+1$	1	2	$\mathcal{O}(n2^{3k^2})$
GRUNDY NUMBER	$1 \leq q \leq 1 + k\log n$	1	2	$\mathcal{O}(n^{3k^2})$
ITERATED DOM. REMOVAL	$1 \leq q \leq 1 + k\log n$	1	2	$\mathcal{O}(n^{3k^2})$

in section 3.1, T has $n-k$ Primitive nodes, $n-k$ Reduce nodes, and $n-k-1$ Join nodes. The algorithm finds the binary parse tree T, traverses it bottom-up executing the respective operation at each of its nodes, and performs Optimize-Root-Table at the root.

Theorem 5.7. *The time complexity for solving an $\exists D_q$ problem, entries of D_q cofinite, with vertex state set A, on a partial k-tree G with n vertices, given a width k tree-decomposition of G, is $\mathcal{O}(nkq|A|^{2k+1})$. If the augmented degree constraint matrix AD_q has $|A_\sigma| = \max_i\{|AD_q[i,i]|\}$ and $|A_\rho| = \max_{i\neq j}\{|AD_q[i,j]|\}$ it can be expressed as $\mathcal{O}(nq^{2(k+1)}|A_\sigma|^{2k+1}|A_\rho|^{(2k+1)(q-1)})$.*

Proof. The first bound holds since the most expensive operation is Join-Tables which costs $\mathcal{O}(kq|I_k||I_{k+1}|)$ where $|I_k| = |A|^k$, and there are less than n Join-Table nodes in the binary parse tree. The refined bound holds since $|A| = q|A_\sigma||A_\rho|^{q-1}$. □

Note that the last bound holds in particular when AD_q has all diagonal entries equal to A_σ and all off-diagonal entries equal to A_ρ.

5.4. Extensions. Here we mention a few natural extensions of the problems described above: partition maximization and minimization, construction of a D_q-partition, complexity of vertex subset problems, optimization over a partition class cardinality, and, finally, implications on optimizations problems without a constant bound.

Recall that given a sequence of degree constraint matrices, $D_1, D_2, \ldots$, partition minimization or maximization problems involve finding an extremal value of q for which a D_q-partition exists in the input graph. To solve such problems with an upper bound $f(n,k)$ on the parameter in question for n-vertex partial k-trees, we need at most $f(n,k)$ calls to the $\exists D_q$ algorithm, for different values of q. Several parameters are bounded by the treewidth only, e.g., chromatic number and domatic number are bounded by $k+1$ on partial k-trees. We call a partition maximization (respectively, minimization) parameter *monotone* if existence of a D_q-partition implies the existence of a D_{q-1}-partition (respectively, a D_{q+1}-partition). For monotone properties we can apply binary search so that $\log f(n,k)$ calls to the $\exists D_q$ algorithm suffices. Resulting time bounds for specific problems are shown in Table 5.1 and also discussed in the following sections.

To construct a D_q-partition, in case of a positive answer for the $\exists D_q$ problem, we add pointers from a positive table entry to the table entries of children which updated it positively.

Table 4.1 lists some vertex subset properties, which we called $[\rho,\sigma]$-properties, expressible by a degree constraint matrix D_2. Various $\mathcal{NP}$-hard optimization problems ask for an extremal value of the cardinality of a vertex subset with some $[\rho,\sigma]$

property. In an earlier paper [28], we give algorithms on partial k-trees for solving these problems.

THEOREM 5.8 (see [28]). *Given a tree-decomposition of width k of a graph G, any optimization problem over a $[\rho, \sigma]$-property, with both ρ and σ cofinite, can be solved on G in $\mathcal{O}(n(|\beta(\rho)| + |\beta(\sigma)|)^{2k+1})$ steps.*

Problems defined over properties derived from Table 4.1 have complexity $\mathcal{O}(n2^{4k})$ (for parameterized properties we assume $p \leq 2$). Those algorithms are very similar to the ones given here, with values of table entries defined to be

$$Table_u[\mathbf{s}] \overset{df}{=} \begin{cases} \perp & \text{if } \Psi = \emptyset, \\ optimum_{\{V_1,V_2\}\in\Psi}\{|V_1|\} & \text{otherwise,} \end{cases}$$

and table operations altered similarly to optimize this value. Any vertex partitioning problem optimizing over the cardinality of a partition class can be solved in a similar manner. The time complexity of the resulting algorithm for a problem given by the degree constraint matrix D_q remains as given in Theorem 5.7.

In section 4 we discussed several new problems, including the general classes of $[\rho, \sigma]$ uniform partition problems, $[\rho, \sigma]$ iterated removal problems, H-coloring, and H-covering problems. All these problems are encompassed by Theorem 5.7. Polynomial-time algorithms for partition maximization or minimization problems on partial k-trees are constructible in this manner only if an appropriate bound holds for the parameter in question. In the next section we first show that the Grundy number of an n-vertex partial k-tree has an upper bound logarithmic in n and then construct a polynomial-time solution algorithm.

6. Grundy number algorithm. The Grundy number of a graph, defined in section 4.3, is a tight upper bound on the number of colors used by the following "naive greedy coloring" algorithm: repeatedly select an uncolored vertex and color it with the least available positive integer. The Grundy number is the highest color thus assigned to any vertex, maximized over all orderings of vertices, with the vertex partitioning iterated removal definition based on the fact that the set of vertices with color i form an independent dominating set in the graph induced by vertices with color i or higher.

Computing the Grundy number of an undirected graph is $\mathcal{NP}$-complete even for bipartite graphs and for chordal graphs [22]. A binomial tree on 2^{q-1} vertices, defined in section 4.2, has Grundy number q [13]. In general the nonexistence of an $f(k)$ upper bound on the Grundy number of a partial k-tree explains the lack of a description of this problem in EMSOL [20] (note that [2] mistakenly gives a different impression). For trees there exists a linear time algorithm [13], but until now it was an open question whether polynomial-time algorithms existed even for 2-trees.

The definition of a Grundy number as a vertex partitioning problem requires all partition classes to be nonempty. In this section, we first show how the algorithm template of section 5 can be easily adjusted to enforce this requirement. For a partial k-tree G with n vertices, we prove a logarithmic in n upper bound on the Grundy number of G. These results suffice to show the polynomial-time complexity of computing the Grundy number of any partial k-tree for fixed k.

To facilitate the presentation of these results, we reverse the ordering of the partition classes in the definition of a Grundy number from section 4; this is expressed by the degree constraint matrix D_q with diagonal entries $\{0\}$, above-diagonal entries $\mathbb{P}$, and below-diagonal entries $\mathbb{N}$. Thus, for a graph G, the Grundy number $GN(G)$ is the largest value of q such that its vertices $V(G)$ can be partitioned into nonempty

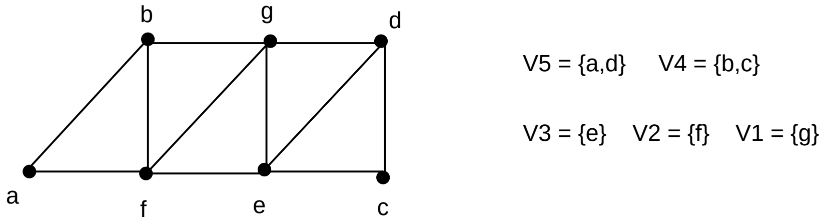


FIG. 6.1. *A* 2*-tree on* 7 *vertices with Grundy number* 5 *and an appropriate partition* $V1, V2, \ldots, V5$.

classes $V_1, V_2, \ldots, V_q$ with the constraint that for $i = 1, \ldots, q$, V_i is an independent set and every vertex in V_i has at least one neighbor in each of the sets $V_{i+1}, V_{i+2}, \ldots, V_q$ (see Figure 6.1). Note that if we have at least one vertex $v \in V_1$ then this guarantees that every partition class is nonempty, since D_q requires v to have at least one neighbor in each of $V_2, V_3, \ldots, V_q$. In the algorithm for deciding whether a partial k-tree has a D_q-partition with nonempty classes, with D_q as described above, we extend the value of a table entry $Table_u[\mathbf{s}]$ by a single extra bit called *nonempty*. This bit will record whether there exists any partition $V_1, \ldots, V_q$ respecting G_u and the separator state $\mathbf{s}$ such that $V_1 \neq \emptyset$. In the following, we use notation as given in section 5, with the definition of table entries

$$Table_u[\mathbf{s}] = \begin{cases} \langle 0,0\rangle & \text{if } \Psi = \emptyset, \\ \langle 1,0\rangle & \text{if } \Psi \neq \emptyset, \text{ but } \nexists V_1, V_2, \ldots, V_q \in \Psi \text{ with } V_1 \neq \emptyset, \\ \langle 1,1\rangle & \text{if } \Psi \neq \emptyset, \text{ and } \exists V_1, V_2, \ldots, V_q \in \Psi \text{ with } V_1 \neq \emptyset. \end{cases}$$

The two-step Initialize-Table procedure becomes

(1) $\forall \mathbf{s} \in I_{k+1} : Table_u[\mathbf{s}] := \langle 0,0\rangle$,

(2) $\forall V_1, V_2, \ldots, V_q \in Partition(B_u)$: if in $G[B_u]$, for $i = 1, \ldots, k+1$, we have $state_{V_1,\ldots,V_q}(w_i) = s_i \in A$, then for $\mathbf{s} = s_1, \ldots, s_{k+1}$
if $V_1 = \emptyset$ set $Table_u[\mathbf{s}] := \langle 1,0\rangle$
else if $V_1 \neq \emptyset$ set $Table_u[\mathbf{s}] := \langle 1,1\rangle$.

Note that for a leaf u of the binary parse tree of G, all vertices of G_u are sources so the separator state $\mathbf{s}$, in step (2) above, contains the information determining if V_1 is empty. The Reduce-Table procedure remains as given in section 5, except that the disjunction is taken pairwise over both bits in the values of table entries, i.e., $\langle a,b\rangle \vee \langle c,d\rangle = \langle a \vee c\rangle, \langle b \vee d\rangle$. For the Join-Table procedure, the concepts of compatibility and combining of pairs are unchanged, and the two-step update procedure becomes

(1) $\forall \mathbf{s} \in I_{k+1} : Table_u[\mathbf{s}] := \langle 0,0\rangle$,

(2) $\forall(\mathbf{p} \in I_{k+1}, \mathbf{r} \in I_k)$: if $compatible(\mathbf{p},\mathbf{r})$ and $Table_a[\mathbf{p}] = \langle 1,x\rangle$ and $Table_b[\mathbf{r}] = \langle 1,y\rangle$ and $Table_u[combine(\mathbf{p},\mathbf{r})] = \langle z,w\rangle$, then $Table_u[combine(\mathbf{p},\mathbf{r})] := \langle 1, x \vee y \vee w\rangle$.

Optimize-Root-Table becomes

YES if $\exists \mathbf{s} = s_1, \ldots, s_k \in I_k$ such that $Table_r[\mathbf{s}] = \langle 1,1\rangle$ and $s_i \in F$ for $1 \leq i \leq k$,

NO otherwise.

It is easy to see that the time complexity of the resulting algorithm remains as described by Theorem 5.7.

We now turn to the bound on the Grundy number $GN(G)$ of a partial k-tree G. Since the Grundy number of a graph may increase when some edges of the graph are removed, we cannot restrict our attention to k-trees, but must consider partial k-trees. A tree (i.e., a 1-tree) with Grundy number q, witnessed by a (Grundy) partition $V_1, \dots, V_q$, must have at least 2^{q-1} vertices since each vertex of the set $\bigcup_{1 \le i < j} V_i$ has a unique neighbor in V_j, thus doubling the size of $\bigcup_{1 \le i \le j} V_i$ for each consecutive $1 < j \le q$. This argument relies on the fact that 1-trees do not have cycles. For a partial k-tree G with $k \ge 2$ and Grundy number q, we cannot guarantee the existence of a perfect elimination ordering of vertices that respects a $V_q, \dots, V_1$ Grundy partition of $V(G)$, as in the 1-tree example above. See Figure 6.1 for an example of a 2-tree on 7 vertices with Grundy number 5 that does not have a perfect elimination ordering respecting the partial order given by any Grundy partition $V_5, V_4, \dots, V_1$. Hence, the upper bound given below has a somewhat less trivial proof than the 1-tree case.

THEOREM 6.1. *The Grundy number of a partial k-tree G on n vertices, $n \ge k \ge 1$, is at most $1 + k \log_2 n$.*

Proof. Let the Grundy number of G be $GN(G) = q$, with $V_1, V_2, \dots, V_q$ an appropriate partition of $V(G)$ as described above. For $1 \le i \le q$, define G_i to be the graph $G \setminus (\cup V_j, j > i)$. Thus $G_q = G$ and, in general, G_i is the graph induced by vertices $V_1 \cup V_2 \cup \cdots \cup V_i$, with V_i a dominating set of G_i. Let $n_i = |V(G_i)|$ and $m_i = |A(G_i)|$. By induction on i from k to q we show that in this range

$$n_i \ge \left(\frac{k+1}{k}\right)^{i-1}.$$

For the base case $i = k$ we have $(2/1)^0 \le 1 \le n_1$ and $(3/2)^1 < 2 \le n_2$ and for $k \ge 3$ $(1 + 1/k)^{k-1} \le (1 + 1/k)^k \le e < 3 \le n_k$. Note that the inequality is strict for $k \ge 2$. We continue with the inductive step of the proof, with the inductive assumption that the inequality holds for j in the range k to $i-1$ and establish the inequality for $j = i$. Note that $m_i - m_{i-1}$ counts the number of edges in G_i, with at least one endpoint in V_i. Since every vertex in $V(G_{i-1}) = V_1 \cup V_2 \cup \cdots \cup V_{i-1}$ has at least one G_i-neighbor in V_i, we get a lower bound on m_i

$$m_i \ge m_{i-1} + n_{i-1}.$$

G_i is a subgraph of a k-tree, and if $i \ge k$, then it is a partial k-tree on $n_i \ge k$ vertices. It is well-known that G_i is then a subgraph of a k-tree on n_i vertices, and from the iterative construction of k-trees it is easy to show that we have

$$m_i \le \frac{k(k-1)}{2} + (n_i - k)k.$$

Rearranging terms, we get the following bound on n_i for $k \le i \le q$:

$$n_i \ge \frac{m_i}{k} + \frac{k+1}{2}.$$

Repeatedly substituting the m_i bound in the above, we get

$$n_i \ge \frac{m_{i-1} + n_{i-1}}{k} + \frac{k+1}{2} \ge \cdots \ge \frac{n_k + n_{k+1} + \cdots + n_{i-2} + n_{i-1}}{k} + \frac{m_k}{k} + \frac{k+1}{2}.$$

In the right-hand side we substitute, for all n_j, the inductive bound $n_j \geq (\frac{k+1}{k})^{j-1}$ to get

$$n_i \geq \frac{1}{k} \sum_{j=k-1}^{i-2} \left(\frac{k+1}{k}\right)^j + \frac{m_k}{k} + \frac{k+1}{2} = \left(\frac{k+1}{k}\right)^{i-1} - \left(\frac{k+1}{k}\right)^{k-1} + \frac{m_k}{k} + \frac{k+1}{2}.$$

Since V_j is a dominating set in G_j for $1 \leq j \leq k$, we must have $m_k \geq (k-1)k/2$, which we substitute in the above to get the desired bound

$$n_i \geq \left(\frac{k+1}{k}\right)^{i-1} - \left(\frac{k+1}{k}\right)^{k-1} + k \geq \left(\frac{k+1}{k}\right)^{i-1}.$$

Note that the last bound is strict for $k \geq 2$. For $i = q$, we thus get $q \leq 1 + \log_{(k+1)/k} n_q$ (note that $q = GN(G)$ and $n_q = n$), which is a tight bound for $k = 1$. For $k \geq 2$, the base is not an integer and, because of the strict inequality mentioned above, we can apply the floor function to the log. Converting bases of the logarithm we get $GN(G) \leq 1 + (\log_2 \frac{k+1}{k})^{-1} \log_2 n \leq 1 + k \log_2 n$. □

THEOREM 6.2. *Given a partial k-tree G on n vertices its Grundy number can be found in $\mathcal{O}(n^{3k^2})$ time.*

Proof. First note that a tree-decomposition can be found in time linear in n [6]. Define the Grundy number problem using the degree constraint matrix D_q with diagonal entries $\{0\}$, above-diagonal entries $\mathbb{P}$, and below-diagonal entries $\mathbb{N}$. We then use the algorithm from section 6.2.2 extended with the *nonempty* information as described above. The correctness of each table operation procedure is easily established, so that by induction over the parse tree we can conclude that the root-optimization procedure will correctly give the answer YES if and only if the input graph has an appropriate partition $V_1, \ldots, V_q$ with nonempty classes. An affirmative answer implies that $GN(G) \geq q$. Using the bound $GN(G) \leq 1 + k \log_2 n$, we run the $\exists D_q$ algorithm for descending values of q starting with $q = 1 + k \log_2 n$ and halting as soon as an affirmative answer is given. The complexity of this algorithm is then given by appropriately applying Theorem 5.7, with $|A_\sigma| = 1$ and $|A_\rho| = 2$. □

Consider any maximum iterated $[\rho, \sigma]$ removal problem with $\rho = \mathbb{P}$, asking how many times we can remove a σ-constrained dominating set from a graph (compare with a Grundy number which removes independent dominating sets). This translates to a partition maximization problem where the degree constraint matrix has diagonal entries σ, above-diagonal entries $\mathbb{N}$, and below-diagonal entries $\mathbb{P}$. Note that the proof of the logarithmic bound on the Grundy number in Theorem 6.1 does not use the fact that the classes V_i of the partition are independent sets, only the fact that they are dominating sets in the remaining graph. Thus we get a logarithmic upper bound also on these generalized maximum dominating iterated removal parameters on partial k-trees and a polynomial-time algorithm for computing these parameters for fixed k.

7. Conclusions.

In this paper, we have presented a design methodology for practical solution algorithms on partial k-trees and a characterization of a class of vertex partitioning problems. These results were combined by adapting the algorithm design methodology on partial k-trees to vertex partitioning problems, yielding the first algorithms for these problems with reasonable time complexity as a function of treewidth.

Implementation of the resulting algorithms is a project at the University of Bergen [17]. The program for solving the Independent Set problem: maximize $|V_1|$ over partitions (V_1, V_2) satisfying

$$D_2 = \begin{pmatrix} \{0\} & \mathbb{N} \\ \mathbb{N} & \mathbb{N} \end{pmatrix}$$

is about 1000 lines of C++ code. Less than 100 of these lines are problem-specific, i.e., to produce a solution algorithm for any other vertex subset problem requires changing only a handful of functions.

The actual running time behaves as predicted by the bounds given in this paper, e.g., to solve the independent set problem on an n-node partial k-tree (using a 150 Mhz alpha processor-based digital computer) it takes roughly $10^{-5} \cdot n \cdot 2^k$ seconds. For example, on a 3000-node graph with treewidth 5, we solve the maximum independent set problem in about 1 second.

Various improvements can be made to these algorithms to reduce the average, if not worst-case, running time. For example, one can use parse trees with smaller bags in "thin" parts of the graph or computing table entries can be based only on nonzero table entries in the children.

A recent result [29] shows that control-flow graphs of structured (goto-free) programs have small treewidth, e.g., treewidth at most 3 for Pascal programs and treewidth at most 6 for C programs. Moreover, a tree-decomposition of the control-flow graph can be easily computed from the program structure (in fact from the 3-address code), making our algorithms, which require a k-tree embedding (tree-decomposition), relatively easily applicable in various compiler optimization settings.

REFERENCES

[1] S. Arnborg, S. T. Hedetniemi, and A. Proskurowski, eds., *Special issue on efficient algorithms and partial k-trees*, Discrete Appl. Math., 54 (2-3), 1994, pp. 97–291.

[2] S. Arnborg, J. Lagergren, and D. Seese, *Easy problems for tree-decomposable graphs*, J. Algorithms, 12 (1991), pp. 308–340.

[3] S. Arnborg and A. Proskurowski, *Linear time algorithms for NP-hard problems on graphs embedded in k-trees*, Discrete Appl. Math., 23 (1989), pp. 11–24.

[4] M. W. Bern, E. L. Lawler, and A. L. Wong, *Linear-time computation of optimal subgraphs of decomposable graphs*, J. Algorithms, 8 (1987), pp. 216–235.

[5] H. L. Bodlaender, *Dynamic programming on graphs with bounded treewidth*, in Proceedings ICALP 88, Lecture Notes in Comput. Sci. 317, Springer-Verlag, New York, 1988, pp. 105–119.

[6] H. L. Bodlaender, *A linear time algorithm for finding tree-decompositions of small treewidth*, in Proceedings STOC'93, 25th Annual ACM Symposium on Theory of Computing, ACM, New York, 1993, pp. 226–234; SIAM J. Comput., 25 (1996), pp. 1305–1317.

[7] R. B. Borie, R. G. Parker, and C. A. Tovey, *Automatic generation of linear algorithms from predicate calculus descriptions of problems on recursive constructed graph families*, Algorithmica, 7 (1992), pp. 555–582.

[8] E. J. Cockayne, B. L. Hartnell, S. T. Hedetniemi, and R. Laskar, *Perfect domination in graphs*, J. Combin. Inform. System Sci., 18 (1993), pp. 136–146.

[9] D. G. Corneil and J. Keil, *A dynamic programming approach to the dominating set problem on k-trees*, SIAM J. Alg. Discrete Meth., 8 (1987), pp. 535–543.

[10] B. Courcelle, *The monadic second-order logic of graphs* I*: Recognizable sets of finite graphs*, Inform. and Comput., 85 (1990), pp. 12–75.

[11] B. Courcelle and M. Mosbah, *Monadic second-order evaluations on tree-decomposable graphs*, Theoret. Comput. Sci., 109 (1993), pp. 49–82.

[12] M. R. Garey and D. S. Johnson, *Computers and Intractability*, W. H. Freeman and Co., San Francisco, CA, 1978.

[13] S. M. Hedetniemi, S. T. Hedetniemi, and T. Beyer, *A linear algorithm for the Grundy* (*coloring*) *number of a tree*, Congr. Numer., 36 (1982), pp. 351–362.

[14] S. T. Hedetniemi and R. Laskar, *Bibliography on domination in graphs and some basic definitions of domination parameters*, Discrete Math., 86 (1990), pp. 257–277.

[15] P. Heggernes and J. A. Telle, *Partitioning graphs into generalized dominating sets*, in Proceedings of XV International Conference of the Chilean Computer Society, Arica, Chile, 1995, pp. 241–252.

[16] P. Hell and J. Nešetřil, *On the complexity of H-colouring*, J. Combin. Theory B, 48 (1990), pp. 92–110.

[17] B. Hiim, *Implementing and Testing of Algorithms for Tree-like Graphs*, forthcoming Technical Report, Dept. of Informatics, University of Bergen, Bergen, Norway.

[18] T. Kloks, *Treewidth*, Lecture Notes in Comput. Sci. 842, Springer-Verlag, New York, 1994.

[19] J. Kratochvíl, A. Proskurowski, and J. A. Telle, *On the complexity of graph covering problems*, in Proceedings of 20th International Workshop on Graph-Theoretic Concepts in Computer Science 1994 — WG '94, Lecture Notes in Comput. Sci. 903, Springer-Verlag, New York, 1995, pp. 93–105.

[20] J. Lagergren, *private communication*, 1994.

[21] J. van Leeuwen, *Graph algorithms*, in Handbook of Theoretical Computer Science, Vol. A, Elsevier, Amsterdam, 1990, p. 550.

[22] A. McRae, *Generalizing NP-Completeness Proofs for Bipartite and Chordal Graphs*, Ph.D. thesis, Clemson University, Clemson, SC, 1994.

[23] A. Proskurowski and M. Sysło, *Efficient computations in tree-like graphs*, Comput. Suppl., 7 (1990), pp. 1–15.

[24] N. Robertson and P. D. Seymour, *Graph minors* II*: Algorithmic aspects of treewidth*, J. Algorithms, 7 (1986), pp. 309–322.

[25] K. Takamizawa, T. Nishizeki, and N. Saito, *Linear-time computability of combinatorial problems on series-parallel graphs*, J. ACM, 29 (1982), pp. 623–641.

[26] J. A. Telle, *Complexity of domination-type problems in graphs*, Nordic J. Comput., 1 (1994), pp. 157–171.

[27] J. A. Telle, *Vertex Partitioning Problems: Characterization, Complexity and Algorithms on Partial k-Trees*, Ph.D. thesis, University of Oregon, Eugene, OR, 1994.

[28] J. A. Telle and A. Proskurowski, *Practical algorithms on partial k-trees with an applicationto domination-type problems*, in Proceedings Workshop on Algorithms and Data Structures, Montreal, 1993, Lecture Notes in Comput. Sci. 709, Springer-Verlag, New York, 1993, pp. 610–621.

[29] M. Thorup, *Structured Programs Have Small Treewidth and Good Register Allocation*, Technical Report DIKU-TR-95/18, Department of Computer Science, University of Copenhagen, Denmark, 1995; to appear in Proceedings 23rd Intl. Workshop on Graph-Theoretic Concepts in Computer Science, Lecture Notes in Comput. Sci. 1198, Springer-Verlag, New York, 1997.

[30] T. Wimer, *Linear Time Algorithms on k-terminal Graphs*, Ph.D. thesis, Clemson University, Clemson, SC, 1988.

SIAM J. Discrete Math.
Vol. 10, No. 4, pp. 551–572, November 1997

GLOBAL PRICE UPDATES HELP*

ANDREW V. GOLDBERG† AND ROBERT KENNEDY‡

Abstract. Periodic global updates of dual variables have been shown to yield a substantial speed advantage in implementations of push-relabel algorithms for the maximum flow and minimum cost flow problems. In this paper, we show that in the context of the bipartite matching and assignment problems, global updates yield a theoretical improvement as well. For bipartite matching, a push-relabel algorithm that uses global updates runs in $O\Big(\sqrt{n}m\frac{\log(n^2/m)}{\log n}\Big)$ time (matching the best bound known) and performs worse by a factor of $\sqrt{n}$ without the updates. A similar result holds for the assignment problem, for which an algorithm that assumes integer costs in the range $[\,-C,\ldots,C\,]$ and that runs in time $O(\sqrt{n}m\log(nC))$ (matching the best cost-scaling bound known) is presented.

Key words. assignment problem, cost scaling, bipartite matching, dual update, push-relabel algorithm, zero-one flow

AMS subject classifications. 90C08, 68Q20, 68R10

PII. S0895480194281185

1. Introduction.

The push-relabel method [10, 13] is the best currently known method for solving the maximum flow problem [1, 2, 19]. This method extends to the minimum cost flow problem using cost-scaling [10, 14], and an implementation of this technique has proven very competitive on a wide class of problems [11]. In both contexts, the idea of periodic global updates of node distances or prices has been critical in obtaining the best running times in practice.

Several algorithms for the bipartite matching problem run in $O(\sqrt{n}m)$ time.[1] The first algorithm proved to achieve this bound was proposed by Hopcroft and Karp [15]. Karzanov [17, 16] and Even and Tarjan [5] proved that the blocking flow algorithm of Dinic [4] runs in this time when applied to the bipartite matching problem. Two-phase algorithms based on a combination of the push-relabel method [13] and the augmenting path method [7] were proposed in [12, 20].

Feder and Motwani [6] give a "graph compression" technique that combines with the algorithm of Dinic to yield an $O\left(\sqrt{n}m\frac{\log(n^2/m)}{\log n}\right)$ algorithm. This is the best time bound known for the problem.

The most relevant theoretical results on the assignment problem are as follows. The best currently known strongly polynomial time bound of $O\big(n(m+n\log n)\big)$ is achieved by the classical Hungarian method of Kuhn [18]. Under the assumption that the input costs are integers in the range $[\,-C,\ldots,C\,]$, Gabow and Tarjan [9] use cost-scaling and blocking flow techniques to obtain an $O\big(\sqrt{n}m\log(nC)\big)$ time algorithm. An algorithm using an idea similar to global updates with the same running time appeared in [8]. Two-phase algorithms with the same running time appeared in [12, 20]. The first phase of these algorithms is based on the push-relabel method and the second phase is based on the successive augmentation approach. Our algorithm

*Received by the editors December 2, 1994; accepted for publication (in revised form) September 11, 1996. This work was supported in part by ONR young investigator award N00014-91-J-1855 and NSF grant CCR-9307045.

http://www.siam.org/journals/sidma/10-4/28118.html

†NEC Research Institute, Inc., Princeton, NJ 08540 (avg@research.nj.nec.com).

‡Silicon Graphics, Inc., 2011 North Shoreline Blvd., MS 178, Mountain View, CA 94043 (robert@cs.stanford.edu).

[1] Here n and m denote the number of nodes and edges, respectively.

for the assignment problem runs in $O\big(\sqrt{n}m\log(nC)\big)$, and like the other algorithms with this time bound, it is based on cost-scaling, assumes that the input costs are integers, and is not strongly polynomial.

We show that algorithms based on the push-relabel method with global updates match the best bounds for the bipartite matching and assignment problems. Our results are based on the following new selection strategies: the *minimum distance* strategy in the bipartite matching case and *minimum price change* strategy in the assignment problem case. We also prove that the algorithms perform significantly worse without global updates. Similar results can be obtained for maximum and minimum cost flows in networks with unit capacities. Our results are a step toward a theoretical justification of the use of global update heuristics in practice.

This paper is organized as follows. Section 2 gives definitions relevant to bipartite matching and maximum flow. Section 3 outlines the push-relabel method for maximum flow and shows its application to bipartite matching. In section 4, we present an $O(\sqrt{n}m)$ time bound for the bipartite matching algorithm with global updates, and in Section 5 we show how to apply Feder and Motwani's techniques to improve the algorithm's performance to $O\left(\sqrt{n}m\frac{\log(n^2/m)}{\log n}\right)$. Section 6 shows that without global updates, the bipartite matching algorithm performs poorly. Section 7 gives definitions relevant to the assignment problem and minimum cost flow. In section 8, we describe the cost-scaling push-relabel method for minimum cost flow and apply the method to the assignment problem. Sections 9 and 10 generalize the bipartite matching results to the assignment problem. In section 11, we give our conclusions and suggest directions for further research.

2. Bipartite matching and maximum flow. Let $\overline{G} = (\overline{V} = X \cup Y, \overline{E})$ be an undirected bipartite graph, let $n = |\overline{V}| + 2$ (the additive constant being, for notational convenience, in the reduction to come), and let $m = |\overline{E}|$. A *matching* in $\overline{G}$ is a subset of edges $M \subseteq \overline{E}$ that have no node in common. The *cardinality* of the matching is $|M|$. The *bipartite matching problem* is to find a maximum cardinality matching.

The conventions we assume for the maximum flow problem are as follows: Let $G = (\{s,t\} \cup V, E)$ be a digraph with an integer-valued *capacity* $u(a)$ associated with each arc[2] $a \in E$. We assume that $a \in E \Rightarrow a^R \in E$ (where a^R denotes the reverse of arc a). A *pseudoflow* is a function $f : E \to \mathbf{R}$ satisfying the following for each $a \in E$:

- $f(a) = -f(a^R)$ (*flow antisymmetry* constraints);
- $f(a) \leq u(a)$ (*capacity* constraints).

The antisymmetry constraints are for notational convenience only, and we will often take advantage of this fact by mentioning only those arcs with nonnegative flow; in every case, the antisymmetry constraints are satisfied simply by setting the reverse arc's flow to the appropriate value. For a pseudoflow f and a node v, the *excess flow into* v, denoted $e_f(v)$, is defined by $e_f(v) = \sum_{(u,v)\in E} f(u,v)$. A *preflow* is a pseudoflow with the property that the excess of every node except s is nonnegative. A node $v \neq t$ with $e_f(v) > 0$ is called *active*.

A *flow* is a pseudoflow f such that, for each node $v \in V$, $e_f(v) = 0$. Observe that a preflow f is a flow if and only if there are no active nodes. The *maximum flow problem* is to find a flow maximizing $e_f(t)$.

[2] Sometimes we refer to an arc a by its endpoints, e.g., (v,w). This is ambiguous if there are multiple arcs from v to w. An alternative is to refer to v as the tail of a and to w as the head of a, which is precise but inconvenient.

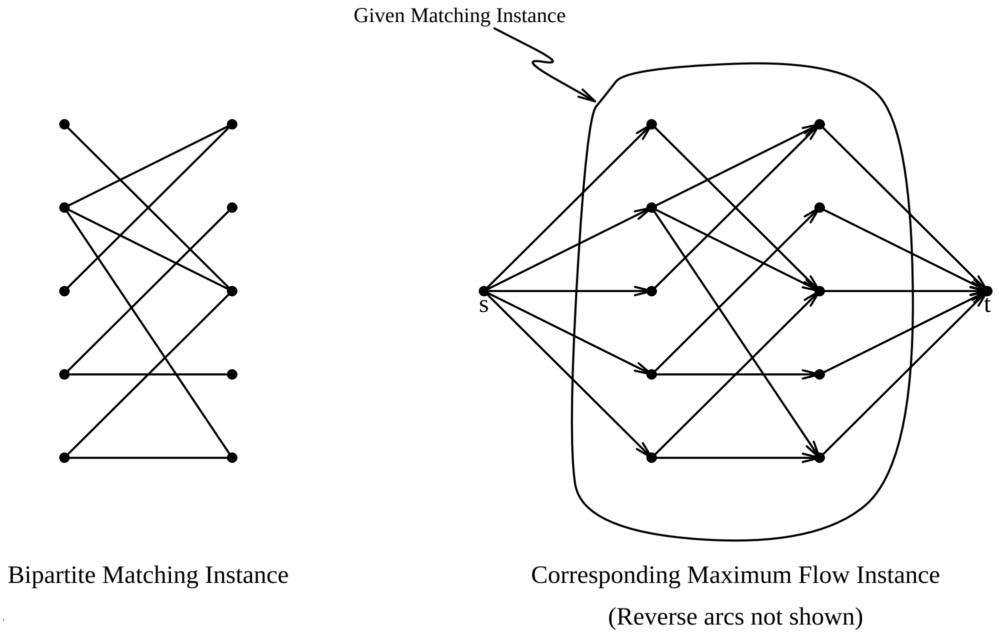

Given Matching Instance
s
t
Bipartite Matching Instance
Corresponding Maximum Flow Instance
(Reverse arcs not shown)

FIG. 3.1. *Reduction from bipartite matching to maximum flow.*

3. The push-relabel method for bipartite matching. We reduce the bipartite matching problem to the maximum flow problem in a standard way. For brevity, we mention only the "forward" arcs in the flow network; to each such arc we give unit capacity. The "reverse" arcs have capacity zero. Given an instance $\overline{G} = \big(\overline{V} = X \cup Y, \overline{E}\big)$ of the bipartite matching problem, we construct an instance $\big(G = (\{s,t\} \cup V, E), u\big)$ of the maximum flow problem by

- setting $V = \overline{V}$;
- for each node $v \in X$, placing arc (s, v) in E;
- for each node $v \in Y$, placing arc (v, t) in E;
- for each edge $\{v, w\} \in \overline{E}$ with $v \in X$ and $w \in Y$, placing arc (v, w) in E.

A graph obtained by this reduction is called a *matching network* (see Figure 3.1). Note that if G is a matching network, then for any integral pseudoflow f and for any arc $a \in E$, $u(a), f(a) \in \{0, 1\}$. Indeed, any integral flow in G can be interpreted conveniently as a matching in $\overline{G}$; the matching is exactly the edges corresponding to those arcs $a \in X \times Y$ with $f(a) = 1$. It is a well-known fact [7] that a maximum flow in G corresponds to a maximum matching in $\overline{G}$.

For a given pseudoflow f, the *residual capacity* of an arc $a \in E$ is $u_f(a) = u(a) - f(a)$. The set of *residual arcs* E_f contains the arcs $a \in E$ with $f(a) < u(a)$. The *residual graph* $G_f = (V, E_f)$ is the graph induced by the residual arcs. The *augmented residual graph* $G_{\overline{f}}^{=}$ has the same nodes and arcs as G but is associated with the capacity function u_f. The point of defining $G_{\overline{f}}^{=}$ is to meaningfully discuss pseudoflows that obey the residual capacity constraints. Since the residual graph lacks arcs a with $u_f(a) = 0$, it can lack reverse arcs that are assumed by the definition of a pseudoflow.

A *distance labeling* is a function $d : V \to \mathbf{Z}^+$. We say a distance labeling d is *valid* with respect to a pseudoflow f if $d(t) = 0$, $d(s) = n$ and, for every arc $(v, w) \in E_f$,

```
PUSH$(v, w)$.
    send a unit of flow from $v$ to $w$.
end.


RELABEL$(v)$.
    replace $d(v)$ by $\min_{(v,w)\in E_f}\{d(w)+1\}$
end.
```

FIG. 3.2. *The* push *and* relabel *operations.*

$d(v) \leq d(w)+1$. Those residual arcs (v, w) with the property that $d(v) = d(w)+1$ are called *admissible* arcs, and the *admissible graph* $G_A = (V, E_A)$ is the graph induced by the admissible arcs. It is straightforward to see that G_A is acyclic for any valid distance labeling.

We begin with a high-level description of the generic push-relabel algorithm for maximum flow specialized to the case of matching networks. The algorithm starts with the zero flow, then sets $f(s, v) = 1$ for every $v \in X$. For an initial distance labeling, the algorithm sets $d(s) = n$ and $d(t) = 0$ and, for every $v \in V$, sets $d(v) = 0$. Then the algorithm applies *push* and *relabel* operations in any order until the current pseudoflow is a flow. The *push* and *relabel* operations, described below, preserve the properties that the current pseudoflow f is a preflow and that the current distance labeling d is valid with respect to f.

The *push* operation applies to an admissible arc (v, w) whose tail node v is active. It consists of "pushing" a unit of flow along the arc, i.e., increasing $f(v, w)$ by one, increasing $e_f(w)$ by one, and decreasing $e_f(v)$ by one. The *relabel* operation applies to an active node v that is not the tail of any admissible arc. It consists of changing v's distance label so that v is the tail of at least one admissible arc, i.e., setting $d(v)$ to the largest value that preserves the validity of the distance labeling. See Figure 3.2.

Our analysis of the push-relabel method is based on the following facts. (See [13] for details; note that arcs in a matching network have unit capacities and thus PUSH(v, w) saturates the arc (v, w)).

- For all nodes v, we have $0 \leq d(v) \leq 2n$.
- Distance labels do not decrease during the computation.
- *relabel*(v) increases $d(v)$.
- The number of *relabel* operations during the computation is $O(n)$ per node.
- The work involved in *relabel* operations is $O(nm)$.
- If a node v is relabeled t times during a computation segment, then the number of pushes from v is at most $(t + 1) \times$ *degree*(v).
- The number of *push* operations during the computation is $O(nm)$.

The above facts imply that any push-relabel algorithm runs in $O(nm)$ time given that the work involved in selecting the next operation to apply does not exceed the work involved in applying these operations. This can be easily achieved using the following simple data structure (see [13] for details). We maintain a *current arc* for every node. Initially, the first arc in the node's arc list is current. When pushing flow excess out of a node v, we push it on v's current arc if the arc is admissible, or advance the current arc to the next arc on the arc list. When there are no more arcs on the list, we relabel v and set v's current arc to the first arc on v's arc list.

4. Global updates and the minimum distance discharge algorithm. In this section, we specify an ordering of the *push* and *relabel* operations that yields certain desirable properties. We also introduce global distance updates and show that the algorithm resulting from our operation ordering and global update strategy runs in $O(\sqrt{n}m)$ time.

For any nodes v, w, let $d_w(v)$ denote the breadth-first-search distance from v to w in the (directed) residual graph of the current preflow. If w is unreachable from v in the residual graph, $d_w(v)$ is infinite. Setting $d(v) = \min\{d_t(v), n + d_s(v)\}$ for every node $v \in V$ is called a *global update operation.* This operation also sets the current arc of every node to the node's first arc. Such an operation can be accomplished with $O(m)$ work that amounts to two breadth-first-search computations. Validity of the resulting distance labeling is a straightforward consequence of the definition. Note that a global update cannot decrease any node's distance label [13].

The ordering of operations we use is called *minimum distance discharge.* It consists of repeatedly choosing an active node whose distance label is minimum among all active nodes and, if there is an admissible arc leaving that node, pushing a unit of flow along the admissible arc; otherwise we relabel the node. For the sake of efficient implementation and easy generalization to the weighted case, we formulate this selection strategy in a slightly different (but equivalent) way and use this formulation to guide the implementation and analysis. The intuition is that we select a unit of excess at an active node with a minimum distance label and process that unit of excess until a relabeling occurs or the excess reaches s or t. In the event of a relabeling, the new distance label may be small enough to guarantee that the same excess still has the minimum label; if so, we avoid the work associated with finding the next excess to process. This scheme's important properties generalize to the weighted case, and it allows us to show easily that the work done in active node selection is not too great.

To implement this selection rule, we maintain a collection of buckets, $b_0, \ldots, b_{2n}$; each b_i contains the active nodes with distance label i, except possibly one which is currently being processed. During execution, we maintain μ, which is the index of the bucket from which we selected the most recent unit of excess. If the new distance label is no more than μ when we relabel a node, we know that node still has a minimum distance label among the active nodes, so we continue processing the same unit of excess.

In addition, we perform periodic global updates. The first global update is performed immediately after the preflow is initialized. After each *push* and *relabel* operation, the algorithm checks the following two conditions and performs a global update if both conditions hold:

- Since the most recent update, at least one unit of excess has reached s or t.
- Since the most recent update, the algorithm has done at least m work in *push* and *relabel* operations.

Immediately after each global update, we rebuild the buckets in $O(n)$ time and set μ to zero. The following lemma shows that the algorithm does little extra work in selecting nodes to process.

LEMMA 4.1. *Between two consecutive global updates, the algorithm does $O(n)$ work in examining empty buckets.*

Proof. The proof is immediate, because μ decreases only when it is set to zero after an update, and there are $2n + 1 = O(n)$ buckets. □

We will denote by $\Gamma(f, d)$ (or simply Γ) the minimum distance label of an active node with respect to the pseudoflow f and the distance labeling d. We let $\Gamma_{\max}$ denote

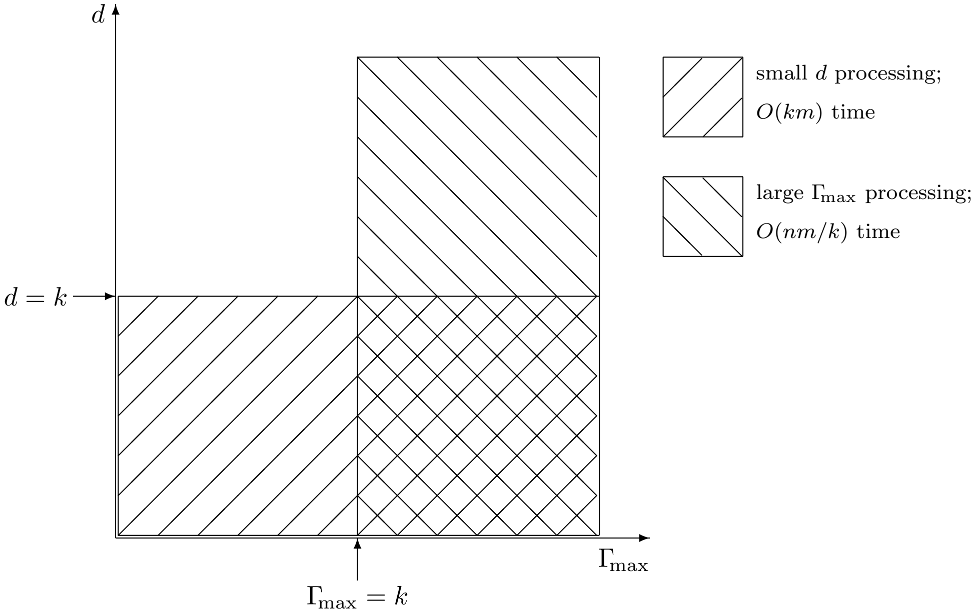


FIG. 4.1. *Accounting for work when* $0 \leq \Gamma_{\max} \leq n$.

the maximum value reached by Γ during the algorithm so far. Note that $\Gamma_{\max}$ is often equal to μ; we use separate names mainly to emphasize that μ is maintained by the implementation, while $\Gamma_{\max}$ is an abstract quantity with relevance to the analysis regardless of the implementation details.

Figure 4.1 represents the structure underlying our analysis of the minimum distance discharge algorithm. (Strictly speaking, the figure shows only half of the analysis; the other half, when $\Gamma_{\max} > n$, is essentially similar.) The horizontal axis corresponds to the value of $\Gamma_{\max}$, which increases as the algorithm proceeds, and the vertical axis corresponds to the distance label of the node currently being processed. Our analysis hinges on a parameter k, in the range $2 \leq k \leq n$, to be chosen later. We divide the execution of the algorithm into four stages. In the first two stages, excesses are moved to t; in the final two stages, excesses that cannot reach t return to s. We analyze the first stage of each pair using the following lemma.

LEMMA 4.2. *The minimum distance discharge algorithm expends* $O(km)$ *work during the periods when* $\Gamma_{\max} \in [0, k]$ *and* $\Gamma_{\max} \in [n, n + k]$.

Proof. First, note that if $\Gamma_{\max}$ falls in the first interval of interest, Γ must lie in that interval as well. This relationship also holds for the second interval after a global update is performed, since $\Gamma_{\max} \geq n$ implies that no excess can reach t. Because the work from the beginning of the second interval until the price update is performed is $O(m)$, it is enough to show that the time spent by the algorithm during periods when $\Gamma \in [0, k]$ and $\Gamma \in [n, n + k]$ is in $O(km)$. Note that the periods defined in terms of Γ may not represent contiguous intervals during the execution of the algorithm.

Each node can be relabeled at most $k + 1$ times when $\Gamma \in [0, k]$ and similarly for $\Gamma \in [n, n + k]$. Hence the relabelings and pushes require $O(km)$ work. The observations that a global update requires $O(m)$ work and that during each period there are $O(k)$ global updates complete the proof. □

To study the behavior of the algorithm during the remainder of its execution, we exploit the structure of matching networks by appealing to a combinatorial lemma. The following lemma is a special case of a well-known decomposition theorem [7] (also see [5]). The proof depends mainly on the fact that for a matching network G, the in-degree of $v \in X$ in G_f is $1 - e_f(v)$ and the out-degree of $w \in Y$ in G_f is $1 + e_f(w)$ for any integral pseudoflow f.

LEMMA 4.3. *Any integral pseudoflow f in the augmented residual graph of an integral flow g in a matching network can be decomposed into cycles and simple paths that are pairwise node-disjoint, except at the endpoints of the paths, such that each element in the decomposition carries one unit of flow. Each path is from a node v with $e_f(v) < 0$ (v can be t) to a node w with $e_f(w) > 0$ (w can be s).*

Lemma 4.3 allows us to show that when $\Gamma_{\max}$ is outside the intervals covered by Lemma 4.2, the amount of excess the algorithm must process is small.

Given a preflow f, we define the *residual flow value* to be the total excess that can reach t in G_f.

LEMMA 4.4. *If $\Gamma_{\max} \geq k > 2$, the residual flow value is at most $n/(k-1)$ if G is a matching network.*

Proof. Note that the residual flow value never increases during an execution of the algorithm, and consider the pair (f, d) such that $\Gamma(f, d) \geq k$ for the first time during the execution. Let f^* be a maximum flow in G, and let $f' = f^* - f$. Now $-f'$ is a pseudoflow in $G^{=}_{\overline{f^*}}$ and can therefore be decomposed into cycles and paths as in Lemma 4.3. Such a decomposition of $-f'$ induces the obvious decomposition on f' with all the paths and cycles reversed and excesses negated. Because $\Gamma \geq k$ and d is a valid distance labeling with respect to f, any path in G_f from an active node to t must contain at least $k+1$ nodes. In particular, the excess-to-t paths in the decomposition of f' contain at least $k+1$ nodes each and are node-disjoint except for their endpoints. Since G contains only n nodes, there can be no more than $(n-2)/(k-1) < n/(k-1)$ such paths. Since f^* is a maximum flow, the amount of excess that can reach t in G_f is no more than $n/(k-1)$. □

The proof of the next lemma is similar.

LEMMA 4.5. *If $\Gamma_{\max} \geq n + k > n + 2$ during an execution of the minimum distance discharge algorithm with global updates on a matching network, the total excess at nodes in V is at most $n/(k-1)$.*

The following lemma shows an important property of the rules we use to trigger global update operations, namely, that during a period when the algorithm does $\Theta(m)$ work at least one unit of excess is guaranteed to reach s or t.

LEMMA 4.6. *Between any two consecutive global update operations, the algorithm does $\Theta(m)$ work.*

Proof. According to the two conditions that trigger a global update, it suffices to show that immediately after an update, the work done in moving a unit of excess to s or t is $O(m)$. For every node v, at least one of $d_s(v)$, $d_t(v)$ is finite. Therefore, immediately after a global update at least one admissible arc leaves every node except s and t, by definition of the global update operation. Recall that the admissible graph is acyclic, so the first unit of excess processed by the algorithm immediately after a global update arrives at t or at s before any relabeling occurs, and does so along a simple path. Consider the path taken by the flow unit to s or t. The work performed while moving the unit along the path is proportional to the length of the path plus the number of times current arcs of nodes on the path are advanced. This $O(n + m) = O(m)$ work is performed before the first condition for a global update is met.

Following an amount of additional work bounded above by $m + O(n)$, plus work proportional to that for a *push* or *relabel* operation, another global update operation will be triggered. Clearly a *push* or *relabel* takes $O(m)$ work and the lemma follows. □

We are ready to prove the main result of this section.

THEOREM 4.7. *The minimum distance discharge algorithm with global updates computes a maximum flow in a matching network (and hence a maximum cardinality bipartite matching) in* $O(\sqrt{n}m)$ *time.*

Proof. By Lemma 4.2, the total work done by the algorithm when $\Gamma_{\max} \in [0, k]$ and $\Gamma_{\max} \in [n, n+k]$ is $O(km)$. By Lemmas 4.4 and 4.5, the amount of excess processed when $\Gamma_{\max}$ falls outside these bounds is at most $2n/(k-1)$. From Lemma 4.6 we conclude that the work done in processing this excess is $O(nm/k)$. Hence the time bound for the minimum distance discharge algorithm is $O\big(km + nm/k\big)$. Choosing $k = \Theta(\sqrt{n}\,)$ to balance the two terms, we see that the minimum distance discharge algorithm with global updates runs in $O(\sqrt{n}m)$ time. □

5. Improved performance through graph compression. Feder and Motwani [6] give an algorithm that runs in $o(\sqrt{n}m)$ time and produces a *compressed representation* $\overline{G}^* = (\overline{V} \cup W, \overline{E}^*)$ of a bipartite graph in which all adjacency information is preserved, but that has asymptotically fewer edges if the original graph $\overline{G} = (\overline{V}, \overline{E})$ is dense. This graph consists of all the original nodes of X and Y, as well as a set of additional nodes W. An edge $\{x, y\}$ appears in $\overline{E}$ if and only if either $\{x, y\} \in \overline{E}^*$ or $\overline{G}^*$ contains a length-two path from x to y through some node of W.

The following theorem is slightly specialized from Feder and Motwani's Theorem 3.1 [6], which details the performance of their algorithm *Compress*.

THEOREM 5.1. *Let* $\delta \in (0, 1)$ *and let* $\overline{G} = (\overline{V} = X \cup Y, \overline{E})$ *be an undirected bipartite graph with* $|X| = |Y| = n$ *and* $|\overline{E}| = m \geq n^{2-\delta}$*. Then algorithm* Compress *computes a compressed representation* $\overline{G}^* = (\overline{V} \cup W, \overline{E}^*)$ *of* $\overline{G}$ *with* $m^* = |\overline{E}^*| = O\left(m\delta^{-1}\frac{\log(n^2/m)}{\log n}\right)$ *in time* $O(mn^\delta \log^2 n)$*. The number of nodes in* W *is* $O(mn^{\delta-1})$*.*

In particular, we choose a constant $\delta < 1/2$; then the compressed representation is computed in time $o(\sqrt{n}m)$ and has $m^* = O\left(m\frac{\log(n^2/m)}{\log n}\right)$ edges.

Given a compressed representation $\overline{G}^*$ of $\overline{G}$, we can compute a flow network G^* in which there is a correspondence between flows in G^* and matchings in $\overline{G}$. The only differences from the reduction of section 3 are that each edge $\{x, w\}$ with $x \in X$ and $w \in W$ gives an arc (x, w), and each edge $\{w, y\}$ with $w \in W$ and $y \in Y$ gives an arc (w, y). As in section 3, we have a relationship between matchings in the original graph $\overline{G}$ and flows in G^*, but now the correspondence is not one-to-one as it was before. Nevertheless, it remains true here that given a flow f with $e_f(t) = c$ in G^*, we can find a matching of cardinality c in $\overline{G}$ using only $O(n)$ time in a straightforward way.

The performance improvement that we gain comes from using the graph compression step as preprocessing; we will show that the minimum distance discharge algorithm with global updates runs in time $O(\sqrt{n}m^*)$ on the flow network G^* corresponding to the compressed representation $\overline{G}^*$ of a bipartite graph $\overline{G}$. In other words, the speedup results only from the reduced number of edges, not from changes within the minimum distance discharge algorithm.

To prove the performance bound, we must generalize certain lemmas from section 4 to networks with the structure of compressed representations. Let $n^* = n + |W|$ be the number of nodes in the maximum flow problem derived from the compressed representation of the input graph. Lemma 4.2 is independent of the input network's structure, as are Lemma 4.6 and Lemma 4.1. These three lemmas give us their conclusions for compressed representations where we substitute n^* for n and m^* for m in their statements and proofs. An analogue to Lemma 4.3 holds in a flow network derived from a compressed representation; this will extend Lemmas 4.4 and 4.5, allowing us to conclude the improved time bound.

Lemma 5.2. *Any integral pseudoflow f in the augmented residual graph of an integral flow g in the flow graph of a compressed representation can be decomposed into cycles and simple paths that are pairwise node-disjoint at nodes of X and Y, except at the endpoints of the paths, such that each element of the decomposition carries one unit of flow. Each path is from a node v with $e_f(v) < 0$ (v can be t) to a node w with $e_f(w) > 0$ (w can be s).*

Proof. As with matching networks, the in-degree of $v \in X$ is $1 - e_f(v)$ and the out-degree of $y \in Y$ is $1 + e_f(y)$, so the standard proof of Lemma 4.3 extends to this case. □

The following lemma is analogous to Lemma 4.4.

Lemma 5.3. *If $\Gamma_{\max} \geq k > 2$, the residual flow value is at most $2n/(k-2)$ if G^* is a compressed representation.*

Proof. The proof follows as in the case of Lemma 4.4, except that here an excess-to-t path in the decomposition of f' must contain at least $k/2$ nodes of $\overline{V}$. Since $\overline{V}$ contains only n nodes, there can be no more than $2n/(k-2)$ such paths, and so because f^* is a maximum flow, the amount of excess that can reach t in G^*_f is no more than $2n/(k-2)$. □

The following lemma is analogous to Lemma 4.5, and its proof is similar to the proof of Lemma 5.3.

Lemma 5.4. *If $\Gamma_{\max} \geq n^* + k > n^* + 2$ during an execution of the minimum distance discharge algorithm with global updates on a compressed representation, the total excess at nodes in $\overline{V} \cup W$ is at most $2n/(k-2)$.*

Using the same reasoning as in Theorem 4.7, we have the following theorem.

Theorem 5.5. *The minimum distance discharge algorithm with global updates computes a maximum flow in the network corresponding to a compressed representation with m^* edges in $O(\sqrt{n}m^*)$ time.*

To complete our time bound for the bipartite matching problem we must dispense with some technical restrictions in Theorem 5.1, namely, the requirements that $|X| = |Y| = n$ and that $m \geq n^{2-\delta}$. The former condition is easily met by adding nodes to whichever of X, Y is the smaller set, so their cardinalities are equal. These "dummy" nodes are incident to no edges. As for the remaining condition, observe that our time bound does not suffer if we simply forego the compression step and apply the result of section 4 in the case where $m < n^{2-\delta}$. To see this, recall that we chose $\delta < 1/2$, and note that $1 \leq m < n^{2-\delta}$ implies $\frac{\log(n^2/m)}{\log n} = \Theta(1)$. So we have the following theorem.

Theorem 5.6. *The minimum distance discharge algorithm with graph compression and global updates computes a maximum cardinality bipartite matching in $O\left(\sqrt{n}m\frac{\log(n^2/m)}{\log n}\right)$ time.*

This bound matches that of Feder and Motwani for Dinic's algorithm.

6. Minimum distance discharge algorithm without global updates.

In this section we describe a family of graphs on which the minimum distance discharge

```
1. Initialization establishes $|X|$ units of excess, one at each node of $X$;
2. Nodes of $X$ are relabeled one-by-one, so all $v \in X$ have $d(v) = 1$;
3. While $e_f(t) < |Y|$,
    3.1. a unit of excess moves from some node $v \in X$ to some node $w \in Y$ with
         $d(w) = 0$;
    3.2. $w$ is relabeled so that $d(w) = 1$;
    3.3. The unit of excess moves from $w$ to $t$, increasing $e_f(t)$ by one.
4. A single node, $x_1$ with $e_f(x_1) = 1$, is relabeled so that $d(x_1) = 2$.
5. $\ell \leftarrow 1$.
6. While $\ell \leq n$,
         Remark: All nodes $v \in V$ now have $d(v) = \ell$ with the exception of the one
         node $x_\ell \in X$, which has $d(x_\ell) = \ell + 1$ and $e_f(x_\ell) = 1$; all excesses are at nodes
         of $X$;
    6.1. All nodes with excess, except the single node $x_\ell$, are relabeled one-by-one so
         that all $v \in X$ with $e_f(v) = 1$ have $d(v) = \ell + 1$;
    6.2. While some node $y \in Y$ has $d(y) = \ell$,
         6.2.1. A unit of excess is pushed from a node in $X$ to $y$;
         6.2.2. $y$ is relabeled so $d(y) = \ell + 1$;
         6.2.3. The unit of excess at $y$ is pushed to a node $x \in X$ with $d(x) = \ell$;
         6.2.4. $x$ is relabeled so that if some node in $Y$ still has distance label $\ell$,
                 $d(x) = \ell + 1$;
                otherwise
                 $d(x) = \ell + 2$ and $x_{\ell+1} \leftarrow x$;
    6.3. $\ell \leftarrow \ell + 1$;
7. Excesses are pushed one-by-one from nodes in $X$ (labeled $n + 1$) to $s$.
```

FIG. 6.1. *The minimum distance discharge execution on bad examples.*

algorithm *without* global updates requires $\Omega(nm)$ time (for values of m between $\Theta(n)$ and $\Theta(n^2)$). This shows that the updates improve the worst-case running time of the algorithm. The goal of our construction is to admit an execution of the algorithm in which each relabeling changes a node's distance label by $O(1)$. Under this condition the execution will have to perform $\Omega(n^2)$ relabelings, and these relabelings will require $\Omega(nm)$ time.

Given $\tilde{n} \in \mathbf{Z}$ and $\tilde{m} \in [1, \tilde{n}^2/4]$, we construct a graph $\overline{G}$ as follows: $\overline{G}$ is the complete bipartite graph with $\overline{V} = X \cup Y$, where

$$X = \left\{ 1, 2, \ldots, \left\lceil \frac{\tilde{n} + \sqrt{\tilde{n}^2 - 4\tilde{m}}}{2} \right\rceil \right\} \quad \text{and} \quad Y = \left\{ 1, 2, \ldots, \left\lfloor \frac{\tilde{n} - \sqrt{\tilde{n}^2 - 4\tilde{m}}}{2} \right\rfloor \right\}.$$

It is straightforward to check that this graph has $n = \tilde{n} + O(1)$ nodes and $m = \tilde{m} + O(\tilde{n})$ edges. Note that $|X| > |Y|$.

Figure 6.1 describes an execution of the minimum distance discharge algorithm on G—the matching network derived from $\overline{G}$—that requires $\Omega(nm)$ time. With more complicated but unilluminating analysis, it is possible to show that every execution of the minimum distance discharge algorithm on G requires $\Omega(nm)$ time.

It is straightforward to verify that in the execution outlined, all processing takes place at active nodes whose distance labels are minimum among the active nodes. The algorithm performs poorly because during the execution, no relabeling changes a distance label by more than two. Hence the execution uses $\Theta(nm)$ work in the course of its $\Theta(n^2)$ relabelings, and we have the following theorem.

THEOREM 6.1. *For any function $m(n)$ in the range $n \leq m(n) < n^2/4$, there exists an infinite family of instances of the bipartite matching problem having $\Theta(n)$ nodes and $\Theta\big(m(n)\big)$ edges on which the minimum distance discharge algorithm without global updates runs in $\Omega\big(nm(n)\big)$ time.*

7. Minimum cost circulation and assignment problems.

Given a weight function $\bar{c} : \overline{E} \to \mathbf{R}$ and a set of edges M, we define the weight of M to be the sum of weights of edges in M. The *assignment problem* is to find a maximum cardinality matching of minimum weight. We assume that the costs are integers in the range $[\,0,\ldots,C\,]$ where $C \geq 1$. (Note that we can always make the costs nonnegative by adding an appropriate number to all arc costs.)

For the minimum cost circulation problem, we adopt the following framework. We are given a graph $G = (V, E)$, with an integer-valued capacity function as before. In addition to the capacity function, we are given an integer-valued *cost* $c(a)$ for each arc $a \in E$.

We assume $c(a) = -c(a^R)$ for every arc a. A *circulation* is a pseudoflow f with the property that $e_f(v) = 0$ for every node $v \in V$. (The absence of a distinguished source and sink accounts for the difference in nomenclature between a circulation and a flow.) We will say that a node v with $e_f(v) < 0$ has a *deficit*.

The cost of a pseudoflow f is given by $c(f) = \sum_{f(a)>0} c(a)f(a)$. The *minimum cost circulation problem* is to find a circulation of minimum cost.

8. The push-relabel method for the assignment problem.

We reduce the assignment problem to the minimum cost circulation problem as follows. As in the unweighted case, we mention only "forward" arcs, each of which we give unit capacity. The "reverse" arcs have zero capacity and obey cost antisymmetry. Given an instance $\big(\overline{G} = (\overline{V} = X \cup Y, \overline{E}), \bar{c}\big)$ of the assignment problem, we construct an instance $\big(G = (\{s,t\} \cup V, E), u, c\big)$ of the minimum cost circulation problem by

- creating special nodes s and t, and setting $V = \overline{V} \cup \{s,t\}$;
- for each node $v \in X$, placing arc (s,v) in E and defining $c(s,v) = -nC$;
- for each node $v \in Y$, placing arc (v,t) in E and defining $c(v,t) = 0$;
- for each edge $\{v,w\} \in \overline{E}$ with $v \in X$, placing arc (v,w) in E and defining $c(v,w) = \bar{c}(v,w)$;
- placing $n/2$ arcs (t,s) in E and defining $c(t,s) = 0$.

If G is obtained by this reduction (see Figure 8.1), we can interpret an integral circulation in G as a matching in $\overline{G}$ just as we did in the bipartite matching case. Furthermore, it is easy to verify that a minimum cost circulation in G corresponds to a maximum matching of minimum weight in $\overline{G}$.

A *price function* is a function $p : V \to \mathbf{R}$. For a given price function p, the *reduced cost* of an arc (v,w) is $c_p(v,w) = c(v,w) + p(v) - p(w)$.

Let $U = X \cup \{t\}$. Note that all arcs in E have one endpoint in U and one endpoint in its complement. Define E_U to be the set of arcs whose tail node is in U.

For a constant $\epsilon \geq 0$, a pseudoflow f is said to be *ϵ-optimal with respect to a price function p* if, for every residual arc $a \in E_f$, we have

$$\begin{cases} a \in E_U \Rightarrow c_p(a) \geq 0, \\ a \notin E_U \Rightarrow c_p(a) \geq -2\epsilon. \end{cases}$$

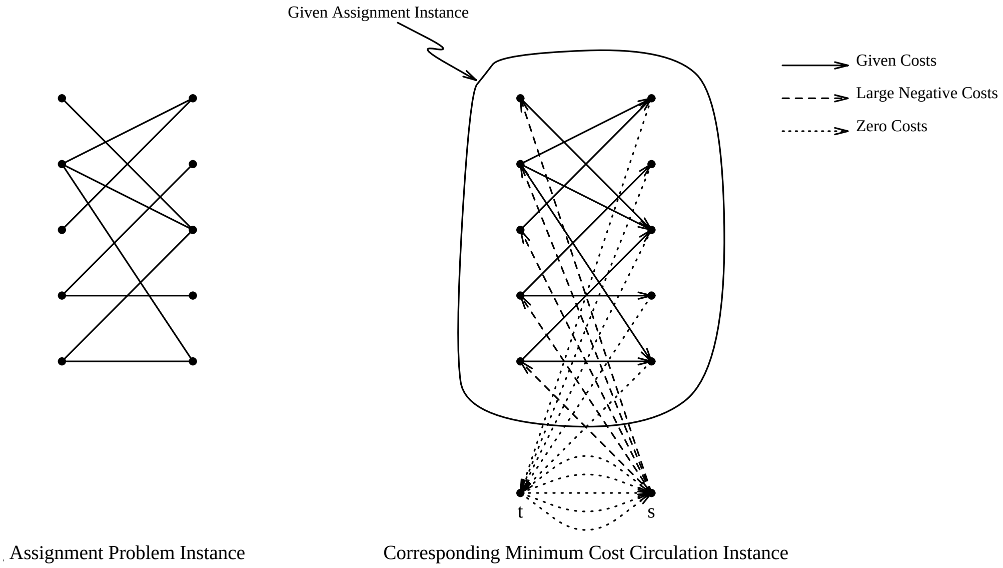

Given Assignment Instance
Given Costs
Large Negative Costs
Zero Costs
t
s
Assignment Problem Instance
Corresponding Minimum Cost Circulation Instance

FIG. 8.1. *Reduction from assignment to minimum cost circulation.*

A pseudoflow f is *ϵ-optimal* if f is ϵ-optimal with respect to some price function p. If the arc costs are integers and $\epsilon < 1/n$, any ϵ-optimal circulation is optimal.

For a given f and p, an arc $a \in E_f$ is *admissible* iff

$$\begin{cases} a \in E_U \quad \text{and} \quad c_p(a) < \epsilon \quad \text{or} \\ a \notin E_U \quad \text{and} \quad c_p(a) < -\epsilon. \end{cases}$$

The *admissible graph* $G_A = (V, E_A)$ is the graph induced by the admissible arcs.

These asymmetric definitions of ϵ-optimality and admissibility are natural in the context of the assignment problem. They have the benefit that the complementary slackness conditions are violated on $O(n)$ arcs (corresponding to the matched arcs). For the symmetric definition, complementary slackness can be violated on $\Omega(m)$ arcs.

First we give a high-level description of the successive approximation algorithm (see Figure 8.2). The algorithm starts with $\epsilon = C$, $f(a) = 0$ for all $a \in E$, and $p(v) = 0$ for all $v \in V$. At the beginning of every iteration, the algorithm divides ϵ by a constant factor α and saturates all arcs a with $c_p(a) < 0$. The iteration modifies f and p so that f is a circulation that is (ϵ/α)-optimal with respect to p. When $\epsilon < 1/n$, f is optimal and the algorithm terminates. The number of iterations of the algorithm is $1 + \lfloor \log_\alpha(nC) \rfloor$.

Reducing ϵ is the task of the subroutine *refine*. The input to *refine* is ϵ, f, and p such that (except in the first iteration) circulation f is ϵ-optimal with respect to p. The output from *refine* is $\epsilon' = \epsilon/\alpha$, a circulation f, and a price function p such that f is ϵ'-optimal with respect to p. At the first iteration, the zero flow is not C-optimal with respect to the zero price function, but because every simple path in the residual graph has cost of at least $-nC$, standard results about *refine* remain true.

The generic *refine* subroutine (described in Figure 8.3) begins by decreasing the value of ϵ and setting f to saturate all residual arcs with negative reduced cost.

This converts f into an ϵ-optimal pseudoflow (indeed, into a 0-optimal pseudoflow). Then the subroutine converts f into an ϵ-optimal circulation by applying a

```
**procedure** MIN-COST($V, E, u, c$);
    [initialization]
    $\epsilon \leftarrow C$ ; $\forall v, \quad p(v) \leftarrow 0; \quad \forall a, \quad f(a) \leftarrow 0;$
    [loop]
    **while** $\epsilon \geq 1/n$ **do**
        $(\epsilon, f, p) \leftarrow$ *refine*$(\epsilon, f, p)$;
    **return**$(f)$;
**end.**
```

FIG. 8.2. *The cost-scaling algorithm.*

```
**procedure** REFINE($\epsilon, f, p$);
    [initialization]
    $\epsilon \leftarrow \epsilon/\alpha$;
    $\forall a \in E$ with $c_p(a) < 0, \quad f(a) \leftarrow u(a)$;
    [loop]
    **while** $f$ is not a circulation
        apply a *push* or a *relabel* operation;
    **return**$(\epsilon, f, p)$;
**end.**
```

FIG. 8.3. *The generic* refine *subroutine.*

```
PUSH($v, w$).
    send a unit of flow from $v$ to $w$.
**end.**


RELABEL($v$).
    **if** $v \in U$
        **then** replace $p(v)$ by $\max_{(v,w) \in E_f} \{p(w) - c(v, w)\}$
        **else** replace $p(v)$ by $\max_{(v,w) \in E_f} \{p(w) - c(v, w) - 2\epsilon\}$
**end.**
```

FIG. 8.4. *The* push *and* relabel *operations.*

sequence of *push* and *relabel* operations, each of which preserves ϵ-optimality. The generic algorithm does not specify the order in which these operations are applied. Next, we describe the *push* and *relabel* operations for the unit-capacity case.

As in the maximum flow case, a *push* operation applies to an admissible arc (v, w) whose tail node v is active, and consists of pushing one unit of flow from v to w. A *relabel* operation applies to an active node v that is not the tail of any admissible arc. The operation sets $p(v)$ to the smallest value allowed by the ϵ-optimality constraints, namely $\max_{(v,w) \in E_f} \{p(w) - c(v, w)\}$ if $v \in U$, or $\max_{(v,w) \in E_f} \{p(w) - c(v, w) - 2\epsilon\}$ otherwise. Figure 8.4 gives the *push* and *relabel* operations.

The analysis of cost-scaling push-relabel algorithms is based on the following facts [12, 14]. During a scaling iteration

- no node price increases;
- every relabeling decreases a node price by at least ϵ;
- for any $v \in V$, $p(v)$ decreases by $O(n\epsilon)$.

9. Global updates and the minimum change discharge algorithm. In this section, we generalize the ideas of minimum distance discharge and global updates to the context of the minimum cost circulation problem and analyze the algorithm that embodies these generalizations.

We analyze a single execution of *refine*, and to simplify our notation, we make some assumptions that do not affect the results. We assume that the price function is identically zero at the beginning of the iteration. Our analysis goes through without this assumption, but the required condition can be achieved at no increased asymptotic cost by replacing the arc costs with their reduced costs and setting the node prices to zero in the first step of *refine*.

Under the assumption that each iteration begins with the zero price function, the *price change* of a node v during an iteration is $\delta(v) = \lfloor -p(v)/\epsilon \rfloor$. By analogy to the matching case, we define $\Gamma(f,p) = \min_{e_f(v)>0}\{\delta(v)\}$, and let $\Gamma_{\max}$ denote the maximum value attained by $\Gamma(f,p)$ so far in this iteration. The *minimum change discharge* strategy consists of repeatedly selecting a unit of excess at an active node v with $\delta(v) = \Gamma$ and processing that unit until it cancels some deficit or until a relabeling occurs. We implement this strategy as in the unweighted case. Observe that no node's price changes by more than $2\alpha n\epsilon$ during refine, so a collection of $2\alpha n+1$ buckets $b_0, \ldots, b_{2\alpha n}$ is sufficient to keep every active node v in $b_{\delta(v)}$. As before, the algorithm maintains the index μ of the lowest-numbered nonempty bucket and avoids bucket access except immediately after a deficit is canceled or a relabeling of a node v sets $\delta(v) > \mu$.

In the weighted context, a global update takes the form of setting each node price so that G_A is acyclic, there is a path in G_A from every excess to some deficit (a node v with $e_f(v) < 0$) and every node reachable in G_A from a node with excess lies on such a path. This amounts to a modified shortest-paths computation and can be done in $O(m)$ time using ideas from Dial's work [3]. At every *refine*, the first global update is performed immediately after saturating all residual arcs with negative reduced cost. After each *push* and *relabel* operation, the algorithm checks the following two conditions and performs a global update if both conditions hold:

- Since the most recent update, at least one unit of excess has canceled some deficit.
- Since the most recent update, the algorithm has done at least m work in *push* and *relabel* operations.

We developed global updates from an implementation heuristic for the minimum cost circulation problem [11], but in retrospect they prove similar in the assignment context to the one-processor Hungarian Search technique developed in [8].

Immediately after each global update, the algorithm rebuilds the buckets in $O(n)$ time and sets μ to zero. As in the unweighted case, we have the following easy bound on the extra work done by the algorithm in selecting nodes to process.

LEMMA 9.1. *Between two consecutive global updates, the algorithm does $O(n)$ work in examining empty buckets.*

Figure 9.1 represents the main ideas behind our analysis of an iteration of the minimum change discharge algorithm. The diagram differs from Figure 4.1 because we must account for pushes and relabelings that occur at nodes with large values of δ when $\Gamma_{\max}$ is small. Such operations could not arise in the matching algorithm but are possible here.

We begin our analysis with a lemma that is essentially similar to Lemma 4.2.

LEMMA 9.2. *The algorithm does $O(km)$ work in the course of* relabel *operations*

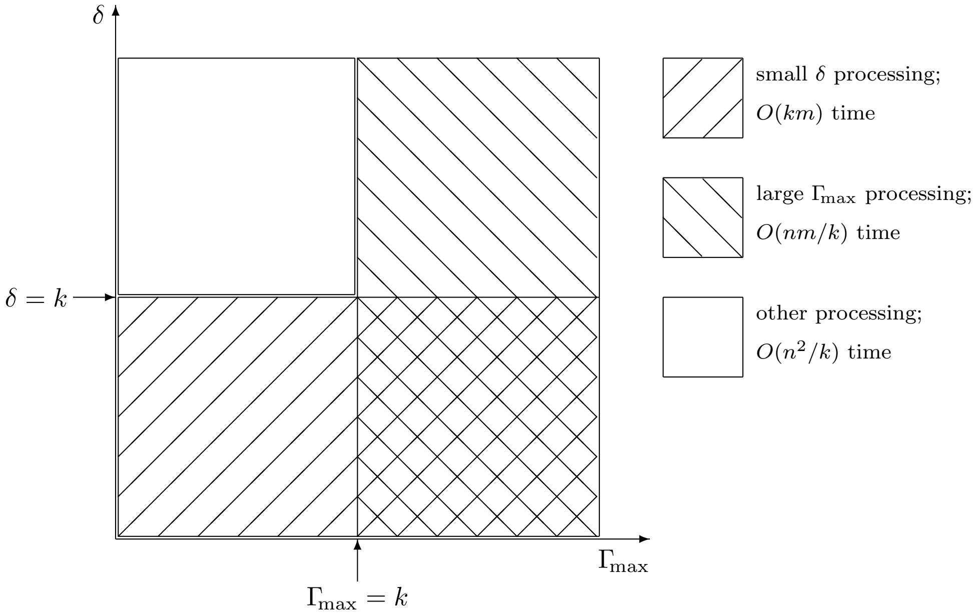


FIG. 9.1. *Accounting for work in the minimum change discharge algorithm.*

on nodes v obeying $\delta(v) \leq k$ and push *operations from those nodes.*

Proof. A node v can be relabeled at most $k + 1$ times while $\delta(v) \leq k$, so the relabelings of such nodes and the pushes from them require $O(km)$ work. □

To analyze our algorithm for the assignment problem, we must overcome two main difficulties that were not present in the matching case. First, we can do *push* and *relabel* operations at nodes whose price changes are large even when $\Gamma_{\max}$ is small; this work is not bounded by Lemma 9.2 and we must account for it. Second, our analysis of the period when $\Gamma_{\max}$ is large in the unweighted case does not generalize because it is not true that $\delta(v)$ gives a bound on the breadth-first-search distance from v to a deficit in the residual graph.

Lemma 9.4 is crucial in resolving both of these issues, and to prove it we use the following standard result which is analogous to Lemma 4.3.

LEMMA 9.3. *Given a matching network G and an integral circulation g, any integral pseudoflow f in G_g can be decomposed into*

- *cycles, and*
- *paths, each from a node u with $e_f(u) < 0$ to a node v with $e_f(v) > 0$,*

where all the elements of the decomposition are pairwise node-disjoint except at s, t, and the endpoints of the paths, and each element carries one unit of flow.

We denote a path from node u to node v in such a decomposition by $(u \rightsquigarrow v)$.

The following lemma is similar in spirit to those in [8] and [12], although the single-phase push-relabel framework of our algorithm changes the structure of the proof. Let $\mathcal{E}(f)$ denote the total excess in pseudoflow f, i.e., $\sum_{e_f(v)>0} e_f(v)$. When no confusion arises, we simply use $\mathcal{E}$ to denote the total excess in the current pseudoflow. The lemma depends on the $(\alpha\epsilon)$-optimality of the circulation produced by the previous iteration of *refine*, so it holds only in the second and subsequent scaling iterations. Because the zero circulation is not C-optimal with respect to the zero price function, we need different phrasing to accomplish the same task in the first iteration. The differences are mainly technical, so the first-iteration lemmas and their proofs are confined to Appendix A.

LEMMA 9.4. *At any point during an execution of* refine *other than the first,* $\mathcal{E} \times \Gamma_{\max} \leq 2\big((5+\alpha)n - 1\big)$.

Proof. Let c denote the (reduced) arc cost function at the beginning of this execution of *refine*, and let $G = (V, E)$ denote the augmented residual graph at the same instant. For simplicity in the following analysis, we view a pseudoflow as an entity in this graph G. Let f', p' be the current pseudoflow and price function, and let f, p be the pseudoflow and price function at the most recent point during the execution of *refine* when $\Gamma(f,p) = \Gamma_{\max}$. Since $\mathcal{E}(f) \geq \mathcal{E}(f')$ and $\Gamma(f,p) \geq \Gamma(f',p')$, it is enough to prove the lemma for f, p. We have

$$\mathcal{E}(f) \times \Gamma_{\max} \leq \sum_{e_f(v)>0} \delta(v) e_f(v).$$

From the definition of δ, then,

$$\mathcal{E}(f) \times \Gamma_{\max} \times \epsilon \leq -\sum_{e_f(v)>0} p(v) e_f(v).$$

We will complete our proof by showing that

$$-\sum_{e_f(v)>0} p(v) e_f(v) = c_p(f) - c(f)$$

and then deriving an upper bound on this quantity.

By the definition of the reduced costs,

$$c_p(f) - c(f) = \sum_{f(v,w)>0} \big(p(v) - p(w)\big) f(v,w).$$

Letting $\mathcal{P}$ be a decomposition of f into paths and cycles according to Lemma 9.3 and noting that cycles make no contribution to the sum, we can rewrite this expression as

$$\sum_{(u \rightsquigarrow v) \in \mathcal{P}} (p(u) - p(v)).$$

Since nodes u with $e_f(u) < 0$ are never relabeled, $p(u) = 0$ for such a node, and we have

$$c_p(f) - c(f) = -\sum_{(u \rightsquigarrow v) \in \mathcal{P}} p(v).$$

Because the decomposition $\mathcal{P}$ must account for all of f's excesses and deficits, we can rewrite

$$c_p(f) - c(f) = -\sum_{e_f(v)>0} p(v) e_f(v).$$

Now we derive an upper bound on $c_p(f) - c(f)$. It is straightforward to verify that for any matching network G and integral circulation g, the residual graph G_g has exactly n arcs $a \notin E_U$, and so from the fact that the execution of *refine* begins with the augmented residual graph of an $(\alpha\epsilon)$-optimal circulation, we deduce that there are at most n negative-cost arcs in E. Because each of these arcs has cost at least $-2\alpha\epsilon$, we have $c(f) \geq -2\alpha n\epsilon$. Hence $c_p(f) - c(f) \leq c_p(f) + 2\alpha n\epsilon$.

Now consider $c_p(f)$. Clearly, $f(a) > 0 \Rightarrow a^R \in E_f$, and ϵ-optimality of f with respect to p says that $a^R \in E_f \Rightarrow c_p(a^R) \geq -2\epsilon$. Since $c_p(a^R) = -c_p(a)$, we have $f(a) > 0 \Rightarrow c_p(a) \leq 2\epsilon$. Recalling our decomposition $\mathcal{P}$ into cycles and paths from deficits to excesses, observe that $c_p(f) = \sum_{P \in \mathcal{P}} c_p(P)$. Let $\nu(P)$ denote the interior of a path P, i.e., the path minus its endpoints and initial and final arcs, and let $\partial(P)$ denote the set containing the initial and final arcs of P. If P is a cycle, $\nu(P) = P$ and $\partial(P) = \emptyset$. We can write

$$c_p(f) = \sum_{P \in \mathcal{P}} c_p\big(\nu(P)\big) + \sum_{P \in \mathcal{P}} c_p\big(\partial(P)\big).$$

The total number of arcs not incident to s or t in the cycles and path interiors is at most n by node-disjointness, and the number of arcs incident to s or t is at most $2n - 1$. Also, the total excess is never more than n, so the initial and final arcs of the paths number no more than $2n$. And because each arc carrying positive flow has reduced cost at most 2ϵ, we have $c_p(f) \leq (n + 2n - 1 + 2n)2\epsilon = (5n - 1)2\epsilon$.

Therefore, $c_p(f) - c(f) \leq 2\big((5 + \alpha)n - 1\big)\epsilon$, and we have $\mathcal{E}(f) \times \Gamma_{\max} \leq 2\big((5 + \alpha)n - 1\big)$. □

COROLLARY 9.5. *$\Gamma_{\max} \geq k$ implies $\mathcal{E} = O(n/k)$.*

We use the following lemma to show that when $\Gamma_{\max}$ is small, we do a limited amount of work at nodes whose price changes are large.

LEMMA 9.6. *While $\Gamma_{\max} \leq k$, the amount of work done in relabelings at nodes v with $\delta(v) > k$ and pushes from those nodes is $O(n^2/k)$.*

Proof. For convenience, we say a node that gets relabeled under the conditions of the lemma is a *bad* node. We process a given node v either because we selected a unit of excess at v or because the most recent operation was a *push* from one of v's neighbors to v. If a unit of v's excess is selected, we have $\delta(v) \leq \Gamma_{\max}$ (indeed without global updates, $\delta(v) = \Gamma_{\max}$), which implies $\delta(v) \leq k$, so v cannot be a bad node. In the second case, the unit of excess just pushed to v will remain at v until $\Gamma_{\max} \geq \delta(v)$ because the condition $\delta(v) > \mu$ will cause excess at a different node to be selected immediately after v is relabeled. We cannot select v's excess until $\Gamma_{\max} \geq \delta(v)$, and at such a time, Corollary 9.5 shows that the total excess remaining is $O(n/k)$. Since each relabeling of a bad node leaves a unit of excess that must remain at that node until $\Gamma_{\max} \geq k$, the number of relabelings of bad nodes is $O(n/k)$. Because every node has degree at most n, the work done in pushes and relabelings at bad nodes is $O(n^2/k)$. □

Recall that the algorithm initiates a global update only after a unit of excess has canceled some deficit since the last global update. The next lemma, analogous to Lemma 4.6, shows that this rule cannot introduce too great a delay.

LEMMA 9.7. *Between any two consecutive global update operations, the algorithm does $\Theta(m)$ work.*

Proof. As in the unweighted case, it suffices to show that the algorithm does $O(m)$ work in canceling a deficit immediately after a global update operation, and $O(m)$ work in selecting nodes to process. The definition of a global update operation suffices to ensure that a unit of excess reaches some deficit immediately after a global update and before any relabeling occurs, and Lemma 9.1 shows that the extra work done between global updates in selecting nodes to process is $O(n)$. □

Lemmas 9.2 and 9.6 show that the algorithm takes $O(km + n^2/k)$ time when $\Gamma_{\max} \leq k$. Corollary 9.5 says that when $\Gamma_{\max} \geq k$, the total excess remaining is $O(n/k)$, and Lemma 9.7 shows that $O(m)$ work suffices to cancel each unit of excess remaining. Therefore the total work in an execution of *refine* is $O\big(km+n^2/k+nm/k\big)$, and choosing $k = \Theta(\sqrt{n}\,)$ gives a $O(\sqrt{n}m)$ time bound on an execution of *refine*. The overall time bound follows from the $O(\log(nC))$ bound on the number of scaling iterations, giving the following theorem.

THEOREM 9.8. *The minimum change discharge algorithm with global updates computes a minimum cost circulation in a matching network in* $O\big(\sqrt{n}m\log(nC)\big)$ *time.*

Graph compression methods [6] do not apply to graphs with weights because the compressed graph preserves only adjacency information and cannot encode arbitrary edge weights. Hence the Feder–Motwani techniques cannot improve performance in the assignment problem context.

10. Minimum change discharge algorithm without global updates. We present a family of assignment instances on which we show that *refine*, without global updates, performs $\Omega(nm)$ work in the first scaling iteration under the minimum change discharge selection rule. Hence this family of matching networks suffices to show that global updates account for an asymptotic difference in running time.

The family of assignment instances on which we show that *refine*, without global updates, takes $\Omega(nm)$ time is structurally the same as the family of bad examples we used in the unweighted case, except that each weighted example has two additional nodes and one additional edge. The costs of the edges present in the unweighted example are zero, and there are two extra nodes connected only to each other, sharing an edge with cost α. These two nodes and the edge between them are present only to establish the initial value of ϵ and the costs of arcs introduced in the reduction, and are ignored in our description of the execution.

At the beginning of the first scaling iteration, $\epsilon = \alpha$. The iteration starts by setting $\epsilon = 1$. From this point on, the execution is similar to the execution of the minimum distance discharge algorithm given in section 6, but the details differ because of the asymmetric definitions of ϵ-optimality and admissibility that we use in the weighted case.

Figure 9.2 details an execution of the minimum change discharge algorithm without global updates. As in the unweighted case, every relabeling changes a node price by at most two and the algorithm does $\Omega(n^2)$ relabelings. Consequently, the relabelings require $\Omega(nm)$ work, and we have the following theorem.

THEOREM 10.1. *For any function* $m(n)$ *in the range* $n \leq m(n) < n^2/4$*, there exists an infinite family of instances of the assignment problem having* $\Theta(n)$ *nodes and* $\Theta\big(m(n)\big)$ *edges on which the minimum change discharge implementation of* refine *without global updates runs in* $\Omega\big(nm(n)\big)$ *time.*

11. Conclusions and open questions. We have presented algorithms that achieve the best time bounds known for bipartite matching, i.e., $O\left(\sqrt{n}m\frac{\log(n^2/m)}{\log n}\right)$, and for the assignment problem in the cost-scaling context, i.e., $O\left(\sqrt{n}m\log(nC)\right)$. We have also given examples to show that without global updates, the algorithms perform worse. Hence we conclude that global updates can be a useful tool in the theoretical development of algorithms.

```
1. Initialization establishes $|X|$ units of excess, one at each node of $X$.
2. While some node $w \in Y$ has no excess,
   2.1. a unit of excess moves from a node of $X$ to $w$;
   2.2. $w$ is relabeled so that $p(w) = -2$.
   Remark: Now every node of $Y$ has one unit of excess.
3. Active nodes in $X$ are relabeled one-by-one so that each has price $-2$.
4. A unit of excess moves from the most recently relabeled node of $X$ to a node of $Y$,
   then to $t$, and on to cancel a unit of deficit at $s$.
5. While more than one node of $Y$ has excess,
   5.1. A unit of excess moves to $t$ and thence to $s$ from a node of $Y$;
6. The remaining unit of excess at a node of $Y$ moves to a node $v \in X$ with $p(v) = 0$,
   and $v$ is relabeled so that $p(v) = -2$.
7. $\ell \leftarrow 1$.
8. While $\ell \leq \alpha n/2 - 1$,
        Remark: All excesses are at nodes of $X$, and these nodes have price $-2\ell$; all
        other nodes in $X$ have price $-2\ell + 2$; all nodes in $Y$ have price $-2\ell$.
   8.1. A unit of excess is selected, and while some node $x \in X$ has $p(x) = -2\ell + 2$,
          • the selected unit moves from some active node $v$ to $w$, a neighbor of $x$ in
            $G_f$ (for a given $x$ there is a unique such $w$);
          • the unit of excess moves from $w$ to $x$;
          • $x$ is relabeled so $p(x) = -2\ell$.
        Remark: Now all nodes in $X \cup Y$ have price $-2\ell$; all excesses are at nodes of
        $X$.
   8.2. While some node $w \in Y$ has $p(w) = -2\ell$ and some node $v \in X$ has $e_f(v) = 1$,
          • a unit of excess moves from $v$ to $w$;
          • $w$ is relabeled so $p(w) = -2\ell - 2$.
        Remark: The following loop is executed only if $|X| < 2|Y|$. All active nodes
        in $Y$ have price $-2\ell - 2$, and all other nodes in $Y$ have price $-2\ell$.
   8.3. If a node in $Y$ has price $-2\ell$, a unit of excess is selected, and while some node
        $y \in Y$ has $p(y) = -2\ell$,
          • the selected unit moves from some $w \in Y$ with $e_f(w) = 1$ to $v \in X$ with
            $p(v) = -2\ell$, and then to $y$;
          • $y$ is relabeled so $p(y) = -2\ell - 2$.
        Remark: The following loop is executed only if $|X| > 2|Y|$.
   8.4. For each node $v \in X$ with $e_f(v) = 1$,
          • $v$ is relabeled so $p(v) = \max\{-2\ell - 2, -\alpha n\}$.
   8.5. For each node $w \in Y$ with $e_f(w) = 1$,
          • a unit of excess moves from $w$ to $v \in X$ with $p(v) = -2\ell$;
          • $v$ is relabeled so $p(v) = \max\{-2\ell - 2, -\alpha n\}$.
   8.6. $\ell \leftarrow \ell + 1$.
9. Excesses move one-by-one from active nodes in $X$ (which have price $-\alpha n$) to $s$.
```

FIG. 9.2. *The minimum change discharge execution on bad examples.*

We have shown a family of assignment instances on which *refine*, without global updates, performs poorly, but the poor performance seems to hinge on details of the reduction, so it happens only in the first scaling iteration. An interesting open question is the existence of a family of instances of the assignment problem on which *refine* uses $\Omega(nm)$ time in *every* scaling iteration.

Appendix A. The first scaling iteration. Let G be the network produced by reducing an assignment problem instance to the minimum cost circulation problem as in section 8. When *refine* initializes by saturating all negative arcs in this network, the only deficit created will be at s by our assumption that the input costs are nonnegative.

For a pseudoflow f in G, define $\mathcal{E}_t(f)$ to be the amount of f's excess that can reach s by passing through t. $\mathcal{E}_t(f)$ corresponds to the residual flow value in the unweighted case (see section 4).

The $(\alpha\epsilon)$-optimality of the initial flow and price function played an important role in the proof of Lemma 9.4, specifically by lower-bounding the initial cost of any arc that currently carries a unit of flow. In contrast, the first scaling iteration may have many arcs that carry flow and have extremely negative costs relative to ϵ, specifically those arcs of the form (s, v) introduced by the reduction. But to counter this difficulty, the first iteration has an advantage that later iterations lack: an *upper* bound (in terms of ϵ) on the initial cost of *every* residual arc in the network. Specifically, recall that the value of ϵ in the first iteration is C/α, where C is the largest cost of an edge in the given assignment instance. So for any arc a other than the (v, s) arcs introduced by the reduction, $c(a) \leq \alpha\epsilon$ in the first scaling iteration.

LEMMA A.1. *At any point during the first execution of* refine, $\mathcal{E}_t \times \Gamma_{\max} \leq n(2 + \alpha)$.

Proof. Let f', p' be the current pseudoflow and price function, and as in the proof of Lemma 9.4, let f, p be the pseudoflow and price function at the most recent point when $\Gamma(f, p) = \Gamma_{\max}$. As before, it is enough to prove the lemma for f, p; this will imply the claim holds for f', p'.

Let f^* be a minimum cost circulation in G, and let $f' = f^* - f$. Recall that the costs on the (s, v) arcs are negative enough that f^* must correspond to a matching of maximum cardinality. Therefore, f' moves $\mathcal{E}_t(f)$ units of f's excess to s through t and returns the remainder to s without its passing through t. Now $-f'$ is a pseudoflow in G_{f^*} and can be decomposed into cycles and paths according to Lemma 9.3; as in the proof of Lemma 4.4, let $\mathcal{P}$ denote the induced decomposition of f'. Let $\mathcal{Q} \subseteq \mathcal{P}$ be the set of paths that pass through t, and note that $\mathcal{E}_t(f) = |\mathcal{Q}|$. Let $e_f^t(v)$ denote the number of paths of $\mathcal{Q}$ beginning at node v. The only deficit in f is at s, so $e_f^t(v)$ is precisely the amount of v's excess that reaches s by passing through t if we imagine augmenting f along the paths of $\mathcal{P}$. Of particular importance is that no path in $\mathcal{Q}$ uses an arc of the form (s, v) or (v, s) for $v \neq t$.

Observe that

$$\mathcal{E}_t(f) \times \Gamma_{\max} \leq \sum_{e_f^t(v) > 0} e_f^t(v)\delta(v),$$

so by the definition of δ,

$$\epsilon \times \mathcal{E}_t(f) \times \Gamma_{\max} \leq -\sum_{e_f^t(v) > 0} e_f^t(v)p(v).$$

Now note that for any path P from v to s, we have $p(v) = c_p(P) - c(P)$ because $p(s) = 0$. Every arc used in the decomposition $\mathcal{P}$ appears in G_f. By ϵ-optimality of f, each of the n or fewer arcs a in G_f with negative reduced cost has $c_p(a) \geq -2\epsilon$, so we have $\sum_{P \in \mathcal{Q}} c_p(P) \geq -2n\epsilon$. Next we use the upper bound on the initial costs to note that $\sum_{P \in \mathcal{Q}} c(P) \leq \alpha n\epsilon$, so

$$\epsilon \times \mathcal{E}_t(f) \times \Gamma_{\max} \leq -\sum_{e_f^t(v) > 0} e_f^t(v)p(v) \leq 2n\epsilon + \alpha n\epsilon = n(2 + \alpha)\epsilon,$$

and the lemma follows. □

LEMMA A.2. *At any point during the first execution of* refine*,* $\mathcal{E} \times (\Gamma_{\max} - \alpha n) \leq n(2 + \alpha)$.

Proof. The proof is essentially the same as the proof of Lemma A.1, except that if $\Gamma_{\max} > \alpha n$, each path from an excess to the deficit at s will include one arc of the form (v, s), and each such arc has original cost $-nC = -\alpha n\epsilon$. □

Lemmas A.1 and A.2 allow us to split the analysis of the first scaling iteration into four stages, much as we did with the minimum distance discharge algorithm for matching. Specifically, the analysis of section 9 holds up until the point where $\Gamma_{\max} \geq \alpha n$, with Lemma A.1 taking the place of Lemma 9.4. Straightforward extensions of the relevant lemmas show that the algorithm does $O(km + n^2/k)$ work when $\Gamma_{\max} \in [\alpha n, \alpha n + k]$, and when $\Gamma_{\max} > \alpha n + k$, Lemma A.2 bounds the algorithm's work by $O(nm/k)$. The balancing works as before: choosing $k = \Theta(\sqrt{n})$ gives a bound of $O(\sqrt{n}m)$ time for the first scaling iteration.

Acknowledgment. The authors would like to thank an anonymous referee whose careful reading led us to several corrections and improvements.

REFERENCES

[1] R. J. ANDERSON AND J. C. SETUBAL, *Goldberg's algorithm for the maximum flow in perspective: A computational study*, in Network Flows and Matching: First DIMACS Implementation Challenge, D. S. Johnson and C. C. McGeoch, eds., AMS, Providence, RI, 1993, pp. 1–18.

[2] U. DERIGS AND W. MEIER, *Implementing Goldberg's max-flow algorithm — A computational investigation*, ZOR—Math. Methods Oper. Res., 33 (1989), pp. 383–403.

[3] R. B. DIAL, *Algorithm* 360*: Shortest path forest with topological ordering*, Comm. ACM, 12 (1969), pp. 632–633.

[4] 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.

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

[6] T. FEDER AND R. MOTWANI, *Clique partitions, graph compression and speeding-up algorithms*, in Proc. 23rd Annual ACM Symposium on Theory of Computing, New Orleans, LA, ACM, New York, 1991, pp. 123–133.

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

[8] H. N. GABOW AND R. E. TARJAN, *Almost-optimal speed-ups of algorithms for matching and related problems*, in Proc. 20th Annual ACM Symposium on Theory of Computing, Chicago, IL, ACM, New York, 1988, pp. 514–527.

[9] H. N. GABOW AND R. E. TARJAN, *Faster scaling algorithms for network problems*, SIAM J. Comput., 18 (1989), pp. 1013–1036.

[10] A. V. GOLDBERG, *Efficient Graph Algorithms for Sequential and Parallel Computers*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 1987. (Also available as Technical Report TR-374, Lab. for Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 1987.)

[11] A. V. GOLDBERG, *An efficient implementation of a scaling minimum-cost flow algorithm*, in Proc. 3rd Integer Prog. and Combinatorial Opt. Conf., Erice, Italy, 1993, pp. 251–266.

[12] A. V. GOLDBERG, S. A. PLOTKIN, AND P. M. VAIDYA, *Sublinear-time parallel algorithms for matching and related problems*, J. Algorithms, 14 (1993), pp. 180–213.

[13] A. V. GOLDBERG AND R. E. TARJAN, *A new approach to the maximum flow problem*, J. Assoc. Comput. Mach., 35 (1988), pp. 921–940.

[14] A. V. GOLDBERG AND R. E. TARJAN, *Finding minimum-cost circulations by successive approximation*, Math. Oper. Res., 15 (1990), pp. 430–466.

[15] J. E. HOPCROFT AND R. M. KARP, *An* $n^{5/2}$ *algorithm for maximum matching in bipartite graphs*, SIAM J. Comput., 2 (1973), pp. 225–231.

[16] A. V. KARZANOV, *On finding maximum flows in networks with special structure and some applications*, in Matematicheskie Voprosy Upravleniya Proizvodstvom, vol. 5, Moscow State University Press, Moscow, 1973 (in Russian).

[17] A. V. Karzanov, *The exact time bound for a maximum flow algorithm applied to the set representatives problem*, in Problems in Cybernetics, vol. 5, Nauka, Moscow, 1973, pp. 66–70 (in Russian).

[18] H. W. Kuhn, *The Hungarian method for the assignment problem*, Naval Res. Logist. Quart., 2 (1955), pp. 83–97.

[19] Q. C. Nguyen and V. Venkateswaran, *Implementations of Goldberg-Tarjan maximum flow algorithm*, in Network Flows and Matching: First DIMACS Implementation Challenge, D. S. Johnson and C. C. McGeoch, eds., AMS, Providence, RI, 1993, pp. 19–42.

[20] J. B. Orlin and R. K. Ahuja, *New scaling algorithms for the assignment and minimum cycle mean problems*, Math. Programming, 54 (1992), pp. 41–56.

SIAM J. Discrete Math.
Vol. 10, No. 4, pp. 573–598, November 1997

004

TASK SCHEDULING IN NETWORKS*

CYNTHIA PHILLIPS[†], CLIFFORD STEIN[‡], AND JOEL WEIN[§]

Abstract. Scheduling a set of tasks on a set of machines so as to yield an efficient schedule is a basic problem in computer science and operations research. Most of the research on this problem incorporates the potentially unrealistic assumption that communication between the different machines is instantaneous. In this paper we remove this assumption and study the problem of *network scheduling*, where each job originates at some node of a network, and in order to be processed at another node must take the time to travel through the network to that node.

Our main contribution is to give approximation algorithms and hardness proofs for fully general forms of the fundamental problems in network scheduling. We consider two basic scheduling objectives: minimizing the makespan and minimizing the average completion time. For the makespan, we prove small constant factor hardness-to-approximate and approximation results. For the average completion time, we give a log-squared approximation algorithm for the most general form of the problem. The techniques used in this approximation are fairly general and have several other applications. For example, we give the first nontrivial approximation algorithm to minimize the average weighted completion time of a set of jobs on related or unrelated machines, with or without a network.

Key words. scheduling, approximation algorithm, NP-completeness, networks

AMS subject classifications. 68M10, 90B12, 68Q10, 68Q22, 68Q25, 90B35, 68M20

PII. S0895480194279057

1. Introduction.

Scheduling a set of tasks on a set of machines so as to yield an efficient schedule is a basic problem in computer science and operations research. It is also a difficult problem, and hence, much of the research in this area has incorporated a number of potentially unrealistic assumptions. One such assumption is that communication between the different machines is instantaneous. In many application domains, however, such as a network of computers or a set of geographically scattered repair shops, decisions about when and where to move the tasks are a critical part of achieving efficient resource allocation. In this paper we remove the assumption of instantaneous communication from the traditional parallel machine models and study the problem of *network scheduling*, in which each job originates at some node of a network, and in order to be processed at another node must take the time to travel through the network to that node.

Until this work, network scheduling problems had either loose [2, 4] or no approximation algorithms. Our main contribution is to give approximation algorithms and hardness proofs for fully general forms of the fundamental problems in network

[*] Received by the editors December 21, 1994; accepted for publication (in revised form) September 11, 1996. A preliminary version of this paper appeared in the Proceedings of the Fourth Scandinavian Workshop on Algorithm Theory, 1994, pp. 290–301. Much of this work was done while the second and third authors were visiting Sandia National Labs, the third author was visiting DIMACS, and the second author was visiting Polytechnic University.

http://www.siam.org/journals/sidma/10-4/27905.html

[†] Sandia National Labs, Albuquerque, NM 87185-1110 (caphill@cs.sandia.gov). This author was supported by U.S. Department of Energy contract DE-AC04-76DP00789.

[‡] Department of Computer Science, Dartmouth College, Hanover, NH 03755-3551 (cliff@cs.dartmouth.edu). This research was partially supported by NSF award CCR-9308701, a Walter Burke Research Initiation Award, and a Dartmouth College Research Initiation Award.

[§] Department of Computer Science, Polytechnic University, Brooklyn, NY 11201-2990(wein@mem.poly.edu). This research was partially supported by NSF research initiation award CCR-9211494 and a grant from the New York State Science and Technology Foundation through its Center for Advanced Technology in Telecommunications.

scheduling. Our upper bounds are robust, as they depend on general characteristics of the jobs and the underlying network. In particular, our algorithmic techniques to optimize average completion time yield other results, such as the first nontrivial approximation algorithms for a combinatorial scheduling question: minimization of average *weighted* completion time on unrelated machines. They also give the first approximation algorithm for a problem motivated by satellite communication systems. (To differentiate our network scheduling models from the traditional parallel machine models, we will refer to the latter as *combinatorial* scheduling models.)

Our results not only yield insight into the network scheduling problem, but also demonstrate contrasts between the complexity of certain combinatorial scheduling problems and their network variants, shedding light on their relative difficulty.

An instance $\mathcal{N} = (G, \ell, \mathcal{J})$ of the network scheduling problem consists of a network $G = (V, E)$, $|V| = m$, with nonnegative edge lengths ℓ; we define $\ell_{\max}$ to be the maximum edge length. At each vertex v_i in the network is a machine M_i. We are also given a set of n jobs, $J_1, \ldots, J_n$. Each job J_j originates, at time 0, on a particular *origin machine* M_{o_j} and has a processing requirement p_j; we define $p_{\max}$ to be $\max_{1 \le j \le n} p_j$. Each job must be processed on one machine without interruption. Job J_j is not available to be processed on a machine M' until time $d(M_{o_j}, M')$, where $d(M_i, M_k)$ is the length of the shortest path in G between M_i and M_k. We assume that the M_i are either identical (J_j takes time p_j on every machine) or that they are *unrelated* (J_j takes time p_{ij} on M_i, and the p_{ij} may all be different). In the unrelated machines setting, we define $p_{\max} = \max_{1 \le i \le m, 1 \le j \le n} p_{ij}$. The identical and unrelated machine models are fundamental in traditional parallel machine scheduling and are relatively well understood [3, 10, 11, 12, 15, 17, 25]. Unless otherwise specified, in this paper the machines in the network are assumed to be identical.

An alternative view of the network scheduling model is that each job J_j has a *release date*, a time before which it is unavailable for processing. In previous work on traditional scheduling models, a job's release date was defined to be the same on all machines. The network model can be characterized by allowing a job J_j's release date to be different on different machines; J_j's release date on M_k is $d(M_{o_j}, M_k)$. One can generalize further and consider problems in which a job's release date can be chosen arbitrarily for all m machines and need not reflect any network structure. Almost all of our upper bounds apply in this more general setting, whereas our lower bounds all apply when the release dates have network structure.

We study algorithms to minimize the two most basic objective functions. One is the *makespan* or *maximum completion time* of the schedule; that is, we would like all jobs to finish by the earliest time possible. The second is the *average completion time*. We define an α-approximation algorithm to be a polynomial-time algorithm that gives a solution of cost no more than α times optimal.

1.1. Previous work.

The problem of network scheduling has received some attention, mostly in the distributed setting. Deng et al. [4] considered a number of variants of the problem. In the special case in which each edge in the network is of unit length, all job processing times are the same, and the machines are identical, they showed that the off-line problem is in $\mathcal{P}$. It is not hard to see that the problem is $\mathcal{NP}$-complete when jobs are allowed to be of different sizes; they give an off-line $O(\log(m\ell_{\max}))$-approximation algorithm for this. They also give a number of results for the distributed version of the problem when the network topology is completely connected, a ring or a tree.

Awerbuch, Kutten, and Peleg [2] considered the distributed version of the problem under a novel notion of on-line performance, which subsumes the minimization of both average and maximum completion time. They give distributed algorithms with polylogarithmic performance guarantees in general networks. They also characterize the performance of feedback-based approaches. In addition they derived off-line approximation results similar to those of Deng et al. [2, 20]. Alon et al. [1] proved an $\Omega(\log m)$ lower bound on the performance of any distributed scheduler that is trying to minimize schedule length. Fizzano et al. [5] give a distributed 4.3-approximation algorithm for schedule length in the special case in which the network is a ring.

Our work differs from these papers by focusing on the centralized off-line problem and by giving approximations of higher quality. In addition, our approximation algorithms work in a more general setting, that of unrelated machines.

1.2. Summary of results. We first focus on the objective of minimizing the makespan and give a 2-approximation algorithm for scheduling jobs on networks of unrelated machines; the algorithm gives the same performance guarantee for identical machines as a special case. The 2-approximation algorithm matches the best-known approximation algorithm for scheduling unrelated machines with no underlying network [17]. Thus it is natural to ask whether the addition of a network to a combinatorial scheduling problem actually makes the problem any harder. We resolve this question by proving that the introduction of the network to the problem of scheduling identical machines yields a qualitatively harder problem. We show that for the network scheduling problem, no polynomial-time algorithm can do better than a factor of $\frac{4}{3}$ times optimal unless $\mathcal{P} = \mathcal{NP}$, even in a network in which all edges have length one. Comparing this with the polynomial approximation scheme of Hochbaum and Shmoys [10] for parallel machine scheduling, we see that the addition of a network does indeed make the problem harder.

Although the 2-approximation algorithm runs in polynomial time, it may be rather slow [21]. We thus explore whether a simpler strategy might also yield good approximations. A natural approach to minimizing the makespan is to construct schedules with no unforced idle time. Such strategies provide schedules of length a small constant factor times optimal, at minimal computational cost, for a variety of scheduling problems [6, 7, 15, 24]. We call such schedules *busy schedules*, and show that for the network scheduling problem their quality degrades significantly; they can be as much as an $\Omega\left(\sqrt{\frac{\log m}{\log\log m}}\right)$ factor longer than the optimal schedule.

This is in striking contrast to the combinatorial model (for which Graham showed that a busy strategy yields a 2-approximation algorithm [6]). In fact, even when release dates are introduced into the identical machine scheduling problem, if each job's release date is the same on all machines, busy strategies still give a $(2 - \frac{1}{m})$-approximation guarantee [8, 9]. Our result shows that when the release dates of the jobs are allowed to be different on different machines busy scheduling degrades significantly as a scheduling strategy. This provides further evidence that the introduction of a network makes scheduling problems qualitatively harder. However, busy schedules are of some quality; we show that they are of length a factor of $O\left(\frac{\log m}{\log\log m}\right)$ longer than optimal. This analysis gives a better bound than the $(O(\log m\ell_{\max}))$ bound of previously known approximation algorithms for identical machines in a network [2, 4, 20].

We then turn to the $\mathcal{NP}$-hard problem of the minimization of average completion time. Our major result for this optimality criterion is a $O(\log^2 n)$-approximation algorithm in the general setting of unrelated machines. It formulates the problem

Table 1

Summary of main algorithms and hardness results. The notation $x < \alpha \leq y$ means that we can approximate the problem within a factor of y, but unless $\mathcal{P} = \mathcal{NP}$ we cannot approximate the problem within a factor of x. Unreferenced results are new results found in this paper.

	Combinatorial	Network
min. makespan, identical machines	$\alpha < (1+\epsilon)$[10]	$4/3 < \alpha \leq 2$
min. makespan, identical machines, Busy schedules	$\alpha = 2 - \frac{1}{m}$[6]	$O\left(\frac{\log m}{\log\log m}\right)$,$\Omega\left(\sqrt{\frac{\log m}{\log\log m}}\right)$
min. makespan, unrelated machines	$3/2 < \alpha \leq 2$ [17]	$3/2 < \alpha \leq 2$
min. avg. completion time unrelated machines	$\alpha = 1$ [12]	$1 < \alpha \leq O(\log^2 n)$
min. avg. wtd. completion time unrelated machines, release dates	$1 < \alpha$[16] $\alpha \leq O(\log^2 n)$	$1 < \alpha \leq O(\log^2 n)$

as a hypergraph matching integer program and then approximately solves a relaxed version of the integer program. We can then find an integral solution to this relaxation, employing as a subroutine the techniques of Plotkin, Shmoys, and Tardos [21]. In combinatorial scheduling, a schedule with minimum average completion time can be found in polynomial time, even if the machines are unrelated.

The techniques for the average completion time algorithm are fairly general, and yield an $O(\log^2 n)$-approximation for minimizing the average *weighted* completion time. A special case of this result is an $O(\log^2 n)$-approximation algorithm for the $\mathcal{NP}$-hard problem of minimizing average weighted completion time for unrelated machines with no network; no previous approximation algorithms were known, even in the special case for which the machines are just of different speeds [3, 15]. Another special case is the first $O(\log^2 n)$-approximation algorithm for minimizing the average completion time of jobs with release dates on unrelated machines. No previous approximation algorithms were known, even for the special case of *just one machine* [15]. The technique can also be used to give an approximation algorithm for a problem motivated by satellite communication systems [18, 26].

We also give a number of other results, including polynomial-time algorithms for several special cases of the above-mentioned problems and a $\frac{5}{2}$-approximation for a variant of network scheduling in which each job has not only an origin, but also a destination.

A summary of some of these upper bounds and hardness results appears in Table 1.

A line of research which is quite different from ours, yet still has some similarity in spirit, was started by Papadimitriou and Yannakakis [19]. They modeled communication issues in parallel machine scheduling by abstracting away from particular networks and rather describing the communication time between *any* two processors by one network-dependent constant. They considered the scheduling of precedence-constrained jobs on an infinite number of identical machines in this model; the issues involved and the sorts of theorems proved are quite different from our results.

Although all of our algorithms are polynomial-time algorithms, they tend to be rather inefficient. Most rely on the work of [21] as a subroutine. As a result we will not discuss running times explicitly for the rest of the paper.

2. Makespan. In this section we study the problem of minimizing the makespan for the network scheduling problem. We first give an algorithm that comes within a factor of 2 of optimal. We then show that this is nearly the best we can hope for, as it is $\mathcal{NP}$-hard to approximate the minimum makespan within a factor of better than $\frac{4}{3}$ for identical machines in a network. This hardness result contrasts sharply with the combinatorial scenario, in which there is a polynomial approximation scheme [10]. The 2-approximation algorithm is computationally intensive, so we consider simple strategies that typically work well in parallel machine scheduling. In another sharp contrast to parallel machine scheduling, we show that the performance of such strategies degrades significantly in the network setting; we prove an $\Omega\left(\sqrt{\frac{\log m}{\log\log m}}\right)$ lower bound on the performance of any such algorithm. We also show that greedy algorithms do have some performance guarantee, namely $O(\frac{\log m}{\log\log m})$. Finally we consider a variant of the problem in which each job has not only an origin but also a destination, and give a $\frac{5}{2}$-approximation algorithm.

2.1. A 2-approximation algorithm for makespan. In this section we describe a 2-approximation algorithm to minimize the makespan of a set of jobs scheduled on a network of unrelated machines; the same bound for identical machines follows as a special case. Let $\mathcal{U}' = (G, \ell, \mathcal{J}')$ be an instance of the unrelated network scheduling problem with optimal schedule length D. Assuming that we know D, we will show how to construct a schedule of length at most $2D$. This can be converted, via binary search, into a 2-approximation algorithm for the problem in which we are not given D [10].

In the optimal schedule of length D, we know that the sum of the time each job spends travelling and being processed is bounded above by D. Thus, job J_j may run on machine M_i in the optimal schedule only if

$$d(M_{o_j}, M_i) + p_{ij} \leq D. \tag{1}$$

In other words, the length of an optimal schedule is not altered if we allow job J_j to run only on the machines for which (1) is satisfied. Formally, for a given job J_j, we will denote by $Q(J_j)$ the set of machines that satisfy (1). If we restrict each J_j to only run on the machines in $Q(J_j)$, the length of the optimal schedule remains unchanged.

Form combinatorial unrelated machines scheduling problem $(\mathcal{Z})$ as follows:

$$p'_{ij} = \begin{cases} p_{ij} & \text{if } M_i \in Q(J_j), \\ \infty & \text{otherwise.} \end{cases} \tag{2}$$

If the optimal schedule for the unrelated network scheduling problem has length D, then the optimal solution to the unrelated parallel machine scheduling problem (2) is at most D. We will use the 2-approximation algorithm of Lenstra, Shmoys and Tardos [17] to assign jobs to machines. The following theorem is easily inferred from [17].

THEOREM 2.1 (see [17]). *Let $\mathcal{Z}$ be an unrelated parallel machine scheduling problem with optimal schedule of length D. Then there exists a polynomial-time algorithm that finds a schedule $\mathcal{S}$ of length $2D$. Further, $\mathcal{S}$ has the property that no job starts after time D.*

THEOREM 2.2. *There exists a polynomial-time 2-approximation algorithm to minimize makespan in the unrelated network scheduling problem.*

Proof. Given an instance of the unrelated network scheduling problem, with shortest schedule of length D, form the unrelated parallel machine scheduling problem $\mathcal{Z}$ defined by (2) and use the algorithm of [17] to produce a schedule $\mathcal{S}$ of length $2D$. This schedule does not immediately correspond to a network schedule because some jobs may have been scheduled to run before their release dates. However, if we allocate D units of time for sending all jobs to the machines on which they run, and then allocate $2D$ units of time to run schedule S, we immediately get a schedule of length $3D$ for the network problem.

By being more careful, we can create a schedule of length $2D$ for the network problem. In schedule $\mathcal{S}$, each machine M_i is assigned a set of jobs S_i. Let $|S_i|$ be the sum of the processing times of the jobs in S_i and let $S_i^{\max}$ be the job in S_i with largest processing time on machine i; call its processing time $p_i^{\max}$. By Theorem 2.1 and the fact that the last job run on machine i is no longer than the longest job run, we know that $|S_i| - p_i^{\max} \leq D$. Let $S_i^{'}$ denote the set of jobs $S_i - S_i^{\max}$. We form the schedule for each machine i by running job $S_i^{\max}$ at time $D - p_i^{\max}$, followed by the jobs in $S_i^{'}$.

In this schedule the jobs assigned to any machine clearly finish by time $2D$; it remains to be shown that all jobs can be routed to the proper machines by the time they need to run there. Job $S_i^{\max}$ must start at time $D - p_i^{\max}$; conditions (1) and (2) guarantee that it arrives in time. The remaining jobs need only arrive by time D; conditions (1) and (2) guarantee this as well. Thus we have produced a valid schedule of length $2D$. □

Observe that this approach is fairly general and can be applied to any problem that can be characterized by a condition such as (2). Consider, for example the following very general problem, which we call *generalized network scheduling with costs.* In addition to the usual unrelated network scheduling problem, the time that it takes for job J_j to travel over an edge is dependent not only on the endpoints of the edge but also on the job. Further, there is a cost c_{ij} associated with processing job J_j on machine M_i. Given a schedule in which job J_j runs on machine $M_{\pi(j)}$, the cost of a schedule is $\sum_j c_{\pi(j),j}$. Given any target cost C, we define $s(C)$ to be the minimum length schedule of cost at most C.

THEOREM 2.3. *Given a target cost C, we can, in polynomial time, find a schedule for the generalized network scheduling problem with makespan at most $2s(C)$ and of cost C if a schedule of cost C exists.*

Proof. We use similar techniques to those used for Theorem 2.2. We first modify condition (1) so that $d(\cdot,\cdot)$ depends on the job as well. We then use a generalization of the algorithm of Lenstra, Shmoys, and Tardos for unrelated machine scheduling, due to Shmoys and Tardos [25] which, given a target cost C, finds a schedule of cost C and length at most twice that of the shortest schedule of cost C. The schedule returned also has the property that no job starts after time D, so the proof of Theorem 2.2 goes through if we use this algorithm in place of the algorithm of [17]. □

2.2. Nonapproximability.

THEOREM 2.4. *It is $\mathcal{NP}$-complete to determine if an instance of the identical network scheduling problem has a schedule of length 3, even in a network with $\ell_{\max} = 1$.*

Proof. For the proof see the appendix. □

COROLLARY 2.5. *There does not exist an α-approximation algorithm for the network scheduling problem with $\alpha < 4/3$ unless $\mathcal{P} = \mathcal{NP}$, even in a network with $\ell_{\max} = 1$.*

Proof. Any algorithm with $\alpha < 4/3$ would have to give an exact answer for a problem with a schedule of length 3 since an approximation of 4 would have too high a relative error. □

It is not hard to see, via matching techniques, that it is polynomial-time decidable whether there is a schedule of length 2. We can show that this is not the case when the machines in the network can be unrelated. Lenstra, Shmoys, and Tardos proved that it is $\mathcal{NP}$-complete to determine if there is a schedule of length 2 in the traditional combinatorial unrelated machine model [17]. If we allow multiple machines at one node, their proof proves Theorem 2.6. If no zero length edges are allowed, i.e., each machine is forced to be at a different network node, this proof does not work, but we can give a different proof of hardness, which we do not include in this paper.

THEOREM 2.6. *There does not exist an α-approximation algorithm for the unrelated network scheduling problem with $\alpha < 3/2$ unless $\mathcal{P} = \mathcal{NP}$, even in a network with $\ell_{\max} = 1$.*

2.3. Naive strategies. The algorithms in section 2.1 give reasonably tight bounds on the approximation of the schedule length. Although these algorithms run in polynomial time, they may be rather slow [21]. We thus explore whether a simpler strategy might also yield good approximations.

A natural candidate is a *busy* strategy: construct a *busy schedule*, in which, at any time t there is no idle machine M_i and idle job J_j so that job J_j can be started on M_i at time t. Busy strategies and their variants have been analyzed in a large number of scheduling problems (see [15]) and have been quite effective in many of them. For combinatorial identical machine scheduling, Graham showed that such strategies yield a $(2 - \frac{1}{m})$ approximation guarantee [6]. In this section we analyze the effectiveness of busy schedules for identical machine network scheduling. Part of the interest of this analysis lies in what it reveals about the relative hardness of scheduling with and without an underlying network; namely, the introduction of an underlying network can make simple strategies much less effective for the problem.

2.3.1. A lower bound. We construct a family of instances of the network scheduling problem, and demonstrate, for each instance, a busy schedule which is $\Omega\left(\sqrt{\frac{\log m}{\log\log m}}\right)$ longer than the shortest schedule for that instance. The network $G = (V, E)$ consists of ℓ levels of nodes, with level $i, 1 \le i \le \ell$, containing ρ^{i-1} nodes. Each node in level $i, 1 \le i < \ell - 1$, is connected to every node in level $i + 1$ by an edge of length 1. Each machine in levels $1, \dots, \ell - 1$ receives ρ jobs of size 1 at time 0. The machines in level ℓ initially receive no jobs. The optimal schedule length for this instance is 2 and is achieved by each machine in level $i, 2 \le i \le \ell$, taking exactly one job from level $i - 1$. We call this instance $\mathcal{I}$; see Figure 1.

The main idea of the lower bound is to construct a busy schedule in which machine M always processes a job which originated on M, if such a job is available. This greediness "prevents" the scheduler from making the much larger assignment of jobs to machines at time 2 in which each job is assigned to a machine one level away.

To construct a busy schedule S, we use algorithm $\mathcal{B}$, which in Step t constructs the subschedule of S at time t.

Step t:

Phase 1*:* Each machine M processes one job that originated at M, if any such jobs remain. We call such jobs *local* to machine M.

Phase 2*:* Consider the bipartite graph $G^* = (X, Y, A)$, where X has one vertex representing each job that is unprocessed after Phase 1 of time t, Y contains one vertex representing each machine which has not had a job assigned to it in Phase 1 of Step t, and $(x, y) \in A$ if and only if job x originated a distance no more than $t - 1$ from machine y. Complete the construction of S at time t by processing jobs on machines based on any maximum matching in G^*. It is clear that S is busy.

When we apply algorithm $\mathcal{B}$ to instance $\mathcal{I}$, the behavior follows a well-defined

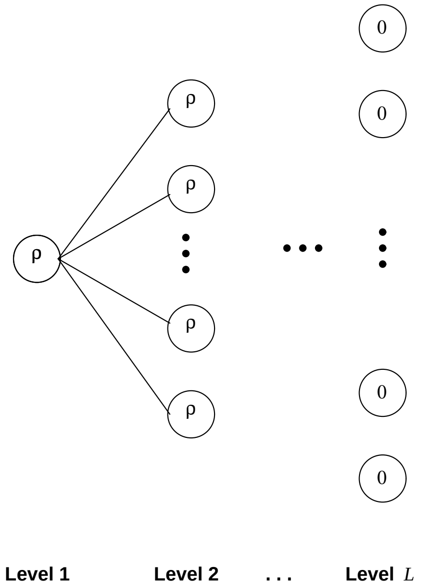


FIG. 1. *Lower bound instance for Theorem* 2.8. *Circles represent processors, and the numbers inside the circles are the number of jobs which originate at that processor at time* 0. *Levels i and i + 1 are completely connected to each other. The optimal schedule is of length* 2 *and is achieved by shifting each job to a unique processor one level to its right.*

pattern. In Phase 2 of Step 2, all unprocessed jobs that originated in level $\ell - 1$ are processed by distinct processors in level ℓ. During Phase 2 of Step 3, all unprocessed jobs that originated in levels $\ell - 2$ and $\ell - 3$ are processed by machines in levels $\ell - 1$ and ℓ. This continues, so that at Step i an additional $(i - 1)$ levels pass their jobs to higher levels and all these jobs are processed. This continues until either level 1 passes its jobs, or processes its own jobs. We characterize the behavior of the algorithm more formally in the following lemma.

LEMMA 2.7. *Let $j(i, t)$ be the number of local jobs of processor i still unprocessed after Phase* 2 *of Step t and let $lev(i)$ be the level number of processor i. Then for all times $t \geq 2$, if $\rho \geq t$, then*

$$j(i,t) = \begin{cases} 0 & \text{if } lev(i) \geq \ell - t(t-1)/2, \\ j(i,t-1) - 1 & \text{otherwise.} \end{cases} \tag{3}$$

Proof. We prove the lemma by induction on t. During Phase 2 of Step 2, the only edges in the graph G^* connect levels ℓ and $\ell - 1$. There are $\rho^{\ell-1}$ nodes in level ℓ and $\rho^{\ell-2}(\rho - 1)$ remaining jobs local to machines in level $\ell - 1$, so the matching assigns all the unprocessed jobs in level $\ell - 1$ to level ℓ. Machines in levels 1 to $\ell - 1$ all process local jobs during Phase 1. As a result, all the neighbors of machines in levels 1 to $\ell - 2$ are busy in Phase 1 and cannot process jobs local to these machines during Phase 2. The number of local jobs on these machines, therefore, decreases only by 1. Thus the base case holds.

Assume the lemma holds for all $t < t'$. Then $j(i, t' - 1) = 0$ for levels greater than $b \equiv \ell - (t' - 1)(t' - 2)/2$, and $j(i, t') = 0$ for levels greater than b as well. We now show that $j(i, t') = 0$ if $lev(i) \geq \ell - t'(t' - 1)/2$. For $1 \leq x \leq t' - 1$, level $b + x$ has ρ^{b+x-1} processors. Level $b + x - (t' - 1)$ has at most $\rho \cdot \rho^{b+x-(t'-1)-1} = \rho^{b+x-t'+1}$ local jobs remaining. If $t' \geq 2$, then there are enough machines on level $b + x$ to process all the remaining jobs local to level $b + x - (t' - 1)$. Therefore another $t' - 1$ of the highest-numbered levels have their local jobs completed during time t'. Thus at time t' we have $j(i, t') = 0$ if $lev(i) \geq \ell - t'(t' - 1)/2$.

Since we assumed sufficiently large initial workloads on all processors on levels $1 \ldots (\ell - 1)$, then by the induction hypothesis, for all machines in levels less than $\ell - t'(t' - 1)/2$, all machines within distance $t' - 1$ of them have local jobs remaining after time $t' - 1$ and will be assigned a local job during Phase 1 of Step t'. Therefore all machines i such that $lev(i) < \ell - t'(t' - 1)/2$ cannot pass any jobs to higher levels and $j(i, t') = j(i, t' - 1) - 1$. □

Depending on the relative values of ρ and ℓ, either the machine in level 1 processes all of the jobs which originated on it, or some of those jobs are processed by machines in higher-numbered levels. Balancing these two cases we get the following theorem.

THEOREM 2.8. *For the family of instances of the identical machine network scheduling problem defined above, there exist busy schedules whose length exceeds the optimal length by a factor* $\Omega\left(\sqrt{\frac{\log m}{\log\log m}}\right)$.

Proof. The first case in (3) will apply to level 1 when $1 \geq \ell - t(t - 1)/2$. This inequality does not hold when $t = \sqrt{2\ell}$, but it does hold when $t = \sqrt{2\ell} + 1$. Thus, if $\rho > \sqrt{2\ell}$ then the schedule length is $\sqrt{2\ell}$, while if $\rho < \sqrt{2\ell}$ then the jobs in level 1 will be totally processed in their level, which takes ρ time. Therefore the makespan of S is at most $\min(\sqrt{2\ell}, \rho)$. Given that the total number of machines is $m = \theta(\rho^{\ell-1})$, a simple calculation reveals that $\min(c\sqrt{\ell}, \rho)$ is maximized at $\ell = \theta(\frac{\log m}{\log\log m})$. Thus S is a busy schedule of length $\theta\left(\sqrt{\frac{\log m}{\log\log m}}\right)$ longer than optimal. □

Note that this example shows that several natural variants of busy strategies, such as scheduling a job on the machine on which it will finish first, or scheduling a job on the closest available processor, also perform poorly.

2.3.2. An upper bound. In contrast to the lower bound of the previous subsection, we can prove that busy schedules are of some quality. Given an instance $\mathcal{I}$ of the network scheduling problem, we define $C^*_{\max}(\mathcal{I})$ to be the length of a shortest schedule for $\mathcal{I}$ and $C^A_{\max}(\mathcal{I})$ to be the length of the schedule produced by algorithm A; when it causes no confusion we will drop the $\mathcal{I}$ and use the notation $C^*_{\max}$.

DEFINITION 2.9. *Consider a busy schedule S for an instance $\mathcal{I}$ of the identical machines network scheduling problem. Let $p_j(t)$ be the number of units of job J_j remaining to be processed in schedule S at time t, and $W_t = \sum_{k=1}^{j} p_k(t)$ be the total work remaining to be processed in schedule S at time t.*

LEMMA 2.10. $W_{iC^*_{\max}} \leq \frac{W_0}{2i!}$ *for* $i \geq 1$.

Proof. We partition schedule S into consecutive blocks $B_1, B_2, \ldots$ of length $C^*_{\max}(\mathcal{I})$ and compare what happens in each block of schedule S to an optimal schedule S^* of length $C^*_{\max}$ for instance $\mathcal{I}$.

Consider a job J_j that was not started by time $C^*_{\max}$ in schedule S, and let M_j be the machine on which job J_j is processed in schedule S^*. This means that in block B_1 machine M_j is busy for p_j units of time during job J_j's *slot* in schedule S^*—the period of time during which job J_j was processed on machine M_j in schedule S^*. Hence for every job J_j that is not started in block B_1 there is an equal amount of unique work which we can identify that *is* processed in block B_1, implying that $W_{C^*_{\max}} \leq W_0/2$. Successive applications of this argument yields $W_{iC^*_{\max}} \leq W_0/2^i$ for $i \geq 1$, which proves the lemma for $i = 1, 2$.

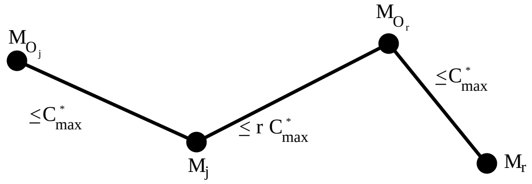


FIG. 2. *If J_r takes J_j's slot in B_r, then the machine on which J_j originates, M_{o_j}, is at most a distance of $(r+2)C^*_{\max}$ from M_r, the machine on which J_r runs in S^*. Thus J_j could have been run in J_r's slot in block i, $i \ge (r+2)$.*

To obtain the stronger bound $W_{iC^*_{\max}} \le \frac{1}{2}(W_0/i!)$, we increase the amount of processed work which we identify with each unstarted job. Choose $i \ge 3$ and consider a job J_j which is unstarted in schedule S at the start of block B_{i+1}, namely at time $iC^*_{\max}$. Assume for the sake of simplicity that in every block B_k of schedule S, only one job is processed in job J_j's slot (the time during which job J_j would be processed if block B_k was schedule S^*). Assume also that this job is exactly of the same size as job J_j; if multiple jobs are processed the argument is essentially the same. Let J_r be the job that took job J_j's slot in block B_r, for $r \le i-2$. We will show that J_j could have been processed in J_r's slot in block B_i for all $1 \le r \le i-2$. Figure 2 illustrates the network structure used in this argument.

Assume that job J_j originated on machine M_{o_j}, that job J_r originated on machine M_{o_r}, and that job J_j was processed on machine M_j in schedule S^*. Then $d(M_{o_j}, M_j) \le C^*_{\max}$ since job J_j was processed on machine M_j in schedule S^*, and $d(M_{o_r}, M_j) \le rC^*_{\max}$ since job J_r was processed in job J_j's slot in block B_r. Thus $d(M_{o_j}, M_{o_r}) \le (r+1)C^*_{\max}$ and consequently J_j could have run in job J_r's slot in any of blocks $B_{r+2}, \ldots, B_i$. We focus on block B_i. Since J_j was not processed in block B_i and schedule S is busy, some job must have been processed during job J_r's slot in block B_i for $1 \le r \le (i-2)$. We identify this work with job J_j ; note that no work is ever identified with more than one job.

When we consider the $(i-2)$ different jobs which were processed in J_j's slot in blocks $B_1, \ldots, B_{i-2}$, and consider the jobs that were processed in their slots in B_i , we see that with each job J_j unstarted at time $iC^*_{\max}$, we can uniquely identify $(i-2)p_j$ units of work that was processed in block B_i. If all these slots were not full in block B_i, then job J_j would have been started in one of them. Including the work processed during job J_j's slot in block B_i, we obtain

$$W_{iC^*_{\max}} \le \frac{1}{i} W_{(i-1)C^*_{\max}}. \qquad \square$$

COROLLARY 2.11. *During time $iC^*_{\max}$ to $(i+1)C^*_{\max}$ at most $m/(2i!)$ machines are completely busy.*

Proof. We have $W_0 \le mC^*_{\max}$. Therefore, by Lemma 2.10, we have $W_{iC^*_{\max}} \le mC^*_{\max}/(2i!)$. A machine that is completely busy from time $iC^*_{\max}$ to time $(i+1)C^*_{\max}$ does $C^*_{\max}$ work during that time and therefore at most $m/(2i!)$ machines can be completely busy. □

To get a stopping point for the recurrence, we require the following lemma.

LEMMA 2.12. *In any busy schedule, if at time t all remaining unprocessed jobs originated on the same machine, the schedule is no longer than $t + 2C^*_{\max}$.*

Proof. Let M be the one machine with remaining local jobs. Let $W^*_{M_i}$ be the amount of work from machine M that is done by machine M_i in the optimal schedule.

Clearly $\sum_i W^*_{M_i}$ equals the amount of work that originated on machine M. Because there is no work left that originated on machines other than M, each machine M_i can process at least $W^*_{M_i}$ work from machine M in the next $C^*_{\max}$ time steps. If after $C^*_{\max}$ steps, all the work originating on machine M is done, then we have finished. Otherwise, some machine M_i processed less than $W^*_{M_i}$ work during this time, which means there was no more work for it to take. Therefore after $C^*_{\max}$ steps all the jobs that originated on machine M have started. Because no job is longer than $C^*_{\max}$, another $C^*_{\max}$ time suffices to finish all the jobs that have started. □

We are now ready to prove the upper bound.

Theorem 2.13. *Let A be any busy scheduling algorithm and $\mathcal{I}$ an instance of the identical machine network scheduling problem. Then $C^A_{\max}(\mathcal{I}) = O(\frac{\log m}{\log\log m} C^*_{\max}(\mathcal{I}))$.*

Proof. If a machine ever falls idle, all of its local work must be started. Otherwise it would process remaining local work. Thus by Corollary 2.11, in $O(\frac{\lg m}{\lg \lg m})C^*_{\max}$ time, the number of processors with local work remaining is reduced to 1. By Lemma 2.12, when the number of processors with remaining local work is down to one, a constant number of extra blocks suffice to finish. □

2.4. Scheduling with origins and destinations. In this subsection we consider a variant of the (unrelated machine) network scheduling problem in which each job, after being processed, has a *destination machine* to which it must travel. Specifically, in addition to having an origin machine M_{o_j}, job J_j also has a terminating machine M_{t_j}. Job J_j begins at machine M_{o_j}, travels distance $d(M_{o_j}, M_{d_j})$ to machine M_{d_j}, the machine it gets processed on, and then proceeds to travel for $d(M_{d_j}, M_{t_j})$ units of time to machine M_{t_j}. We call this problem the *point-to-point scheduling problem.*

Theorem 2.14. *There exists a polynomial-time $\frac{5}{2}$-approximation algorithm to minimize makespan in the point-to-point scheduling problem.*

Proof. We construct an unrelated machines scheduling problem as in the proof of Theorem 2.2. In this setting the condition on when a job J_j can run on machine M_i depends on the time for J_j to get to M_i, the time to be processed there, and the time to proceed to the destination machine. Thus a characterization of when job J_j is able to run on machine M_i in the optimal schedule is that

$$d(M_{o_j}, M_i) + p_{ij} + d(M_i, M_{t_j}) \leq D. \tag{4}$$

Now, for a given job J_j, we define $Q(J_j)$ to be the set of machines that satisfy (4). We can then form a combinatorial unrelated machines scheduling problem as follows:

$$p'_{ij} = \begin{cases} p_{ij} & \text{if } M_i \in Q(J_j), \\ \infty & \text{otherwise.} \end{cases} \tag{5}$$

We then approximately solve this problem using [17] to obtain an assignment of jobs to machines. Pick any machine M_i and let $\mathcal{J}_i$ be the set of jobs assigned to machine M_i. By Theorem 2.1 we know that the sum of the processing times of all of the jobs in $\mathcal{J}_i$ except the longest is at most D. We partition the set of jobs $\mathcal{J}_i$ into three groups, and place each job into the lowest numbered group which is appropriate:

1. $\mathcal{J}^0_i$ contains the job in $\mathcal{J}_i$ with the longest processing time,
2. $\mathcal{J}^1_i$ contains jobs for which $d(M_{o_j}, M_i) \leq D/2$,
3. $\mathcal{J}^2_i$ contains jobs for which $d(M_{o_j}, M_i) \geq D/2$.

Let $p(\mathcal{J}^k_i)$ be the sum of the processing times of the jobs in group $\mathcal{J}^k_i$, $k = 1, 2$. As noted above, $p(\mathcal{J}^1_i) + p(\mathcal{J}^2_i) \leq D$. We will always schedule $\mathcal{J}^1_i \cup \mathcal{J}^2_i$ in a block of

D consecutive time steps, which we call B. The first $p(\mathcal{J}_i^1)$ time steps will be taken up by jobs in $\mathcal{J}_i^1$ while the last $p(\mathcal{J}_i^2)$ time steps will be taken up by jobs in $\mathcal{J}_i^2$. Note that there may be idle time in the interior of the block.

We consider two possible scheduling strategies based on the relative sizes of $p(\mathcal{J}_i^1)$ and $p(\mathcal{J}_i^2)$.

Case 1. ($p(\mathcal{J}_i^1) \leq p(\mathcal{J}_i^2)$). In this case we first run the long job in $\mathcal{J}_i^0$; by condition (4) it finishes by time D. We then run block B from time D to $2D$. Since $p(\mathcal{J}_i^1) \leq D/2$, the jobs in $\mathcal{J}_i^1$ all finish by time $3D/2$ and by condition (4) reach their destinations by time $5D/2$. By the definition of $\mathcal{J}_i^2$, for any job $J_j \in \mathcal{J}_i^2, d(M_i, M_{t_j}) \leq D/2$. Since every $J_j \in \mathcal{J}_i^2$ is scheduled to complete processing by time $2D$, it will arrive at its destination by time $5D/2$.

Case 2. ($p(\mathcal{J}_i^1) \geq p(\mathcal{J}_i^2)$). We first run block B from time $D/2$ to $3D/2$. We then start the long job in $\mathcal{J}_i^0$ at time $3D/2$; by condition (4) it arrives at its destination by time $5D/2$. Since $p(\mathcal{J}_i^2) \leq D/2$, machine M_i need not start processing any job in $\mathcal{J}_i^2$ until time D and hence we are guaranteed that they have arrived at machine M_i by that time. By definition of $\mathcal{J}_i^1$ all of its jobs are available by time $D/2$; it is straightforward from condition (4) that all jobs arrive at their destinations by time $5D/2$. □

We can also show that the analysis of this algorithm is tight, for algorithms in which we assign jobs to processors using the linear program defined in [17] using the processing times specified by equation 5. Let D be the length of the optimal schedule. Then we can construct instances for which any such schedule S has length at least $5/2D - 1$. Consider a set of $k + 1$ jobs and a particular machine M_i. We specify the largest of these jobs to have size D and to have M_i as both its origin and its destination machine. We specify that each of the other k jobs are of size D/k and have distance $D(k - 1)/2k$ from M_i to both their origin and destination machines. The combinatorial unrelated machines algorithm may certainly assign all of these jobs to M_i, but it is clear that any schedule adopted for this machine will have completion time at least $(\frac{5}{2} - \frac{1}{2k})D$.

3. Average completion time.

3.1. Background.

We turn now to the network scheduling problem in which the objective is to minimize the average completion time. Given a schedule S, let C_j^S be the time that job J_j finishes running in S. The average completion time of S is $\frac{1}{n}\sum_j C_j^S$, whose minimization is equivalent to the minimization of $\sum_j C_j^S$. Throughout this section we assume without loss of generality that $n \geq m$.

We have noted in section 1 that our network scheduling model can be characterized by a set of n jobs J_j and a set of release dates r_{ij}, where J_j is not available on m_i until time r_{ij}. We noted that this is a generalization of the traditional notion of release dates, in which $r_{ij} = r_{i'j}\ \forall i, i'$. We will refer to the latter as *traditional* release dates; the unmodified phrase *release date* will refer to the general r_{ij}.

The minimization of average completion time when the jobs have no release dates is polynomial-time solvable [3, 12], even on unrelated machines. The solution is based on a bipartite matching formulation, in which one side of the bipartition has jobs and the other side (machine, position) pairs. Matching J_j to (m_i, k) corresponds to scheduling J_j in the *kth-from-last* position on m_i; this edge is weighted by kp_{ij}, which is J_j's contribution to the average completion time if J_j is kth from last.

When release dates are incorporated into the scheduling model, it seems difficult to generalize this formulation. Clearly it can not be generalized precisely for arbitrary release dates, since even the one machine version of the problem of minimizing average completion time of jobs with release dates is strongly $\mathcal{NP}$-hard [3]. Intuitively, even the approximate generalization of the formulation seems difficult, since if all jobs are not available at time 0, the ability of J_j to occupy position k on m_i is dependent on which jobs precede it on m_i and when. Release dates associated with a network structure do not contain traditional release dates as a subclass even for one machine, so the $\mathcal{NP}$-completeness of the network scheduling problem does not follow immediately from the combinatorial hardness results; however, not surprisingly, minimizing average completion time for a network scheduling problem is $\mathcal{NP}$-complete.

THEOREM 3.1. *The network scheduling problem with the objective of minimum average completion time is $\mathcal{NP}$-complete even if all the machines are identical and all edge lengths are* 1.

Proof. For the proof see the appendix. □

In what follows we will develop an approximation algorithm for the most general form of this problem. We will follow the basic idea of utilizing a bipartite matching formulation; however we will need to explicitly incorporate time into the formulation. In addition, for the rest of the section we will consider a more general optimality criterion: average *weighted* completion time. With each J_j we associate a weight w_j, and the goal is to minimize $\sum_{j=1}^{j=n} w_j C_j$. All of our algorithms handle this more general case; in addition they allow the nm release dates r_{ij} to be arbitrary and not necessarily derived from the network structure.

3.2. Unit-size jobs. We consider first the special case of unit-size jobs.

THEOREM 3.2. *There exists a polynomial-time algorithm to schedule unit-size jobs on a network of identical machines with the objective of minimizing the average weighted completion time.*

Proof. We reduce the problem to minimum-weight bipartite matching. One side of the bipartition will have a node for each job J_j, $1 \leq j \leq n$, and the other side will have a node $[m_i, t]$ for $1 \leq i \leq m$, $t \in T_i$ with T_i to be described below. An edge $(J_j, [m_i, t])$ of weight $w_j(t+1)$ is included if J_j is available on m_i at time t, and the inclusion of that edge in the matching will represent the scheduling of J_j on m_i from time t to $t+1$. Release dates are included in the model by excluding an edge $(J_j, [m_i, t])$ if J_j will not be available on m_i by time t.

To determine the necessary sets T_i, we observe that there is no advantage in unforced idle time. Since each job is only one unit long, there is no reason to make it wait for a job of higher weight that is about to be released. It is clear, therefore, that setting $T_i = \{t | r_{ij} \leq t \leq r_{ij} + n \ \forall j\}$ would suffice, since no job would need to be scheduled more than n time later than its release date. This gives $|T_i| = O(n^2)$; this can be reduced to $O(n)$, but we omit the details for the sake of brevity. □

By excluding edges which do not give job J_j enough time to travel between the machine on which J_j runs and the destination machine M_{d_j}, we can prove a similar theorem for the point-to-point scheduling problem, defined in section 2.4.

THEOREM 3.3. *There exists a polynomial-time algorithm to solve the point-to-point scheduling problem with the objective of minimizing the average weighted completion time of unit-size jobs.*

3.3. Polynomial-size jobs. We now turn to the more difficult setting of jobs of different sizes and unrelated machines. The minimization of average weighted completion time in this setting is strongly $\mathcal{NP}$-hard, as are many special cases. For example, the minimization of average completion time of jobs with release dates on one machine is strongly $\mathcal{NP}$-hard [16]; no approximation algorithms were known for this special case, to say nothing of parallel identical or unrelated machines, or weighted completion times. If there are no release dates, namely all jobs are available at time 0, then minimization of average weighted completion time is $\mathcal{NP}$-hard for parallel identical machines. A small constant factor approximation algorithm was known for this problem [14], but no approximation algorithms were known for the more general cases of machines of different speeds or unrelated machines. We introduce techniques which yield the first approximation algorithms for several other problems as well, which we discuss in section 3.5.

Our approximation algorithm for minimum average completion time begins by formulating the scheduling problem as a *hypergraph matching problem.* The set of vertices will be the union of two sets, J and M, and the set of hyperedges will be denoted by F. J will contain n vertices J_j, one for each job, and M will contain mT vertices, where T is an upper bound on the number of time units that will be needed to schedule this instance. The time units will range over $\mathcal{T} = \{t | \exists r_{ij} \text{ with } r_{ij} \leq t \leq r_{ij} + np_{\max}\}$. M will have a node for each (machine, time) pair; we will denote the node that corresponds to machine M_i at time t as $[m_i, t]$. A hyperedge $e \in F$ represents scheduling a job J_j on machine M_i from time t_1 to t_2 by including nodes $J_j, [m_i, t_1], [m_i, t_1 + 1], \ldots, [m_i, t_2]$. The cost of an edge e, denoted by c_e, will be the weighted completion time of job J_j if it is scheduled in the manner represented by e. There will be one edge in the hypergraph for each feasible scheduling of a job on a machine; we exclude edges that would violate the release date constraints.

The problem of finding the minimum cost matching in the hypergraph can be phrased as the following integer program $\mathcal{I}$. We use decision variable $x_e \in \{0, 1\}$ to denote whether hyperedge e is in the matching.

$$\begin{aligned} &\text{minimize} \sum_e x_e c_e \\ &\qquad\text{subject to} \\ &\qquad\qquad \sum_{J_j \in e} x_e = 1, \qquad j = 1, \ldots, n, \\ (6) \qquad &\qquad\qquad \sum_{(i,t) \in e} x_e \leq 1 \qquad \forall (i, t) \in M, \\ &\qquad\qquad x_e \in \{0, 1\}. \end{aligned}$$

Two considerations suggest that this formulation might not be useful. The formulation is not of polynomial size in the input size, and in addition the following theorem suggests that calculating approximate solutions for this integer program may be difficult.

THEOREM 3.4. *Consider an integer program in the form $\mathcal{I}$ which is derived from an instance of the network scheduling problem with identical machines, with the c_e allowed to be arbitrary. Then there exists no polynomial-time algorithm $\mathcal{A}$ to approximate $\mathcal{I}$ within any factor unless $\mathcal{P} = \mathcal{NP}$.*

Proof. For an arbitrary instance of the network scheduling problem construct the hypergraph matching problem in which an edge has weight $W >> n$ if it corresponds to a job being completed later than time 3 and give all other edges weight 1. If there is a schedule of length 3 then the minimum weight hypergraph matching is of weight n; otherwise the weight is at least W; therefore an α-approximation algorithm with

$\alpha < \frac{W}{n}$ would give a polynomial-time algorithm to decide if there was a schedule of length 3 for the network scheduling problem, which by Theorem 2.4 would imply $\mathcal{P} = \mathcal{NP}$. □

In order to overcome this obstacle, we need to seek a different kind of approximation to the hypergraph matching problem. Typically, an approximate solution is a feasible solution, i.e., one that satisfies all the constraints, but whose objective value is not the best possible. We will look for a different type of solution, one that satisfies a *relaxed* set of constraints. We will then show how to turn a solution that satisfies the relaxed set of constraints into a schedule for the network scheduling problem, while only introducing a bounded amount of error into the quality of the approximation.

We will assume for now that $p_{\max} \leq n^3$. This implies that the size of program $\mathcal{I}$ is polynomial in the input size. We will later show how to dispense with the assumption on the size of $p_{\max}$ via a number of rounding and scaling techniques.

We begin by turning the objective function of $\mathcal{I}$ into a constraint. We will then use the standard technique of applying bisection search to the value of the objective function. Hence for the remainder of this section we will assume that C, the optimal value to integer program $\mathcal{I}$, is given. We can now construct approximate solutions to the following integer linear program ($\mathcal{J}$):

$$\sum_{J_j \in e} x_e \quad = \quad 1, \qquad j = 1, \ldots, n, \tag{7}$$

$$\sum_{(i,t) \in e} x_e \quad \leq \quad 1 \qquad \forall (i,t) \in M, \tag{8}$$

$$\sum_{e} x_e c_e \quad \leq \quad C, \tag{9}$$

$$x_e \quad \in \quad \{0,1\}.$$

This integer program is a packing integer program, and as has been shown by Raghavan [22], Raghavan and Thompson [23] and Plotkin, Shmoys, and Tardos [21], it is possible to find provably good approximate solutions in polynomial time. We briefly review the approach of [21], which yields the best running times.

Plotkin, Shmoys, and Tardos [21] consider the following general problem.

The Packing Problem: $\exists ? x \in P$ such that $Ax \leq b$, where A is an $m \times n$ nonnegative matrix, $b > 0$, and P is a convex set in the positive orthant of R^n.

They demonstrate fast algorithms that yield approximately optimal integral solutions to this linear program. All of their algorithms require a fast subroutine to solve the following problem.

The Separation Problem: Given an m-dimensional vector $y \geq 0$, find $\tilde{x} \in P$ such that $c\tilde{x} = \min(cx : x \in P)$, where $c = y^t A$.

The subroutine to solve this problem will be called the *separating subroutine.*

An approximate solution to the packing problem is found by considering the relaxed problem

$$\exists ? x \in P \text{ such that} Ax \leq \lambda b$$

and approximating the minimum λ such that this is true. Here the value λ characterizes the "slack" in the inequality constraints, and the goal is to minimize this slack.

Our integer program can be easily put in the form of a packing problem; the equality constraints (7) define the polytope P and the inequality constraints (8,9) make up $Ax \leq b$. The quality of the integral solutions obtained depends on the *width* of P relative to $Ax \leq b$, which is defined by

$$\rho = \max_i \max_{x \in P} \frac{a_i x}{b_i}. \tag{10}$$

It also depends on d, where d is the smallest integer such that any solution returned by the separating routine is guaranteed to be an integral multiple of $\frac{1}{d}$.

Applying equation (10) to compute ρ for polytope P (defined by (7)) yields a value that is at least n, as we can create matchings (feasible schedules) whose cost (average completion time) is much greater than C, the optimal average completion time.

In fact, many other packing integer programs considered in [21] also, when first formulated, have large width. In order to overcome this obstacle, [21] gave several techniques to reduce the width of integer linear programs. We discuss and then use one such technique here, namely that of decomposing a polytope into n lower-dimensional polytopes, each of which has smaller width. The intuition is that all the nonzero variables in each equation of the form (7) are associated with only one particular job. Thus we will be able to decompose the polytope into n polytopes, one for each job. We will then be able to optimize individually over each polytope and use only the inequality constraints (8) and (9) to describe the relationships between different jobs.

We now proceed in more detail. We say that a polytope P can be decomposed into a product of n polytopes $P^1 \times P^2 \times \cdots \times P^n$ if the coordinates of each vector x can be partitioned into $(x^1, \ldots, x^n)$, and $x \in P$ if and only if $x^l \in P^l$ for $l = 1, \ldots, n$. If our polytope can be decomposed in this way, and we can solve the separation problem for each polytope P^l, then we can apply a theorem of [21] to give an approximately optimal solution in polynomial time. In particular, let λ^* be the minimum possible value of λ for which there exists a feasible solution to the relaxed version of $\mathcal{J}$. The following theorem is a specialization of Theorem 2.11 in [21] to our problem and describes the quality of integral solutions that can be obtained for such integer programs.

THEOREM 3.5 (see [21]). *Let ρ^l be the width of P^l and $\bar{\rho} = \max_l \rho^l$. Let γ be the number of constraints in $Ax \leq b$, and let $\lambda' = \max(\lambda^*, (\bar{\rho}/d)\log\gamma)$. Given a polynomial-time separating subroutine for each of the P^l, there exists a polynomial-time algorithm for $\mathcal{J}$ which gives an integral solution with $\lambda \leq \lambda^* + O\left(\sqrt{\lambda'(\bar{\rho}/d)\log(\gamma n d)}\right)$.*

We will now show how to reformulate $\mathcal{J}$ so that we will be able to apply this theorem. Polytope P (from equation 6) can indeed be decomposed into n different polytopes, $P^1, P^2, \ldots, P^n$, where P^j corresponds to those equality constraints which include only J_j. In order to keep the width of the P^j small, we also include into the definition of P^j the constraint $x_e = 0$ for each edge e which includes J_j and has $c_e > C$; this does not increase the optimal value of the integer program. We integrate each of these new constraints into the appropriate polytope P^j, and decompose $x = (x^1, x^2, \ldots, x^n)$, where x^j consists of those components of x which represent edges that include J_j. In other words, P^l is defined by

$$\begin{aligned} \sum_{J_l \in e} x_e &= 1, & \\ x_e &= 0 & \qquad \text{if } c_e > C \text{ and } J_l \in e. \end{aligned}$$

This yields the following relaxation $\mathcal{L}$:

minimize λ

subject to

$$\begin{aligned} x^l &\in P^l, && 1 \le l \le n, \\ \sum_{(i,t)\in e} x_e &\le \lambda && \forall (i,t) \in M, \qquad (11) \\ \sum_e x_e c_e &\le \lambda C, && \qquad (12) \\ x &= (x^1, x^2, \ldots, x^n) \in \{0,1\}^{|F|}. && \qquad (13) \end{aligned}$$

To apply Theorem 3.5 we must (1) demonstrate a polynomial-time separating subroutine and (2) calculate $\bar{\rho}$, d and γ. The decomposition of P into n separate polytopes makes this task much easier. The separating subroutine must find $x^l \in P^l$ that minimizes cx^l; however, since the vector that is 1 in the eth component and 0 in all other components is in P^l for all e such that $J_l \in e$ and $c_e \le C$, the separating routine reduces merely to finding the minimum component $c_{e'}$ of c and returning the vector with a 1 in position e' and 0 everywhere else. An immediate consequence of this is that $d = 1$. Recall as well that the assumption that $p_{\max} \le n^3$ implies that γ is upper bounded by a polynomial in n.

To compute $\bar{\rho}$, recall that we compute $\bar{\rho}$ relative to the polytope defined by $\sum_{(i,t)\in e} x_e \le 1$ and $\sum_e x_e c_e \le C$, as the relaxed versions of these constraints appear in (11) and (12) above. It is thus not hard to see that $\bar{\rho}$ is 1 and therefore

$$\lambda \le \lambda^* + O\left(\sqrt{(\bar{\rho}/d)\log\gamma(\bar{\rho}/d)\log(\gamma n d)}\right)$$

$$\le 1 + O(\log n) = O(\log n).$$

By employing binary search over C and the knowledge that the optimal solution has $\lambda = 1$, we can obtain an invalid "schedule" in which as many as $O(\lambda)$ jobs are scheduled at one time. If $p_{\max}$ is polynomial in n and m then we have a polynomial-time algorithm; therefore we have proven the following lemma.

LEMMA 3.6. *Let C^* be the solution to the integer program $\mathcal{I}$ and assume that $|M|$ is bounded by mn^4. There exists a polynomial-time algorithm that produces a solution x^* such that*

$$\begin{aligned} \sum_{j\in e} x_e^* &= 1, && j = 1, \ldots, n, \\ \sum_{(i,t)\in e} x_e^* &= O(\log n) && \forall (i,t) \in M, \qquad (14) \\ \sum_e x_e^* c_e &= O(C^* \log n), \\ x_e^* &\in \{0,1\}. \end{aligned}$$

This relaxed solution is not a valid schedule, since $O(\log n)$ jobs are scheduled at one time; however, it can be converted to a valid schedule by use of the following lemma.

LEMMA 3.7. *Consider an invalid schedule S for a set of jobs with release dates on m unrelated parallel machines, in which at most λ jobs are assigned to each machine at any time. If W is the average weighted completion time of S, then there exists a schedule of average weighted completion time at most λW, in which at most one job is assigned to each machine at any time.*

Proof. Consider a job J_j scheduled in S; let its completion time be C_j^S. If we schedule the jobs on each machine in the order of their completion times in S, never starting one before its release date, then in the resulting schedule

1. J_j is started no earlier than its release date,
2. J_j finishes by time at most λC_j^S.

Statement 1 is true by design of the algorithm. Statement 2 is true since at most $\lambda C_j^S - p_{ij}$ work from other jobs can complete no later than C_j^S in schedule S, and jobs run simultaneously in schedule S can run back-to-back with no intermediate idle time in our expanded schedule. Therefore, job J_j is started by time $\lambda C_j^S - p_{ij}$ and completed by time λC_j^S. □

Combining the last two lemmas with the observation that $p_{\max} \leq n^3$ implies $|M| \leq mn^4$ yields the following theorem.

THEOREM 3.8. *There is a polynomial-time $O(\log^2 n)$-approximation algorithm for the minimization of average weighted completion time of a set of jobs with machine-varying release dates on unrelated machines, under the assumption that the maximum job sizes are bounded by $p_{\max} \leq n^3$.*

3.4. Large jobs. Since the p_{ij} are input in binary and in general need not be polynomial in n and m, the technique of the last section can not be applied directly to all instances, since it would yield superpolynomial-size formulations. Therefore we must find a way to handle very large jobs without impacting significantly on the quality of solution.

It is a standard technique in combinatorial scheduling to partition the jobs into a set of large jobs and a set of small jobs, schedule the large jobs, which are scaled to be in a polynomially bounded range, and then schedule the small jobs arbitrarily and show that their net contribution is not significant, (see, e.g., [24]). In the minimization of average weighted completion time, however, we must be more careful, since the small jobs may have large weights and can not be scheduled arbitrarily.

We employ several steps, each of which increases the average weighted completion time by a small constant factor. With more care we could reduce the constants introduced by each step; however, since our overall bound is $O(\log^2 n)$ we dispense with this precision for the sake of clarity of exposition.

The basic idea is to characterize each job by the minimum value, taken over all machines, of its (release date + processing time) on that machine. We then group the jobs together based on the size of their minimum $r_{ij} + p_{ij}$. The jobs in each group can be scaled down to be of polynomial size and thus we can construct a schedule for the scaled down versions of each group. We then scale the schedules back up, correct for the rounding error, and show that this does not affect the quality of approximation by more than a constant factor. We then apply Lemma 3.9 (see below) to show that the makespan can be kept short simultaneously.

The resulting schedules will be scheduled consecutively. However, since we have kept the makespan from growing too much, we have an upper bound on the start time of each subsequent schedule and thus we can show that the net disturbance of the initial schedules to the latter schedules will be minimal.

We now proceed in greater detail. Let $m(J_j) = \min_i(p_{ij} + r_{ij})$, and $\mathcal{J}^i =$

$\{J_j | n^{i-1} \leq m(J_j) \leq n^i\}$. Note that there are at most n nonempty $\mathcal{J}^i$, one for each of the n jobs. We will employ the following lemma in order to keep the makespan from growing too large.

LEMMA 3.9. *A schedule S for $\mathcal{J}^k$ can be converted, in polynomial time, to a schedule T of makespan at most $2n^{k+1}$ such that $C_j^T \leq 2C_j^S$ $\forall j \in \mathcal{J}^k$.*

Proof. Remove all jobs from S that complete later than time n^{k+1}, and, starting at time n^{k+1}, schedule them arbitrarily on the machine on which they run most quickly. This will take at most n^{k+1} time, so therefore any rescheduled job J_j satisfies $C_j^T \leq 2n^{k+1} \leq 2C_j^S$. □

We now turn to the problem of scheduling each $\mathcal{J}^l$ with a bounded guarantee on the average completion time.

LEMMA 3.10. *There exists an $O(\log^2 n)$-approximation algorithm to schedule each $\mathcal{J}^l$. In addition the schedule for $\mathcal{J}^l$ has makespan at most $2n^{l+1}$.*

Proof. Let $\mathcal{A}$ be the algorithm referred to in Theorem 3.8. We will use $\mathcal{A}$ to find an approximately optimal solution S^l for each $\mathcal{J}^l$. $\mathcal{A}$ cannot be applied directly to $\mathcal{J}^l$ since the sizes of the jobs involved may exceed n^3, so we apply $\mathcal{A}$ to a scaled version of $\mathcal{J}^l$.

For all j such that $J_j \in \mathcal{J}^l$, and for all i, set $p'_{ij} = \lfloor \frac{p_{ij}}{n^{l-2}} \rfloor$ and $r'_{ij} = \lfloor \frac{r_{ij}}{n^{l-2}} \rfloor$. Note that on at least one machine i , for each job J_j, $p'_{ij} \in [0, n^2]$ and $r'_{ij} \in [0, n^2]$.

We use $\mathcal{A}$ to obtain an approximate solution to the scaled version of $\mathcal{J}^l$ of average weighted completion time W. Although some of the p'_{ij} may still be large, Lemma 3.9 indicates that restricting the hypergraph formulation constructed by A to allow completion times no later than time $\lfloor \frac{2n^{l+1}}{n^{l-2}} \rfloor = 2n^3$ can only affect the quality of approximation by at most a factor of 2. Therefore $|M|$, the number of (machine, time) pairs, is $O(mn^3)$. Note that some of the p'_{ij} may be 0, but it is still important to include an edge in the hypergraph formulation for each job of size 0.

Now we argue that interpreting the solution of the scaled instance as a solution to the original instance $\mathcal{J}^l$ does not degrade the quality of approximation by more than a constant factor. The conversion from the scaled instance to the original instance is carried out by multiplying $p^*_{ij} = n^{l-2}p'_{ij}, r^*_{ij} = n^{l-2}r'_{ij}$ (which has no impact on quality of approximation) and then adding to each r^*_{ij} and p^*_{ij} the residual amount that was lost due to the floor operation.

The additional residual amounts of the release dates contribute at most a total of n^{l-1} time to the makespan of the schedule, since $|r_{ij} - r^*_{ij}| < n^{l-2}$, and therefore the entire contribution to the makespan is bounded above by $n \times n^{l-2} = n^{l-1}$. By a similar argument, the entire contribution of the residual amounts of the processing times to the makespan is bounded above by n^{l-1}.

So in the conversion from p^*_{ij}, r^*_{ij} to p_{ij}, r_{ij} we add at most $2n^{l-1}$ to the makespan of the schedule for $\mathcal{J}^l$. However, n^{l-1} is a lower bound on the completion time of any job in $\mathcal{J}^l$. Therefore, even if this additional time were added to the completion time of every job, the restoration of the residual amounts of the r_{ij} and p_{ij} degrades the quality of the approximation to average completion time by at most a constant factor. Finally, to satisfy the makespan constraint, we apply Lemma 3.9. □

We now construct two schedules S^{o} and S^{e}. In S^{o} we consecutively schedule $S^1, S^3, S^5, \ldots$, and in S^{e} we consecutively schedule $S^2, S^4, S^6, \ldots$. For the sake of clarity our schedule will have time of length $2n^{i+1}$ dedicated to each S^i even if S^i has no jobs.

LEMMA 3.11. *Let $\mathcal{J}^{o}$ be the set of jobs scheduled in S^{o} and $\mathcal{J}^{e}$ the set of jobs scheduled in S^{e}. The average weighted completion time of S^{o} is within a factor of*

$O(\log^2 n)$ *of the best possible for* $\mathcal{J}^o$, *and similarly for* S^e *and* $\mathcal{J}^e$.

Proof. The subschedule for any set $\mathcal{J}^i$ scheduled in S^{o} or S^{e} begins by time $(2+o(n))n^{i-1}$, since $\mathcal{J}^i$ is scheduled after $\mathcal{J}^{i-2}, \mathcal{J}^{i-4}, \ldots$, and the makespan of $\mathcal{J}^l$ is at most $2n^{l+1}$. Since n^{i-1} is a lower bound on the completion time of any job in $\mathcal{J}^i$, in the combined schedule S^{o} or S^{e}, each job completes within a small constant factor of its completion time in S^i. □

We now combine S^{o} and S^{e} by superimposing them over the same time slots. This creates an infeasible schedule in which the sum of completion times is just the sum of the completions times in S^{o} and S^{e}, but in which there may be two jobs scheduled simultaneously. We then use Lemma 3.7 to combine S^{o} and S^{e} to obtain a schedule S^{α} for all the jobs, whose average weighted completion time is within a factor of $O(\log^2 n)$ of optimal.

THEOREM 3.12. *There is a polynomial-time* $O(\log^2 n)$*-approximation algorithm for the minimization of average weighted completion time of a set of jobs with machine-varying release dates on unrelated machines.*

3.5. Scheduling with periodic connectivity.

The hypergraph formulation of the scheduling problem can model time-varying connectivity between jobs and machines; e.g., a job can only be processed during certain times on each machine. In this section we show how to apply our techniques to scheduling problems of *periodic* connectivity under some modest assumptions on the length of the period and job sizes.

DEFINITION 3.13. *The* periodic scheduling problem *is defined by* n *jobs,* m *unrelated machines, a period* P*, and for each time unit of* P *a specification of which jobs are allowed to run on which machines at that time.*

THEOREM 3.14. *Let* $\mathcal{I}$ *be an instance of the periodic scheduling problem in which* $p_{\max}$ *is polynomial in* n *and* m*, and let the optimum makespan of* $\mathcal{I}$ *be* $\mathcal{L}$*. There exists a polynomial-time algorithm which delivers a schedule of makespan* $O(\log n)(\mathcal{L}+P)$.

Proof. As above, we assume that $\mathcal{L}$ is known in advance, and then use binary search to complete the algorithm.

We construct the integer program

$$\sum_{J_j \in e} x_e = 1, \qquad j = 1, \ldots, n, \tag{15}$$

$$\sum_{(i,t) \in e} x_e \leq 1 \qquad \forall (i,t) \in M, \tag{16}$$

$$x_e \in \{0,1\},$$

where $M = \{(i,t) | 1 \leq i \leq m, 1 \leq t \leq \mathcal{L}\}$. We include an edge in the formulation if and only if it is valid with respect to the connectivity conditions. We then use Theorem 3.8 to produce a relaxed solution that satisfies

$$\begin{aligned} \sum_{j \in e} x_e^* &= 1, & j &= 1, \ldots, n, \\ \sum_{(i,t) \in e} x_e^* &= O(\log n) & \forall (i,t) &\in M, \\ x_e^* &\in \{0,1\}. \end{aligned}$$

Let the length of this relaxed schedule be L; $L \leq \mathcal{L}$. We construct a valid schedule of length $O(\log n)(L+P)$ by concatenating $O(\log n)$ blocks of length L. At the end of each block we will have to wait until the start of the next period to begin the next block; hence we obtain an overall bound of $O(\log n)(L + P)$. □

Note that we are assuming that the entire connectivity pattern of P is input explicitly; if it is input in some compressed form then we must assume that P is polynomial in n and m.

One motivation for such problems is the domain of *satellite communication systems* [18, 26]. One is given a set of sites on Earth and a set of satellites (in Earth orbit). Each site generates a sequence of communication requests; each request is potentially of a different duration and may require communication with any one of the satellites. A site can only transmit to certain satellites at certain times, based on where the satellite is in its orbit. The connectivity pattern of communication opportunities is periodic, due to the orbiting nature of the satellites.

The goal is to satisfy all communication requests as quickly as possible. We can use our hypergraph formulation technique to give an $O(\log n)$-approximation algorithm for the problem under the assumption that the p_j are bounded by a polynomial, since the rounding techniques do not generalize to this setting.

Appendix.

Proof of Theorem 2.4. The reduction is similar to the techniques used by Lenstra, Shmoys, and Tardos [17] to show that no algorithm can approximate the optimal makespan for unrelated parallel machines by better than a factor of $\frac{3}{2}$ unless $\mathcal{P} = \mathcal{NP}$.

Let A, B, C be disjoint sets, each with n elements, and let T be a set of m *triples*, $T = \{(a_i, b_j, c_k) : a_i \in A, b_j \in B, \text{and } c_k \in C\}$. We say that triple (a_i, b_j, c_k) *covers* a_i, b_j, and c_k, and define a *perfect matching* as a set of n triples that covers every element of A, B, and C exactly once. The problem of determining whether there exists a perfect matching given A, B, C, T is known as 3-dimensional matching and is $\mathcal{NP}$-complete [13]. We will refer to this problem as 3DM.

We will convert an instance $M = (A, B, C, T)$ of 3DM to an instance $(G, \ell, \mathcal{J})$ of the network scheduling problem that has a schedule of length 3 if and only if instance M has a perfect matching. We construct $\mathcal{N} = (G, \ell, \mathcal{J})$ as follows. To construct $G = (V, E)$, we associate a machine with each triple $t \in T$ (the *triple machines*) and a machine with each element of sets A, B, and C (the A *machines,* B *machines*, and C *machines,* respectively). Thus there are $3n + m$ machines. For each triple $t = (a_i, b_j, c_k)$, we create three edges: one from machine t to machine a_i of length 1, one from machine t to machine b_j of length 1, and one from machine t to machine c_k of length 2 (see Figure 3) This yields a network with $3m$ edges.

(In order to obtain a construction with only unit-length edges we introduce new nodes v_e, one for each edge of length 2, and replace each edge e from t to c_k by a path t to v_e to c_k. Each node v_e receives a job of size 3 at time 0. Clearly, in a schedule of length 3 this functions exactly as an edge of length 2, so for ease of exposition we use edges of length 2.)

The initial job distribution $\mathcal{J}$ is defined as follows. For each element $a_i \in A$, let $t(a_i)$ be the number of triples which contain element a_i. On A machine a_i we place $t(a_i)$ jobs:

1. $t(a_i) - 1$ jobs J_j with $p_j = 2$, the *dummy jobs*,
2. 1 job J_j with $p_j = 3$.

On each B machine, we place 2 jobs:

1. 1 job J_j with $p_j = 1$,
2. 1 job J_j with $p_j = 3$.

On each C machine, we place 2 jobs:

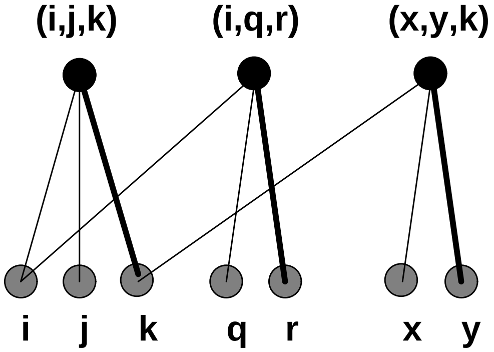


FIG. 3. *Subgraph of $\mathcal{N}$ corresponding to two triples (i, j, k) and (p, q, r). Dark edges correspond to edges of length* 2.

1. 1 job J_j with $p_j = 1$,
2. 1 job J_j with $p_j = 3$.

The basic idea behind the construction is that in any schedule of length 3, each machine corresponding to triple (a_i, b_j, c_k) runs one of only two possible schedules: either one dummy job from machine a_i (a *dummy schedule*) or the two unit-size jobs from machines b_j and c_k, respectively (a *matching schedule*). Each machine a_i is adjacent to exactly one machine running the latter schedule, and therefore these machines correspond to a perfect matching. If there is no perfect matching, a schedule of length 3 cannot exist. We now proceed in more detail.

We first argue that if M has a perfect matching, then the corresponding network scheduling problem $\mathcal{N}$ has a schedule of length 3. Each A, B, and C machine runs its job of size 3 from time 0 to 3. The remaining $m - n$ jobs of size 2 from the A machines ($t(a_i) - 1$ from machine a_i), the n unit-size jobs from the B machines and the n unit-size jobs from the C machines are scheduled as follows. Let $T_p \subset T$ be the perfect matching. Each machine corresponding to $t \in T_p$ runs a matching schedule, specifically the unit-size job from machine b_j and the unit-size job from machine c_k. Since these jobs are available to their triple machines at times 1 and 2, respectively, this schedule is feasible, and all jobs starting on B or C machines have been scheduled. Because the matching T_p contains exactly one triple for each a_i, there are $t(a_i) - 1$ unutilized machines adjacent to machine a_i. Each such machine runs one of the size-2 jobs from machine a_i starting at time 1. Since any job starting on machine a_i can arrive at these machines at time 1, this schedule is feasible and all jobs originating on the A machines have been scheduled. Therefore we have scheduled every job validly in 3 units of time.

We now show that if instance $\mathcal{N}$ has a schedule of length 3, then 3DM instance M has a perfect matching. We argue that any schedule of length 3 must have the form described above where each triple machine runs either a matching schedule or a dummy schedule; the set of machines running matching schedules correspond to a perfect matching for instance M.

First observe that in the schedule created above, each machine started processing a job as early as possible, and then was busy until the schedule completed. We call this property the *nonidleness* property. Clearly in any schedule of length 3 each A, B, and C machine must run only the size 3 job that originates there. In addition, a simple counting argument shows that the triple machines are idle for one unit of time and must be busy the remainder of the time. Thus in any schedule of length 3, all the $m-n$ size-2 dummy jobs from the A machines, n unit-size jobs from the B machines, and the n unit-size jobs from the C machines must be run by the triple machines.

We also observe that each job that is not run on its originating machine must run on an adjacent machine. We call this the *locality* property. The size 2 dummy jobs cannot travel more than 1 unit away in a length 3 schedule. Because all edges adjacent to each C machine have length 2, these unit-size jobs cannot travel more than one edge in a length-3 schedule. Finally, the unit-size jobs from the B machines must travel distance at least 3 to reach a nonadjacent triple machine, which is impossible in a schedule of length 3.

We now argue that each triple machine must run either a dummy schedule or a matching schedule. Each A machine a_i must send all of its $t(a_i)-1$ size-2 jobs to the triple machines adjacent to it. In a length-3 schedule, no machine can process two size-2 jobs. Therefore, the $t(a_i)-1$ jobs will be sent to $t(a_i)-1$ distinct triple machines. They will all run from time 1 through 3, and hence no other jobs can run on those $t(a_i)-1$ machines.

There are now exactly n triple machines not running dummy jobs and by construction of the network each has a different first element (the a_i's are all distinct). There are $2n$ unit-size jobs remaining to be scheduled, so by the nonidleness property, each such machine must run a unit-size job at time 1 and at time 2. Each edge adjacent to a C machine has length 2. Therefore, no job originating at a C machine can be processed elsewhere before time 2. Since there are n such jobs and n triple machines remaining to process them, each triple machine $t=(a_i,b_j,c_k)$ must run a job from a C machine at time 2. Furthermore, this C job must correspond to element c_k by the locality property. Therefore, each triple machine must process the job from the B machine adjacent to it at time slot 1. Therefore, the set of machines which run matching schedules cover all elements of sets A, B, and C. □

Proof of Theorem 3.1. We show how to convert an instance $M=(A,B,C,T)$ of the 3-dimensional matching problem to an instance $\mathcal{N}=(G,\ell,\mathcal{J})$ of the network scheduling problem. Instance $\mathcal{N}$ will have an average completion time equal to a certain value if and only if instance M has a perfect matching.

We construct $\mathcal{N}=(G,\ell,\mathcal{J})$ as follows. To construct the graph $G=(V,E)$, we associate a machine with each triple $t\in T$ (the triple machines), and a machine with each element of sets A, B, and C. For each triple $t=(a_i,b_j,c_k)$, we create three paths: one from machine t to machine a_i of length 1, one from machine t to machine b_j of length 3, and one from machine t to machine c_k of length 1. On the intermediate nodes of the path of length 3 we place machines (called the path machines), thus yielding a network with $m+5n$ nodes (machines) and $5m$ edges.

The initial job distribution $\mathcal{J}$ is defined as follows. For each element $x\in A\cup B\cup C$, let $t(x)$ be the number of triples which contain element x. Let $L=10nm$.

1. On each A machine a_i, we place two jobs, one with processing time 2 and one with processing time L.
2. On each B machine b_j, we place two jobs, each with processing time L.
3. On each path machine, we place a job of length L.
4. On each C machine c_k we place $t(c_k)$ jobs, all of length L.

Let I_j be the total idle time experienced by J_j before being processed. A schedule of minimum average completion time will minimize $\beta=\sum_j I_j$, the sum of the idle times. Because the value of L is so large, an optimal schedule will minimize the number of times quantities with the value L contribute to β. In particular, to avoid any job experiencing an idle time of L, in an optimal schedule once a machine runs a job of size L, it does not run any jobs afterward. Thus, at most one job of size L runs on any machine, and that job must be the last one. Since there are $m+5n$ jobs of size L and $m+5n$ machines, every machine must run exactly one size-L job.

We now compute a lower bound on β. First, observe that all but one of the jobs that originate on a machine c_k must run on another machine, since no machine can run two jobs of size L. Thus, each of these jobs must travel at least one edge for a total of $m-n$ idle time. Next, observe that of the two jobs that start on an A machine a_i, they either both run on a_i, with idle time at least 2, or one runs on another machine for an idle time of at least 1, and n overall. Now consider a B machine and its associated path machines. The combined idle time of the jobs originating on these machines must be at least 3. Thus we have a lower bound on idle time β of $m-n+n+3n=m+3n$.

We now show that a schedule with total idle time of β can be achieved if and only if there is a perfect 3D matching. If there is a matching, then each of the n triple machines that correspond to a matched edge will run a size-2 job from the A machine at time 1 and one of the size L jobs from the B machine at time 3. The $m-n$ unmatched triple machines will run a job from the corresponding C machine. Since there is a perfect matching there are exactly $t(c_k)-1$ such machines. All other jobs run on their originating machines at time 0, thus giving us a schedule with $\beta=m+3n$.

Now we show that this is the only such schedule of this length and hence must imply that a perfect matching exists. By the above lower bound arguments, each C machine c_k must send out $t(c_k)-1$ jobs, thus contributing at least $t(c_k)-1$ to the idle time. If any of these machines contributes more to the idle time, the total idle time must exceed β. The only way this lower bound can be achieved is for each of these jobs to travel exactly 1 edge and run at time 1. Therefore, in any schedule with idle time β, $m-n$ of the jobs of size L from C machines travel to adjacent triple machines and are run at time 1. These triple machines cannot run any other jobs. By construction of the network, the remaining set of triple machines T cover the set C.

Again by the above lower bound arguments, each A machine must contribute at most 1 to the idle time. Keeping both jobs incurs an idle time of 2, and therefore the global lower bound is exceeded. Thus in any schedule with $m+3n$ idle time, exactly one of the jobs from each A machine travels exactly 1 unit of time and is run at time 1. It must be the job of size 2, because each A machine must run a job of size L. Because the only adjacent machines are triple machines, all of the size-2 A jobs run on adjacent triple machines at time 1. Because there are exactly n machines in set T, each running exactly one A job, the set T covers set A.

Now consider a B machine and its associated path machines. The lower bound argument above shows that the combined idle time of the jobs originating on these machines must be at least 3. There are many ways to achieve this amount of idle time, each one places a job of size L on a triple machine at time x, where $x\in\{1,2,3\}$. But by the arguments above about the placement of the A jobs, we see that the size-L job that makes it from one of the B or path machines cannot run before time 3. It is straightforward to show that in an optimal schedule, a job arrives from a B or path machine at a triple machine at exactly time 3, and this job must run immediately upon arrival (otherwise the idle time would exceed 3). Therefore, each triple machine in T processes exactly one size-L job from a B machine, as this is the only job that can arrive at exactly time 3 without causing any additional idle time. This is only possible if set T covers set B. Thus the set of triples in T is a perfect matching. □

Acknowledgments. We are grateful to Phil Klein for several helpful discussions early in this research, to David Shmoys for several helpful discussions, especially about the upper bound for average completion time, to David Peleg and Baruch Awerbuch for explaining their off-line approximation algorithm to us, and to Perry Fizzano for reading an earlier draft of this paper. We also thank the anonymous referee for providing the example which demonstrates that Theorem 2.14 is tight.

REFERENCES

[1] N. Alon, G. Kalai, M. Ricklin, and L. Stockmeyer, *Lower bounds on the competitive ratio for mobile user tracking and distributed job scheduling*, in Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1992, pp. 334–343.

[2] B. Awerbuch, S. Kutten, and D. Peleg, *Competitive distributed job scheduling*, in Proceedings of the 24th Annual ACM Symposium on Theory of Computing, ACM, New York, 1992, pp. 571–580.

[3] J. Bruno, E. Coffman, and R. Sethi, *Scheduling independent tasks to reduce mean finishing time*, Comm. ACM, 17 (1974), pp. 382–387.

[4] X. Deng, H. Liu, J. Long, and B. Xiao, *Competitive analysis of network load balancing*, J. Parallel Distrib. Comput., 40 (1997), pp. 162–172.

[5] P. Fizzano, D. Karger, C. Stein, and J. Wein, *Job scheduling in rings*, in Proceedings of the 1994 ACM Symposium on Parallel Algorithms and Architectures, ACM, New York, 1994, pp. 210–219.

[6] R. Graham, *Bounds for certain multiprocessor anomalies*, Bell System Technical Journal, 45 (1966), pp. 1563–1581.

[7] R. Graham, *Bounds on multiprocessing anomalies*, SIAM J. Appl. Math., 17 (1969), pp. 263–269.

[8] D. Gusfield, *Bounds for naive multiple machine scheduling with release times and deadlines*, J. Algorithms, 5 (1984), pp. 1–6.

[9] L. Hall and D. B. Shmoys, *Approximation schemes for constrained scheduling problems*, in Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1989, pp. 134–141.

[10] D. Hochbaum and D. Shmoys, *Using dual approximation algorithms for scheduling problems: theoretical and practical results*, J. ACM, 34 (1987), pp. 144–162.

[11] D. Hochbaum and D. Shmoys, *A polynomial approximation scheme for machine scheduling on uniform processors: using the dual approximation approach*, SIAM J. Comput., 17 (1988), pp. 539–551.

[12] W. Horn, *Minimizing average flow time with parallel machines*, Oper. Res., 21 (1973), pp. 846–847.

[13] R. M. Karp, *Reducibility among combinatorial problems*, in Complexity of Computer Computations, Plenum Press, New York, 1972, pp. 85–103.

[14] T. Kawaguchi and S. Kyan, *Worst case bound of an LRF schedule for the mean weighted flow-time problem*, SIAM J. Comput., 15 (1986), pp. 1119–1129.

[15] E. Lawler, J. Lenstra, A. R. Kan, and D. Shmoys, *Sequencing and scheduling: Algorithms and complexity*, in Handbooks in Operations Research and Management Science, Vol. 4, Logistics of Production and Inventory, S. Graves, A. R. Kan, and P. Zipkin, eds., North-Holland, Amsterdam, 1993, pp. 445–522.

[16] J. Lenstra, A. R. Kan, and P. Brucker, *Complexity of machine scheduling problems*, Ann. Discrete Math., 1 (1977), pp. 343–362.

[17] J. Lenstra, D. Shmoys, and E. Tardos, *Approximation algorithms for scheduling unrelated parallel machines*, Math. Programming, 46 (1990), pp. 259–271.

[18] J. H. Lodge, *Mobile satellite communication systems: Toward global personal communications*, IEEE Communications Magazine, 30 (1991), pp. 24–31.

[19] C. H. Papadimitriou and M. Yannakakis, *Towards an architecture-independent analysis of parallel algorithms*, SIAM J. Comput., 19 (1990), pp. 322–328.

[20] D. Peleg, *private communication*, 1992.

[21] S. Plotkin, D. B. Shmoys, and E. Tardos, *Fast approximation algorithms for fractional packing and covering problems*, Math. Oper. Res., 20 (1995), pp. 257–301.

[22] P. Raghavan, *Probabilistic construction of deterministic algorithms: approximating packing integer programs*, J. Comput. System Sci., 37 (1988), pp. 130–143.

[23] P. Raghavan and C. D. Thompson, *Randomized rounding: a technique for provably good algorithms and algorithmic proofs*, Combinatorica, 7 (1987), pp. 365–374.

[24] D. B. Shmoys, C. Stein, and J. Wein, *Improved approximation algorithms for shop scheduling problems*, SIAM J. Comput., 23 (1994), pp. 617–632.

[25] D. B. Shmoys and E. Tardos, *An approximation algorithm for the generalized assignment problem*, Math. Programming A, 62 (1993), pp. 461–474.

[26] P. Wood, *Mobile satellite services for travelers*, IEEE Communications Magazine, 30 (1991), pp. 32–35.

SIAM J. DISCRETE MATH.
Vol. 10, No. 4, pp. 599–625, November 1997

STACK AND QUEUE LAYOUTS OF POSETS[*]

LENWOOD S. HEATH[†] AND SRIRAM V. PEMMARAJU[‡]

Abstract. The stacknumber (queuenumber) of a poset is defined as the stacknumber (queuenumber) of its Hasse diagram viewed as a directed acyclic graph. Upper bounds on the queuenumber of a poset are derived in terms of its jumpnumber, its length, its width, and the queuenumber of its covering graph. A lower bound of $\Omega(\sqrt{n})$ is shown for the queuenumber of the class of n-element planar posets. The queuenumber of a planar poset is shown to be within a small constant factor of its width. The stacknumber of n-element posets with planar covering graphs is shown to be $\Theta(n)$. These results exhibit sharp differences between the stacknumber and queuenumber of posets as well as between the stacknumber (queuenumber) of a poset and the stacknumber (queuenumber) of its covering graph.

Key words. poset, queue layout, stack layout, book embedding, Hasse diagram, jumpnumber

AMS subject classifications. 05C99, 68R10, 94C15

PII. S0895480193252380

1. Introduction.

Stack and queue layouts of undirected graphs appear in a variety of contexts such as VLSI, fault-tolerant processing, parallel processing, and sorting networks (Pemmaraju [16]). In a new context, Heath, Pemmaraju, and Ribbens [10] use queue layouts as the basis of an efficient scheme to perform matrix computations on a data driven network. Bernhart and Kainen [1] introduce the concept of a stack layout, which they call *book embedding*. Chung, Leighton, and Rosenberg [3] study stack layouts of undirected graphs and provide optimal stack layouts for a variety of classes of graphs. Heath and Rosenberg [13] develop the notion of queue layouts and provide optimal queue layouts for many classes of undirected graphs. Heath, Leighton, and Rosenberg [8] study relationships between queue and stack layouts of undirected graphs. In some applications of stack and queue layouts, it is more realistic to model the application domain with directed acyclic graphs (dags) or with posets, rather than with undirected graphs. Various questions that have been asked about stack and queue layouts of undirected graphs acquire a new flavor when there are directed edges (arcs). This is because the direction of the arcs imposes restrictions on the node orders that can be considered. Heath and Pemmaraju [9] and Heath, Pemmaraju, and Trenk [11, 12] initiate the study of stack and queue layouts of dags and provide optimal stack and queue layouts for several classes of dags.

In this paper, we focus on stack and queue layouts of posets. Posets are ubiquitous mathematical objects, and various measures of their structure have been defined. Some of these measures are bumpnumber, jumpnumber, length, width, dimension, and thickness [2, 7]. Nowakowski and Parker [15] define the stacknumber of a poset as the stacknumber of its Hasse diagram viewed as a dag. They derive a general lower bound on the stacknumber of a planar poset and an upper bound on the stacknumber of a lattice. Nowakowski and Parker [15] conclude by asking whether the stacknumber of the class of planar posets is unbounded. Hung [14] shows that there exists a planar

[*] Received by the editors July 22, 1993; accepted for publication (in revised form) September 19, 1996. This research was partially supported by National Science Foundation grant CCR-9009953.
http://www.siam.org/journals/sidma/10-4/25238.html

[†] Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0106 (heath@cs.vt.edu).

[‡] Department of Computer Science, University of Iowa, Iowa City, IA 52242 (sriram@cs.uiowa.edu).

poset with stacknumber 4; moreover, no planar poset with stacknumber 5 is known. Sysło [17] provides a lower bound on the stacknumber of a poset in terms of its bumpnumber. He also shows that, while posets with jumpnumber 1 have stacknumber at most 2, posets with jumpnumber 2 can have an arbitrarily large stacknumber.

The organization of this paper is as follows. Section 2 contains definitions. In section 3, we derive upper bounds on the queuenumber of a poset in terms of its jumpnumber, its length, its width, and the queuenumber of its covering graph. In section 4, we show that the queuenumber of the class of planar posets is unbounded. In a complementary upper bound result, we show that the queuenumber of a planar poset is within a small constant factor of its width. In section 5, we show that the stacknumber of the class of n-element posets with planar covering graphs is $\Theta(n)$. In section 6, the decision problem of recognizing a 4-queue poset is defined; Heath and Pemmaraju [9] and Heath, Pemmaraju, and Trenk [11] show that the problem is NP-complete. In section 7, we present several open questions and conjectures concerning stack and queue layouts of posets.

2. Definitions.

This section contains the definitions of stack and queue layouts of undirected graphs, dags, and posets. Other measures of the structure of posets are also defined.

Let $G = (V, E)$ be an undirected graph without multiple edges or loops. A *k-stack layout* of G consists of a total order σ on V along with an assignment of each edge in E to one of k stacks, $s_1, s_2, \ldots, s_k$. Each stack s_j operates as follows. The vertices of V are scanned in left-to-right (ascending) order according to σ. When a vertex v is encountered, any edges assigned to s_j that have v as their right endpoint must be at the top of the stack and are popped. Any edges that are assigned to s_j and have left endpoint v are pushed onto s_j in descending order (according to σ) of their right endpoints. The *stacknumber* $SN(G)$ of G is the smallest k such that G has a *k-stack layout.* G is said to be a *k-stack* graph if $SN(G) = k$. The *stacknumber* of a class of graphs $\mathcal{C}$, denoted by $SN_{\mathcal{C}}(n)$, is the function of the natural numbers that equals the least upper bound of the stacknumber of all graphs in $\mathcal{C}$ with at most n vertices. We are interested in the asymptotic behavior of $SN_{\mathcal{C}}(n)$ or in whether $SN_{\mathcal{C}}(n)$ is bounded above by a constant.

A *k-queue layout* of G consists of a total order σ on V along with an assignment of each edge in E to one of k queues, $q_1, q_2, \ldots, q_k$. Each queue q_j operates as follows. The vertices of V are scanned in left-to-right (ascending) order according to σ. When a vertex v is encountered, any edges assigned to q_j that have v as their right endpoint must be at the front of the queue and are dequeued. Any edges that are assigned to q_j and have left endpoint v are enqueued into q_j in ascending order (according to σ) of their right endpoints. The *queuenumber* $QN(G)$ of G is the smallest k such that G has a *k-queue layout.* The *queuenumber* of a class of graphs $\mathcal{C}$, denoted by $QN_{\mathcal{C}}(n)$, is the function of the natural numbers that equals the least upper bound of the queuenumber of all graphs in $\mathcal{C}$ with at most n vertices. We are interested in the asymptotic behavior of $QN_{\mathcal{C}}(n)$ or in whether $QN_{\mathcal{C}}(n)$ is bounded above by a constant.

For a fixed order σ on V, we identify sets of edges that are obstacles to minimizing the number of stacks or queues. A *k-rainbow* is a set of k edges $\{(a_i, b_i) \mid 1 \leq i \leq k\}$ such that

$$a_1 <_\sigma a_2 <_\sigma \cdots <_\sigma a_{k-1} <_\sigma a_k <_\sigma b_k <_\sigma b_{k-1} <_\sigma \cdots <_\sigma b_2 <_\sigma b_1;$$

i.e., a rainbow is a *nested* matching. Any two edges in a rainbow are said to *nest.*

A *k-twist* is a set of k edges $\{(a_i, b_i) \mid 1 \le i \le k\}$ such that

$$a_1 <_\sigma a_2 <_\sigma \cdots <_\sigma a_{k-1} <_\sigma a_k <_\sigma b_1 <_\sigma b_2 <_\sigma \cdots <_\sigma b_{k-1} <_\sigma b_k,$$

i.e., a twist is a *fully crossing* matching. Any two edges in a twist are said to *cross*.

A rainbow is an obstacle for a queue layout because no two edges that nest can be assigned to the same queue, while a twist is an obstacle for a stack layout because no two edges that cross can be assigned to the same stack. Intuitively, we can think of a stack layout or a queue layout of a graph as a drawing of the graph in which the vertices are laid out on a horizontal line and the edges appear as arcs above the line. In a stack layout no two edges that intersect can be assigned to the same stack, while in a queue layout no two edges that nest can be assigned to the same queue. Clearly, the size of the largest twist (rainbow) in a layout is a lower bound on the number of stacks (queues) required for that layout. Heath and Rosenberg [13] show that the size of the largest rainbow in a layout equals the minimum queue requirement of the layout.

PROPOSITION 2.1 (Heath and Rosenberg, [Theorem 2.3, 13]). *Suppose $G = (V, E)$ is an undirected graph, and σ is a fixed total order on V. If G has no rainbow of more than k edges with respect to σ, then G has a k-queue layout with respect to σ.*

In contrast, the size of the largest twist in a layout may be strictly less than the minimum stack requirement of the layout (see [13, Proposition 2.4]).

The definitions of stack and queue layouts are now extended to dags by requiring that the layout order be a topological order. Following a common distinction, we use *vertices* and *edges* for undirected graphs, but *nodes* and *arcs* for directed graphs. Suppose that $G = (V, E)$ is an undirected graph and that $\vec{G} = (V, \vec{E})$ is a dag whose arc set $\vec{E}$ is obtained by directing the edges in E. A *topological order* of $\vec{G}$ is a total order σ on V such that $(u, v) \in \vec{E}$ implies $u <_\sigma v$. A k-stack (k-queue) layout of the dag $\vec{G} = (V, \vec{E})$ is a k-stack (k-queue) layout of the graph G such that the total order is a *topological order* of $\vec{G}$. As before, $SN(\vec{G})$ is the smallest k such that $\vec{G}$ has a k-stack layout, and $QN(\vec{G})$ is the smallest k such that $\vec{G}$ has a k-queue layout.

A *partial order* is a reflexive, transitive, antisymmetric binary relation. A *poset* $P = (V, \le)$ is a set V with a partial order $\le$ (see Birkhoff [2] or Davey and Priestly [4]). The cardinality $|P|$ of a poset P equals $|V|$. We only consider posets with finite cardinality in this paper. We write $u < v$ if $u \le v$ and $u \ne v$. The *Hasse diagram* $\vec{H}(P) = (V, \vec{E})$ of a poset $P = (V, \le)$ is a dag with arc set

$$\vec{E} = \{(u, v) \mid u < v \text{ and there is no } w \text{ such that } u < w < v\}$$

(see Davey and Priestly [4]). A Hasse diagram is a minimal representation of a poset because it contains none of the arcs implied by transitivity of $\le$. The stacknumber $SN(P)$ of a poset P is $SN(\vec{H}(P))$, the stacknumber of its Hasse diagram. Similarly, the queuenumber $QN(P)$ of a poset P is $QN(\vec{H}(P))$, the queuenumber of its Hasse diagram. Figure 2.1 gives an example of a 2-stack poset, while Fig. 2.2 gives an example of a 2-queue poset. The underlying undirected graph, $H(P)$, of $\vec{H}(P)$ is called the *covering graph* of P. Clearly, for any poset P, we have

$$SN(H(P)) \le SN(P)$$

and

$$QN(H(P)) \le QN(P).$$

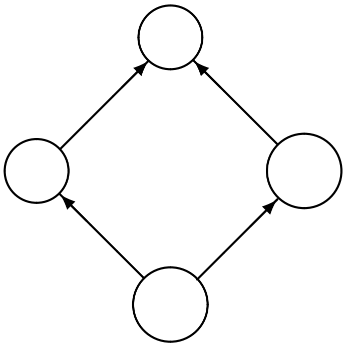


FIG. 2.1. *A 2-stack poset.*

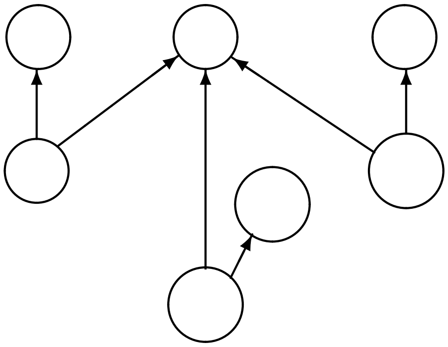


FIG. 2.2. *A 2-queue poset.*

The stacknumber and the queuenumber of the covering graphs of the posets in both Fig. 2.1 and Fig. 2.2 are 1. A poset P is *planar* if its Hasse diagram $\vec{H}(P)$ has a planar embedding in which all arcs are drawn as straight line segments with the tail of each arc strictly below its head with respect to a Cartesian coordinate system; call such an embedding of any dag an *upwards embedding.* Without loss of generality, we may always assume that no two nodes of $\vec{H}(P)$ are on the same horizontal line. (If two nodes are on the same horizontal line, a slight vertical perturbation of either of them yields another upwards embedding with the nodes on different horizontal lines.) Given an upwards embedding of a dag, the y coordinates of the nodes give a topological order on the nodes from lowest to highest called the *vertical order.* Note that the covering graph $H(P)$ may be planar even though the poset P is not. Figure 2.3 shows an example of a nonplanar poset whose covering graph is planar.

Let γ be a fixed topological order on $\vec{H}(P)$. Two elements u and v are *adjacent* in γ if there is no w such that $u <_\gamma w <_\gamma v$ or $v <_\gamma w <_\gamma u$. A *spine arc* in $\vec{H}(P)$ with respect to γ is an arc (u, v) in $\vec{H}(P)$ such that u and v are adjacent in γ. A *break* in $\vec{H}(P)$ with respect to γ is a pair (u, v) of adjacent elements such that $u <_\gamma v$ and (u, v) is not an arc in $\vec{H}(P)$. A *connection* C in $\vec{H}(P)$ with respect to γ is a maximal sequence of elements $u_1 <_\gamma u_2 <_\gamma \cdots <_\gamma u_k$ such that (u_i, u_{i+1}) is a spine arc for all $i, 1 \leq i < k$; in other words a connection is a maximal path of spine arcs without a break. Since $\vec{H}(P)$ contains no transitive arcs, there can be no nonspine arcs between nodes in a connection. The *breaknumber* $BN(\gamma, P)$ of a topological order γ of $\vec{H}(P)$

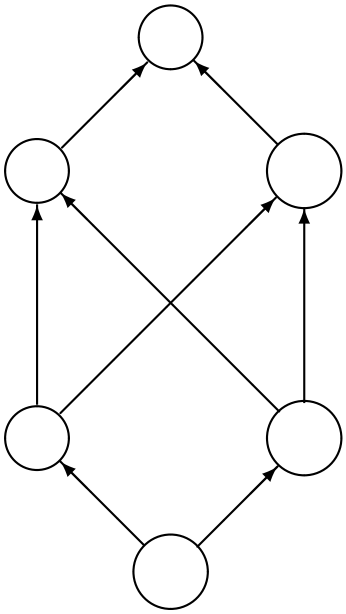


FIG. 2.3. *A nonplanar poset whose covering graph is planar.*

is the number of breaks in $\vec{H}(P)$ with respect to γ. The *jumpnumber of* P, denoted by $JN(P)$, is the minimum of $BN(\gamma, P)$ over all topological orders γ on $\vec{H}(P)$.

A *chain* in a poset P is a set of elements $\{u_1, u_2, \ldots, u_k\}$ such that $u_1 < u_2 < \cdots < u_k$. The *length* $L(P)$ of a poset P is the maximum cardinality of any chain in P. An *antichain* in a poset P is a subset of elements of S that does not contain a chain of size 2. The *width* $W(P)$ of a poset P is the maximum cardinality of any antichain in P.

3. Upper bounds on queuenumber.

In this section we derive upper bounds on the queuenumber of a poset in terms of its jumpnumber, its length, its width, and the queuenumber of its covering graph.

3.1. Jumpnumber and queuenumber.

Sysło [17] proves the following relationship between the jumpnumber and the stacknumber of posets.

PROPOSITION 3.1 (Sysło [17]). *For any poset P with $JN(P) = 1$, we have $SN(P) \leq 2$. If $\mathcal{J}_2$ is the infinite class of posets having jumpnumber 2, then $SN_{\mathcal{J}_2}(n) = \Omega(n)$.*

In contrast, we show that, for any poset P, the queuenumber of P is at most the jumpnumber of P plus 1. Moreover, we show that this bound is tight within a small constant factor.

THEOREM 3.2. *For any poset P, $QN(P) \leq JN(P) + 1$. For every $n \geq 2$, there exists a poset P such that $|P| = 2n$ and $JN(P)/2 < QN(P)$.*

Proof. For the upper bound on queuenumber, suppose that P is any poset and that $JN(P) = k$. Let γ be a topological order on $\vec{H}(P)$ that has exactly k breaks and $k+1$ connections. Lay out $\vec{H}(P)$ according to γ and label these connections $C_0, C_1, \ldots, C_k$ from left to right. Let (u_1, v_1) and (u_2, v_2) be any two nonspine arcs such that u_1 and u_2 are in C_i and v_1 and v_2 are in C_j, where $1 \leq i < j \leq k$. If (u_1, v_1) and (u_2, v_2) nest, then one of (u_1, v_1) and (u_2, v_2) (the arc that nests over the other arc) is a transitive arc. Since $\vec{H}(P)$ contains no transitive arcs, (u_1, v_1) and (u_2, v_2) do not nest. This suggests the following assignment of arcs to queues: Assign all nonspine arcs between pairs of connections C_i and C_j, where $|i-j| = \ell, 1 \leq \ell \leq k$, to queue q_ℓ. Assign all the spine arcs to a queue q_0. Hence, we use k queues for nonspine arcs and one queue for spine arcs, for a total of $k+1$ queues.

For the lower bound on queuenumber, construct the Hasse diagram of a poset P from the complete bipartite graph $K_{n,n} = (V_1, V_2, E)$ by directing all the edges from vertices in V_1 to vertices in V_2. All topological orders on $\vec{H}(P)$ yield isomorphic layouts. Hence, $JN(P) = 2(n-1)$, $QN(P) = n$, and

$$QN(P) = \frac{n}{2(n-1)} JN(P).$$

The lower bound follows. □

Proposition 3.1 and Theorem 3.2 lead to the following corollary.

Corollary 3.3. *There exists a class of posets $\mathcal{P}$ for which the ratio*

$$\frac{SN_{\mathcal{P}}(n)}{QN_{\mathcal{P}}(n)} = \Omega(n).$$

Looking ahead, Theorem 4.2 shows the existence of a class of posets $\mathcal{P}$ for which the reciprocal ratio $QN_{\mathcal{P}}(n)/SN_{\mathcal{P}}(n)$ is unbounded.

3.2. Length and queuenumber. To prove the next theorem, we need the following lemma that gives a bound on the queuenumber of a layout of a graph whose vertices have been rearranged in a limited fashion.

Lemma 3.4 (Pemmaraju [16]). *Suppose that σ is an order on the vertices of an m-partite graph $G = (V_1, V_2, \ldots, V_m, E)$ that yields a k-queue layout of G. Let σ' be an order on the vertices of G in which the vertices in each set V_i, $1 \leq i \leq m$, appear consecutively and in the same order as in σ. Then σ' yields a layout of G in $2(m-1)k$ queues.*

Theorem 3.5, the main result of this section, gives an upper bound on the queuenumber of a poset in terms of its length and the queuenumber of its covering graph.

Theorem 3.5. *For any poset P,*

$$QN(P) \ \leq \ 2 \cdot (L(P) - 1) \cdot QN(H(P)).$$

There exists an infinite class of posets $\mathcal{P}$ such that $L_{\mathcal{P}}(n) = 2$ and, for all $P \in \mathcal{P}$,

$$\left\lceil \frac{QN(P)}{2} \right\rceil = (L(P) - 1) \cdot QN(H(P)).$$

Proof. Suppose P is any poset, $\vec{H}(P) = (V, \vec{E})$, and $QN(H(P)) = k$. Let σ be a total order on V that yields a k-queue layout of $H(P)$. The nodes of $\vec{H}(P)$ can be labeled by a function $l : V \to \{1, \ldots, L(P)\}$ such that $l(u) < l(v)$ if $u < v$ in P, as follows. Let $\vec{H}_0 = \vec{H}(P)$. Label all the nodes with indegree 0 in $\vec{H}_0$ with the label 1. Delete all the labeled nodes in $\vec{H}_0$ to obtain $\vec{H}_1$. In general, label the nodes with indegree 0 in $\vec{H}_i$ with the label $i+1$. Delete the labeled nodes in $\vec{H}_i$ to obtain $\vec{H}_{i+1}$. By an inductive proof, it can be checked that the labeling so obtained satisfies the required conditions. Let $V_i = \{u \in V \mid l(u) = i\}$. For any arc $(u, v) \in \vec{E}$, if $u \in V_i$ and $v \in V_j$, then $i < j$. Therefore $\vec{H}(P) = (V_1, V_2, \ldots, V_{L(P)}, \vec{E})$ is an $L(P)$-partite dag. Define the total order γ on the nodes of $\vec{H}(P)$ by the following:

1. The elements in each set V_i, $1 \leq i \leq L(P)$, occur contiguously and in the order prescribed by σ.
2. The elements in V_i occur before the elements in V_{i+1} for all i, $1 \leq i < L(P)$.

Since every arc in $\vec{H}(P)$ is from a node in V_i to a node in V_j, $1 \leq i < j \leq L(P)$, γ is a topological order on $\vec{H}(P)$. By Lemma 3.4 γ yields a layout that requires no more than $2 \cdot (L(P) - 1) \cdot k$ queues.

We now prove the second part of the theorem. For each $n \geq 2$, let $p = \lfloor n/2 \rfloor$ and $q = \lceil n/2 \rceil$. Let the complete bipartite graph $K_{p,q} = (V_1, V_2, E)$ be such that $|V_1| = p$ and $|V_2| = q$. We get the Hasse diagram of a poset P of size n by directing the edges in $K_{p,q}$ from V_1 to V_2. Clearly, $L(P) = 2$ and $QN(P) = p$. Heath and Rosenberg [13] and Pemmaraju [16] present different proofs of the following formula that gives the precise queuenumber of an arbitrary complete bipartite graph:

$$QN(K_{r,s}) = \min(\lceil r/2 \rceil, \lceil s/2 \rceil).$$

Since $p \leq q$, $QN(K_{p,q}) = \lceil p/2 \rceil$. Therefore,

$$\left\lceil \frac{QN(P)}{2} \right\rceil = (L(P) - 1) \cdot QN(H(P)).$$

Let $\mathcal{P}$ be the class of all posets constructed in the manner described above. The second part of the theorem follows. □

Note that Theorem 3.5 holds for dags as well as for posets as its proof does not rely on the absence of transitive arcs. Theorem 3.5 leads to the following corollary.

COROLLARY 3.6. *For any poset P,*

$$QN(H(P)) \;\leq\; QN(P) \;\leq\; 2 \cdot (L(P) - 1) \cdot QN(H(P)).$$

Suppose $\mathcal{P}$ is a class of posets such that there exists a constant K with $L(P) \leq K$, for all $P \in \mathcal{P}$. Then $QN_{\mathcal{P}}(n) = \Theta(QN_{H(\mathcal{P})}(n))$.

We conjecture, but have been unable to show, that the upper bound in Theorem 3.5 is tight, within constant factors, for larger values of $L(P)$ also.

3.3. Width and queuenumber. In this section, we establish an upper bound on the queuenumber of a poset in terms of its width. We need the following result of Dilworth.

LEMMA 3.7 (Dilworth [5]). *Let $P = (V, \leq)$ be a poset. Then V can be partitioned into $W(P)$ chains.*

For a poset $P = (V, \leq)$, let $Z_1, Z_2, \ldots, Z_{W(P)}$ be a partition of V into $W(P)$ chains. Define an *i-chain arc* as an arc in $\vec{H}(P)$, both of whose end points belong to chain Z_i, $1 \leq i \leq W(P)$. An *(i,j)-cross arc*, $i \neq j$, is an arc whose tail belongs to chain Z_i and whose head belongs to chain Z_j.

THEOREM 3.8. *The largest rainbow in any layout of a poset P is of size no greater than $W(P)^2$. Hence, the queuenumber of any layout of P is at most $W(P)^2$.*

Proof. Fix an arbitrary topological order of $\vec{H}(P)$. Let $Z_1, Z_2, \ldots, Z_{W(P)}$ be a partition of V into $W(P)$ chains. For any i, no two i-chain arcs nest, since $\vec{H}(P)$ contains no transitive arcs. Therefore, the largest rainbow of chain arcs has size no greater than $W(P)$. If $i \neq j$ then no two (i,j)-cross arcs can nest without one of them being a transitive arc. Therefore, the largest rainbow of cross arcs has size no greater than $W(P)(W(P) - 1)$. The size of the largest rainbow is at most $W(P) + W(P)(W(P) - 1) = W(P)^2$. By Proposition 2.1, the theorem follows. □

The bound established in the above theorem is not known to be tight. In fact, we believe that the queuenumber of a poset is bounded above by its width (see Conjecture 1 in Section 7).

4. The queuenumber of planar posets.

In this section, we first show that the queuenumber of the class of planar posets is unbounded. We then establish an upper bound on the queuenumber of a planar poset in terms of its width.

4.1. A lower bound on the queuenumber of planar posets.

We construct a sequence of planar posets P_n with $|P_n| = 3n + 3$ and $QN(P_n) = \Theta(\sqrt{n})$. In fact, we determine the queuenumber of P_n almost exactly. To prove the theorem, we need the following result of Erdös and Szekeres.

LEMMA 4.1 (Erdös and Szekeres [6]). *Let $(x_i)_{i=1}^n$ be a sequence of distinct elements from a set X. Let δ be a total order on X. Then $(x_i)_{i=1}^n$ either contains a monotonically increasing subsequence of size $\lceil \sqrt{n} \rceil$ or a monotonically decreasing subsequence of size $\lceil \sqrt{n} \rceil$ with respect to δ.*

The proof of Theorem 4.2 constructs the desired sequence of posets.

THEOREM 4.2. *For each $n \geq 1$, there exists a planar poset P_n with $3n + 3$ elements such that*

$$\lceil \sqrt{n+1} \rceil \leq QN(P_n) \leq \lceil \sqrt{n} \rceil + 1.$$

Proof. Suppose $n \geq 1$. Define three disjoint sets U, V, and W as follows:

$$\begin{aligned} U &= \{u_i \mid 0 \leq i \leq n\}, \\ V &= \{v_i \mid 0 \leq i \leq n\}, \\ W &= \{w_i \mid 0 \leq i \leq n\}. \end{aligned}$$

Let $S = U \cup V \cup W$. The planar poset $P_n = (S, \leq)$ is given by

$$\begin{aligned} u_i &< u_{i-1}, \\ v_{i-1} &< v_i, \end{aligned}$$

for $1 \leq i \leq n$, and

$$u_i \quad < \quad w_i \quad < \quad v_i,$$

for $0 \leq i \leq n$. Figure 4.1 shows the Hasse diagram of P_4. Let σ be an arbitrary order on the elements of S. The elements of $U \cup V \cup \{w_0\}$ appear in the order $u_n, u_{n-1}, \ldots, u_0, w_0, v_0, v_1, \ldots, v_n$ in σ, and all elements of W appear between u_n and v_n. Define a total order δ on the elements of W by $w_i <_\delta w_j$ if $i < j$. Suppose that

$$w_{i_1}, w_{i_2}, \ldots, w_{i_k}$$

is an increasing sequence of nodes in W with respect to δ. Since w_{i_1} appears after u_{i_1} in any topological order of $\vec{H}(P_n)$, the following sequence of nodes is a subsequence of σ:

$$u_{i_k}, u_{i_{k-1}}, \ldots, u_{i_1}, w_{i_1}, w_{i_2}, \ldots, w_{i_k}.$$

Therefore, the set $\{(u_{i_j}, w_{i_j}) \mid 1 \leq j \leq k\}$ is a k-rainbow in σ. Similarly, if

$$w_{i_1}, w_{i_2}, \ldots, w_{i_k}$$

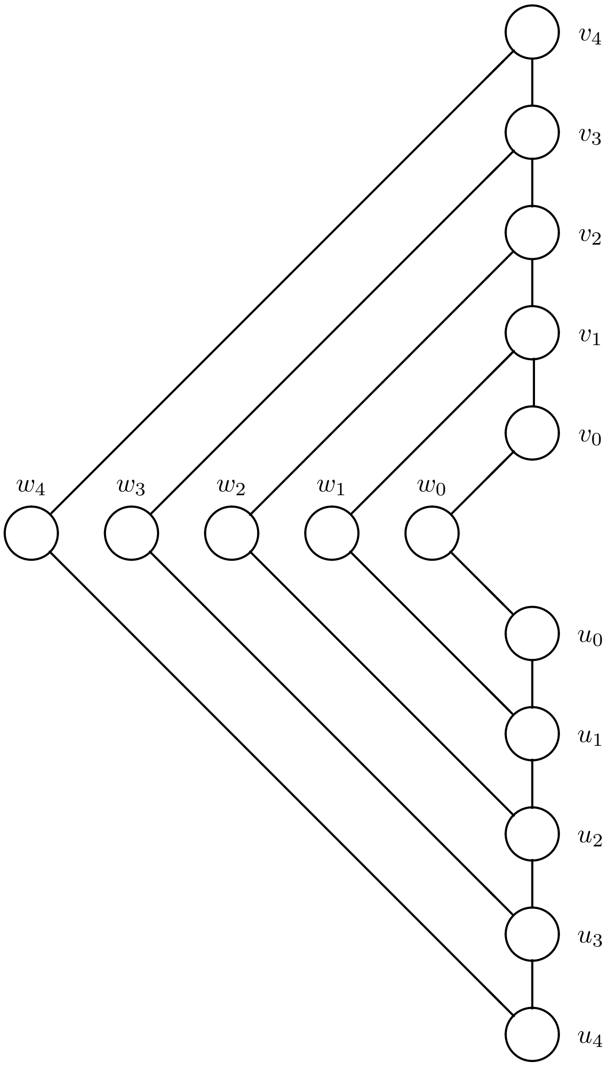


FIG. 4.1. *The planar poset P_4.*

is a decreasing sequence of nodes in W with respect to δ, then the set $\{(w_{i_j}, v_{i_j}) \mid 1 \leq j \leq k\}$ is a k-rainbow in σ. By Lemma 4.1, in σ, there is an increasing subsequence of size $\lceil \sqrt{n+1} \rceil$ or a decreasing subsequence of size $\lceil \sqrt{n+1} \rceil$ with respect to δ. Thus there is a rainbow of size $\lceil \sqrt{n+1} \rceil$ in any topological order on $\vec{H}(P_n)$. Therefore, $QN(P_n) \geq \lceil \sqrt{n+1} \rceil$. This is the desired lower bound.

To prove the upper bound, we give a layout of P_n in $\lceil \sqrt{n} \,\rceil + 1$ queues. Let $s = \lceil \sqrt{n} \,\rceil$, and let $t = \lceil n/s \rceil \leq \lceil \sqrt{n} \,\rceil$. Partition $W - \{w_0\}$ into s nearly equal-sized subsets

$$W_1, W_2, \ldots, W_s$$

as follows:

$$W_i = \begin{cases} \{w_j \mid (i-1)t + 1 \leq j \leq it\}, & 1 \leq i \leq s-1, \\ \{w_j \mid (s-1)t + 1 \leq j \leq n\}, & i = s. \end{cases}$$

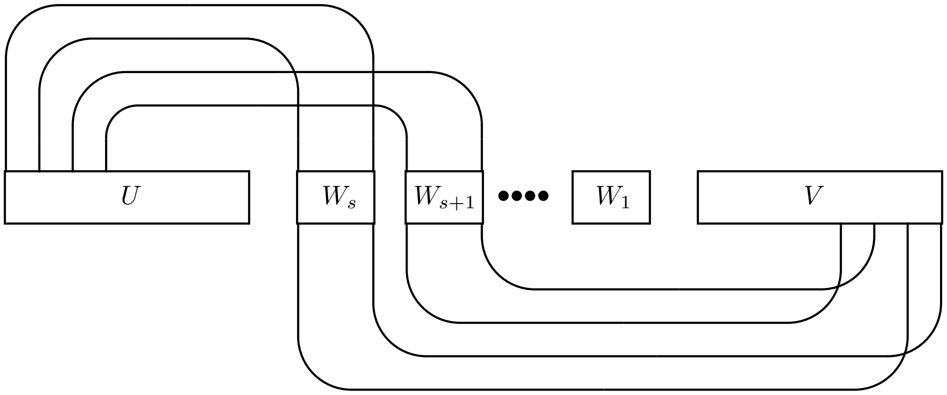


FIG. 4.2. *Schematic layout of planar poset P_n.*

Construct an order σ on the elements of S by first placing the elements in $U \cup V \cup \{w_0\}$ in the order

$$u_n, u_{n-1}, \ldots, u_0, w_0, v_0, v_1, \ldots, v_n.$$

Now place the elements of $W - \{w_0\}$ between u_0 and v_0 such that the elements belonging to each set W_i appear contiguously and the sets themselves appear in the order

$$W_s, W_{s-1}, \ldots, W_1.$$

Within each set W_i, $1 \leq i \leq s$, place the elements in increasing order with respect to δ. Figure 4.2 schematically represents the constructed order. The arcs from U to W form s mutually intersecting rainbows each of size at most t. Therefore, t queues suffice for these arcs. The arcs from W to V form s nested twists each of size at most t. Therefore s queues suffice for these arcs. Since no two arcs, one from U to W and the other from W to V nest, they can all be assigned to the same set of s queues. An additional queue is required for the remaining arcs. This is a layout of P_n in $\lceil \sqrt{n} \rceil + 1$ queues. Therefore, $QN(P_n) \leq \lceil \sqrt{n} \rceil + 1$, as desired. □

We believe that the upper bound in the above proof can be tightened to exactly match the lower bound. In fact, we have been able to show that for $m^2 \leq n \leq m(m+1)$, $QN(P_n) = m + 1 = \lceil \sqrt{n+1} \rceil$.

The situation for stacknumber of planar posets is somewhat different in that there is no known example of a sequence of planar posets with unbounded stacknumber. Two observations about the sequence P_n in Theorem 4.2 are in order. The first observation is that $SN(P_n) = 2$. A 2-stack layout of $\vec{H}(P_4)$ is shown in Fig. 4.3. The second observation is that the stacknumber *and* the queuenumber of $H(P_n)$ is 2. A 2-queue layout of $H(P_4)$ is shown in Fig. 4.4. Theorem 4.2 and the above observations imply the following corollaries.

COROLLARY 4.3. *There exists a class $\mathcal{P}$ of planar posets such that*

$$\frac{QN_{\mathcal{P}}(n)}{SN_{\mathcal{P}}(n)} = \Omega\left(\sqrt{n}\right).$$

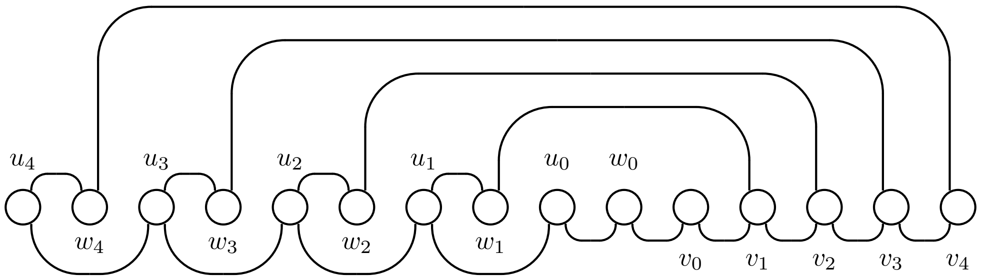


FIG. 4.3. *A 2-stack layout of the planar poset P_4.*

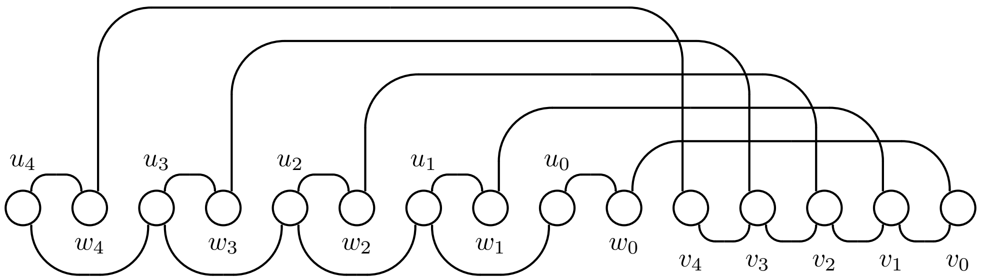


FIG. 4.4. *A 2-queue layout of the covering graph of P_4.*

COROLLARY 4.4. *There exists a class $\mathcal{P}$ of planar posets such that*

$$\frac{QN_{\mathcal{P}}(n)}{QN_{H(\mathcal{P})}(n)} = \Omega\left(\sqrt{n}\right).$$

While Theorem 4.2 establishes a lower bound of $\Omega(\sqrt{n})$ on the queuenumber of the class of n-element planar posets, a matching upper bound is not known (see Conjecture 2 in section 7).

4.2. An upper bound on the queuenumber of planar posets. In this section, we show that the queuenumber of a planar poset is bounded above by a small constant multiple of its width. The bound is a consequence of the following theorem, the proof of which occupies the remainder of the section.

THEOREM 4.5. *For any planar poset P where $\vec{H}(P)$ contains at least one arc and for any upward embedding of $\vec{H}(P)$, the layout of $\vec{H}(P)$ given by the vertical order σ has queuenumber less than $4W(P)$.*

Before the proof of Theorem 4.5, we present some definitions, some observations, and a series of three lemmas. First, we fix notation and terminology to use throughout the section. Suppose that $P = (V, \leq_P)$ is a planar poset with a given upwards embedding of $\vec{H}(P)$. Let σ be the vertical order on V. Now suppose that the size of a largest rainbow in the vertical order of $\vec{H}(P)$ is $k \geq 1$. By Proposition 2.1, the queuenumber of this layout is k. Focus on a particular k-rainbow whose arcs are $(a_1, b_1), (a_2, b_2), \ldots, (a_k, b_k)$. Call these arcs the *rainbow arcs;* in particular, the arc (a_i, b_i) is the *rainbow arc of* a_i and *of* b_i. The nodes in the set $A = \{a_1, a_2, \ldots, a_k\}$ are *bottom nodes,* and the nodes in the set $B = \{b_1, b_2, \ldots, b_k\}$ are *top nodes.* Let $y(v)$ denote the y-coordinate of a node v in the upwards embedding. Suppose that (a_i, b_i) and (a_j, b_j) are distinct rainbow arcs. Since these arcs nest in the vertical order σ,

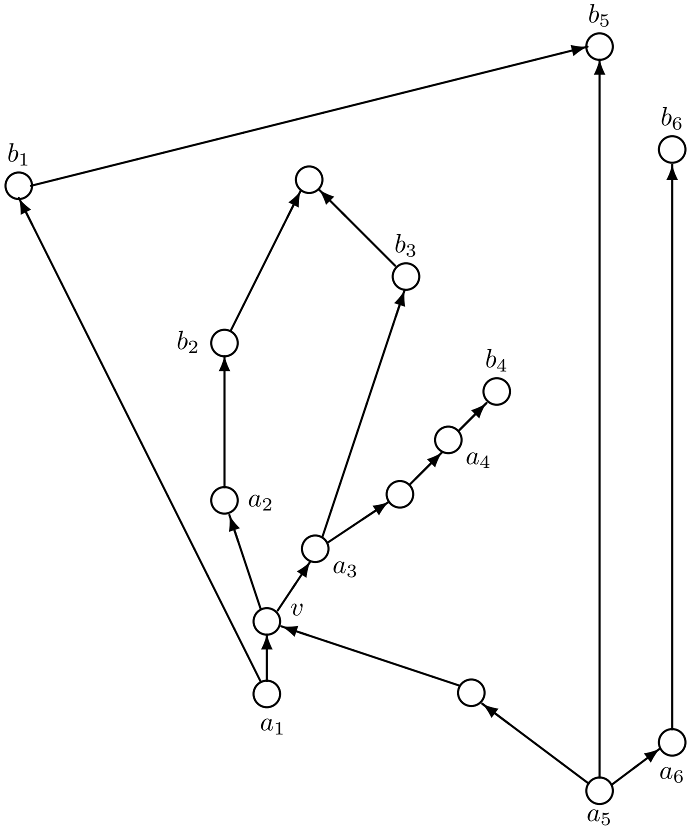


FIG. 4.5. *An example of rainbow arcs.*

we know that $\max\{y(a_i), y(a_j)\} < \min\{y(b_i), y(b_j)\}$. More generally,

$$y_1 = \max_{1\leq i\leq k} y(a_i) < \min_{1\leq i\leq k} y(b_i) = y_2.$$

The horizontal line defined by the equation $y = (y_1 + y_2)/2$ intersects every (a_i, b_i). In moving along this line from left to right, we encounter these intersections in a definite order. By re-indexing the rainbow arcs, we may assume that these intersections are encountered in the order $(a_1, b_1), (a_2, b_2), \ldots, (a_k, b_k)$; call this the left-to-right order of the rainbow arcs. Figure 4.5 illustrates an upwards embedding of a Hasse diagram with $k = 6$. The arcs are indexed in left-to-right order.

Define the *left-to-right total order* $\leq_{LR}$ on A (respectively, B) by $a_i \leq_{LR} a_j$ (respectively, $b_i \leq_{LR} b_j$) if $i \leq j$. If $a_i \leq_{LR} a_j$, we say that a_i is to the *left* of a_j and that a_j is to the *right* of a_i. These notions of left and right do not always correspond to our normal understanding of these notions when looking at an upwards embedding. For example, in Fig. 4.5, the x-coordinate of a_1 is greater than that of a_2, though $a_1 <_{LR} a_2$ and hence a_1 is to the left of a_2. We consistently use left and right with respect to the order $\leq_{LR}$.

A *bottom chain* is any chain of bottom nodes, and a *top chain* is any chain of top nodes. In Fig. 4.5, the set $\{a_1, a_3, a_4\}$ is a bottom chain, while the set $\{a_2, a_3, a_5\}$ is not. If C is a chain of P and $u, v \in V$, then the *closed interval* from u to v is the subchain $C[u, v] = \{w \in C \mid u \leq_P w \leq_P v\}$, and the *open interval* from u to v is the subchain $C(u,v) = \{w \in C \mid u <_P w <_P v\}$. Subchains $C(u,v]$ and $C[u,v)$, the corresponding *half-open intervals*, are defined analogously. For any bottom chain C, the *extent* of C is

$$\langle C\rangle = \left(\max_{a_i \in C} i\right) - \left(\min_{a_j \in C} j\right);$$

that is, the extent is the distance from the leftmost node in C to the rightmost node in C, measured in rainbow arcs. The extent of a top chain is defined analogously. Suppose C is any chain. We say that C *covers* the nodes it contains. If D is a path in $\vec{H}(P)$ that contains every node of C, then D *covers* C. Note that there must be at least one path in $\vec{H}(P)$ that covers C.

In what follows, we show that more than $k/4$ chains are required to cover the set $A \cup B$. Since $W(P)$ is the minimum number of chains required to cover all the nodes in the poset, it follows that $k/4 < W(P)$ and therefore $QN(P) < 4W(P)$. As the proof is long and tedious, we give here an informal overview. Start with a partition $\mathcal{C}_A$ of A into bottom chains and a partition $\mathcal{C}_B$ of B into top chains. Because each element of $\mathcal{C}_A \cup \mathcal{C}_B$ is a chain, there is a path in $\vec{H}(P)$ covering it. Thinking of each such path as a vertex, we construct a graph G that contains an edge connecting a pair of vertices if the corresponding paths in $\vec{H}(P)$ are connected by a rainbow arc. It is easy to see that G is planar if the paths in $\vec{H}(P)$ covering the chains in $\mathcal{C_A} \cup \mathcal{A_B}$ are pairwise nonintersecting. The construction of a collection of pairwise nonintersecting paths that cover the chains of $\mathcal{C}_A \cup \mathcal{C}_B$ is not always possible. This leads us to the weaker notion of a crossing of two chains and to the construction of G from chains rather than paths. Since the final step of the proof requires G to be planar, we first show (Lemmas 4.7 and 4.8) that all crossings between pairs of chains can be eliminated. Applying Euler's formula to the resulting planar G finally yields the bound in Theorem 4.5.

At this point, we restrict our argument to bottom nodes, as the corresponding argument for top nodes is similar. If C is any bottom chain, the order in which its elements appear with respect to $\leq_P$ is constrained by the rainbow arcs. In particular, we make the following observation.

Observation 1. Suppose that C is a bottom chain whose nodes occur in the following order with respect to $\leq_P$:

$$c_1 \leq_P c_2 \leq_P \cdots \leq_P c_t.$$

For any i with $1 \leq i \leq t-1$, if $c_i <_{LR} c_{i+1}$, then $c_i <_{LR} c_j$ for all $j \geq i+1$. Similarly, for any i with $1 \leq i \leq t-1$, if $c_i >_{LR} c_{i+1}$, then $c_i >_{LR} c_j$ for all $j \geq i+1$.

Intuitively, if the chain starts going to the right after c_i, then the remainder of the chain must be to the right of the rainbow arc of c_i. The rainbow arc of c_i is a barrier to the chain reaching a bottom node to the left of c_i. For example, in Fig. 4.5, the rainbow arc (a_5, b_5) is a barrier to any path originating at a_6. Since $a_5 <_P a_6$ and $a_5 <_{LR} a_6$, no bottom chain containing both a_5 and a_6 has a node $a_i >_P a_6$ to the left of a_5.

By Lemma 3.7, there is a partition of A into at most $W(P)$ chains. Let $\mathcal{C}_A$ be such a partition. Let $C_1 \in \mathcal{C}_A$ have the order

$$c_1 \leq_P c_2 \leq_P \cdots \leq_P c_m,$$

and let $C_2 \in \mathcal{C}_A$, $C_1 \neq C_2$, have the order

$$d_1 \leq_P d_2 \leq_P \cdots \leq_P d_n.$$

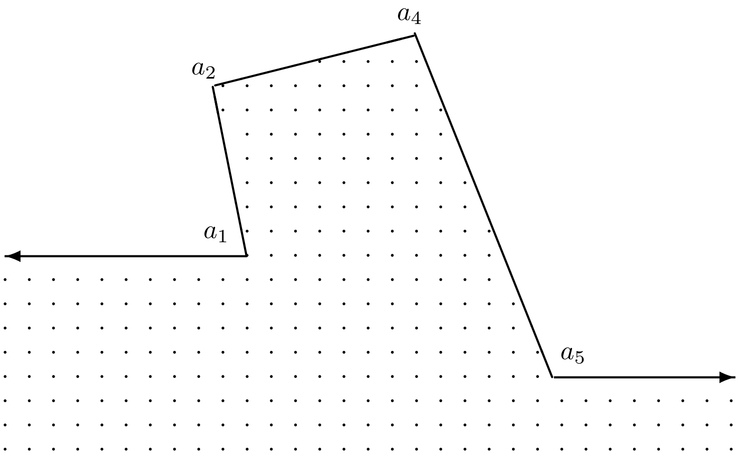


FIG. 4.6. *The region R.*

These two bottom chains *cross* if there exist $c_p, c_q \in C_1$ and $d_r, d_s \in C_2$ such that $c_p <_{LR} d_r <_{LR} c_q <_{LR} d_s$ or $c_p >_{LR} d_r >_{LR} c_q >_{LR} d_s$; in such a case, the 4-tuple (c_p, c_q, d_r, d_s) is a *crossing* of C_1 and C_2. Since c_p and c_q are related by $\leq_P$, there is a directed path D_1 in $\vec{H}(P)$ between c_p and c_q that covers $C_1[c_p, c_q]$. Similarly, there is a directed path D_2 in $\vec{H}(P)$ between d_r and d_s that covers $C_2[d_r, d_s]$.

LEMMA 4.6. *D_1 and D_2 have at least one node in common.*

Proof. Without loss of generality, assume that $c_p <_{LR} d_r <_{LR} c_q <_{LR} d_s$. Consider the polygonal path consisting of the horizontal ray from c_p to $-\infty$, followed by the line segments (c_p, d_r), (d_r, c_q), and (c_q, d_s), and completed by the horizontal ray from d_s to ∞. Let R be the region of the plane consisting of this polygonal path and all points below it. (Figure 4.6 illustrates the region R derived from Fig. 4.5 with crossing (a_1, a_4, a_2, a_5).) Topologically, R is a 2-dimensional ball with a single boundary point removed. Topologically, D_1 and D_2 are paths in the plane with endpoints on the boundary of R. By Observation 1, neither path can cross either of the two infinite rays. Also, neither path can pass above the rainbow arc of c_q or d_r, because every top node is higher than any bottom nodes in the upwards embedding of $\vec{H}(P)$. Hence, if either path crosses one of the three line segments of the polygonal path and proceeds outside of R, then that path must return to the polygonal path at a higher point on the *same line segment.* In essence, we can disregard any excursions outside of R and assume, from a topological viewpoint, that both paths remain within R. The nodes of D_1 and D_2 alternate along the polygonal path. Hence, these paths must intersect topologically, and D_1 and D_2 must have at least one node in common. □

A node that D_1 and D_2 have in common is an *intersection* of C_1 and C_2. Note that an intersection need not be a bottom node. In Fig. 4.5, the chains $\{a_1, a_3, a_4\}$ and $\{a_5, a_2\}$ cross and have the intersection v, which is not a bottom node.

Observation 2. Since, with respect to $\leq_P$, an intersection associated with the crossing (c_p, c_q, d_r, d_s) is between c_p and c_q and between d_r and d_s, we have these relations:

$$\min_P\{c_p, c_q\} <_P \max_P\{d_r, d_s\},$$
$$\min_P\{d_r, d_s\} <_P \max_P\{c_p, c_q\}.$$

The following observation is helpful in constructing pairs of noncrossing chains.

Observation 3. Suppose that $C_1 - \{c_i\}$ and C_2 do not cross. If no $d_j \in C_2$ is between c_{i-1} and c_i with respect to $\leq_{LR}$ or if no $d_j \in C_2$ is between c_i and c_{i+1} with respect to $\leq_{LR}$, then C_1 and C_2 do not cross.

We wish to be able to assume that $\mathcal{C}_A$ does not contain a pair of crossing chains. The first of two steps in justifying that assumption is to show that we can replace two crossing chains with two noncrossing chains according to the following lemma. The replacing pair is further constrained to satisfy the five properties in the lemma. The need for properties 1, 2, and 3 is clear. Property 4 states that, if the original pair crosses, then the replacing pair is smaller, in a precise technical sense, than the original pair; hence the process of replacement of a crossing pair by a noncrossing pair cannot be repeated forever. Property 5 allows us to identify the minima in the replacing pair; this property is a technical condition useful only within the inductive proof of the lemma.

LEMMA 4.7. *Suppose C_1 and C_2 are disjoint bottom chains. Then there exists a function $\mathcal{NC}$ that yields a pair of bottom chains $(C_1', C_2') = \mathcal{NC}(C_1, C_2)$ with these properties*:

1. $C_1' \cup C_2' = C_1 \cup C_2$;
2. C_1' *and* C_2' *are disjoint;*
3. C_1' *and* C_2' *do not cross;*
4. *the sum of extents does not increase:*

$$\langle C_1' \rangle + \langle C_2' \rangle \leq \langle C_1 \rangle + \langle C_2 \rangle;$$

if equality holds and if C_1 and C_2 cross, then the minimum extent decreases:

$$\min\{\langle C_1' \rangle, \langle C_2' \rangle\} < \min\{\langle C_1 \rangle, \langle C_2 \rangle\};$$

and

5. *chain minima are preserved:*

$$\begin{array}{rcccl} c_1 & = & \min_P C_1' & = & \min_P C_1, \\ d_1 & = & \min_P C_2' & = & \min_P C_2. \end{array}$$

Proof. In addition to our previous notation for C_1 and C_2, we define

$$\begin{aligned} \alpha &= \min\nolimits_{LR} C_1, \\ \beta &= \max\nolimits_{LR} C_1, \\ \gamma &= \min\nolimits_{LR} C_2, \\ \delta &= \max\nolimits_{LR} C_2. \end{aligned}$$

By Observation 1, either $c_1 = \alpha$ or $c_1 = \beta$, and either $d_1 = \gamma$ or $d_1 = \delta$. If $c_1 = \alpha$, choose a path D_1 from c_1 to β that covers the subchain $C_1[c_1, \beta]$; if $c_1 = \beta$, choose a path D_1 from c_1 to α that covers the subchain $C_1[c_1, \alpha]$. Similarly, if $d_1 = \gamma$, choose a path D_2 from d_1 to δ that covers the subchain $C_2[d_1, \delta]$; if $d_1 = \delta$, choose a path D_2 from d_1 to γ that covers the subchain $C_2[d_1, \gamma]$. By Observation 1, both paths are monotonic with respect to $\leq_{LR}$.

We proceed to show the lemma by induction on the pair (m, n). Recall that m is the cardinality of C_1 and n is the cardinality of C_2. The base cases are all pairs (m, n) with either $m = 1$ or $n = 1$. In these cases, C_1 and C_2 do not cross, and setting $\mathcal{NC}(C_1, C_2) = (C_1, C_2)$ yields the desired pair of bottom chains.

For the inductive case, we assume that $m \geq 2$, that $n \geq 2$, and that the lemma holds for (m', n') whenever $m' < m$ and $n' \leq n$ or whenever $m' \leq m$ and $n' < n$. We show that the lemma then holds for C_1 and C_2. Without loss of generality, we assume $\alpha <_{LR} \gamma$. There are now three main cases depending on the relative order of α, β, γ, and δ with respect to $<_{LR}$.

Case 1. $\alpha <_{LR} \beta <_{LR} \gamma <_{LR} \delta$. In this case, C_1 and C_2 do not cross and the lemma trivially holds.

Case 2. $\alpha <_{LR} \gamma <_{LR} \beta <_{LR} \delta$. In this case, C_1 and C_2 necessarily cross. There are four subcases.

Case 2.1. $c_1 = \alpha$ *and* $d_1 = \gamma$. Paths D_1 and D_2 necessarily contain at least one intersection. Let v be the intersection that occurs first in going from α to β on D_1. The subpath D_1' of D_1 from c_1 to v does not meet the subpath D_2' of D_2 from d_1 to v until v. Hence, unless D_2' consists only of d_1 (that is, $d_1 = v$), one of D_1' and D_2' is above the other in the upwards embedding. D_1' cannot be above D_2', because the rainbow arc of d_1 is a barrier to D_1' going above d_1. Hence, either D_2' consists only of d_1 or D_2' is above D_1'. There are two subcases, depending on the relative order of c_2 and v according to P.

Case 2.1.1. $c_2 <_P v$. Since c_2 is on D_1' and the rainbow arc of c_2 must not be a barrier for D_2', we have $c_2 <_{LR} d_1$. Let $(C_1', C_2') = \mathcal{NC}(C_1 - \{c_2\}, C_2)$. Since $c_1 \in C_1'$, we set $\mathcal{NC}(C_1, C_2) = (C_1' \cup \{c_2\}, C_2')$. For this case only, we provide a full proof that the lemma holds for $\mathcal{NC}(C_1, C_2)$, leaving the details for the remaining cases to the reader. We employ the properties that hold for (C_1', C_2') by the inductive hypothesis. By property 5 of the inductive hypothesis, C_1' and C_2' are bottom chains with $c_1 \in C_1'$ and $d_1 \in C_2'$. We must show that $C_1' \cup \{c_2\}$ is a bottom chain. If $d_j \in C_2[v, d_n]$, we have $c_2 <_P d_j$. If $d_j \in C_2[d_1, v)$ and $c_1 <_P d_j$, then $c_2 <_P d_j$, since any path in $\vec{H}(P)$ between c_1 and d_j must cross D_1' between c_2 and v. In any case, for any $d_j \in C_2'$, if $c_1 <_P d_j$, then $c_2 <_P d_j$. Hence, $(C_1' \cup \{c_2\}, C_2')$ is a pair of bottom chains, as required. We now establish that $\mathcal{NC}(C_1, C_2)$ satisfies the 5 properties.

1. By property 1 of the inductive hypothesis, $C_1' \cup C_2' = (C_1 - \{c_2\}) \cup C_2$. Hence, $(C_1' \cup \{c_2\}) \cup C_2' = C_1 \cup C_2$.
2. By property 2 of the inductive hypothesis, C_1' and C_2' are disjoint. Since $c_2 \notin C_1' \cup C_2'$, $C_1' \cup \{c_2\}$ and C_2' are disjoint.
3. By property 3 of the inductive hypothesis, C_1' and C_2' do not cross. Since $c_2 <_{LR} d_1$, there is no node of C_2 between c_1 and c_2. Also, by Observation 1 there is no node in C_1 that is between c_1 and c_2. Therefore there is no node in C_2' between c_1 and c_2 and hence by Observation 3, $C_1' \cup c_2$ and C_2' do not cross. Since there is no node of C_2' between c_1 and c_2 with respect to $\leq_{LR}$, $C_1' \cup \{c_2\}$ and C_2' do not cross by Observation 3.
4. To be definite, let $\alpha = a_p$, $\beta = a_q$, $\gamma = a_r$, and $\delta = a_s$. Then, by property 4 of the induction hypothesis and the fact that $c_1 <_{LR} c_2 <_{LR} \beta$, we have

$$\begin{aligned} \langle C_1' \rangle + \langle C_2' \rangle &\leq \langle C_1 - \{c_2\} \rangle + \langle C_2 \rangle \\ &= (q - p) + (s - r) \\ &= \langle C_1 \rangle + \langle C_2 \rangle, \end{aligned}$$

and, if equality holds and if $C_1 - \{c_2\}$ and C_2 cross,

$$\min\{\langle C_1' \rangle, \langle C_2' \rangle\} < \min\{\langle C_1 - \{c_2\} \rangle, \langle C_2 \rangle\}.$$

If $\langle C_1' \rangle + \langle C_2' \rangle < \langle C_1 \rangle + \langle C_2 \rangle$, then we are done. So assume that $\langle C_1' \rangle + \langle C_2' \rangle = \langle C_1 \rangle + \langle C_2 \rangle$. Calculate $\langle C_1 \rangle = q - p$ and $\langle C_2 \rangle = s - r$. We obtain

$\langle C_1'\rangle + \langle C_2'\rangle = (q-p)+(s-r)$. If $\delta \in C_2'$, then $\langle C_2'\rangle = s-r = \langle C_2\rangle$, $\langle C_1'\rangle = \langle C_1\rangle = q-p$, and $a_q = \beta \in C_1'$, a contradiction to C_1' and C_2' not crossing. Hence, $\delta \in C_1'$, $\langle C_1'\rangle = s-p$, $\langle C_2'\rangle = q-r$, and $\langle C_1' \cup \{c_2\}\rangle = s-p$. Then we have

$$\begin{aligned}\min\{\langle C_1' \cup \{c_2\}\rangle, \langle C_2'\rangle\} &= \min\{s-p, q-r\}\\ &= q-r\\ &< \min\{q-p, s-r\}\\ &= \min\{\langle C_1\rangle, \langle C_2\rangle\}.\end{aligned}$$

Hence, property 4 holds for $(C_1' \cup \{c_2\}, C_2')$.

5. By property 5 of the inductive hypothesis, $c_1 = \min_P C_1' = \min_P C_1 - \{c_2\}$ and $d_1 = \min_P C_2' = \min_P C_2$. Since $c_1 <_P c_2$, we obtain $c_1 = \min_P C_1' \cup \{c_2\} = \min_P C_1$ and $d_1 = \min_P C_2' = \min_P C_2$, as required.

This completes the full proof for the case $c_2 <_P v$.

Case 2.1.2. $v \leq_P c_2$. Hence, $d_1 <_{LR} c_2$. For this case, let $c_1 = a_p$, $c_2 = a_q$, $\beta = a_x$, $d_1 = a_r$, $d_2 = a_s$, and $\delta = a_y$. We have $p < r < q \leq x < y$ and $r < s \leq y$. Consider the relative left-to-right positions of c_2 and d_2.

First suppose that $d_2 <_{LR} c_2$. Since $d_2 <_{LR} c_2 <_{LR} \delta$, no node in $C_2(d_2, d_n]$ is between d_1 and d_2. Since the subpath of D_2 from d_2 to δ must go below or through c_2, c_2 must be above d_2 in the vertical order. Hence, no node of C_1 is between d_1 and d_2. Let $(C_1', C_2') = \mathcal{NC}(C_1, C_2[d_2, d_n])$. By Observation 3, $(C_1', C_2' \cup \{d_1\})$ is a pair of noncrossing chains. Set $\mathcal{NC}(C_1, C_2) = (C_1', C_2' \cup \{d_1\})$. We need to show that $(C_1', C_2' \cup \{d_1\})$ satisfies property 4. By property 4 of the inductive hypothesis,

$$\langle C_1'\rangle + \langle C_2'\rangle \leq \langle C_1\rangle + \langle C_2[d_2, d_n]\rangle.$$

Calculate $\langle C_1\rangle = x-p$, $\langle C_2\rangle = y-r$, $\langle C_2[d_2, d_n]\rangle = y-s$, and

$$\begin{aligned}\langle C_1'\rangle + \langle C_2' \cup \{d_1\}\rangle &= \langle C_1'\rangle + \langle C_2'\rangle + (s-r)\\ &\leq \langle C_1\rangle + \langle C_2[d_2, d_n]\rangle + (s-r)\\ &= (x-p)+(y-s)+(s-r)\\ &= (x-p)+(y-r)\\ &= \langle C_1\rangle + \langle C_2\rangle.\end{aligned}$$

If $\langle C_1'\rangle + \langle C_2' \cup \{d_1\}\rangle < \langle C_1\rangle + \langle C_2\rangle$, then property 4 holds. So assume $\langle C_1'\rangle + \langle C_2' \cup \{d_1\}\rangle = \langle C_1\rangle + \langle C_2\rangle$. If $|C_1'| = 1$ (that is $C_1' = \{c_1\}$), then $1 = \min\{\langle C_1'\rangle, \langle C_2' \cup \{d_1\}\rangle\} < 2 \leq \min\{\langle C_1\rangle, \langle C_2\rangle\}$, and again property 4 holds. Otherwise, $|C_1'| \geq 2$. Since $d_2 \in C_2'$ and C_1' and C_2' do not cross, $\delta \in C_1'$. Hence, $\langle C_1'\rangle = y-p$, $\langle C_2'\rangle = x-s$, and $\langle C_2' \cup \{d_1\}\rangle = x-r$. We have

$$\begin{aligned}\min\{\langle C_1'\rangle, \langle C_2' \cup \{d_1\}\rangle\} &= x-r\\ &< \min\{x-p, y-r\}\\ &= \min\{\langle C_1\rangle, \langle C_2\rangle\}.\end{aligned}$$

Hence, property 4 holds.

Now suppose that $c_2 <_{LR} d_2$. There are finally three subcases to consider.

Case 2.1.2.1. *$d_2 <_{LR} \delta$ and $c_2 <_{LR} \beta$.* Let $(C_1', C_2') = \mathcal{NC}(\{c_1\} \cup C_2[d_2, d_n], C_1[c_2, c_m])$. There are no nodes of $C_1 \cup C_2$ between d_1 and c_2. So set

$\mathcal{NC}(C_1, C_2) = (C_1', C_2' \cup \{d_1\})$. By Observation 3, $\{d_1\} \cup C_2'$ is a chain that does not cross C_1'. By property 4 of the inductive hypothesis,

$$\langle C_1' \rangle + \langle C_2' \rangle \leq \langle \{c_1\} \cup C_2[d_2, d_n] \rangle + \langle C_1[c_2, c_m] \rangle,$$

and, if equality holds, then either $\{c_1\} \cup C_2[d_2, d_n]$ and $C_1[c_2, c_m]$ do not cross, or

$$\min\{\langle C_1' \rangle, \langle C_2' \rangle\} < \min\{\langle \{c_1\} \cup C_2[d_2, d_n] \rangle, \langle C_1[c_2, c_m] \rangle\}.$$

We proceed to show that property 4 holds for C_1' and $\{d_1\} \cup C_2'$. Since $\langle \{d_1\} \cup C_2' \rangle = \langle C_2' \rangle + (q - r)$, we have

$$\begin{aligned}\langle C_1' \rangle + \langle \{d_1\} \cup C_2' \rangle &= \langle C_1' \rangle + \langle C_2' \rangle + (q - r)\\ &\leq \langle \{c_1\} \cup C_2[d_2, d_n] \rangle + \langle C_1[c_2, c_m] \rangle + (q - r)\\ &= (\langle C_2 \rangle - (s - r) + (s - p)) + (\langle C_1 \rangle - (q - p)) + (q - r)\\ &= \langle C_1 \rangle + \langle C_2 \rangle,\end{aligned}$$

and hence

$$\langle C_1' \rangle + \langle \{d_1\} \cup C_2' \rangle \leq \langle C_1 \rangle + \langle C_2 \rangle.$$

If this inequality is strict, then property 4 holds. If equality holds, then one of two possibilities holds. First suppose that $\{c_1\} \cup C_2[d_2, d_n]$ and $C_1[c_2, c_m]$ do not cross. In that case, we have $\beta <_{LR} d_2 <_{LR} \delta$ and

$$\begin{aligned}\min\{\langle C_1' \rangle, \{d_1\} \cup \langle C_2' \rangle\} &= \min\{y - p, x - r\}\\ &= x - r\\ &< \min\{x - p, y - r\}\\ &= \min\{\langle C_1 \rangle, \langle C_2 \rangle\}.\end{aligned}$$

Second suppose that

$$\begin{aligned}\min\{\langle C_1' \rangle, \langle C_2' \rangle\} &< \min\{\langle \{c_1\} \cup C_2[d_2, d_n] \rangle, \langle C_1[c_2, c_m] \rangle\}\\ &= x - q.\end{aligned}$$

Then

$$\begin{aligned}\min\{\langle C_1' \rangle, \langle \{d_1\} \cup C_2' \rangle\} &\leq \min\{\langle C_1' \rangle, \langle C_2' \rangle\} + (q - r)\\ &< (x - q) + (q - r)\\ &= x - r\\ &< \min\{\langle C_1 \rangle, \langle C_2 \rangle\}.\end{aligned}$$

For both possibilities, property 4 holds. We conclude that $\mathcal{NC}(C_1, C_2) = (C_1', \{d_1\} \cup C_2')$ gives the desired pair of chains.

Case 2.1.2.2. *$d_2 <_{LR} \delta$ and $c_2 = \beta$.* Since $c_1 <_{LR} d_2$ and $d_1 <_{LR} c_2$, both $\{c_1\} \cup C_2[d_2, d_n]$ and $\{d_1\} \cup C_1[c_2, c_m]$ are chains, and they do not cross. Setting $\mathcal{NC}(C_1, C_2) = (\{c_1\} \cup C_2[d_2, d_n], \{d_1\} \cup C_1[c_2, c_m])$ gives the desired pair of chains. Since

$$\begin{aligned}\langle \{c_1\} \cup C_2[d_2, d_n] \rangle + \langle \{d_1\} \cup C_1[c_2, c_m] \rangle &= (y - p) + (q - r)\\ &= (x - p) + (y - r)\\ &= \langle C_1 \rangle + \langle C_2 \rangle,\end{aligned}$$

and

$$\begin{aligned}\min\{\langle\{c_1\}\cup C_2[d_2,d_n]\rangle,\langle\{d_1\}\cup C_1[c_2,c_m]\rangle\} &= \min\{y-p,q-r\}\\ &= q-r\\ &< \min\{q-p,y-r\}\\ &= \min\{\langle C_1\rangle,\langle C_2\rangle\},\end{aligned}$$

property 4 holds.

Case 2.1.2.3. $d_2=\delta$. Let $(C_1',C_2')=\mathcal{NC}(C_2[d_2,d_n],\{d_1\}\cup C_1[c_2,c_m])$. Since c_1 is leftmost and d_2 rightmost in $C_1\cup C_2$, the pair $(C_1'\cup\{c_1\},C_2')$ is also noncrossing. Let $a_z=\min_{LR}C_2[d_2,d_n]$. We have $r<z\le y$ and $\langle C_2[d_2,d_n]\rangle=y-z$. By property 4 of the inductive hypothesis, we have

$$\begin{aligned}\langle C_1'\rangle+\langle C_2'\rangle &\le \langle C_2[d_2,d_n]\rangle+\langle\{d_1\}\cup C_1[c_2,c_m]\rangle\\ &= (y-z)+(x-r).\end{aligned}$$

We proceed to show property 4 for $(C_1'\cup\{c_1\},C_2')$. First,

$$\begin{aligned}\langle C_1'\cup\{c_1\}\rangle+\langle C_2'\rangle &= \langle C_1'\rangle+\langle C_2'\rangle+(z-p)\\ &\le \langle C_2[d_2,d_n]\rangle+\langle\{d_1\}\cup C_1[c_2,c_m]\rangle+(z-p)\\ &= (y-z)+(x-r)+(z-p)\\ &= (x-p)+(y-r)\\ &= \langle C_1\rangle+\langle C_2\rangle.\end{aligned}$$

If this inequality is strict, then we are done. Otherwise, $\langle C_1'\cup\{c_1\}\rangle+\langle C_2'\rangle=(x-p)+(y-r)$ and $\langle C_2'\rangle=x-r$. We have

$$\begin{aligned}\min\{\langle C_1'\cup\{c_1\}\rangle,\langle C_2'\rangle\} &= \min\{y-p,x-r\}\\ &= x-r\\ &< \min\{y-r,x-p\}\\ &= \min\{\langle C_1\rangle,\langle C_2\rangle\}.\end{aligned}$$

Hence, Property 4 holds for $(C_1'\cup\{c_1\},C_2')$.

Case 2.2. $c_1=\alpha$ *and* $d_1=\delta$. In this case, C_1 and C_2 always cross. If we succeed in replacing these with two noncrossing chains C_1' and C_2' having the same nodes, then $\max_{LR}C_1'<\min_{LR}C_2'$. Hence, property 4 follows easily for every $(C_1',C_2')=\mathcal{NC}(C_1,C_2)$ constructed for this case.

Again, let v be the first intersection of D_1 and D_2. If $v\in A$, then all of $C_1(v,c_m]$ is to the right of v, and all of $C_2(v,d_n]$ is to the left of v. If $v\notin C_1\cup C_2$, then setting $\mathcal{NC}(C_1,C_2)=(C_1[c_1,v)\cup C_2(v,d_n],C_2[d_1,v)\cup C_1(v,c_m])$ gives the desired pair of chains. If $v\in C_1\cup C_2$, then setting $\mathcal{NC}(C_1,C_2)=(C_1[c_1,v]\cup C_2(v,d_n],C_2[d_1,v)\cup C_1(v,c_m])$ gives the desired pair of chains. In either case, C_1' and C_2' do not cross.

If $v\notin A$, then the argument is a bit more involved. Otherwise, if $c_2<_{LR}\gamma$, then let $(C_1',C_2')=\mathcal{NC}(C_1-\{c_2\},C_2)$. Setting $\mathcal{NC}(C_1,C_2)=(C_1'\cup\{c_2\},C_2')$ gives the desired pair of chains. If $\beta<_{LR}d_2$, then let $(C_1',C_2')=\mathcal{NC}(C_1,C_2-\{d_2\})$. Setting $\mathcal{NC}(C_1,C_2)=(C_1',C_2'\cup\{d_2\})$ gives the desired pair of chains. Hence, suppose $\gamma<_{LR}c_2$ and $d_2<_{LR}\beta$. Since the rainbow arcs of c_2 and d_2 are barriers, we have $v\le_P c_2$, $d_2\le_P v$, and $d_2<_{LR}c_2$. By Observation 1, there are four possibilities.

Case 2.2.1. *$C_1(c_2, c_m]$ is to the left of c_2 and $C_2(d_2, d_n]$ is to the left of d_2.* If $C_1(c_2, c_m]$ remains to the right of d_2, then set $\mathcal{NC}(C_1, C_2) = (\{c_1\} \cup C_2[d_2, d_n], \{d_1\} \cup C_1[c_2, c_m])$. Otherwise, (c_2, c_m, d_1, d_2) is a crossing, and, by Observation 2, $c_2 <_P d_j$, for all $j \geq 2$. Hence $C_2 \cup \{c_2\}$ is a chain, and there are no nodes of $C_1 \cup C_2$ between c_2 and d_1. Let $(C_1', C_2') = \mathcal{NC}(C_1 - \{c_2\}, C_2)$. Setting $\mathcal{NC}(C_1, C_2) = (C_1', C_2' \cup \{c_2\})$ gives the desired pair of chains.

Case 2.2.2. *$C_1(c_2, c_m]$ is to the left of c_2 and $C_2(d_2, d_n]$ is to the right of d_2.* Let $(C_1', C_2') = \mathcal{NC}(C_1[c_2, c_m], C_2[d_2, d_n])$. Setting $\mathcal{NC}(C_1, C_2) = (\{c_1\} \cup C_2', \{d_1\} \cup C_1')$ gives the desired pair of chains.

Case 2.2.3. *$C_1(c_2, c_m]$ is to the right of c_2 and $C_2(d_2, d_n]$ is to the left of d_2.* Here $C_1[c_2, c_m]$ and $C_2[d_2, d_n]$ do not cross. Setting $\mathcal{NC}(C_1, C_2) = (\{c_1\} \cup C_2[d_2, d_n], \{d_1\} \cup C_1[c_2, c_m])$ gives the desired pair of chains.

Case 2.2.4. *$C_1(c_2, c_m]$ is to the right of c_2 and $C_2(d_2, d_n]$ is to the right of d_2.* This is the left-to-right mirror image of 2.2.1. The same argument applies, *mutatis mutandis.*

Case 2.3. *$c_1 = \beta$ and $d_1 = \gamma$.* This case cannot occur because the rainbow arcs of c_1 and d_1 are barriers to the paths D_1 and D_2. It would require both D_1 to go below d_1 and D_2 to go below c_1, which is impossible.

Case 2.4. *$c_1 = \beta$ and $d_1 = \delta$.* This case is the left-to-right mirror image of Case 2.1. The same argument applies, *mutatis mutandis.*

Case 3. *$\alpha <_{LR} \gamma <_{LR} \delta <_{LR} \beta$.* In this case, C_1 and C_2 may cross. There are again four subcases.

Case 3.1. *$c_1 = \alpha$ and $d_1 = \gamma$.* First suppose $c_2 <_{LR} d_1$. Let $(C_1', C_2') = \mathcal{NC}(C_1 - \{c_1\}, C_2)$. The desired pair of chains is $\mathcal{NC}(C_1, C_2) = (C_1' \cup \{c_1\}, C_2')$. Suppose that $d_1 <_{LR} c_2 <_{LR} \delta$. Then D_1 and D_2 necessarily have an intersection before c_2 and before δ. This is handled as in Case 2.1. Suppose that $\delta <_{LR} c_2$ and $c_2 \neq \beta$. Then $C_1(c_2, c_m]$ is to the right of c_2, C_2 is between c_1 and c_2, and C_1 and C_2 do not cross. Finally, suppose $\delta <_{LR} c_2$ and $c_2 = \beta$. Let $(C_1', C_2') = \mathcal{NC}(C_1 - \{c_1\}, C_2)$. Since all of $C_1(c_2, c_m] \cup C_2$ is between c_1 and c_2 with respect to $\leq_{LR}$, and since $c_2 \in C_1'$, it follows that $C_1' \cup \{c_1\}$ and C_2' do not cross. The desired pair of chains is $\mathcal{NC}(C_1, C_2) = (C_1' \cup \{c_1\}, C_2')$. It is necessary to justify property 4. Let $\tau = \min_{2 \leq i \leq m} c_i$, where min is taken with respect to $\leq_{LR}$. There are three subcases.

Case 3.1.1. *$\tau <_{LR} \gamma$.* Note that all of $C_2(d_1, d_n]$ is to the right of $d_1 = \gamma$. If τ and γ are unrelated with respect to $\leq_P$ or if $\tau <_P \gamma$, then $\tau \notin C_2'$, since $\gamma = \min_P C_2'$. If $\gamma <_P \tau$, then τ, being to the left of γ, is unrelated to every node in $C_2[d_2, d_n]$; again $\tau \notin C_2'$. Since $\tau \in C_1'$, we have $\langle C_1' \rangle = \langle C_1[c_2, c_m] \rangle$. Applying property 4, we must have $\langle C_2' \rangle < \langle C_2 \rangle$ if $C_1[c_2, c_m]$ and C_2 cross. It follows that $\langle C_1' \cup \{c_1\} \rangle + \langle C_2' \rangle < \langle C_1 \rangle + \langle C_2 \rangle$ if C_1 and C_2 cross.

Case 3.1.2. *$\gamma <_{LR} \tau <_{LR} \delta$.* For $C_2[c_2, c_m]$ and C_1, this case is the same as Case 2.2. For all the possibilities in that case, we get that $\langle C_2' \rangle < \langle C_2 \rangle$. Hence,

$$\begin{aligned} \langle C_1' \cup \{c_1\} \rangle + \langle C_2' \rangle &= (\beta - \alpha) + \langle C_2' \rangle \\ &< (\beta - \alpha) + \langle C_2 \rangle \\ &= \langle C_1 \rangle + \langle C_2 \rangle, \end{aligned}$$

as desired.

Case 3.1.3. *$\delta <_{LR} \tau$.* In this case, $C_1' = C_1[c_2, c_m]$ and $C_2' = C_2$ do no cross. Hence, neither do $C_1' \cup \{c_1\}$ and C_2'.

Case 3.2. *$c_1 = \alpha$ and $d_1 = \delta$.* In this case, C_1 and C_2 do not cross, as the rainbow arc of d_1 is a barrier to D_1 crossing D_2.

Case 3.3. *$c_1 = \beta$ and $d_1 = \gamma$.* This case is the left-to-right mirror image of Case 3.2.

Case 3.4. *$c_1 = \beta$ and $d_1 = \delta$.* This case is the left-to-right mirror image of Case 3.1. □

The second and last step in justifying the assumption converts any $\mathcal{C}_A$ into a $\mathcal{C}'_A$ that has no pair of crossing chains.

Lemma 4.8. *Suppose $\mathcal{C}_A$ is a set of disjoint bottom chains of minimum cardinality that covers A. Then there exists a set $\mathcal{C}'_A$ of disjoint bottom chains that covers A such that $|\mathcal{C}'_A| = |\mathcal{C}_A|$ and no pair of chains in $\mathcal{C}'_A$ cross.*

Proof. If $\mathcal{C}_A$ contains no pair of crossing chains, then $\mathcal{C}'_A = \mathcal{C}_A$ is the set required for the lemma.

Otherwise, let $C_1, C_2 \in \mathcal{C}_A$ be a pair of chains that cross. By Lemma 4.7, there exist chains C'_1 and C'_2 such that by substituting these chains for C_1 and C_2, we get the set $\mathcal{C}''_A = \mathcal{C}_A \cup \{C'_1, C'_2\} - \{C_1, C_2\}$, which is also a set of bottom chains of minimum cardinality that covers A. By property 4, either

(i) the sum of the extents of chains in $\mathcal{C}''_A$ is strictly less than the sum of the extents of chains in $\mathcal{C}_A$ or,

(ii) $\min\{\langle C'_1\rangle, \langle C'_2\rangle\} < \min\{\langle C_1\rangle, \langle C_2\rangle\}$.

Since every chain has extent at least 0, repeated substitution of a pair of crossing chains by a pair of noncrossing chains must eventually reduce the sum of the extents of the chains. Again, since every chain has extent at least 0, the sum of the extents of the chains cannot reduce infinitely, and hence we must eventually arrive at a set $\mathcal{C}'_A$ that contains no pair of noncrossing chains. This set $\mathcal{C}'_A$ is the set required for the lemma. □

We are finally prepared to prove our main result.

Proof of Theorem 4.5. By Lemma 4.8, we may assume that $\mathcal{C}_A$ contains no pair of crossing chains. Now let $\mathcal{C}_B$ be a partition of B into at most $W(P)$ chains. Similarly, we may assume that $\mathcal{C}_B$ contains no pair of crossing chains.

Consider an arbitrary bottom chain C and an arbitrary top chain C'. It is possible that a rainbow arc connects a node in C to a node in C'. However, it is not possible for more than one rainbow arc to connect C and C', for then one of the rainbow arcs (the “longest” one) would be a transitive arc in $\vec{H}(P)$. For example, in Fig. 4.5, we cannot have a bottom chain $C = \{a_1, a_2\}$ and a top chain $C' = \{b_1, b_2\}$, for then there is a path from b_2 to b_1 and (a_1, b_1) is a transitive arc.

We now construct a bipartite graph $G = (\mathcal{C}_A, \mathcal{C}_B, E)$, where E contains an edge between $C \in \mathcal{C}_A$ and $C' \in \mathcal{C}_B$ if there is a rainbow arc connecting C to C'. Since every rainbow arc connects exactly one bottom chain to exactly one top chain, there is exactly one edge in G for every rainbow arc; that is, $|E| = k$. Since there is no pair of crossing bottom chains and no pair of crossing top chains, G is planar. As an example, Fig. 4.7 illustrates a graph $G = (\mathcal{C}_A, \mathcal{C}_B, E)$ obtained from the poset of Fig. 4.5. In particular,

$$\mathcal{C}_A = \Big\{\{a_1, a_2\}, \{a_3, a_4\}, \{a_5, a_6\}\Big\}$$

and

$$\mathcal{C}_B = \Big\{\{b_1, b_5\}, \{b_2\}, \{b_3\}, \{b_4\}, \{b_6\}\Big\}.$$

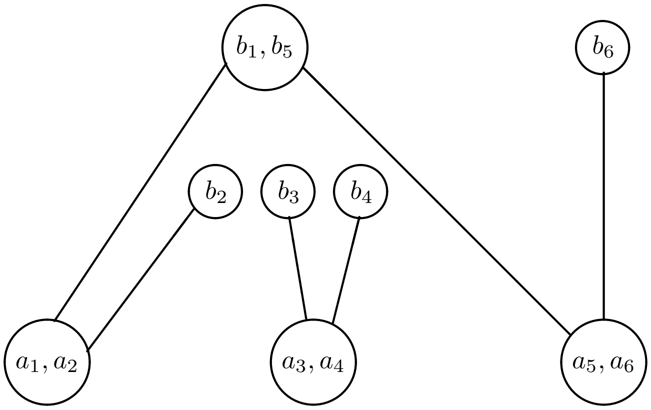


FIG. 4.7. *A bipartite planar graph $G = (\mathcal{C}_A, \mathcal{C}_B, E)$ corresponding to the poset in Fig.* 4.5.

According to Euler's formula for planar graphs, we have

$$|\mathcal{C}_A| + |\mathcal{C}_B| - |E| + f = 1 + t, \tag{4.1}$$

where f is the number of faces in a planar embedding of G and t is its number of connected components. If G consists of a single edge, then $k = 1 \leq W(P)$ and $k < 4W(P)$, as desired. Otherwise, since G is bipartite, we have the following inequality:

$$\begin{aligned} 4f &\leq 2|E|, \\ f &\leq \frac{|E|}{2}. \end{aligned} \tag{4.2}$$

Combining equations 4.1 and 4.2, we obtain

$$\begin{aligned} |\mathcal{C}_A| + |\mathcal{C}_B| - |E| + \frac{|E|}{2} &\geq 1 + t, \\ |\mathcal{C}_A| + |\mathcal{C}_B| &\geq 1 + t + \frac{|E|}{2}, \\ 2 + \frac{|E|}{2} &\leq |\mathcal{C}_A| + |\mathcal{C}_B|. \end{aligned} \tag{4.3}$$

We know that $|E| = k$ and that both $|\mathcal{C}_A|$ and $|\mathcal{C}_B|$ are at most $W(P)$. Substituting into equation 4.3, we obtain

$$k + 4 \leq 4W(P).$$

Hence, the queuenumber of $\vec{H}(P)$ with respect to σ is less than $4W(P)$. □

COROLLARY 4.9. *For any planar poset P where $\vec{H}(P)$ contains at least one arc, $QN(P) < 4W(P)$.*

We believe that this result can be improved to show that, for *any* poset P, there exists a $W(P)$-queue layout of $\vec{H}(P)$; see Conjecture 1 in section 7.

5. Stacknumber of posets with planar covering graphs. In this section we construct, for each $n \geq 1$, a $3n$-element poset R_n such that $H(R_n)$ is planar and hence has stacknumber at most 4 (see Yannakakis [21]), but such that the stacknumber of the class $\mathcal{R} = \{R_n \mid n \geq 1\}$ is not bounded.

THEOREM 5.1. *For each $n \geq 1$, there exists a poset R_n such that $|R_n| = 3n$, $H(R_n)$ is planar, and*

$$\left\lceil \frac{n}{2} \right\rceil \;\leq\; SN(R_n) \;\leq\; n.$$

Proof. Suppose $n \geq 1$. Define three disjoint sets U, V, and W as follows:

$$\begin{aligned} U &= \{u_i \mid 1 \leq i \leq n\}, \\ V &= \{v_i \mid 1 \leq i \leq n\}, \\ W &= \{w_i \mid 1 \leq i \leq n\}. \end{aligned}$$

The poset $R_n = (U \cup V \cup W, \leq)$ is given by

$$\begin{aligned} u_i &< u_{i+1}, \\ v_i &< v_{i+1}, \\ w_i &< w_{i+1}, \end{aligned}$$

for $1 \leq i \leq n-1$,

$$u_i \quad < \quad w_i \quad < \quad v_i,$$

for $1 \leq i \leq n$, and

$$u_n < v_1.$$

Figure 5.1 shows $H(R_4)$.

Aside. While the covering graph $H(R_n)$ is clearly planar, the poset R_n is not planar. This can be seen as follows. In any upward embedding of $\vec{H}(R_n)$ in the plane, the nodes

$$u_1, u_2, \ldots, u_n, v_1, v_2$$

have increasing y-coordinates. Thus, any point in the plane whose y-coordinate is between the y-coordinates of u_1 and v_2 lies either on the left or on the right of the path

$$D = u_1, u_2, \ldots, u_{n-1}, u_n, v_1, v_2.$$

Now add the nodes w_1 and w_2 to the embedding. Their y-coordinates are between the y-coordinates of u_1 and v_2 because of $u_1 < w_1 < v_1 < v_2$ and $u_1 < u_2 < w_2 < v_2$. If both w_1 and w_2 are embedded on the same side of D, then the paths u_1, w_1, v_1 and u_2, w_2, v_2 must cross somewhere. If w_1 and w_2 are embedded on different sides of D, then the line segment (w_1, w_2) crosses a line segment in D. *End Aside.*

To prove the lower bound on $SN(R_n)$, let σ be any topological order on $\vec{H}(R_n)$. The order σ contains the elements of $U \cup V$ in the order $u_1, u_2, \ldots, u_n, v_1, v_2, \ldots, v_n$, and the elements of W in the order $w_1, w_2, \ldots, w_n$. The elements of W are mingled among the elements of $U \cup V$. Suppose $w_1, w_2, \ldots, w_k$ occur before u_n in σ, while $w_{k+1}, w_{k+2}, \ldots, w_n$ occur after u_n. Then the arcs

$$(w_1, v_1), (w_2, v_2), \ldots, (w_k, v_k)$$

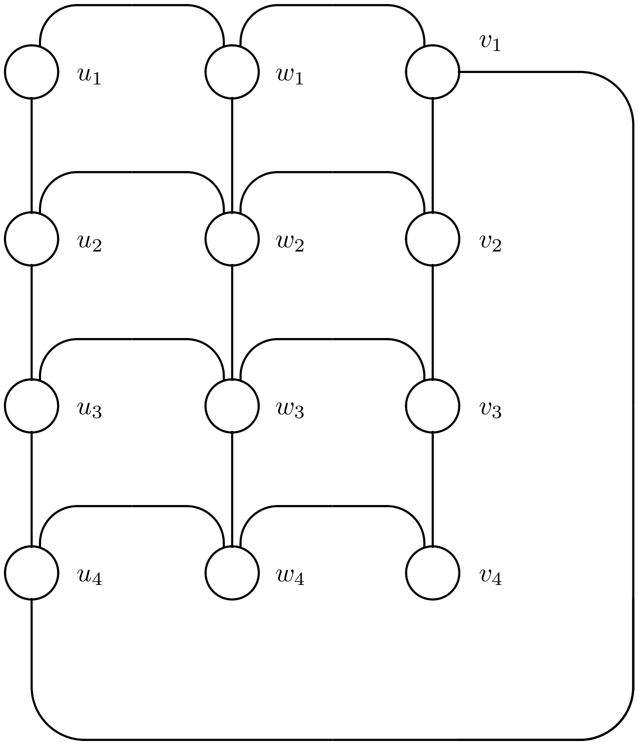


FIG. 5.1. *The covering graph of R_4.*

form a k-twist, while the arcs

$$(u_{k+1}, w_{k+1}), (u_{k+2}, w_{k+2}), \ldots, (u_n, w_n)$$

form an $(n-k)$-twist. Hence,

$$SN(R_n) \geq \max(k, n-k) \geq \lceil n/2 \rceil .$$

Therefore, $SN(R_n) \geq \lceil n/2 \rceil$, as desired.

The proof of the upper bound is constructive. An n-stack layout of R_n is obtained by laying out the elements of $U \cup V$ in the only possible order, and then placing each w_i immediately after u_i for all i, $1 \leq i \leq n$. The assignment of arcs to stacks is as follows. Assign each arc in the set $\{(u_i, w_i), (w_i, v_i), (w_i, w_{i+1})\}$ to stack s_i for all i, $1 \leq i \leq n-1$ and assign each arc in the set $\{(u_n, w_n), (w_n, v_n)\}$ to stack s_n. Note that no two arcs assigned to the same stack intersect. The only arcs remaining to be assigned are the arcs in the set

$$\{(u_i, u_{i+1}) \mid 1 \leq i \leq n-1\} \cup \{(v_i, v_{i+1}) \mid 1 \leq i \leq n-1\} \cup \{(u_n, v_1)\}.$$

The arcs (v_i, v_{i+1}) for i, $1 \leq i \leq n-1$, do not intersect any other arc and can be assigned to any stack. Each arc (u_i, u_{i+1}), $1 \leq i \leq n-1$, is assigned to stack s_{i+1} and arc (u_n, v_1) is assigned to stack s_1. An n-stack layout of R_n is obtained. The upper bound follows. □

Two observations about the poset R_n constructed in the above proof are in order. The first observation is that $QN(R_n) = 2$. A 2-queue layout of R_4 is shown in Fig. 5.2.

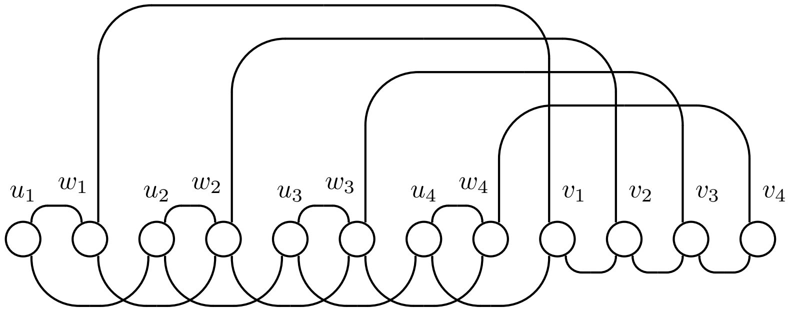


FIG. 5.2. *A 2-queue layout of R_4.*

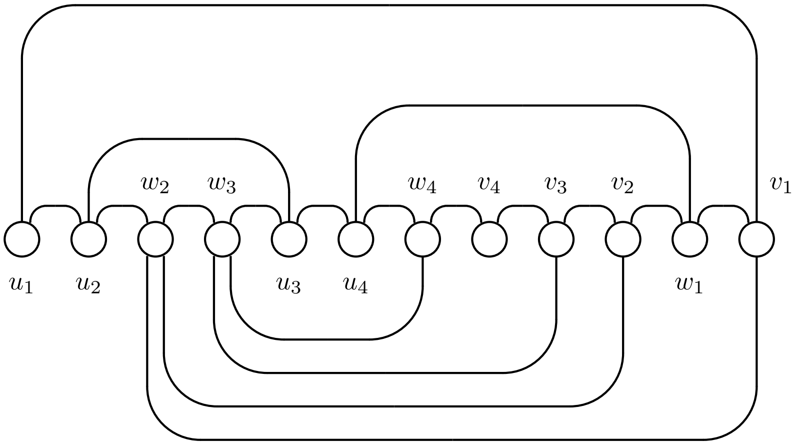


FIG. 5.3. *A 2-stack layout of the covering graph of R_4.*

In general, the total order used in the n-stack layout of R_n described in the above proof yields a 2-queue layout of R_n. The second observation is that the stacknumber and the queuenumber of the covering graph $H(R_n)$ is 2. A 2-stack layout of $H(R_4)$ is shown in Fig. 5.3. In general, a 2-stack layout of $H(R_n)$ can be obtained because $H(R_n)$ is a hamiltonian planar graph [1].

Theorem 5.1 and the above observations lead to the following corollaries.

COROLLARY 5.2. *There exists a class $\mathcal{R} = \{R_n \mid n \geq 1\}$ of posets such that $|R_n| = 3n$, $H(R_n)$ is planar, and*

$$\frac{SN_{\mathcal{R}}(n)}{QN_{\mathcal{R}}(n)} = \Omega(n).$$

COROLLARY 5.3. *There exists a class $\mathcal{R} = \{R_n \mid n \geq 1\}$ of posets R_n such that $|R_n| = 3n$, $H(R_n)$ is planar and*

$$\frac{SN_{\mathcal{R}}(n)}{SN_{H(\mathcal{R})}(n)} = \Omega(n).$$

6. NP-completeness results. Heath and Rosenberg [13] show that the problem of recognizing a 1-queue graph is NP-complete. Since a 1-stack graph is an outerplanar graph, it can be recognized in linear time (Sysło and Iri [18]). But Wigderson

[19] shows that the problem of recognizing a 2-stack graph is NP-complete. Heath and Pemmaraju [9] and Heath, Pemmaraju, and Trenk [11] show that the problem of recognizing a 4-queue poset is NP-complete. They also show that the problem of recognizing a 6-stack dag is NP-complete. We have not been able to extend this NP-completeness result for stack layouts of dags to an analogous result for posets.

Formally, the decision problem for queue layouts of posets is POSETQN.

POSETQN
INSTANCE: A poset P.
QUESTION: Does P have a 4-queue layout?

THEOREM 6.1 ([1, 9]). *The decision problem POSETQN is NP-complete.*

Since the Hasse diagram of a poset is a dag, this result hold for dags in general. This result is in the spirit of the result of Yannakakis [20] that recognizing a 3-dimensional poset is NP-complete.

7. Conclusions and open questions. In this paper, we have initiated the study of queue layouts of posets and have proved a lower bound result for stack layouts of posets with planar covering graph. The upper bounds on the queuenumber of a poset in terms of its jumpnumber, its length, its width, and the queuenumber of its covering graph, proved in section 3, may be useful in proving specific upper bounds on the queuenumber of various classes of posets. We believe that the upper bound of $W(P)^2$ on the queuenumber of an arbitrary poset P, proved in section 3, and the upper bound of $4W(P)$ on the queuenumber of any planar poset P, proved in section 4 are not tight. We have the following conjecture.

CONJECTURE 1. *For any poset P, $QN(P) \leq W(P)$.*

We have established a lower bound of $\Omega(\sqrt{n})$ on the queuenumber of the class of planar posets. We believe that this bound is tight and come to the following conjecture.

CONJECTURE 2. *For any n-element planar poset P, $QN(P) = O(\sqrt{n})$.*

We conjecture that another upper bound on the queuenumber of a planar poset P is given by its length $L(P)$. We believe that it is possible to embed a planar poset in an “almost” leveled-planar fashion with $L(P)$ levels. (See Heath and Rosenberg [13] for a definition of leveled-planar embeddings.) From such an embedding, a queue layout of P in $L(P)$ queues should be obtainable. Therefore we have the following conjecture.

CONJECTURE 3. *For any planar poset P, $QN(P) \leq L(P)$.*

In section 5, we show that the stacknumber of the class of n-element posets having planar covering graphs is $\Theta(n)$. However the stacknumber of the more restrictive class of planar posets is still unresolved.

Acknowledgments. We thank Praveen Paripati for his helpful comments. We are also grateful for the helpful comments of the referees, especially the elucidation of an error in the original statement of Theorem 4.5.

REFERENCES

[1] F. BERNHART AND P. C. KAINEN, *The book thickness of a graph*, J. Combin. Theory Ser. B, 27 (1979), pp. 320–331.

[2] G. BIRKHOFF, *Lattice Theory*, American Mathematical Society, Providence, RI, 1940.

[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. Alg. Discrete Methods, 8 (1987), pp. 33–58.

[4] B. A. Davey and H. A. Priestly, *Introduction to Lattices and Order*, Cambridge University Press, New York, 1990.

[5] R. P. Dilworth, *A decomposition theorem for partially ordered sets*, Ann. of Math., 51 (1950), pp. 161–166.

[6] P. Erdös and G. Szekeres, *A combinatorial problem in geometry*, Compositio Math., 2 (1935), pp. 463–470.

[7] P. C. Fishburn, *Thicknesses of ordered sets*, SIAM J. Discrete Math., 3 (1990), pp. 489–501.

[8] L. S. Heath, F. T. Leighton, and A. L. Rosenberg, *Comparing queues and stacks as mechanisms for laying out graphs*, SIAM J. Discrete Math., 5 (1992), pp. 398–412.

[9] L. S. Heath and S. V. Pemmaraju, *Stack and queue layouts of directed acyclic graphs: Part* II, SIAM J. Comput. Sci., to appear.

[10] L. S. Heath, S. V. Pemmaraju, and C. J. Ribbens, *Sparse Matrix-Vector Multiplication on a Small Linear Array*, Technical Report TR93-11, Dept. of Computer Science, Virginia Polytechnic and State University, Blacksburg, VA, 1993.

[11] L. S. Heath, S. V. Pemmaraju, and A. Trenk, *Stack and queue layouts of directed acyclic graphs*, in Planar Graphs, W. T. Trotter, ed., American Mathematical Society, Providence, RI, 1993, pp. 5–11.

[12] L. S. Heath, S. V. Pemmaraju, and A. Trenk, *Stack and queue layouts of directed acyclic graphs: Part* I, SIAM J. Comput. Sci., to appear.

[13] L. S. Heath and A. L. Rosenberg, *Laying out graphs using queues*, SIAM J. Comput., 21 (1992), pp. 927–958.

[14] L. T. Q. Hung, *A Planar Poset which Requires* 4 *Pages*, Institute of Computer Science, University of Wrocław, typescript, 1989.

[15] R. Nowakowski and A. Parker, *Ordered sets, pagenumbers and planarity*, Order, 6 (1989), pp. 209–218.

[16] S. V. Pemmaraju, *Exploring the Powers of Stacks and Queues via Graph Layouts*, Ph.D. thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, 1992.

[17] M. M. Sysło, *Bounds to the page number of partially ordered sets*, in Proceedings of the 15th International Workshop on Graph-Theoretic Concepts in Computer Science, M. Nagl, ed., Springer-Verlag, Berlin, 1989, pp. 181–195.

[18] M. M. Sysło and M. Iri, *Efficient outerplanarity testing*, Fund. Inform., 2 (1979), pp. 261–275.

[19] A. Wigderson, *The complexity of the Hamiltonian circuit problem for maximal planar graphs*, Tech. Report 82-298, Electrical Engineering and Computer Science Department, Princeton University, Princeton, NJ, 1982.

[20] M. Yannakakis, *The complexity of the partial order dimension problem*, SIAM J. Alg. Discrete Methods, 3 (1982), pp. 351–358.

[21] M. Yannakakis, *Embedding planar graphs in four pages*, J. Comput. System Sci., 38 (1989), pp. 36–67.

SIAM J. DISCRETE MATH.
Vol. 10, No. 4, pp. 626–646, November 1997

MULTIMATROIDS I. COVERINGS BY INDEPENDENT SETS[*]

ANDRÉ BOUCHET[†]

Abstract. Multimatroids are combinatorial structures that generalize matroids and arise in the study of Eulerian graphs. We prove, by means of an efficient algorithm, a covering theorem for multimatroids. This theorem extends Edmonds' covering theorem for matroids. It also generalizes a theorem of Jackson on the Euler tours of a 4-regular graph.

Key words. matroids, Euler tours, coverings

AMS subject classification. 05B35

PII. S0895480193242591

1. Introduction.

Jackson [16, 15] recently solved the following problem in graph theory due to Kotzig [17]. Say that two Euler tours of a connected 4-regular graph are disjoint if no pair of edges is consecutive in both of them. Since there are precisely six pairs of edges incident to each vertex of G and any Euler tour of G uses two of them as consecutive pairs of edges, G has at most three pairwise disjoint Euler tours. The problem is to characterize the 4-regular graphs that realize this upper bound. Jackson's solution implies that the property of having three pairwise disjoint Euler tours is in $NP \cap co - NP$, but it does not give an efficient algorithm to actually construct these Euler tours. The proof relies on an extended submodular inequality and involves quite long computations. Later we presented a structural proof of Jackson theorem with a related efficient algorithm [8]. A generalization of the latter proof is presented here.

A second motivation underlying this paper is the unification of two combinatorial structures, analogous to matroids, which we recently introduced: isotropic systems [6] (used in Jackson's original proof) and delta-matroids [5]. The theory of isotropic systems can be considered as an extension of the theory of binary matroids, whereas delta-matroids extend arbitrary matroids. However delta-matroids do not generalize isotropic systems. For example, a delta-matroid admits two kinds of minors, which correspond to deletions and contractions, whereas an isotropic system admits three kinds of minors.

In section 3 we introduce a new combinatorial structure, which we call multimatroid. For each integer $q \geq 1$ there is a subclass of multimatroids called q-matroids. In particular, 1-matroids can be identified to matroids, 2-matroids are somehow equivalent to delta-matroids, and isotropic systems are particular cases of 3-matroids.

Independent sets, bases, and circuits can be defined in a multimatroid, and they play the same kind of role as in matroid theory. Some multimatroids can be constructed by means of Eulerian graphs. Thus a natural extension of Jackson's theorem is to search for a covering of the ground-set of a multimatroid by a minimal number of independent sets. This problem has been solved by Edmonds [14] for matroids. In the new setting Jackson's theorem and Edmonds' theorem can be seen as identical results for different types of multimatroids. We extend Edmonds' theorem to a large class of multimatroids, which includes q-matroids with $q \geq 3$. Very little is currently

[*] Received by the editors January 11, 1993; accepted for publication September 20, 1996.
http://www.siam.org/journals/sidma/10-4/24259.html

[†] Département d'Informatique, Université du Maine, 72017 Le Mans Cedex, France (bouchet@lium.univ-lemans.fr).

known about the extension of Edmonds' theorem for $q = 2$, and we state some open problems.

The study of multimatroids will be completed in a series of papers [3, 4, 2].

2. Splitters and detachments. Nash-Williams [21, 20] has studied the operation which consists of replacing a vertex v of a graph G by a set of vertices $\{v_1, v_2, \dots, v_k\}$, each edge initially incident to v becoming incident to one of the new vertices v_1, v_2, …, v_k. He defined a detachment of G as a graph obtained by making the preceding operation on the vertices of a subset $W \subseteq V(G)$. We restrict our attention to the case where G is Eulerian and each vertex is replaced by two vertices of even degree. The increase in the number of components, after making the detachment, allows to define a rank function. The properties of this rank function will serve as axioms to define a multimatroid in section 3.

The cardinality of a finite set X is denoted by $|X|$. A set $\{x\}$, of cardinality 1, is often denoted by x.

Basic definitions in graph theory. Our graphs are finite and may contain loops and multiple edges. It is convenient to consider that each edge is incident to two *half-edges*, each half-edge being incident to precisely one vertex. So the *ends* of an edge e are the vertices incident to the half-edges incident to e. The vertex-set and the edge-set of G are denoted by V and E, respectively. For each $v \in V$, we denote by $h(v)$ the set of half-edges incident to v. The *degree* of v is $\deg v = |h(v)|$. The graph G is said to be *Eulerian* (respectively, 4*-regular*) if $\deg v$ is even (respectively, equal to 4) for all $v \in V$.

Given a subset $W \subseteq V$ (respectively, $F \subseteq E$) the subgraph *induced* by G on W (respectively, F), which is denoted by $G[W]$ (respectively, $G[F]$), is obtained by deleting every vertex not in W and every edge whose ends are not in W (respectively, every edge not in F and every vertex that is not an end of an edge in F). If W is a minimal nonempty subset of vertices such that no edge has an end in both W and $V \setminus W$, then $G[W]$ is called a *component* of G. The number of components of G is denoted by $k(G)$. A graph is *connected* if it has just one component. A vertex v of G is called a *cut vertex* if $E(G)$ can be partitioned into two nonempty subsets E_1 and E_2 such that $G[E_1]$ and $G[E_2]$ have just the vertex v in common. A connected graph that has no cut vertex is called a *block*. A *block of* G is a subgraph of G that is a block and is maximal with respect to this property.

Rank of a splitter. We now assume that G is Eulerian. A *local splitter* of G, incident to a vertex v, is a pair $s_v = \{s'_v, s''_v\}$, where s'_v and s''_v are complementary subsets of $h(v)$ having even cardinalities. We call $\{\emptyset, h(v)\}$ the *null local splitter*. If s_v is nonnull the *detachment* of G along s_v is the graph, denoted by $G||s_v$, constructed by replacing v by a vertex v' incident to the half-edges of s'_v and a vertex v'' incident to the half-edges of s''_v. Two local splitters incident to the same vertex v, say $s_v = \{s'_v, s''_v\}$ and $t_v = \{t'_v, t''_v\}$, are *skew* if $|s'_v \cap t'_v|$ is odd (the definition clearly does not depend on the choice of s'_v in s_v and t'_v in t_v).

A *splitter* of G is a set $s = \{s_v : v \in W\}$, where $W \subseteq V(G)$ is the set of vertices *incident* to s and s_v is a nonnull local splitter incident at v. The *detachment* of G along s is the Eulerian graph $G||s = G||s_{v_1}||s_{v_2}||\cdots||s_{v_n}$, where $v_1v_2\cdots v_n$ is an enumeration of W (changing the order of the enumeration does not change $G||s$). The *rank* of s is the integer $r(s) = |s| - k(G||s) + k(G)$.

The union of two disjoint subsets, X and Y, of a set Z is often denoted as $X + Y$. For $x \in Z \setminus X$ we simplify the notation $X \cup \{x\}$ into $X + x$.

Proposition 2.1. *Let s be a splitter of an Eulerian graph G. Let v be a vertex of G not incident to s. The following properties are satisfied:*

(i) $r(\emptyset) = 0$.

(ii) *If s_v is a nonnull local splitter incident to v, then*

$$r(s) \leq r(s + s_v) \leq r(s) + 1.$$

(iii) *If t is a splitter such that $s \cup t$ is also a splitter, then*

$$r(s \cup t) + r(s \cap t) \leq r(s) + r(t).$$

(iv) *If s_v and t_v are skew local splitters incident to v, then*

$$r(s + s_v) - r(s) + r(s + t_v) - r(s) \geq 1.$$

Proof. Property (i) is obvious. Let $s = \{s_v : v \in W\}$ and $G' = G||s$. Property (ii) is equivalent to

$$k(G') \leq k(G'||s_v) \leq k(G') + 1.$$

The first inequality is obvious, and the second one holds because when splitting a vertex into two vertices we augment the number of components by at most 1.

To prove (iii), it is sufficient to verify that

$$\text{(v) } r(s + s_v + s_w) - r(s + s_v) \leq r(s + s_w) - r(s)$$

holds for any pair of local splitters s_v and s_w incident to distinct vertices $v, w \notin W$. Indeed, if we let $s' = s \cap t$, $a = s \setminus t$, and $b = t \setminus s$, then (iii) can be written

$$r(s' + a + b) - r(s' + a) \leq r(s' + b) - r(s'),$$

which can be derived from (v) by induction (first on $|a|$, assuming $|b| = 1$, then on $|b|$). Inequality (v) is equivalent to

$$\text{(vi) } k(G'||s_v||s_w) - k(G'||s_v) \geq k(G'||s_w) - k(G').$$

Assume first that v and w belong to distinct components of G'. We may decompose G' into two proper subgraphs H_1 and H_2, each of them being a union of components, such that v is a vertex of H_1 and w is a vertex of H_2. Then we have

$$\begin{aligned} k(G'||s_v||s_w) - k(G') &= k(H_1||s_v) - k(H_1) + k(H_2||s_w) - k(H_2) \\ &= k(G'||s_v) - k(G') + k(G'||s_w) - k(G'), \end{aligned}$$

which implies (vi). We now assume that v and w belong to the same component Γ of G'. Inequality (vi) is obvious if $k(G'||s_w) = k(G')$. Otherwise w is a cut-vertex of G', and Γ is split into two components Γ' and Γ'' in $G'||s_w$. The vertex w is split into two vertices w' and w'' in $G'||s_w$, and we may assume that $w' \in V(\Gamma')$ and $w'' \in V(\Gamma'')$. The vertex v is incident to precisely one of the components Γ' and Γ'', say Γ'. When constructing $G'||s_w||s_v$ from $G'||s_w$, Γ' is split into at most two components, say Γ'_1 and Γ'_2, allowing $\Gamma'_1 = \Gamma'_2$. One of these components, say Γ'_1, is incident to w'. We can reconstruct $G'||s_v$ from $G'||s_v||s_w$ by identifying w' and w'' into a single vertex w, which merges the two components Γ'' and Γ'_1 into a single component. Thus $k(G'||s_v||s_w) = k(G'||s_v) + 1$, which implies (vi).

Property (iv) is equivalent to

$$k(G'||s_v) - k(G') + k(G'||t_v) - k(G') \leq 1.$$

Suppose that the inequality does not hold. Then we have $k(G'||s_v) > k(G')$ and $k(G'||t_v) > k(G')$. Let $\Gamma_1, \Gamma_2, \ldots, \Gamma_k$ be the blocks of G' incident to v. For $1 \leq i \leq k$ let H_i be the set of half-edges of Γ_i that are incident to v. The set H_i has an even cardinality; otherwise there would be only one vertex of odd degree in Γ_i. Since $G'||s_v$ has more components than G', there exists $I \subseteq \{1, 2, \ldots, k\}$ such that $s_v = \{s'_v, s''_v\}$ with $s'_v = \bigcup(X_i : i \in I)$. Similarly there exists $J \subseteq \{1, 2, \ldots, k\}$ such that $t_v = \{t'_v, t''_v\}$ with $t'_v = \bigcup(H_i : i \in J)$. Thus $|s'_v \cap t'_v| = \sum(|H_i| : i \in I \cap J)$ is even, which is a contradiction because s_v and t_v are skew. □

We say that s is *independent* if $r(s) = |s|$ (so $k(G||s) = k(G)$). We say that s is *complete* if every vertex of G is incident to s (so every vertex of G is split into two vertices in $G||s$). If U is a set of nonnull local splitters of G, then we define a *base* of U as a maximal independent splitter included in U.

COROLLARY 2.2. *Let U be a set of nonnull local splitters of an Eulerian graph G. Assume that each vertex is incident to at least two skew local splitters in U. Then every base of U is complete.*

Proof. Consider a base s. We have $r(s) = |s|$. Assume that s is not complete. Then there exists a vertex v that is not incident to s. According to the assumptions there exist two skew local splitters, s_v and t_v, incident to v. It follows from (iv) in Proposition 2.1 that either $r(s+s_v) \geq r(s)+1 = |s+s_v|$ or $r(s+t_v) \geq r(s)+1 = |s+t_v|$, and so s is not a maximal independent splitter, which is a contradiction. □

Splitters and Euler tours of a 4-regular graph. We now assume, until the end of this section, that G is a connected 4-regular graph. A pair of half-edges incident to the same vertex is called a *transition*. So each local splitter of G is made of two disjoint transitions. There are precisely three local splitters incident to any vertex and these splitters are pairwise skew. We denote by U the set of all the local splitters of G.

Let T be an Euler tour of G. A *transition* of T, incident to a vertex v, is any pair of half-edges successive in T and incident to v. Since v has degree 4, there are precisely two transitions of T incident to v, say s'_v and s''_v. We say that the local splitter $\{s'_v, s''_v\}$ is *used* by T. The set of the local splitters used by T, which is a complete splitter, is denoted by $s(T)$. The succession of the half-edges of T is a circuit in the detachment $G||s(T)$. So we have $k(G) = k(G||s(T)) = 1$, and $s(T)$ is a base of U. Conversely, if s is any base of U, then $G||s$ is a circuit and the succession of the half-edges along this circuit is an Euler tour of G, say T, such that $s = s(T)$. Therefore the mapping $T \mapsto s(T)$ is a bijection from the set of the Euler tours of G onto the set of the bases of U.

Consider now any complete splitter F. Since $U - F$ still satisfies conditions (i) and (ii) of Proposition 2.1, the bases of $U - F$ are complete splitters of G. Note that an Euler tour of G uses no transition in F if and only if it corresponds to a base of $U - F$. So we have the following classical result.

COROLLARY 2.3 (see Kotzig [18]). *Let F be a complete splitter of a connected 4-regular graph G. There exists a Euler tour of G that uses no local splitter in F.*

We say that two Euler tours, T_1 and T_2, are *disjoint* if $s(T_1) \cap s(T_2) = \emptyset$. Since each vertex is incident to precisely three nonnull local splitters there are at most three pairwise disjoint Euler tours in G.

THEOREM 2.4 (see Jackson [15]). *A connected* 4*-regular graph G admits three pairwise disjoint Euler tours if and only if*

$$3(k(G||s)-1) \leq 2|s|$$

holds for all splitters s of G.

This result will be implied by Theorem 6.1 proved in the sequel.

3. Definition of a multimatroid. Consider a partition Ω of a finite set U. Each class $\omega \in \Omega$ is called a *skew class*. Each pair of distinct elements belonging to the same skew class is called a *skew pair*. A *subtransversal* (respectively, *transversal*) of Ω is a subset $S \subseteq U$ such that $|S \cap \omega| \leq 1$ (respectively, $|S \cap \omega| = 1$) holds for all $\omega \in \Omega$. We denote by $\mathcal{S}(\Omega)$ (respectively, $\mathcal{T}(\Omega)$) the set of all subtransversals (respectively, transversals) of Ω.

A *multimatroid* is a triple $Q = (U, \Omega, r)$ with a partition Ω of a finite set U and a *rank function* $r : \mathcal{S}(\Omega) \to \mathbf{N}$ satisfying the four following axioms:

3.1. $r(\emptyset) = 0$.

3.2. *For $A \in \mathcal{S}(\Omega)$ and $x \in U$ such that A is disjoint from the skew class containing x,*

$$r(A) \leq r(A + x) \leq r(A) + 1.$$

3.3. Submodularity inequality: *For $A, B \in \mathcal{S}(\Omega)$ such that $A \cup B \in \mathcal{S}(\Omega)$,*

$$r(A) + r(B) \geq r(A \cup B) + r(A \cap B).$$

3.4. *For $A \in \mathcal{S}(\Omega)$ and any skew pair $\{x, y\}$ included in a skew class disjoint from A,*

$$r(A + x) - r(A) + r(A + y) - r(A) \geq 1.$$

The pair (U, Ω) is called the *carrier* of Q. The *restriction* of Q to a subset $U' \subseteq U$, which we denote by $Q[U']$, is the multimatroid (U', Ω', r') such that $\Omega' = \{\omega \cap U' \neq \emptyset : \omega \in \Omega\}$ and r' is the restriction of r to $\mathcal{S}(\Omega')$. Where $q \geq 1$ is an integer, we call Q a *q-matroid* if all the skew classes have a cardinality equal to q.

Eulerian multimatroids. Let G be an Eulerian graph on the vertex-set V. Choose a set U of local splitters of G such that any two local splitters of U incident to the same vertex are skew. Let Ω_v denote the set of the local splitters of U that are incident to a vertex v. Let $\Omega = \{\Omega_v : v \in V, \Omega_v \neq \emptyset\}$. Note that $\mathcal{S}(\Omega)$ is equal to the set of the splitters of G included in U. Let r be the restriction to $\mathcal{S}(\Omega)$ of the rank function defined on the set of the splitters of G. It follows from Proposition 2.1 that $Q(G, U) = (U, \Omega, r)$ is a multimatroid. We call $Q(G, U)$ an *Eulerian multimatroid.*

The results of the preceding section, when G is 4-regular, can be interpreted as follows. There are precisely three local splitters incident to any vertex of G and these local splitters are pairwise skew. So if we let U be the set of all the local splitters of G, then $Q(G, U)$ is a 3-matroid. If F is any complete splitter of G, then $Q(G, U - F)$ is a 2-matroid. If G is connected, then the bases of $Q(G, U)$ correspond to the Euler tours of G, whereas the bases of $Q(G, U - F)$ correspond to the Euler tours of G using no local splitter in F.

4. Comparison with other combinatorial structures. In this section we show that multimatroids involve matroids, delta-matroids, and isotropic systems. We consider a multimatroid $Q = (U, \Omega, r)$.

Matroids. Let us recall that a *matroid* is a pair $M = (E, r)$ with a finite set E and a *rank function*, $r : \mathcal{P}(E) \to \mathbf{N}$, that satisfies the three following axioms:

4.1. $r(\emptyset) = 0$.

4.2. *For* $A \in \mathcal{P}(E)$ *and* $x \in E - A$,

$$r(A) \leq r(A + x) \leq r(A) + 1.$$

4.3. Submodularity inequality: *For* $A, B \in \mathcal{P}(E)$

$$r(A) + r(B) \geq r(A \cup B) + r(A \cap B).$$

Assume that Q is a 1-matroid. So axiom 3.4 is void, $\mathcal{S}(\Omega) = \mathcal{P}(U)$, and $r(A)$ is defined for all $A \subseteq U$. The first three axioms amount to say that r is the rank function of a matroid. We identify Q to the matroid on U defined by the rank function r. The inverse construction that associates a 1-matroid to a matroid is obvious.

Independent sets, bases and circuits. The similarity between multimatroids and matroids leads us to define independent sets, bases, and circuits, which play similar roles. An *independent set* of Q is a subtransversal I such that $r(I) = |I|$. A *base* is a maximal independent set. A *circuit* is a minimal subtransversal that is not independent. Let $\mathcal{I}(Q)$, $\mathcal{B}(Q)$, and $\mathcal{C}(Q)$ be the sets of the independent sets, bases, and circuits of Q, respectively. For any $A \in \mathcal{S}(\Omega)$,

4.4. $r(A) = \max(|I| : I \subseteq A, I \in \mathcal{I})$

is satisfied. Therefore Q is determined when $\mathcal{I}(Q)$ is known. Accordingly Q is determined when either $\mathcal{B}(Q)$ or $\mathcal{C}(Q)$ is known.

Multimatroid sheltered by a matroid. There is another way to compare multimatroids and matroids. Is it possible to extend the rank function of multimatroid $Q = (U, \Omega, r)$ into a submodular function R defined for every subset of U? Then R is the rank function of a matroid M on U, and we say that M *shelters* Q. The following example, due to Duchamp [13], shows that it is not always possible. Let $U = \{a, a', b, b', c, c', d, d'\}$, $\Omega = \{\{a, a'\}, \{b, b'\}, \{c, c'\}, \{d, d'\}\}$, and $\mathcal{C}(Q) = \{\{a, b', c'\}, \{a', b, c'\}, \{a', b', c\}, \{a, b, c, d'\}\}$. It is not very difficult to check that R cannot exist.

It can be verified that the main applications—for example, to Eulerian graphs—involve multimatroids (U, Ω, r) that can be sheltered by matroids. So, when a problem is stated in terms of multimatroids, it is understood that the sole consideration of the sheltering matroid and the partition matroid associated to Ω does not help to solve the problem. The first remark in section 6 gives a specific example.

Delta-matroids. For two sets A and B we let $A \Delta B = (A \setminus B) \cup (B \setminus A)$ be the symmetric difference of A and B. A *set system* is a pair $(X, \mathcal{F})$ with a finite set X and a subset $\mathcal{F} \subseteq \mathcal{P}(X)$. The set system is said to be *nonempty* if $\mathcal{F} \neq \emptyset$. A *delta-matroid* is a nonempty set system $(X, \mathcal{F})$ satisfying the following *symmetric exchange axiom*:

4.5. *If* $F', F'' \in \mathcal{F}$, *and* $x \in F' \Delta F''$, *then there is* $y \in F' \Delta F''$ *such that* $F' \Delta \{x, y\} \in \mathcal{F}$.

Delta-matroids, and similar structures, have been independently introduced by Chandrasekaran and Kabadi [9], Dress and Havel [11], Qi [22], and Bouchet [5].

Proposition 5.5 in the next section states that any base of a nondegenerate multimatroid on (U, Ω) is a transversal of Ω, provided that each skew class has at least two elements. The following theorem will also be proved in the next section.

THEOREM 4.1. *Let (U, Ω) be a 2-matroid carrier. A nonempty set of transversals $\mathcal{B}$ of Ω is the base-set of a 2-matroid on (U, Ω) if and only if it satisfies the following axiom.*

4.6. Transversal exchange: *If $A', A'' \in \mathcal{B}$, and $p \subseteq A' \Delta A''$ is a skew pair, then there is a skew pair $q \subseteq A' \Delta A''$ such that $A' \Delta (p \cup q) \in \mathcal{B}$.*

Remark. The preceding theorem implies that the structure of a 2-matroid is identical to the structure of a symmetric matroid, introduced in [5].

Given a 2-matroid Q on the carrier (U, Ω) and a transversal X of Ω, the set system $Q \cap X = (X, \mathcal{F})$, where $\mathcal{F} = \{A \cap X : A \in \mathcal{B}(Q)\}$, is called the *section* of Q by X. The 2-matroid Q can be reconstructed, up to an isomorphism, when $Q \cap X$ is known. Indeed, consider a copy $\tilde{X}$ of X satisfying $X \cap \tilde{X} = \emptyset$. Denote by $\tilde{x}$ the copy of any $x \in X$, and, for $F \subseteq X$, let $\tilde{F} = \{\tilde{x} : x \in F\}$. Define the 2-matroid carrier (U, Ω) by $U = X \cup \tilde{X}$ and $\Omega = \{\{x, \tilde{x}\} : x \in X\}$. Then $\{F \cup (\widetilde{X - F}) : F \in \mathcal{F}\}$ is the base-set of a 2-matroid isomorphic to Q (we recall that a multimatroid is determined by its base-set). Using the preceding theorem and the present correspondence between 2-matroids and their sections, it is easy to prove the following result (see [5] for details).

PROPOSITION 4.2. *A set system is a delta-matroid if and only if it is equal to the section of a 2-matroid by a transversal.*

So 2-matroids are another view of delta-matroids.

Isotropic systems. A bilinear form $(A, B) \mapsto \langle A, B \rangle$, defined on a vector space E, is called a *symplectic form* if $\langle A, A \rangle = 0$ holds for all $A \in E$ and no $B \in E - \{0\}$ satisfies $\langle B, A \rangle = 0$ for all $A \in E$. Assume that such a symplectic form is given. Two vectors A and B are said to be *orthogonal* (respectively, *skew*) if $\langle A, B \rangle = 0$ (respectively, $\langle A, B \rangle \neq 0$). A subspace L of E is said to be *totally isotropic* if any two vectors in L are orthogonal. A classical result says that, if L is a maximal totally isotropic subspace of E, $\dim(L) = \dim(E)/2$.

A *binary hyperbolic plane* is a 2-dimensional vector space over GF(2) provided with a symplectic form $(a, b) \mapsto \langle a, b \rangle$. For a direct product of binary hyperbolic planes, $E = \prod(E_v : v \in V)$, and $A, B \in E$ let

$$\langle A, B \rangle = \sum(\langle A_v, B_v \rangle : v \in V).$$

The mapping $(A, B) \mapsto \langle A, B \rangle$ is a symplectic form over E.

An *isotropic system* is defined by a direct product, $E = \prod(E_v : v \in V)$, of binary hyperbolic planes and a maximal totally isotropic subspace L of E. We represent such an isotropic system by the notation $S = (E, L, V)$. (In our original paper [6] we fixed a particular binary hyperbolic plane, denoted by K, and we took $E = K^V$.)

We construct a triple $Q(S) = (U, \Omega, r)$ as follows. Say that a vector $A \in E$ is an *atom* if there exists precisely one element $v \in V$, called the *support* of A, such that $A_v \neq 0$. We let U be the set of the atoms of E. For $v \in V$ we let Ω_v be the set of the atoms supported by v and we note that $|\Omega_v| = 3$ (a binary hyperbolic hyperplane, which has dimension 2, has precisely three nonnull vectors). We let $\Omega = \{\Omega_v : v \in V\}$. For any $s \in \mathcal{S}(\Omega)$ we let $r(s) = |s| - \dim \langle s \rangle \cap L$, where $\langle s \rangle$ denotes the subspace of E generated by s.

Remark. There is a bijective mapping $\alpha : E \to \mathcal{S}(\Omega)$. It satisfies $A = \sum(u : u \in \alpha(A))$.

PROPOSITION 4.3. *If $S = (E, L, V)$ is an isotropic system, then $Q(S)$ is a 3-matroid.*

Proof. We have to verify that $Q = Q(S)$ satisfies axioms 3.1 to 3.4. This is obvious for 3.1. Axiom 3.2 holds because the dimension of $\langle A\rangle$ is increased by at most 1 when replacing A by $A + x$. Axiom 3.3 follows from

$$\begin{aligned}\dim(L \cap \langle A\rangle) + \dim(L \cap \langle B\rangle) &= \dim(L \cap \langle A\rangle + L \cap \langle B\rangle)\\ &\quad + \dim(L \cap \langle A\rangle \cap L \cap \langle B\rangle)\\ &\leq \dim(L \cap \langle A \cup B\rangle) + \dim(L \cap \langle A \cap B\rangle).\end{aligned}$$

Suppose that axiom 3.4 does not hold. Then we can find a subset of atoms $A \in \mathcal{S}(\Omega)$ and a pair of skew atoms, $\{x, y\}$, included in a skew class disjoint from A, such that

$$\dim(L \cap \langle A + x\rangle) - \dim(L \cap \langle A\rangle) + \dim(L \cap \langle A + y\rangle) - \dim(L \cap \langle A\rangle) \geq 2.$$

Consider a vector $P \in L \cap \langle A + x\rangle - L \cap \langle A\rangle$ and a vector $Q \in L \cap \langle A + y\rangle - L \cap \langle A\rangle$. Since the atoms x and y are a skew pair, they have the same support, say w. For $v \in V$ let us denote by A_v the atom contained in A and supported by v if it exists, otherwise let $A_v = 0$. There exists $\lambda \in 2^V$ such that $P = \sum(\lambda_v A_v : v \in V - w) + \lambda_w x$ because $P \in \langle A + x\rangle$. We have $\lambda_w \neq 0$ because $P \notin \langle A\rangle$. Similarly there exists $\mu \in 2^V$ such that $Q = \sum(\mu_v A_v : v \in V - w) + \mu_w y$ and $\mu_w \neq 0$. Then

$$\begin{aligned}\langle P, Q\rangle &= \sum(\lambda_v \mu_v \langle A_v, A_v\rangle : v \in V - w) + \lambda_w \mu_w \langle x, y\rangle\\ &= \lambda_w \mu_w \langle x, y\rangle\end{aligned}$$

because $\langle A_v, A_v\rangle = 0$. Since the atoms x and y have a same support w, we have $\langle x, y\rangle = \langle x_w, y_w\rangle$, which is nonnull because x_w and y_w are distinct nonnull elements and $\langle ., .\rangle$ is a symplectic form over GF(2). This implies $\langle P, Q\rangle \neq 0$, whereas $P, Q \in L$, which is a contradiction. □

Contrary to 1-matroids and 2-matroids, which are new views of already known combinatorial structures, every 3-matroid cannot be derived from an isotropic system. The reader will find details in [3].

Remark. The term multimatroid, which we choose to name the structure, reflects that each restriction $Q[S]$, where S is a subtransversal, is a matroid. This property also occurs by assuming only axioms 3.1 to 3.3. The construction of a multimatroid associated to an isotropic system gives an algebraic interpretation of axiom 3.4. We show in [3] that any two matroids, $Q[s]$ and $Q[t]$, where s and t are disjoint transversals, are somehow orthogonal. Accordingly, a more appropriate name of the full structure is *isotropic multimatroid*. The qualificative isotropic will be implicit (the structure defined by axioms 3.1 to 3.3 is too weak to be interesting).

5. Properties of independent sets, bases, and circuits. Let us recall two classical characterizations of a matroid (see [23] for details).

PROPOSITION 5.1. *A subset $\mathcal{I} \subseteq \mathcal{P}(E)$ is the set of independent sets of a matroid on E if and only if*

(a) $\emptyset \in \mathcal{I}$,

(b) *if $I \in \mathcal{I}$ and $J \subseteq I$, then $J \in \mathcal{I}$,*

(c) Augmentation: *if $I, J \in \mathcal{I}$ and $|I| < |J|$, then $I + x \in \mathcal{I}$ for some $x \in J \setminus I$.*

PROPOSITION 5.2. *A subset $\mathcal{C} \subseteq \mathcal{P}(E)$ is the set of circuits of a matroid on E if and only if*

(a) $\emptyset \notin \mathcal{C}$,

(b) *if $C', C'' \in \mathcal{C}$ and $C' \subseteq C''$, then $C' = C''$,*

(c) Elimination: *if* $C', C'' \in \mathcal{C}$ *and* $x \in C' \cap C''$, *then* $C \subseteq (C' \cup C'') - x$ *for some* $C \in \mathcal{C}$.

The two following characterizations of a multimatroid, by means of independent subsets and circuits, consist of four properties. The first three ones correspond to axioms 3.1 to 3.3, and they amount to say that $Q[S]$ is a matroid for all $S \in \mathcal{S}(\Omega)$. The fourth property corresponds to axiom 3.4. Some ideas used in the proofs of Propositions 5.3, 5.4, and 4.6 come from the thesis of Duchamp [12] on symmetric matroids.

We say that two subtransversals A and B of Ω are *compatible* if $A \cup B$ is also a subtransversal of Ω.

PROPOSITION 5.3. *Let* (U, Ω) *be a multimatroid carrier. A subset* $\mathcal{I} \subseteq \mathcal{S}(\Omega)$ *is the set of the independent sets of a multimatroid on* (U, Ω) *if and only if the following properties are satisfied:*

(a) $\emptyset \in \mathcal{I}$,

(b) *if* $I \in \mathcal{I}$ *and* $J \subseteq I$, *then* $J \in \mathcal{I}$,

(c) Augmentation: *if* $I, J \in \mathcal{I}$ *are compatible and* $|I| < |J|$, *then* $I + x \in \mathcal{I}$ *for some* $x \in J \setminus I$,

(d) *for any* $I \in \mathcal{I}$ *and any pair* $\{x, y\}$ *included in a class* $\omega \in \Omega$ *disjoint from* I, *either* $I + x \in \mathcal{I}$ *or* $I + y \in \mathcal{I}$.

Proof. Suppose that $\mathcal{I}$ is the set of the independent subsets of a multimatroid whose rank function is r. Axiom 3.1 implies (a). Proposition 5.1 (b), applied to the matroid $Q[I]$, implies (b). Proposition 5.1 (c), applied to the matroid $Q[I \cup J]$, implies (c). If condition (d) is not satisfied, we have

$$r(I + x) - r(I) + r(I + y) - r(I) = 0,$$

which contradicts axiom 3.4.

Conversely suppose that $\mathcal{I}$ satisfies (a) to (d). Define r by formula 4.4.

Property (a) implies axiom 3.1. According to (a), (b), (c), and Proposition 5.1, for every $X \in \mathcal{S}(\Omega)$ there is a matroid, say M_X, whose set of independent sets is equal to $\{I : I \subseteq X, I \in \mathcal{I}\}$. According to formula 4.4, the rank function of M_X is equal to the restriction of r to X. Axiom 3.2 is verified in the matroid M_A. Axiom 3.3 is verified in the matroid $M_{A \cup B}$. We now verify axiom 3.4. Let I be a maximal member of $\mathcal{I}$ included in A. According to (d), either $I + x$ or $I + y$ belongs to $\mathcal{I}$. We may suppose $I + x \in \mathcal{I}$. Since $I + x \subseteq A + x$, we have $r(A + x) \geq |I| + 1 = r(A) + 1$, which implies axiom 3.4. □

PROPOSITION 5.4. *Let* (U, Ω) *be a multimatroid carrier. A subset* $\mathcal{C} \subseteq \mathcal{S}(\Omega)$ *is the set of circuits of a multimatroid on* (U, Ω) *if and only if the following properties are satisfied:*

(a) $\emptyset \notin \mathcal{C}$,

(b) *if* $C', C'' \in \mathcal{C}$ *and* $C' \subseteq C''$, *then* $C' = C''$,

(c) Elimination: *if* $C', C'' \in \mathcal{C}$ *are compatible and* $x \in C' \cap C''$, *then* $C \subseteq (C' \cup C'') - x$ *for some* $C \in \mathcal{C}$,

(d) *if* $C_1, C_2 \in \mathcal{C}$, *then* $C_1 \cup C_2$ *cannot include precisely one skew pair.*

Proof. Suppose that $\mathcal{C}$ is the set of the circuits of a multimatroid of rank function r. Property (a) holds because $r(\emptyset) = 0$. Proposition 5.2 (b), applied to the matroid $Q[C'']$, implies (b). Proposition 5.2 (c), applied to the matroid $Q[C' \cup C'']$, implies (c). Suppose for a contradiction that (d) is not satisfied. Let $\{x_1, x_2\}$ be the skew pair included in $C_1 \cup C_2$, and suppose $x_i \in C_i$ for $i = 1, 2$. Let $D_i = C_i - x_i$. Since

C_i is a circuit,

$$\text{(i) } r(C_i) = r(D_i + x_i) = r(D_i)$$

is satisfied. Note that $D_1 \cup D_2 + x_i$ is a subtransversal of Ω for $i = 1, 2$ since $\{x_1, x_2\}$ is the only skew pair in $C_1 \cup C_2$. The submodular inequality 3.3 and (i) imply

$$r(D_1 \cup D_2 + x_i) = r(D_1 \cup D_2),$$

which contradicts axiom 3.4 applied to the subtransversal $D_1 \cup D_2$ and the skew pair $\{x_1, x_2\}$.

Conversely, suppose that $\mathcal{C}$ satisfies (a) to (d). Let $\mathcal{I} = \{I : I \not\supseteq C \text{ for all } C \in \mathcal{C}\}$. We verify that $\mathcal{I}$ satisfies (a) to (d) in Proposition 5.3, and so $\mathcal{C}$ will be the set of the circuits of the multimatroid on (U, Ω) whose set of independent sets is equal to $\mathcal{I}$. Let $S \in \Omega$. Conditions (a), (b), and (c) imply that $\{C \in \mathcal{C} : C \subseteq S\}$ is the set of the circuits of a matroid M_S on S. It follows that $\{I \in \mathcal{I} : I \subseteq S\}$ is the set of the independent sets of M_S. So $\mathcal{I}$ satisfies (a), (b), and (c) in Proposition 5.3. Suppose for a contradiction that $\mathcal{I}$ does not satisfy (d) in Proposition 5.3. Then $I+x$, which does not belong to $\mathcal{I}$, includes some $C_x \in \mathcal{C}$. We have $x \in C_x$ otherwise C_x would be a subset of I. Similarly $I + y$ includes some $C_y \in \mathcal{C}$ such that $y \in C_y$. So $\{x, y\} \subseteq C_x \cup C_y$. Since $C_x \cup C_y \subseteq I + \{x, y\}$, no skew pair distinct from $\{x, y\}$ is included in $C_x \cup C_y$, which contradicts (d). $\square$

We say that a multimatroid Q is *nondegenerate* if no skew class of Q is a singleton.

PROPOSITION 5.5. *The bases of a nondegenerate multimatroid are transversal.*

Proof. Suppose that a base B of a nondegenerate multimatroid $Q = (U, \Omega, r)$ is not transversal. Take any $\omega \in \Omega$ disjoint from B. Since Q is nondegenerate we can chose distinct elements x and y in ω. Condition (d) of Proposition 5.3 implies that either $B + x$ or $B + y$ is independent, and so B cannot be a base. $\square$

COROLLARY 5.6. *Every base of a q-matroid is a transversal if $q \geq 2$.*

PROPOSITION 5.7. *Let B be a base and let ω be a skew class of a multimatroid. Then $B \cup \omega$ includes at most one circuit.*

Proof. Suppose that a circuit C' is included in $B \cup \omega$. Since B is independent, C' must intersect $\omega \setminus B$. Let $p' \in C' \cap (\omega \setminus B)$. Suppose that a second circuit C'' is included in $B \cup \omega$. Let $p'' \in C'' \cap (\omega \setminus B)$. Suppose first $p' = p''$. Then C' and C'' are included in $T = (B \setminus \omega) + p'$, which is a transversal. So C' and C'' are compatible, and, by an elimination of $p' \in C' \cap C''$, we obtain a circuit $C \subseteq (C' \cup C'') - p' \subseteq B$, a contradiction since B is independent. So $p' \neq p''$. This implies that $\{p', p''\}$ is a skew pair included in $C' \cup C''$. No other skew pair can be included in $C' \cup C''$ because $(C' \cup C'') \setminus \omega \subseteq B \setminus \omega$ and B is transversal. This contradicts Proposition 5.4. $\square$

The unique circuit included in $B \cup \omega$, if it exists, will be denoted by $C(B, \omega)$ and called a *fundamental circuit.* We often say that $C(B, \omega)$ is *not defined* if there is no circuit in $B \cup \omega$.

PROPOSITION 5.8. *The set of bases of a nondegenerate multimatroid satisfies the transversal exchange axiom.*

Proof. Consider a nondegenerate multimatroid $Q = (U, \Omega, r)$ and the set $\mathcal{B}$ of its bases. We use the notation of the transversal exchange axiom. If $A'\Delta p$ is a base of Q, the property is proved with $q = p$. Suppose that $A'\Delta p$ is not a base of Q. There exists a circuit $C \subseteq A'\Delta p$. Let $p = \{p', p''\}$ and assume $p' \in A'$ and $p'' \in A''$. So $A'\Delta p = A' - p' + p''$, $p'' \in C$, and $p' \notin C$. There exists a skew pair $q = \{q', q''\} \subseteq A' \cup A''$ such that $q' \in C \cap A'$ and $q'' \in A''$, otherwise every element $q' \in C \cap A'$ would also belong to A'' and the circuit C would be included in A'', a contradiction since A'' is independent. Let $A = A' - p' + p'' - q'$. The subset A does no longer include C and, since C is the (uniquely defined) fundamental circuit included in $A' \cup \omega$, A is independent. The subset $A + q'$ is not independent because it includes C. Therefore $A + q''$ is independent by Proposition 5.3, so that $A + q'' = A' \Delta (p \cup q)$ is a base of Q, which completes the proof. □

The converse of the preceding proposition is false. For a counterexample, consider the carrier $(U = \{x_1, x_2, y_1, y_2, y_3\}, \Omega = \{\{x_1, x_2\}, \{y_1, y_2, y_3\}\})$, and suppose that $\mathcal{B}$ contains only $\{x_1, y_1\}$. Since $|\mathcal{B}| = 1$, the transversal exchange axiom is satisfied. Suppose that $\mathcal{B}$ is the set of the bases of a multimatroid. According to axiom 3.4 applied to $\emptyset \in \mathcal{S}(\Omega)$ and the skew pair $\{y_2, y_3\}$, either y_2 or y_3 is independent, a contradiction since no member of $\mathcal{B}$ contains either y_2 or y_3.

Proof of Theorem 4.1. According to Proposition 5.8 it remains to prove that any nonempty subset $\mathcal{B} \subseteq \mathcal{T}(\Omega)$ satisfying the transversal exchange axiom is the set of bases of a 2-matroid on (U, Ω). For that we prove that $\mathcal{I} = \{I : I \subseteq A \text{ for some } A \in \mathcal{B}\}$ satisfies conditions (a) to (d) of Proposition 5.3. We pass through the intermediate of the set $\mathcal{C}$ of (inclusionwise) minimal members of $\mathcal{S}(\Omega) \setminus \mathcal{I}$.

The set $\mathcal{C}$ clearly satisfies conditions (a) and (b) of Proposition 5.4. We now verify that (c) is satisfied. Consider any two compatible members C_1 and C_2 of $\mathcal{C}$ and an element $x \in C_1 \cap C_2$. We have to find a member $C \in \mathcal{C}$ that is included in $I' = C_1 \cup C_2 - x$. Suppose that C cannot be found. Then $I' \subseteq A'$ for some $A' \in \mathcal{B}$. Let $p = \{x, y\}$ be the skew pair that contains x. The intersection of p and A' is nonempty because A' is transversal. The element x does not belong to A', otherwise A' would include C_1 and C_2. So $y \in A'$ and $C_1 \cup C_2 \subseteq A' \Delta p$. The set $I'' = C_1 \cap C_2$ includes no member of $\mathcal{C}$. Then $I'' \subseteq A''$ for some $A'' \in \mathcal{B}$. The element x, which belongs to I'', also belongs to A'', so that $p \subseteq A' \Delta A''$. According to the transversal exchange axiom there exists a skew pair $q \subseteq A' \Delta A''$ such that $A = A' \Delta (p \cup q) \in \mathcal{B}$. If we assume $p = q$ then $C_1 \cup C_2 \subseteq A' \Delta p = A$, a contradiction. So we have $p \neq q$ and $A = A' \Delta p \Delta q$. Suppose $q \cap C_1 = \emptyset$. Then $C_1 \cap A = C_1 \cap (A' \Delta p) = C_1$, a contradiction because C_1 is not included in A, which is a member of $\mathcal{B}$. So $q \cap C_1 \neq \emptyset$ and, similarly, $q \cap C_2 \neq \emptyset$. Since $C_1 \cup C_2 \in \mathcal{S}(\Omega)$, there is at most one element of q, say z, that can belong to $C_1 \cup C_2$. So $z \in C_1 \cap C_2$, which implies $z \in A' \cap A''$, a contradiction with $q \subseteq A' \Delta A''$.

Since $\mathcal{C}$ satisfies conditions (a) to (c) of Proposition 5.4, $\mathcal{I}$ satisfies conditions (a) to (c) of Proposition 5.3 (consider, for each $S \in \mathcal{S}(\Omega)$ the matroid whose circuits are the members of $\mathcal{C}$ included in S). Any $I \in \mathcal{I}$ is, by definition, included in some $A \in \mathcal{B}$. If $\{x, y\}$ is a skew pair disjoint from I, then $I + x$ or $I + y$ is included in $\mathcal{B}$, and so belongs to $\mathcal{I}$. Therefore, $\mathcal{I}$ also satisfies condition (d) of Proposition 5.3, which completes the proof. □

Circuit indicators and fundamental graphs. Given a multimatroid carrier (U, Ω) it will be useful to define a surjective mapping $sp : U \to V$ such that $\Omega = \{sp^{-1}(v) : v \in V\}$. We say that (U, Ω), and any multimatroid Q defined on (U, Ω), is *indexed* on V. For $v \in V$ we let $\Omega_v = sp^{-1}(v)$. For $X \in \mathcal{S}(\Omega)$ and $v \in V$, we say that X_v is *defined* if $X \cap \Omega_v \neq \emptyset$, and in this case X_v denotes the unique element of $X \cap \Omega_v$.

Example. If $Q = Q(G, U)$ is a Eulerian multimatroid then, where V is the vertex-set of G, there is a natural indexation of Q by V such that Ω_v is the set of the local splitters of U incident to v, for any $v \in V$.

Let A be a base of Q. We define a subtransversal $\bar{A} \in \mathcal{S}(\Omega)$ and a set $\mathrm{Arc}(A) \subseteq V \times V$ as follows:

5.1. *$\bar{A}_v$ is defined if and only if $C(A, \Omega_v)$ is defined.*
5.2. *If $C(A, \Omega_v)$ is defined, then $\bar{A}_v = C(A, \Omega_v)_v$ (which implies $\bar{A}_v \neq A_v$).*
5.3. *If $C(A, \Omega_v)$ is defined, then $\{w : (v, w) \in \mathrm{Arc}(A)\} = sp(C(A, \Omega_v)) - v$.*
5.4. *If $C(A, \Omega_v)$ is not defined, then $\{w : (v, w) \in \mathrm{Arc}(A)\} = \emptyset$.*

We call $\bar{A}$ the *circuit indicator* of A and the pair $(V, \mathrm{Arc}(A))$, which is a digraph, the *fundamental graph* of A.

6. Covering theorem and applications. Let us distinguish two kinds of covering problems, the first one being an instance of the second one.

Simple covering problem. A multimatroid Q is given with an integer $k \geq 2$. We search for k independent sets whose union have maximum cardinality.

Multiple covering problem. A finite family of multimatroids $Q = (Q^j : j \in J)$ defined on a common carrier (U, Ω) is given. We denote by $\mathcal{I}(Q)$ the set of the families $(I^j : j \in J)$, where I^j is an independent set of Q^j. For $I \in \mathcal{I}(Q)$, we let $Cov(I) = \bigcup(I^j : j \in J)$. An element $x \in U$ (respectively, subset $X \subseteq U$) is said to be *covered* by I if $x \in Cov(I)$ (respectively, $X \subseteq Cov(I)$). We search for an $I \in \mathcal{I}(Q)$ that maximizes $|Cov(I)|$. Then I is called an *optimal covering* of U (*by the independent sets of* Q).

Remark. If each Q^j is sheltered by a matroid M^j, then every I in $\mathcal{I}(Q)$ is a covering of U where each I^j is an independent set of M^j. In that case we can reformulate the problem as maximizing $|Cov(I)|$, $I = (I^j : j \in J)$, where I^j is an independent set of M^j and a subtransversal of Ω. This is an instance of the covering problem for matroids solved by Edmonds [14] with an additional constraint. However this property does not seem to help in finding a solution.

For algorithmic purposes we will use a *rank-oracle* to compute the rank function r^j of Q^j, for each $j \in J$, and we assume that the time-complexity to compute $r^j(S)$, for any $S \in \mathcal{S}(\Omega)$, is equal to $O(1)$.

Our main result is the following one, which partially solves the multiple covering problem.

THEOREM 6.1. *Let $Q = (Q^j : j \in J)$ be a finite family of multimatroids defined on a common carrier (U, Ω). Where r^j is the rank function of Q^j, let $r(S) = \sum(r^j(S) : j \in J)$ for all $S \in \mathcal{S}(\Omega)$. If every skew class ω satisfies $3 \leq |\omega| \leq |J|$, then*

$$\min(|U| - |Cov(I)| : I \in \mathcal{I}(Q)) = \max(|S| - r(S) : S \in \mathcal{S}(\Omega)).$$

A pair of solutions, $I \in \mathcal{I}(Q)$ and $S \in \mathcal{S}(\Omega)$, satisfying the equality can be found in polynomial time.

COROLLARY 6.2. *Let $Q = (U, \Omega, r)$ be a multimatroid where each skew class has at least three elements. Let $k \geq 3$ be an integer. The set U can be covered by k independent sets of Q if and only if*

$$|\omega| \leq k \quad \forall \omega \in \Omega$$

and

$$kr(S) \geq |S| \quad \forall S \in \mathcal{S}(\Omega).$$

This corollary, which partially solves the simple covering problem, also holds when Q is a matroid and k is any positive integer (recall that a matroid is identified to a 1-matroid); it is a theorem proved by Edmonds [14].

Removing large skew classes. We show that the assumption $|\omega| \leq |J|$ in Theorem 6.1 is not essential. It only stands to concentrate the attention on the main difficulties. Say that a skew class ω is *large* (respectively, *small*) if $|\omega| > |J|$ (respectively, $|\omega| \leq |J|$). For $I, I' \in \mathcal{I}(Q)$, we write $I \supseteq I'$ if $I^j \supseteq I'^j$ for all $j \in J$.

PROPOSITION 6.3. *For any $I' \in \mathcal{I}(Q)$ there exists $I \supseteq I'$, $I \in \mathcal{I}(Q)$, with the following properties satisfied for each skew class ω disjoint from $Cov(I')$:*

if ω is large, then $|\omega \cap Cov(I)| = |J|$,

if ω is small, then $|\omega \cap Cov(I)| \geq |\omega| - 1$.

Moreover, I can be derived from I' in polynomial time.

Proof. Identify J with the set of integers $\{1, 2, \ldots, t\}$. Let $\Omega' = \{\omega \in \Omega : \omega \cap Cov(I') = \emptyset\}$. Choose any $\omega \in \Omega'$. If ω is large, let $p = t$; otherwise let $p = |\omega| - 1$. We construct a sequence of pairwise distinct elements of ω, say $x^1, x^2, \ldots, x^p$, such that $I^j = I'^j + x^j$ is an independent set of Q^j for any $j = 1, 2, \ldots, p$. Assume that $x^1, x^2, \ldots, x^{j-1}$ has been determined and consider any skew pair $\{x, y\} \subseteq \omega - \{x^1, x^2, \ldots, x^{j-1}\}$, which exists by the definition of p. Since I'^j is independent in Q^j, it follows from Proposition 5.3 that either $I'^j + x$ or $I'^j + y$ is also independent in Q^j. So we may take either $x^j = x$ or $x^j = y$. We construct the sequence $x^1, x^2, \ldots, x^p$ by letting j be successively equal to $1, 2, \ldots, p$. Finally we let $I^j = I'^j$ for each integer j such that $p < j \leq t$. So $I = (I^j : j \in J)$ clearly satisfies $I \supseteq I'$ and the two properties stated in the proposition for the particular ω that has been chosen in Ω'.

We let $\Omega' = \Omega' - \omega$, $I' = I$ and we repeat the preceding construction if $\Omega' \neq \emptyset$. □

Divide Ω into a subset Ω' of small skew classes and a subset Ω'' of large skew classes. Let U' be the union of the small skew classes. Assume that we know an optimal covering I' of $Q[U']$. So $I' \in \mathcal{I}(Q)$. By applying the preceding proposition to I' we get some $I \in \mathcal{I}(Q)$ satisfying $I \supseteq I'$ and $|\omega \cap Cov(I)| = |J|$ for all large skew class ω. Clearly I is an optimal covering of Q.

It follows that, to search for an optimal covering, we may reduce the problem to the set of the small skew classes. A particular extremal case is when every skew class is large. Then the problem is trivially reduced to the empty multimatroid and we can find an optimal covering $I \in \mathcal{I}(Q)$ such that $|\omega \cap Cov(I)| = |J|$ for every skew class ω.

Parity problem. We now discuss the assumption $|\omega| \geq 3$, which is essential for the validity of Theorem 6.1. Indeed we show that the parity problem for matroids can be expressed as a multiple covering problem involving two 2-matroids. Lovász [19] has shown that this problem is nonpolynomial in general. We already discussed a similar question in [7]. For the sake of completeness, we adapt this discussion to 2-matroids.

Given a matroid $M = (X, r)$ and a partition P of X into pairs, the *parity problem* is to find an independent set I of M, having maximal cardinality, that can be expressed as a union of pairs in P. Let $\tilde{X}$ be a copy of X. Assume $X \cap \tilde{X} = \emptyset$. For $x \in X$ let $\tilde{x}$ denote the copy of x. For $A \subseteq X$ let $\tilde{A} = \{\tilde{x} : x \in A\}$. Consider the 2-matroid carrier (U, Ω), where $U = X \cup \tilde{X}$ and $\Omega = \{\{x, \tilde{x}\} : x \in X\}$. Every subtransversal of Ω is uniquely expressible as a union $A \cup \tilde{B}$, where A and B are disjoint subsets of X. Let

$$r_M(A \cup \tilde{B}) = r(A) + r^*(B),$$
$$r_P(A \cup \tilde{B}) = |A| + |B| - p(A, B),$$

where r^* is the rank function of the matroid dual of M and $p(A, B)$ denotes the number of the pairs in P that intersect A and B. It is not difficult to verify that r_M and r_P satisfy axioms 3.1 to 3.4. Therefore $Q_M = (U, \Omega, r_M)$ and $Q_P = (U, \Omega, r_P)$

are 2-matroids. If B_M is a base of Q_M and B_P is a base of Q_P, then $|B_M \cup B_P|$ is maximal if and only if B_M is a solution to the parity problem.

Application to Eulerian graphs. Consider a connected Eulerian graph of minimum degree $2d \geq 4$. For any integer $k \leq 2d-1$ we can find a set Ω_v of k pairwise skew local splitters incident to any vertex v (for example, if h_1, h_2, ..., h_{2d} are the half-edges incident to v, then $\{\{h_1, h_i\}, h(v)-\{h_1, h_i\} : 2 \leq i \leq 2d\}$ is a set of pairwise skew local splitters). Let $U = \bigcup(\Omega_v : v \in V)$. Define a base of U as a splitter $T \subseteq U$ incident to every vertex of G and such that $G||T$ is still connected. Apply Corollary 6.2 to the Eulerian multimatroid $Q(G, U)$. We see that U is a disjoint union of bases of U if and only if

$$k(G||s) \leq 1 + |s|(1 - 1/k).$$

In particular, if $d = 2$ and $k = 3$, U is necessarily equal to the set of all the local splitters of G. Any base of U can be identified to an Euler tour. We retrieve Jackson's Theorem 2.4.

Note also the following property, which follows from Proposition 6.3. If $|\Omega_v| > k$ holds for all $v \in V$ then we can find k pairwise disjoint complete splitters s_1, s_2, ..., s_k such that $G||s_1$, $G||s_2$, ..., $G||s_k$ are connected. This is the trivial case where each skew class is large.

Application to isotropic systems. Consider an isotropic system $S = (E, L, V)$ and its associated 3-matroid $Q(S) = (U, \Omega, r)$. We also consider the bijective mapping $\alpha : E \to \mathcal{S}(\Omega)$ satisfying $A = \sum(u : u \in \alpha(A))$. For $A \in E$ let $\rho(A) = r(\alpha(A))$, which is called the *rank* of A, and say that A is *Eulerian* if $\alpha(A)$ is a base of $Q(S)$.

Corollary 6.4 (of Theorem 6.1). *Consider a direct product of hyperbolic planes* $E = \prod(E_v : v \in V)$ *and, for* $j \in J = \{1, 2, 3\}$, *an isotropic system* $S^j = (E, L^j, V)$ *of rank function* ρ^j. *Denote by* $\mathcal{A}$ *the set of the triples* (A^1, A^2, A^3), *where* A^j *is an Eulerian vector of* S^j *for* $j \in J$. *Then*

$$\min(3|V| - |\alpha(A^1) \cup \alpha(A^2) \cup \alpha(A^3)|) = \max(|B| - \rho^1(B) - \rho^2(B) - \rho^3(B)),$$

where the maximum is taken for $(A^1, A^2, A^3) \in \mathcal{A}$ *and the minimum is taken for* $B \in E$.

Say that two Eulerian vectors, A and B, of the isotropic system S are *disjoint* if $A_v \neq B_v$ holds for all $v \in V$. Equivalently A is disjoint from B if $\alpha(A) \cap \alpha(B) = \emptyset$. (The term *compatible* is used in place of disjoint in [8, 16, 15].)

Corollary 6.5. *An isotropic system* $S = (E, L, V)$, *of rank function* ρ, *has three pairwise disjoint Eulerian vectors if and only if* $3\rho(B) \geq |B|$ *holds for all* $B \in E$.

The preceding corollary has been proved by Jackson [16], who used it in [15] to establish Theorem 2.4. Corollary 6.4 has been proved by Bouchet in [8].

7. Proof of Theorem 6.1.

From now on we follow the notation defined in Theorem 6.1. So every skew class ω satisfies $|\omega| \leq |J|$. A family $I \in \mathcal{I}(Q)$ will be called a *suboptimal covering* if $|\omega \cap Cov(I)| \geq |\omega| - 1$ is satisfied for all $\omega \in \Omega$.

Proposition 7.1. *Every optimal covering is suboptimal.*

Proof. Let I'' be an optimal covering and let $\omega \in \Omega$. Suppose for a contradiction that $|\omega \cap Cov(I'')| < |\omega| - 1$. Apply Proposition 6.3 to $I' := (I''^j \setminus \omega : j \in J)$. We get a new family $I \in \mathcal{I}(Q)$ such that $I \supseteq I'$, $\omega' \cap Cov(I) = \omega' \cap Cov(I')$ for all $\omega' \in \Omega - \omega$ and $|\omega \cap Cov(I)| \geq |\omega| - 1$. So $|Cov(I)| > |Cov(I'')|$, which is a contradiction. □

Let I be a suboptimal covering. Denote by $\nu(I)$ the number of the skew classes that are *not covered* by I, that is $\nu(I) = |\{\omega \in \Omega : |\omega \cap Cov(I)| = |\omega| - 1\}|$. Clearly, we have axiom 7.1.

7.1. $|Cov(I)| = |U| - \nu(I)$.
For each $j \in J$, let A^j be a base of Q^j that includes I^j. Clearly $(A^j : j \in J)$ is still a suboptimal covering of Q, which is optimal if I is optimal. Denote by $\mathcal{B}(Q)$ the set of the suboptimal coverings $A = (A^j : j \in J)$, where A^j is a base of Q^j. According to axiom 7.1, to solve the covering problem it is sufficient to find a suboptimal covering $A \in \mathcal{B}(Q)$ for which $\nu(A)$ is minimal.

Proposition 7.2. *For any suboptimal covering $A \in \mathcal{B}(Q)$ and any $S \in \mathcal{S}(\Omega)$ we have*

$$\nu(A) \geq |S| - r(S).$$

Proof. We have

$$\begin{aligned} r(S) &= \sum (r^j(S) : j \in J) \\ &\geq \sum (r^j(A^j \cap S) : j \in J) \\ &= \sum (|A^j \cap S| : j \in J) \\ &\geq |S| - \nu(A). \qquad \square \end{aligned}$$

So to prove Theorem 6.1 it is sufficient to find $A \in \mathcal{B}(Q)$ and $S \in \mathcal{S}(\Omega)$ such that $\nu(A) = |S| - r(S)$. The proof will follow an algorithm that maintains a suboptimal covering $A \in \mathcal{B}(Q)$. Eventually, A will be optimal.

We first define some procedures whose purpose, at the exception of the first one, is to modify A in order to cover a new skew class that previously was not covered. Each of these procedures will leave covered every skew class that was previously covered. We assume that (U, Ω) is indexed on a set V and we use the notation introduced at the end of section 5.

Let $M = (M_{ij} : 1 \leq i \leq s, 1 \leq j \leq t)$ be a binary matrix. Assume $s \leq t$. An *allowed permutation* of M is a sequence $(i_1, j_1), (i_2, j_2), \ldots, (i_s, j_s)$ such that $M_{i_p j_p} = 0$ for $1 \leq p \leq s$ and each of the sequences $i_1, i_2, \ldots, i_s$ and $j_1, j_2, \ldots, j_s$ is made of pairwise distinct elements (and so every row index appears exactly once in the first sequence).

Procedure `FIND ALLOWED PERMUTATION(`M`).` It yields an allowed permutation of a binary matrix $M = (M_{ij} : 1 \leq i \leq s, 1 \leq j \leq t)$ satisfying the following conditions:

(i) $s \leq t$;
(ii) every column of M has at most one nonnull entry;
(iii) every row of M has at least one null entry.

The algorithm runs as follows.

Let $q = 1$.

Choose a row with a maximal number of nonnull entries, and let i_q be the index of that row. By (iii) there exists a null entry in the row indexed by i_q, say $M_{i_q j_q}$. Let M' be the submatrix obtained by deleting the row indexed by i_q and the column indexed by j_q. We claim that M', if it is nonempty, still satisfies (i) to (iii). This is obvious for (i) and (ii). Suppose that (iii) does not hold. So there exists a row of M', indexed by some i, which has only nonnull entries. Since the row of M indexed by i_q has a maximal number of nonnull entries, it must be equal to the row of M indexed by i, a contradiction with (ii). We increase q by 1, we replace M by M' and we repeat the sequence of instructions as long as $q \leq s$.

The sequence $(i_1, j_1), (i_2, j_2), \ldots, (i_s, j_s)$, constructed by the procedure, is an allowed permutation. □

We say that a skew class Ω_v is *obstructed* if there exists $x \in \Omega_v$ such that $\bar{A}_v^j = x$ for all $j \in J$.

Procedure `COVER NONOBSTRUCTED(`Ω_v`)`. A skew class Ω_v that is not obstructed and not covered by A is given. The procedure returns with Ω_v covered without modifying A_w^j for all $1 \leq j \leq t$ and $w \in V - v$. The algorithm runs as follows.

Label the elements of Ω_v as $x_1, x_2, \ldots, x_s$. Construct the binary matrix, $M = (M_{ij} : 1 \leq i \leq s, 1 \leq j \leq t)$, such that $M_{ij} = 1$ if and only if $\bar{A}_v^j = x_i$. Conditions (i) and (ii) of Procedure `FIND ALLOWED PERMUTATION(`M`)` are clearly satisfied. Condition (iii) is also satisfied because Ω_v is nonobstructed. We call the procedure, and so we obtain an allowed permutation $(i_1, j_1), (i_2, j_2), \ldots, (i_s, j_s)$. For each j_q, $1 \leq q \leq s$, we let A'^{j_q} be the transversal of Ω obtained from A^{j_q} by replacing its element $A_v^{j_q}$ by x_{i_q}. The transversal A'^{j_q} is still a base of Q because it does not contain the fundamental circuit $C(A_v^{j_q}, \Omega_v)$. Since A'^{j_q} contains x_{i_q}, for $1 \leq q \leq s$, and every row index of M appears in the sequence $i_1 i_2, \ldots, i_s$, the subset of the bases $\{A'^{j_1}, A'^{j_2}, \ldots, A'^{j_s}\}$ covers Ω_v. □

A skew class Ω_v is said to be j*-critical*, where $j \in J$, if $\bar{A}_v^k$ is defined for all $k \in J - j$ and $\bar{A}_v^k = A_v^j$.

Procedure `COVER CRITICAL(`Ω_v, j_1, j_2`)`. A j_1-critical skew class Ω_v and an index $j_2 \in J - j_1$ are given. The procedure covers Ω_v without modifying either A^{j_1} or A^{j_2} or A_w^i for $1 \leq i \leq t$, $w \in V - v$. The algorithm runs as follows.

We label the elements of Ω_v as in Procedure `COVER NONOBSTRUCTED(`Ω_v`)`, and we construct the same matrix M. Let $x_{i_1} = A_v^{j_1}$ and $x_{i_2} = A_v^{j_2}$. Since Ω_v is j_1-critical, we have $x_{i_1} = A_v^{j_1} = \bar{A}_v^{j_2} \neq A_v^{j_2} = x_{i_2}$, which implies $i_1 \neq i_2$. Let M' be the submatrix of M obtained by deleting the rows i_1 and i_2 and the columns j_1 and j_2. Since Ω_v is j_1-critical every column of M, not indexed by j_1, has a nonnull entry in row i_1. Accordingly every entry of M' is null, and so the conditions to call Procedure `FIND ALLOWED PERMUTATION(`M'`)` are obviously satisfied. Then we get an allowed permutation of M', which can be used to determine new bases $A'^{j_3}, A'^{j_4}, \ldots, A'^{j_s}$. These bases cover $\Omega_v - \{x_{i_1}, x_{i_2}\}$. So, by adding A^{j_1} and A^{j_2}, the whole class Ω_v is covered. □

Procedure `COVER OBSTRUCTED(`Ω_v, j, Ω_w`)`. An obstructed skew class Ω_v and a non-j-critical skew class Ω_w are given with the property $(v, w) \in \mathrm{Arc}(A^j)$. The procedure returns with Ω_v and Ω_w covered, and A_u^i unchanged for $1 \leq i \leq t$, $u \in V - \{v, w\}$.

Let $X = A_w^j$. Since Ω_w is not j-critical, there exists $k \in J - j$ such that $\bar{A}_w^k \neq X$.

In the case where $C(A^j, \Omega_w)$ is defined, let $Y = \bar{A}_w^j$. Choose $Z \in \Omega_w - X$ if $C(A^j, \Omega_w)$ is not defined, $Z \in \Omega_w - \{X, Y\}$ otherwise (Z can always be chosen because $|\Omega_w| \geq 3$). The transversal $A'^j = A^j \Delta \{X, Z\}$ is a base of Q^j because it does not include the fundamental circuit $C(A^j, \Omega_w)$, if it is defined. Let $x = A_v^j$, denote by y the element of Ω_v such that $\bar{A}_v^i = y$ for all $i \in J$, let $A''^j = A'^j \Delta \{x, y\}$. We claim that A''^j is a base of Q^j.

Suppose not. Then the fundamental circuit $C' = C(A'^j, \Omega)_v$ is defined and included in A''^j, which implies $C'_v = A''^j_v = y$. Consider also the fundamental circuit $C = C(A^j, \Omega_v)$, which satisfies $C_v = \bar{A}_v^j = y$. Note that $C_w = X$ because $(v, w) \in \mathrm{Arc}(A^j)$. For any $u \in V - \{v, w\}$, each of the values C_u and C'_u is either nondefined or equal to A^j_u. Therefore there can exist at most one skew pair included in $C \cup C'$, and this skew pair is included in Ω_w. Since $C \cup C'$ cannot include precisely one skew pair, by Proposition 5.4, either C'_w is not defined or $C'_w = C_w$. If C'_w is not defined, then C' is included in $A^j \cup \Omega_v$, so that $C' = C(A^j, \Omega_v)$, a contradiction since C_w is defined. Suppose finally that $C_w = C'_w$. Since $C' = C(A'^j, \Omega_v)$ we have $C'_w = A'^j_w = Z$. Since $C = C(A^j, \Omega_v)$ we have $C_w = A^j_w = X$, which contradicts $C_w = C'_w$. So A''^j is actually a base. Note that C is now included in $A''^j \cup \Omega_w$, so that $C = C(A''^j, \Omega_w)$.

The procedure runs as follows. It replaces A^j by A''^j, so that we now have $A^j_v = y$ and $C = C(A^j, \Omega_w)$. It can be seen that Ω_v is now j-critical. Furthermore we have $\bar{A}^j_w = C_w = X \neq \bar{A}^k_w$. The procedure calls Procedure `COVER CRITICAL(`Ω_v, j, k`)`, which returns with Ω_v covered without modifying A^j and A^k, so that $\bar{A}^j_w \neq \bar{A}^k_w$ is still satisfied. Therefore Ω_w is nonobstructed and we can call Procedure `COVER NONOBSTRUCTED(`Ω_w`)`. □

Let $\Gamma = (v_0, j_1, v_1, \ldots, j_q, v_q)$ be a sequence of (not necessarily distinct) vertices $v_0, v_1, \ldots, v_q$ and elements $j_1, j_2, \ldots, j_q$ in J. We call Γ a *critical sequence* if it satisfies the following properties: (i) Ω_{v_0} is obstructed; (ii) Ω_{v_s} is covered by A, Ω_{v_s} is j_s-critical, and $(v_{s-1}, v_s) \in \mathrm{Arc}(A^{j_s})$ for $s = 1, 2, \ldots, q$; (iii) $j_s \neq j_{s+1}$ for $s = 1, 2, \ldots, q-1$. We shall say that each of the vertices v_s, $0 \leq s \leq q$, is *accessible.*

We call Γ an *improving sequence* if it satisfies the same conditions as a critical sequence at the exception of Ω_{v_q} which is now required to be not j_q-critical and possibly not covered by A (all the other conditions in (i) to (iii) must be satisfied). A *shortcut* of Γ is a triple (v_r, j, v_q) such that $j \in J$, $0 \leq r \leq q-2$ and the sequence $(v_0, j_1, v_1, \ldots, v_r, j, v_q)$ is an improving sequence (so Ω_{v_q} is not j-critical).

Procedure `COVER SEQUENCE(`Γ`)`. An improving sequence $\Gamma = (v_0, j_1, v_1, \ldots, j_q, v_q)$ is given. The procedure returns with Ω_{v_0} covered and leaves covered every previously covered skew class.

If $q = 1$ the conditions to call Procedure `COVER OBSTRUCTED(`$\Omega_{v_0}, j_1, \Omega_{v_1}$`)` are satisfied. We make this call and we return.

If $q > 1$ we first search for shortcuts of Γ and shorten Γ in accordance. Thus we suppose that Γ has no shortcut. We shall determine a subset $J' \subseteq J$ and a suboptimal covering $A' = (A'^j : j \in J)$ in such a way that $A'^j = A^j$ for all $j \in J - J'$ and the following properties hold:

(i) $j_q \in J'$;
(ii) for every $j \in J'$, Ω_{v_q} is not j-critical with respect to A;
(iii) $A'^j_{v_q} \neq A^j_{v_q}$ and $A'^j_v = A^j_v$ for all $v \in V - v_q$ and all $j \in J'$;
(iv) A' covers Ω_{v_q};
(v) $\Omega_{v_{q-1}}$ is not j_{q-1}-critical with respect to A';
(vi) $(v_{r-1}, v_r) \in \mathrm{Arc}(A'^{j_r})$ for $1 \leq r \leq q-2$.

According to (iii) and (iv), the skew classes covered by A and A' will be the same ones. If Ω_{v_0} is no longer obstructed with respect to A', then we call Procedure `COVER NONOBSTRUCTED(`Ω_{v_0}`)` and we return. Otherwise we consider the minimal value $r > 0$ such that Ω_{v_r} is not j_r-critical with respect to A', which exists by (v). The sequence $\Gamma' = (v_0, j_1, v_1, \ldots, j_r, v_r)$ is improving with respect to A' by (vi). By recursively calling Procedure `COVER SEQUENCE(`Γ'`)`, Ω_{v_0} is eventually covered and every skew class that was previously covered remains covered.

It may happen in the sequel that we refer to some element X_v, for $X \in \mathcal{S}(\Omega)$ and $v \in V$, which may possibly not defined. In order to properly speak of Ω_v, we implicitly consider a new element $\theta \notin U$, and we let $X_v = \theta$.

We now determine J' and A'^j for each $j \in J'$, and we show that (i) to (iv) hold. Let $X = A^{j_q}_{v_q}$. Since Ω_{v_q} is not j_q-critical, there exists $k \in J - j_q$ such that $\bar{A}^k_{v_q} \neq X$. We distinguish four cases.

Case 1. $A^k_{v_q} = X$. We let $J' = \{j_q\}$. Conditions (i) and (ii) are obviously satisfied. Consider any $Z \in \Omega_{v_q} - \{X, \bar{A}^{j_q}_{v_q}\}$ (which exists because $|\Omega_{v_q}| \geq 3$). Let $A'^{j_q} = A^{j_q}\Delta\{X, Z\}$. The set A'^{j_q} is a base because the condition $\bar{A}^{j_q}_{v_q} \neq Z$ implies that the fundamental circuit $C(A^{j_q}, \Omega_{v_q})$, if it exists, is not included in A'^{j_q}. So (iii) is satisfied. When replacing A^{j_q} by A'^{j_q}, the new base no longer covers X, but the base A^k already covered X, and so (iv) is satisfied.

From now on we suppose $A^k_{v_q} \neq X$ and we let $Z = A^k_{v_q}$.

Case 2. $\bar{A}^{j_q}_{v_q} \neq Z$. We let $J' = \{j_q, k\}$, $A'^{j_q} = A^{j_q}\Delta\{X, Z\}$, and $A'^k = A^k\Delta\{X, Z\}$. Clearly (i) is satisfied. The vertex v_q is not k-critical with respect to A because $A^k_{v_q} = Z \neq \bar{A}^{j_q}_{v_q}$. So (ii) is satisfied. As above A'^{j_q} is a base because $\bar{A}^{j_q}_{v_q} \neq Z$. Similarly A'^k is a base because $\bar{A}^k_{v_q} \neq X$. So (iii) is satisfied. Finally we have $\{A'^{j_q}_{v_q}, A'^k_{v_q}\} = \{Z, X\} = \{A^k_{v_q}, A^{j_q}_{v_q}\}$, so that (iv) is satisfied.

From now on we assume $\bar{A}^{j_q}_{v_q} = Z$. Choose an element $Y \in \Omega_{v_q} - \{X, Z\}$, which is possible because $|\Omega_{v_q}| \geq 3$. Since A covers Ω_{v_q} there exists $l \in J - \{j_q, k\}$ such that $A^l_{v_q} = Y$.

Case 3. $\bar{A}^l_{v_q} = Z$. We let $J' = \{j_q, l\}$, $A'^{j_q} = A^{j_q}\Delta\{X, Y\}$, and $A'^l = A^l\Delta\{X, Y\}$. It is clear that (i) is satisfied. The vertex v_q is not l-critical with respect to A because $A^l_{v_q} = Y \neq Z = \bar{A}^{j_q}_{v_q}$. So (ii) is satisfied. The sets A'^{j_q} and A'^l are bases because $Z = \bar{A}^{j_q}_{v_q} \neq Y$ and $Z = \bar{A}^l_{v_q} \neq X$ are satisfied. So (iii) holds. Finally we have $\{A'^{j_q}_{v_q}, A'^l_{v_q}\} = \{Y, X\} = \{A^l_{v_q}, A^{j_q}_{v_q}\}$, so that (iv) is satisfied.

Case 4. $\bar{A}^l_{v_q} \neq Z$. We let $J' = \{j_q, k, l\}$, $A'^{j_q} = A^{j_q}\Delta\{X, Y\}$, $A'^k = A^k\Delta\{Z, X\}$, and $A'^l = A^l\Delta\{Y, Z\}$. Condition (i) is satisfied. The vertex v_q is not k-critical with respect to A because $A^k_{v_q} = Z \neq \bar{A}^l_{v_q}$. The vertex v_q is not l-critical with respect to A because $A^l_{v_q} = Y \neq Z = \bar{A}^{j_q}_{v_q}$. So (ii) holds. The sets A'^{j_q}, A'^k and A'^l are bases because $Z = \bar{A}^{j_q}_{v_q} \neq Y$, $\bar{A}^k_{v_q} \neq X$ and $\bar{A}^l_{v_q} \neq Z$ are satisfied. So (iii) holds. Finally we have $\{A'^{j_q}_{v_q}, A'^k_{v_q}, A'^l_{v_q}\} = \{Y, X, Z\} = \{A^l_{v_q}, A^{j_q}_{v_q}, A^k_{v_q}\}$, so that (iv) is satisfied.

We now prove that (v) and (vi) are satisfied.

Suppose that (v) is not satisfied. Since $\Omega_{v_{q-1}}$ is j_{q-1}-critical with respect to A and A', we have

(vii) $A^{j_{q-1}}_{v_{q-1}} = \bar{A}^{j_q}_{v_{q-1}}$ and $A'^{j_{q-1}}_{v_{q-1}} = \bar{A}'^{j_q}_{v_{q-1}}$.

So the fundamental circuits $C = C(A^{j_q}, \Omega_{v_{q-1}})$ and $C' = C(A'^{j_q}, \Omega_{v_{q-1}})$ exist. Since A'^{j_q} and A^{j_q} coincide on any skew class Ω_v, $v \in V - \{v_{q-1}, v_q\}$, the pair $\{C_v, C'_v\}$ cannot be a skew pair included in $C \cup C'$. The pair $\{C_{v_{q-1}}, C'_{v_{q-1}}\}$ cannot be a skew pair because we have $A'^{j_{q-1}}_{v_{q-1}} = A^{j_{q-1}}_{v_{q-1}}$ by (iii), which implies $C_{v_{q-1}} = \bar{A}^{j_q}_{v_{q-1}} = \bar{A}'^{j_q}_{v_{q-1}} = C'_{v_{q-1}}$ by using (vii). So the pair $\{C_{v_q}, C'_{v_q}\}$ cannot be skew by Proposition 5.4. Since $(v_{q-1}, v_q) \in \mathrm{Arc}(A^{j_q})$, the value C_{v_q} is defined and we have $C_{v_q} = A^{j_q}_{v_q}$. Since $\{C_{v_q}, C'_{v_q}\}$ is not skew, either C'_{v_q} is not defined or $C'_{v_q} = C_{v_q}$. If C'_{v_q} is not defined we have $C' \subseteq A^{j_q} \cup \Omega_{v_{q-1}}$, so that $C' = C$ and $C'_{v_q} = C_{v_q}$, a contradiction. If C'_{v_q} is defined, we have $C'_{v_q} = A'^{j_q}_{v_q}$ and $C_{v_q} = A^{j_q}_{v_q}$. But $j_q \in J'$ implies $A'^{j_q}_{v_q} \neq A^{j_q}_{v_q}$ by (iii), and so $C'_{v_q} \neq C_{v_q}$, again a contradiction.

Let us prove (vi). We have $(v_{r-1}, v_r) \in \mathrm{Arc}(A^{j_r})$. Therefore $C = C(A^{j_r}, \Omega_{v_{r-1}})$

exists and $C_{v_r} = A^{j_r}_{v_r}$. If $A'^{j_r} = A^{j_r}$, the two preceding conditions are still satisfied with respect to A', and so $(v_{r-1}, v_r) \in \mathrm{Arc}(A'^{j_r})$ actually holds. Assume now $A'^{j_r} \neq A^{j_r}$. So we have $j_r \in J'$, which implies Ω_{v_q} is not j_r-critical by (ii). Since Γ has no shortcut, $(v_{r-1}, v_q) \notin \mathrm{Arc}(A^{j_r})$. Therefore C_{v_q} is not defined, which implies $C \subseteq A'^{j_r} \cup \Omega_{v_{r-1}}$. So $C(A'^{j_r}, \Omega_{v_{r-1}})$ is defined and $(v_{r-1}, v_r) \in \mathrm{Arc}(A'^{j_r})$. □

The global algorithm runs as follows. We first construct a suboptimal covering A by using Proposition 6.3 with $I' = (\emptyset : j \in J)$. As long as a condition is satisfied to call one of the three procedures COVER NONOBSTRUCTED, COVER OBSTRUCTED, and COVER SEQUENCE, we call this procedure. When it is no longer possible to call one of the procedures, we construct $S \in \mathcal{S}(\Omega)$ according to the following proposition. Since $\nu(A) = |S| - r(S)$ is satisfied, A is an optimal covering.

Proposition 7.3. *At the end of the algorithm, let $X = \{v : \Omega_v$ is obstructed$\}$ and, for each $j \in J$, let $X^j = \{v : \Omega_v$ is accessible and j-critical$\}$. Let $S \in \mathcal{S}(\Omega)$ be defined as follows*:

$S_v = \bar{A}^j_v, j \in J, v \in X$ (we recall that $\bar{A}^j_v$ does not depend on j when Ω_v is obstructed),

$S_v = A^j_v = \bar{A}^k_v, j \in J, v \in X^j, k \in J - j$ (we recall that $\bar{A}^k_v$ does not depend on k when Ω_v is j-critical),

S_v is not defined otherwise.

Then

$$\nu(A) = r(S) - |S|.$$

Proof. We first verify that $\nu(A) = |X|$. For any $v \notin X$, Ω_v is not obstructed, and so Ω_v is covered by A; otherwise Procedure COVER NONOBSTRUCTED(Ω_v) could be called. For $v \in X$, Ω_v is obstructed, and so there exists $x \in \Omega_v$ that satisfies $\bar{A}^j_v = x$ for all $j \in J$. Since $A^j_v \neq \bar{A}^j_v$ holds for all $j \in J$, the element x cannot be covered by A. This completes the verification. □

Let $\bar{X} = X \bigcup (X^j : j \in J)$.

Claim 7.4. *Let $j \in J$ and $v \in \bar{X} - X^j$. For any $(v, w) \in \mathrm{Arc}(A^j)$ we have $w \in X^j$.*

Proof. Suppose that we can find $v \in \bar{X} - X^j$ and $w \notin X^j$ such that $(v, w) \in \mathrm{Arc}(A^j)$. Consider a critical sequence $\Gamma = (v_0, j_1, v_1, \ldots, j_q, v_q)$ such that $v_q = v$. Let $\Gamma' = (v_0, j_1, v_1, \ldots, j_q, v_q, j, w)$. If w is not j-critical, then Γ' is an improving sequence, so that we may call Procedure COVER SEQUENCE(Γ'), which is a contradiction. If w is j-critical, then Ω_w cannot be obstructed. This implies that Ω_w is covered, otherwise Procedure COVER NONOBSTRUCTED(Ω_w) could be called. Therefore Γ' is a critical sequence that leads from $v_0 \in X$ to w. So w is accessible, which implies $w \in X^j$, a final contradiction. □

Consider any $j \in J$ and any $v \in \bar{X} - X^j$. The fundamental circuit $C = C(A^j, \Omega_v)$ is defined because either $v \in X$ (and so $\bar{A}^j_v$ is defined for all $j \in J$) or $v \in X^k$ for some $k \in J - j$ (and so $\bar{A}^j_v$ is defined because Ω_v is k-critical). It also follows from the definition of S that $C_v = S_v$. Decompose S into two subsets, $S' = \{S_v : v \in X^j\}$ and $S'' = \{S_v : v \in \bar{X} - X^j\}$. So $C_v \in S''$. By axiom 5.3 and the claim we have $sp(C) - v = \{w : (v, w) \in \mathrm{Arc}(A^j)\} \subseteq X^j$. This implies $C - C_v \subseteq S'$. Consider the matroid $M^j = Q^j[S]$ (since S is a subtransversal, M^j is a 1-matroid). The properties $C_v \in S''$ and $C - C_v \subseteq S'$ imply that C is a circuit of M^j (recall that C is a fundamental circuit in Q^j). Since C is a circuit and $C \cap S'' = S_v$, the element S_v belongs to the closure of S' in the matroid M^j. This property holds for all the elements of S''. Therefore $S = S' \cup S''$ is included in the closure of S' in the matroid M^j. This implies $r^j(S) \leq r^j(S') \leq |X^j|$. So $|S| - r(S) = |S| - \sum(r^j(S) : j \in J) \geq |X| = \nu(A)$. We have $\nu(A) \geq |S| - r(S)$ by Proposition 7.2. Therefore $|S| - r(S) = \nu(A)$. □

Where $n = |\Omega|$ and $k = |J|$, we let the reader verify that the time-complexity and the space-complexity, to find a pair of solutions (I, S) satisfying Theorem 6.1, are equal to $O(\max(n, k)(n + k)n^2)$ and $O(n^2 k)$, respectively.

8. Open problems. One knows very little about the simple covering problem for 2-matroids. The basic question is the following one: Does there exist a good characterization of the 2-matroids that can be covered by two bases?

The problem can be specialized to Eulerian multimatroids. Consider a connected 4-regular graph G. Let U be the set of the nonnull local splitters of G and F be a complete splitter of G. Then $Q(G, U - F)$ is a 2-matroid. The question, stated for $Q(G, F)$, is to search for two disjoint Euler tours using no local splitters in F. Jackson [15] shows that Corollary 6.2 cannot hold in general for $Q(G, U - F)$.

There is another interesting way to specialize the problem by using tightness and separators, two notions defined and studied in [10]. Consider a multimatroid $Q = (U, \Omega, r)$. A *separator* of Q is a subset $X \subseteq U$ that is a union of skew classes and satisfies $r(S \cap X) + r(S \setminus X) = r(X)$ for all $S \in \mathcal{S}(\Omega)$. This notion clearly generalizes the similar one for matroids. We denote by $k(Q)$ the number of minimal nonempty separators of Q. The multimatroid Q is said to be *tight* if the union of any base with any skew class includes a circuit. For example the preceding Eulerian 2-matroid $Q(G, U - F)$ is tight if and only if it is possible to provide each half-edge with a sign, $+$ or $-$, in such a way that any two half-edges incident to the same edge have distinct signs and any two half-edges belonging to any pair in any local splitter of F have the same sign. We may imagine that the signs define a direction on each edge, from the positive half-edge towards the negative one, in such a way that precisely two edges leave any vertex. Since G is connected, it is easy to verify that the local splitters of F (which are said to be *antidirected*) define the directions up to a global reversing. Thus tight Eulerian 2-matroids correspond to oriented 4-regular graphs (with the implicit condition that precisely two directed edges leave any vertex).

PROPOSITION 8.1. *If B_1 and B_2 are two bases of a tight 2-matroid $Q = (U, \Omega, r)$ and $S \in \mathcal{S}(\Omega)$, then $|U - (B_1 \cup B_2)| \geq k(Q||S)$.*

The proof of the proposition will be published later. It implies the inequality $kr(s) \geq |s|$ of Corollary 6.2. The reader will see an adaptation of the proposition to the case of oriented 4-regular graphs in a paper with Andersen and Jackson [1]. We do not know whether the min-max equality holds in general.

REFERENCES

[1] L. ANDERSEN, A. BOUCHET, AND B. JACKSON, *Orthogonal A-trails of 4-regular graphs embedded in surfaces of low genus*, J. Combin. Theory Ser. B, 66 (1996), pp. 232–246.

[2] A. BOUCHET, *Multimatroid* IV, *Chain-group representations*, submitted.

[3] A. BOUCHET, *Multimatroids* II, *Minors and connectivity*, submitted.

[4] A. BOUCHET, *Multimatroids* III, *Tightness, fundamental graphs, and pivotings*, submitted.

[5] A. BOUCHET, *Greedy algorithm and symmetric matroids*, Math. Programming, 38 (1987), pp. 147–159.

[6] A. BOUCHET, *Isotropic systems*, European J. Combin., 8 (1987), pp. 231–244.

[7] A. BOUCHET, *Matchings and Δ–matroids*, Discrete Math., 24 (1989), pp. 55–62.

[8] A. BOUCHET, *Compatible Euler tours and supplementary Eulerian vectors*, European J. Combin., 14 (1993), pp. 513–520.

[9] R. Chandrasekaran and S. N. Kabadi, *Pseudomatroids*, Discrete Math., 71 (1988), pp. 205–217.

[10] D. Bénard, A. Bouchet, and A. Duchamp, *Tutte and Martin polynomials*, submitted.

[11] A. Dress and T. Havel, *Some combinatorial properties of discriminants in metric vector spaces*, Adv. Math., 62 (1986), pp. 285–312.

[12] A. Duchamp, *Etudes de quelques notions et propriétés relatives aux matroïdes symétriques: Axiomatiques, Extensions ponctuelles, Quotients, Représentations*, Ph.D. thesis, Université du Maine, 1991.

[13] A. Duchamp, *private communication*, 1993.

[14] J. Edmonds, *Lehman's switching game and a theorem of Tutte and Nash-Williams*, J. Res. Bur. Standards Sect. B, 69B (1965), pp. 73–77.

[15] B. Jackson, *A characterization of graphs having three pairwise compatible Euler tours*, J. Combin. Theory Ser. B, 53 (1991), pp. 80–92.

[16] B. Jackson, *Supplementary Eulerian vectors in isotropic systems*, J. Combin. Theory Ser. B, 53 (1991), pp. 93–105.

[17] A. Kotzig, *private communication to B. Jackson.*

[18] A. Kotzig, *Eulerian lines in finite* 4*-valent graphs and their transformations*, in Theory of Graphs, Erdös and Katona, eds., Academic Press, New York, 1968, pp. 219–230.

[19] L. Lovász, *The matroid matching problem*, in Algebraic Methods in Graph Theory, L. Lovász and V. T. Sos, eds., North-Holland, Amsterdam, 1978, pp. 495–517.

[20] C. Nash-Williams, *Another proof of a theorem concerning detachments of graphs*, European J. Combin., 12 (1991), pp. 245–248.

[21] C. S. J. A. Nash-Williams, *Acyclic detachments of graphs*, in Graph Theory and Combinatorics, Open Univ., Milton Keynes, Pitman, Boston, 1979, pp. 87–97.

[22] L. Qi, *Directed submodularity, ditroids and directed submodular flows*, Math. Programming, 42 (1988), pp. 579–599.

[23] D. Welsh, *Matroid Theory*, Academic Press, New York, 1976.

SIAM J. DISCRETE MATH.
Vol. 10, No. 4, pp. 647–661, November 1997

RANDOMNESS IN PRIVATE COMPUTATIONS*

EYAL KUSHILEVITZ[†] AND YISHAY MANSOUR[‡]

Abstract. We consider the amount of randomness used in *private* distributed computations. Specifically, we show how n players can compute the exclusive-or (xor) of n boolean inputs t-privately, using only $O(t^2 \log(n/t))$ random bits (the best known upper bound is $O(tn)$). We accompany this result by a lower bound on the number of random bits required to carry out this task; we show that any protocol solving this problem requires at least t random bits (again, this significantly improves over the known lower bounds).

For the upper bound, we show how, given m subsets of $\{1, \ldots, n\}$, to construct in (deterministic) polynomial time a probability distribution of n random variables (i.e., a probability distribution over $\{0, 1\}^n$) such that (1) the parity of random variables in each of these m subsets is 0 or 1 with equal probability, and (2) the support of the distribution is of size at most $2m$. This construction generalizes previously considered types of sample spaces (such as k-wise independent spaces and Schulman's spaces [*Sample spaces uniform on neighborhoods*, in Proc. of the 24th Annual ACM Symposium on Theory of Computing, ACM, New York, 1992, pp. 17–25]). We believe that this construction is of independent interest and may have various applications.

Key words. randomness, small probability spaces, privacy, xor function

AMS subject classifications. 94A60, 68R05, 68M10

PII. S0895480196306130

1. Introduction.

There has been a great effort devoted to the study of *randomization*. Initially, the main application of randomization was for solving problems to which deterministic solutions are impossible (e.g., in distributed computing and in cryptography) or unknown (e.g., efficient primality testing). Furthermore, randomization is used to construct both more efficient and, not less significant, much simpler algorithms.

Randomness as a resource was extensively studied in the last decade. One line of research was devoted to a quantitative study of the role of randomness in specific contexts, e.g., [35, 28, 5, 12, 8, 9, 32]; another direction was developing general-purpose methods for saving random bits. These methods range over pseudo-random generators [38, 7, 34], techniques for recycling random bits [22, 19], sources of weak randomness [18, 37, 40], and construction of various kinds of small probability spaces [33, 1, 36, 27, 25, 23]. A particular goal was to allow *derandomization*, i.e., to completely eliminate the use of randomness. For some problems, the best known deterministic algorithms are randomized algorithms which are later derandomized (see, e.g., [24]).

We consider the role of randomness in t-*private* protocols. Informally, a t-private protocol $\mathcal{P}$ for computing a function f is a protocol that allows n players, P_i $(1 \leq i \leq n)$, each holding an individual secret input, x_i, to compute the value of $f(\vec{x})$ in such a way that no coalition of at most t players learns about the initial inputs of other players more than what is revealed by the value of $f(\vec{x})$ and their own inputs. The

* Received by the editors June 12, 1996; accepted for publication (in revised form) September 20, 1996.

http://www.siam.org/journals/sidma/10-4/30613.html

[†] Department of Computer Science, Technion, Technion-City, Haifa 32000, Israel (eyalk@cs.technion.ac.il). The research of this author was supported in part by MANLAM Fund 120-865.

[‡] Department of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel (mansour@cs.tau.ac.il). The research of this author was supported in part by the Israel Science Foundation administered by the Israel Academy of Science and Humanities and by a grant from the Israeli Ministry of Science and Technology.

players are assumed to be honest but curious; namely, they all follow the prescribed protocol $\mathcal{P}$, but they could try to get additional information from the messages they receive during the execution of the protocol. The study of private computations in this setting was initiated by [6, 13] and was the subject of a considerable amount of work, e.g., [3, 14, 31, 4, 20, 15, 16, 17, 29, 30].[1] The use of *randomness* is a crucial ingredient in private protocols; without randomness only degenerate functions can be computed privately.

Protocols for the xor (*exclusive-or*) function (and, more generally, protocols for the modular sum function) are basic building blocks in most private protocols currently known. As a result (and due to its relative simplicity), the task of computing xor t-privately was the subject of previous research [20, 15, 32, 10]. We investigate the amount of randomness required to compute the xor of n input bits t-privately. The known upper bound uses $O(tn)$ random bits.[2] Better upper bounds were known only for the case $t = 1$ (see [32]) in which a single random bit is sufficient. As for lower bounds, Blundo et al. [10] proved that if $t \geq n - c$ (for some constant c) then $\Omega(n^2)$ random bits are required and if $t \geq (2-\sqrt{2})n$ then $\Omega(n)$ bits are required. For smaller values of t it was only known that no deterministic protocol for this task exists.

We significantly improve both the upper bound and the lower bound for this problem. We present a protocol for this task that uses only $O(t^2 \log(\frac{n}{t}))$ random bits. On the other hand, we prove that any t-private protocol for xor requires at least t random bits. This is the best known lower bound for most values of t (i.e., excluding the case where t is very close to n).

For our upper bound, we develop a new construction of small sample spaces that naturally generalizes k-wise independent sample spaces and sample spaces of the type studied by Schulman [36].[3] More precisely, given subsets $T_1, \ldots, T_m \subseteq \{1, \ldots, n\}$ we look for a small sample space over $\{0,1\}^n$ in which the parity of the bit positions in every subset T_j gets the values 0 and 1 with equal probability, i.e., $\frac{1}{2}$. We present a deterministic polynomial-time algorithm that allows constructing such a space for *any* collection of such subsets (in particular, it is important for our application that there is no restriction on the size of the sets). This is a *uniform* space (i.e., a space that consists of a multiset S and the uniform distribution over S) whose size is linear in m (at most $2m$ vectors). We also show how to find such spaces in parallel (in NC). We believe that these constructions are of independent interest and may have various applications.

Relation to other work. The above-mentioned result of Schulman [36] was generalized in various ways [27, 25, 23]. In particular, Karger and Koller [23], presented a construction that can handle parity requirements as in our work. However, there is a significant difference between their construction (as well as those in [27, 25]) and the

[1] This setting is different from the one studied in [39, 21] in which the computational power of the players is restricted and hence intractability assumptions can be used.

[2] This upper bound works as follows. In the first round, each player P_i $(1 \leq i \leq n)$ picks t random bits $r_{i,1}, \ldots, r_{i,t}$ and sends the bit $r_{i,j}$ to P_j $(1 \leq j \leq t)$. Player P_i also sends $x_i + r_{i,1} + \cdots + r_{i,t}$ modulo 2 to P_n. In the second round, each of $P_1, \ldots, P_t$ sends the sum (modulo 2) of the random bits it received in the first round to P_n. Finally, P_n computes the sum (modulo 2) of all the bits he received during the protocol which is exactly the xor of all input bits. The privacy property of this protocol is easy to verify.

[3] Schulman [36] initiated the study of sample spaces that satisfy a list of specific independence constraints. He showed that if we are interested in having n random variables (i.e., a probability distribution over $\{0,1\}^n$) such that only m subsets $H_1, \ldots, H_m$ of them, each of size at most k, will behave uniformly, then one can get a sample space of size $O(m \cdot 2^k)$ (in particular, if $k = O(\log n)$ then the sample space is of polynomial size).

construction presented in this paper: our construction builds a *uniform* sample space, while the constructions of [27, 25, 23] build *nonuniform* sample spaces (i.e., each $s \in S$ is selected with a different probability p_s). While nonuniformity does not matter if the goal is derandomization, it has many disadvantages if we have other applications in mind. In particular, if we want to *sample* in the space then nonuniform sample spaces may not be useful for saving random bits (the goal of the current paper): for sampling in a uniform space $\lceil \log_2 |S| \rceil$ random bits are enough, while for sampling in a nonuniform space we may need much more (depending on the actual values p_s for $s \in S$)[4] and in some cases will not be able to create exactly the same distribution (see [26]). Sampling is needed if, for example, we do not want to pay the $|S|$ penalty in the running time involved with derandomizing an algorithm, but we just want to run the algorithm once. In such a case a nonuniform space does not necessarily reduce the number of random bits required.

In addition, as mentioned, randomness is not only used for achieving efficient algorithms. For example, in cryptographic protocols randomness is used to maintain the secrecy of information. In such applications, we cannot completely eliminate the use of randomness while maintaining the secrecy, so we are just interested in reducing the number of random bits used. For such applications, other constructions such as ϵ-bias (uniform) sample spaces [33, 1] are not useful, as they only guarantee that parities are "almost" balanced. In cryptographic settings this could mean information leakage.[5]

Organization. In section 2 we present the construction of small sample spaces required for our results (its parallel version appears in section 2.4 and other extensions appear in section 2.5). In section 3 we study the question of randomness in private computations. The upper bound that builds on the construction of small sample spaces, appears in section 3.2, while the lower bound appears in section 3.3. (Section 3.1 includes some required definitions.) Finally, the appendix briefly describes an alternative protocol, due to Canetti, to compute xor with a small number of random bits which does not rely on our small sample spaces construction.

2. Constructing small spaces immune to parity tests.

2.1. Preliminaries. Let $S \subseteq \{0,1\}^n$ be a collection of (not necessarily distinct) binary vectors of length n. We denote by $s \in_R S$ a choice of an element of S uniformly at random. The distribution generated by S is the distribution induced by picking $s \in_R S$.

We say that a set S is *immune* to a parity test $T \subseteq \{1,\ldots,n\}$ if

$$\Pr_{s \in_R S}[\oplus_{i \in T} s_i \;=\; 1] \;=\; \frac{1}{2}.$$

Informally, this means that when considering the distribution generated by S, the parity of variables in T is unbiased; i.e., the probability that the parity is 0 (or 1) is exactly $\frac{1}{2}$. Let $T_1,\ldots,T_m$ be m (nonempty) subsets of $\{1,2,\ldots,n\}$. A set S is immune to $T_1,\ldots,T_m$ if it is immune to each T_i. (Note that this does *not* mean that S is immune to *combinations* of tests; such spaces are considered in section 2.5.)

[4] In the above-mentioned work the values p_s ($s \in S$) are obtained via a solution of linear systems (over the reals) and hence they have no particular guarantee.

[5] One can work with ϵ-bias sample spaces and argue (by using the statistical distance measure) that although this does not give a perfect privacy, the information leakage is "small"; note, however, that the cost of making ϵ negligible is in increasing the size of the sample space to super polynomial.

In the sequel, whenever there are operations between elements of $\{0,1\}^n$ they are done in $GF(2^n)$. For example, $x+y$ is simply the bitwise xor of x and y; the inner product is defined by $\langle x,y\rangle = \sum_{i=1}^{n} x_i y_i \bmod 2$. The following definition and lemma play a central role in our constructions.

DEFINITION 2.1. *Let M be an $n\times\ell$ 0–1 matrix. We define the following multiset of size 2^ℓ:*

$$\mathit{space}(M) = \left\{ M\cdot\vec{v}^T \;\mid\; \vec{v}\in\{0,1\}^\ell \right\}.$$

That is, $\mathit{space}(M)$ contains 2^ℓ (not necessarily distinct) vectors/strings where each is n-bit long. The matrix M is referred to as the generating matrix *of $\mathit{space}(M)$. We denote the rows of this matrix by $M_1,\dots,M_n$ and its columns by $M^1,\dots,M^\ell$.*

LEMMA 2.2. *Consider a set T_i, and let M be a matrix as above such that $\sum_{j\in T_i} M_j \neq \vec{0}$. Then $\mathit{space}(M)$ is immune to the set T_i.*

Proof. We need to show that the probability of the parity of the bits in T_i is zero is exactly $\frac{1}{2}$. Note that the parity of the bits of the vector $M\cdot\vec{v}^T$ whose indices are in T_i is simply $1_{T_i}\cdot M\cdot\vec{v}^T$, where 1_{T_i} stands for the characteristic vector of the set T_i. Since $\sum_{j\in T_i} M_j \neq \vec{0}$, then $1_{T_i}\cdot M = \vec{u}$, where $\vec{u}\neq\vec{0}$. The probability that the parity of the bits in T_i is zero is the probability that $\langle\vec{u},\vec{v}\rangle = 0$, where $\vec{v}\in\{0,1\}^\ell$. Since $\vec{u}\neq\vec{0}$, this probability is exactly $\frac{1}{2}$. □

Using this lemma, we reduce the problem of constructing a space which is immune to $T_1,\dots,T_m$ to the problem of constructing a matrix M such that the rows corresponding to each T_i do not sum up to $\vec{0}$. We call such a matrix M *good* (with respect to $T_1,\dots,T_m$). All our constructions make use of this observation; i.e., their aim is to find a good matrix M.

2.2. Randomized construction. Here we present a randomized construction of immune spaces. While a randomized construction is not very useful in the applications, this construction exhibits the possibility of finding such spaces.

Let $\ell = \lceil\log m\rceil + k$ for some parameter k. The construction is simply to select a random 0–1 matrix M of size $n\times\ell$. Note that the construction depends on the number of sets m, but it does *not* depend on the specific sets. Also note that $\mathit{space}(M)$ is of size at most $2\cdot m\cdot 2^k$ and that to verify that the randomly chosen matrix M is good takes polynomial time (in n and m).

LEMMA 2.3. *Let $T_1,\dots,T_m$ be a collection of sets, and let M be a random $0-1$ matrix of size $n\times\ell$. With probability at least $1-\frac{1}{2^k}$, $\mathit{space}(M)$ is immune to all sets $T_1,\dots,T_m$.*

Proof. For convenience, we view the construction as selecting at random, one-by-one, the n rows of the matrix $M_1,\dots,M_n\in\{0,1\}^\ell$. Fix a set $T_i\subseteq\{1,\dots,n\}$ and denote by t the maximal element in T_i. By Lemma 2.2, it is enough that

$$\sum_{j\in T_i} M_j \neq \vec{0}.$$

Consider the tth step in the construction of M, when $M_1,\dots,M_{t-1}$ were already fixed. The only row in the above sum that still should be chosen is M_t, and there is exactly *one* choice, among the 2^ℓ possibilities, that will make this sum equal $\vec{0}$. Hence, the probability of failure for T_i is $\frac{1}{2^\ell}\leq\frac{1}{m\cdot 2^k}$. Therefore, the probability that there exists a set T_i for which we fail is at most m times larger, i.e., at most $\frac{1}{2^k}$. □

2.3. Deterministic construction. In this section we describe a *deterministic* construction. The idea is similar to what we did in the randomized case, but instead of choosing the rows at random we will construct them deterministically entry-by-entry. This time the construction does look at the specific sets $T_1, \ldots, T_m$ in question. Let $\ell = \lceil \log(m+1) \rceil$.

- For $k = 1, \ldots, n$ construct the kth row of M as follows.
  We say that a set T_i is *relevant* for the kth step if k is the maximal element of T_i. The goal in the kth step is to make sure that each of the relevant sets has the property that $\sum_{j \in T_i} M_j \neq \vec{0}$ (the fact that the set is relevant means that k is the only row in the sum that was not fixed yet). For each of these sets there is exactly one value (in $\{0,1\}^\ell$) for the row that violates this property (i.e., with this value $\sum_{j \in T_i} M_j = \vec{0}$). This implies that there are at most m illegal choices for the kth row. However, there are $2^\ell \geq m+1$ possible values for this row; hence, at least one of them satisfies the property with respect to all relevant sets.

LEMMA 2.4. *The above algorithm constructs in time poly(n, m) a matrix M such that space(M) is immune to all sets $T_1, \ldots, T_m$.*

Proof. The correctness follows from the above discussion (particularly, the choice of ℓ and Lemma 2.2). The time complexity is obvious. □

2.4. Parallel construction. In this section we show how immune sample spaces can be constructed in parallel. Obviously, the randomized construction (section 2.2) can be parallelized. Our goal, however, is to show how this can be done deterministically, that is, by an NC-algorithm.

To do so, we construct a matrix M as in section 2.3 but this time in a column-by-column fashion. Also, we think of each set T_i as a vector in $\{0,1\}^n$, which is simply its characteristic vector. Note that if we find a column $M^j \in \{0,1\}^n$ such that the inner product $\langle T_i, M^j \rangle$ is 1 (i.e., $1 = \sum_{k=1}^{n} T_{i,k} \cdot M_k^j = \sum_{k \in T_i} M_k^j$), then T_i has the desired property that $\sum_{j \in T_i} M_j \neq \vec{0}$. We will use for the construction ϵ-bias spaces. These are sets $B \subset \{0,1\}^n$ of size polynomial in n and $\frac{1}{\epsilon}$ such that, for every $x \in \{0,1\}^n$ (and in particular every T_i), $\Pr_{b \in B}(\langle x, b \rangle = 1)$ is at least $\frac{1}{2} - \epsilon$. These spaces were studied in [33, 1], which in particular proved that such spaces can be constructed in NC^1. By a simple counting argument, for every collection of t vectors in $\{0,1\}^n$ there exists $b \in B$ whose inner product with $\frac{1}{2} - \epsilon$ of the vectors gives 1. The idea of the construction is that this vector can be found in parallel. Let $\epsilon = \frac{1}{m}$ and $\ell = \lceil \log_{\frac{2}{1+2\epsilon}}(m+1) \rceil$ (again, the size of the space is $2^\ell = O(m)$).

- Construct an ϵ-bias space B.
- For $j = 1, \ldots, \ell$,
  consider all the sets $T_{i_1}, \ldots, T_{i_t}$ which were not marked as "done" yet. Find (in parallel) a vector $b \in B$ which is good for at least $\frac{1}{2} - \epsilon$ of these sets. Mark these sets as "done." Let $M^j = b$.

As argued above, there exists some vector $b \in B$ which is good for $\frac{1}{2} - \epsilon$ of the sets. In order to find it in parallel, we can check for each b and each T_i whether b is good for T_i (this can be done in $O(\log n)$ time on EREW PRAM).[6] For each b, we need to count the number of T_i's for which it is good (this takes $O(\log m)$ time on an EREW PRAM). Finally, we need to choose the b with the maximum number of good sets (this takes $O(\log n)$ time on EREW PRAM). Therefore, each iteration takes

[6] EREW PRAM = exclusive read exclusive write parallel random access machine.

$O(\log n + \log m)$ time on EREW PRAM. Since there are at most $O(\log m)$ iterations, the total time is $O(\log m \log n + \log^2 m)$, which implies that this is an NC^2 algorithm.

Lemma 2.5. *Let the matrix M be constructed as above. Then, space(M) is immune to all sets $T_1, \ldots, T_m$.*

Proof. By the above discussion, no matter what is the set of unmarked T_i's, there exists b which is good for at least $\frac{1}{2} - \epsilon$ of them. Hence ℓ columns are enough. □

2.5. Extensions of the construction. Many times we are guaranteed some additional properties of the sets $T_1, \ldots, T_m$. Of a particular interest is the case where for every $p \in \{1, \ldots, n\}$ it is known that p belongs to at most d of the sets (see [36]). In this case, we are able to generate a space of size at most $2d$ (instead of size $2m$ guaranteed by the construction of section 2.3).

Lemma 2.6. *Let $T_1, \ldots, T_m$ be a collection of sets such that each element appears in at most d of them. Construct a matrix M using the construction of section 2.3 with the exception that now $\ell = \lceil \log(d+1) \rceil$. Then, space$(M)$ is immune to all sets $T_1, \ldots, T_m$.*

Proof. The proof is similar to the proof of Lemma 2.4, except that now the number of relevant sets for each row k can obviously be bounded by d (instead of m). Since $2^\ell \geq d+1$, then ℓ columns are sufficient. □

Schulman [36] considered the following problem: given sets $H_1, \ldots, H_p$ construct a space whose projection on every H_i yields a uniform distribution. Using our constructions, we get the following corollary, which is a new proof for the results of [36].[7]

Corollary 2.7.[8] *Let $H_1, \ldots, H_p$ be a collection of sets of size at most h. Then* (1) *a space of size $O(p \cdot 2^h)$, whose projection on every H_i is uniform, can be constructed in time polynomial in p, n, and 2^h*; *and* (2) *if in addition every element in $\{1, \ldots, n\}$ appears in at most d of the H_i's, then the construction can be made of size $O(d \cdot 2^h)$.*

Proof. Given sets $H_1, \ldots, H_p$ of size at most h define $\mathcal{T} = \{T |$ for some $i, T \subseteq H_i\}$ (i.e., $\mathcal{T}$ consists of $p \cdot 2^h$ parity tests). Observe that if a space is immune to the sets of $\mathcal{T}$, then its projection on every H_i is uniform. Again this is a standard fact (see, e.g., [2]); to see this, consider a set H_i. When specifying the probability of the parity for each $T \subseteq H_i$, we essentially determine the Fourier transform of the distribution; hence we uniquely determine the probability distribution over H_i. Since we require that all the sets are immune, it implies that the probability distribution over the H_i is the uniform one.

The two parts of the corollary now follow from Lemmas 2.4 and 2.6, respectively. □

It is sometimes useful to have a sample space which is not only immune to a single test T_i but is t-wise immune; namely, if we take $k \leq t$ tests, then we get each combination of the k parities with probability 2^{-k}. This of course is possible only if the k tests are independent (for example, if T_1 and T_2 are disjoint sets, then the result of the parity test $T_3 = T_1 \cup T_2$ is always the sum of the parity tests T_1 and T_2). We show here how to transform our construction into a t-wise immune sample space.[9] Let $T = T_1 \oplus T_2 \oplus \cdots \oplus T_k$ denote the set of elements that appear in an even number of

[7] The construction obtained is *different* from that of [36]. In [36] the vectors of the space are constructed directly without going through what we call the generating matrix. We believe that the construction based on our approach is somewhat simpler.

[8] Both this corollary and Lemma 2.6 are not used in the present paper and appear here only to exemplify the power of our construction.

[9] This transformation is standard and described here for the sake of completeness.

sets in $T_1, \ldots, T_k$. This operation can best be viewed by looking at the characteristic vectors of the sets. Then, the characteristic vector of T is the sum (over $GF(2^n)$) of the characteristic vectors of $T_1, \ldots, T_k$. We say that $T_1, \ldots, T_k$ are *independent* if no subset of them gives $T = \emptyset$ (alternatively, if their characteristic vectors are linearly independent over $GF(2^n)$).

Lemma 2.8. *Given $T_1, \ldots, T_m$, let $\mathcal{T}$ be the collection of all (nonempty) sets of the form $T_{i_1} \oplus \cdots \oplus T_{i_k}$, where $k \leq t$. If a sample space is immune to $\mathcal{T}$, it is t-wise immune to $T_1, \ldots, T_m$.*

Proof. Let $b_1, \ldots, b_m$ be any boolean values, and consider the event in which, for all i, the parity of the inputs in T_i is b_i. Note that such an event is equivalent to the event $\prod_{i=1}^{m}(\chi_{T_i}(x) - b_i)/2 \neq 0$, where $\chi_{T_i}(x) = (-1)^{\Sigma_{j \in T_i} x_i}$ (these are just the Fourier basis functions). Observe that once we multiply out the product in the above expression, we have only terms of the form $\chi_A(x)$, where $A \in \mathcal{T}$, i.e. the event depends only on the parity of sets of the form $A = T_{i_1} \oplus \cdots \oplus T_{i_k}$, where $k \leq t$. □

We can now use each of the constructions presented above. For example, see the following corollary.

Corollary 2.9. *Let $T_1, \ldots, T_m$ be a collection of sets. There exists a sample space which is t-wise immune to the sets $T_1, \ldots, T_m$. The size of the space is bounded by the number of different sets of the form $T_{i_1} \oplus \cdots \oplus T_{i_k}$ (where $k \leq t$) which is at most $O(\sum_{i=0}^{t} \binom{m}{i}) = O(m^t)$. This space can be constructed in time polynomial in n and m^t.*

3. Privacy.

In this section we consider the problem of computing xor t-privately. We prove bounds on the (total) number of random bits used by the n players in order to perform this task. We first present the upper bound (section 3.2) for which we use the results of the previous section. Then (in section 3.3) we give a lower bound for this problem. We start with some required definitions.

3.1. Preliminaries.

A set of n players, $P_1, \ldots, P_n$, each possessing a single bit x_i (known only to P_i), collaborate in a protocol to compute xor (i.e., $\oplus(x_1, \ldots, x_n)$). Each player may toss coins during the protocol. This is formalized as follows: the player P_i holds an infinite tape R_i of random bits. Each such bit gets the value 1 with probability $\frac{1}{2}$ (and the value 0 with probability $\frac{1}{2}$), and the bits are all independent. The number of random bits P_i uses is the position of the right-most cell read by the player P_i on his tape R_i. The number of random bits used by the protocol in a certain execution is the total number of bits used by all players (note that in different executions each player P_i may use a different number of random bits). The randomness complexity of a protocol is the worst-case (over all inputs and all executions) number of random bits. A protocol to compute a function f is said to be *correct* if for every input $\vec{x}$ and for every choice of the random tapes, the protocol terminates with the value $f(\vec{x})$ known to all players.

Next, we define the notion of privacy (we follow, e.g., [6, 14]). For a set of players T (sometimes called *a coalition*), denote by C_T the communication seen by the players in T; that is, all messages the players in T receive during the execution of the protocol (excluding, for convenience, the output messages). A protocol is said to be *t-private* if every coalition T of size at most t sees the same distribution of communication on inputs that look the same for players in T. Formally, for every two inputs $\vec{x}$ and $\vec{y}$ such that $\mathrm{xor}(\vec{x}) = \mathrm{xor}(\vec{y})$ and $x_i = y_i$ for all $i \in T$, for every sequence of messages C, and for every choice of random tapes R_i for players in T, the protocol satisfies

$$\Pr[C_T = C | \{R_i\}_{i \in T}, \vec{x}\,] \quad = \quad \Pr[C_T = C | \{R_i\}_{i \in T}, \vec{y}\,],$$

where the probability ranges over the random tapes of the players not in T.[10]

To simplify the presentation, we also consider a nonstandard model in which, in addition to the n players $P_1, \ldots, P_n$, there is some trusted dealer Q. This dealer has a multiset of vectors S, which are just strings in Σ^n for some set Σ. The trusted dealer participates in the protocol in a very restricted way: Q first tosses some coins and use their outcome to pick a random string from S; he then sends to P_i the ith coordinate of that vector (the P_i's have no other source of randomness); the players $P_1, \ldots, P_n$ proceed without receiving any more messages from Q.[11] We assume that each vector in S has a positive probability to be picked.

The notion of t-privacy is defined in a way similar to the standard model, but here the definition is somewhat simpler since we assume that Q does not participate in any coalition (he is a trusted dealer) and also that all other players are deterministic (i.e., they do not toss coins). That is, for every $T \subseteq \{P_1, \ldots, P_n\}$, every $\vec{x}, \vec{y}$ as above and every sequence of messages C, the protocol satisfies

$$\Pr[C_T = C|\vec{x}] \quad = \quad \Pr[C_T = C|\vec{y}],$$

where the probability ranges over the random choices made by the trusted dealer Q (and C_T includes the random bits received from Q by players in T).

3.2. Upper bound. In this section we present a t-private protocol which computes xor while using a small amount of random bits. We first show how, in the trusted-dealer model, t-private computation of xor can be performed using only $O(t \log(\frac{n}{t}))$ random bits. Then, we modify the protocol to work in the standard model (where no such trusted dealer exists) with a penalty of $O(t)$ in the randomness complexity. Namely, we use $O(t^2 \log(\frac{n}{t}))$ random bits (compared with the previously known upper bound which is $O(tn)$).

We start with the protocol in the trusted-dealer model. We first assume that the trusted dealer Q uses random bits which are uniformly distributed and completely independent. We will analyze this protocol and as a result note that for the proof to go through much weaker requirements regarding the random bits are needed. These requirements are of the type satisfied by the sample spaces constructed above, and hence Q will be able to choose its random bits from such a space. In the following protocol (and throughout this section) all additions are modulo 2.

1. The trusted dealer Q chooses at random $n-1$ random bits $z_1, \ldots, z_{n-1}$. He sends z_i to P_i $(1 \le i \le n-1)$. In addition, Q sends $z_n = \sum_{i=1}^{n-1} z_i$ to P_n.
2. In round i (for $i = 1, \ldots, n-1$), player P_i sums up the message (bit) m_{i-1} he received from P_{i-1} in round $i-1$, the random bit z_i received from Q, and its input x_i, and it sends the result, m_i, to P_{i+1}. (The first player P_1 has to sum only z_1 and x_1 as he receives no other message; hence, we define $m_0 = 0$.) Similarly, in the nth round, P_n computes $m_n = m_{n-1} + x_n + z_n$ and announces m_n as the output.

[10] The definition given above is sometimes called *static* privacy. A more general definition, where the coalition T can be chosen (by an adversary) in an adaptive manner, was defined and discussed in [11].

[11] If we allow the trusted dealer Q to be active, he can collect the inputs of all players and compute the answer. However, we will not be able to transform such a solution to the standard model (with no trusted dealer).

A simple induction shows that, for $1 \leq i \leq n-1$, the message m_i sent by P_i satisfies:

$$m_i = x_i + z_i + m_{i-1} = \sum_{j=1}^{i} x_j + \sum_{j=1}^{i} z_j.$$

This, together with the fact that $z_n = \sum_{i=1}^{n-1} z_i$, implies that

$$m_n = m_{n-1} + x_n + z_n = \sum_{j=1}^{n} x_j,$$

and so the protocol is correct. It remains to prove the privacy of the protocol; namely, that every coalition, given the output of the protocol and the input of coalition members, sees the same distribution of communication. We distinguish between two types of coalitions, depending on whether or not P_n is in the coalition. Consider a coalition $T = \{i_1, i_2, \ldots, i_p\}$, of size $p \leq t$, which does *not* include P_n. The view of this coalition consists of the random bits $z_{i_1}, \ldots, z_{i_p}$ they received from Q in the first step of the protocol and of messages $m_{j_1}, \ldots, m_{j_r}$ $(r \leq t)$ sent from players not in the coalition to players in the coalition. We claim that, for all $\vec{x}$, each assignment of values to these $p+r$ messages is obtained with probability $1/2^{p+r}$. Since this is true for all $\vec{x}$, the privacy with respect to this kind of coalitions follows. For each player $P_{j_k} \notin T$ that sends a message to a member of the coalition, denote by $P_{i(j_k)}$ the last player of the coalition before him (formally, $i(j_k)$ is the maximum value which is smaller than j_k and $P_{i(j_k)} \in T$; if no such index exists then $i(j_k) = 0$). Now, we can express each message m_{j_k} as

$$m_{i(j_k)} + \sum_{i=i(j_k)+1}^{j_k} x_i + Y_{j_k},$$

where Y_{j_k} denotes the sum $\sum_{i=i(j_k)+1}^{j_k} z_i$. Clearly, given $\vec{x}$, the values of $z_{i_1}, \ldots, z_{i_p}$ and $Y_{j_1}, \ldots, Y_{j_r}$ are completely determined by the view of T. Since each of these $p+r$ values depends on different z_i's, they are all equally distributed and independent. Therefore, every communication has probability $1/2^{p+r}$, as claimed.

The case of coalitions that include the player P_n is similar. There, however, the value of the message z_n received by P_n is already determined by the other elements of the view and *the output*. Hence, each communication which is consistent with the output has probability of $1/2^{p-1+r}$.

By the above analysis, it is already clear how to define the sets to which the sample space should be immune: for every $1 \leq i \leq j \leq n-1$, let $T_{i,j} = \{i, \ldots, j\}$. These sets guarantee that each z_{i_j} is uniformly distributed (as we have the singleton T_{i_j,i_j} as a special case) and so is each Y_{j_k} (as it is the parity corresponding to the set $T_{i(j_k)+1,j_k}$). To get the independence among these $p+r \leq 2t$ values, it is sufficient to consider the collection $\mathcal{T}$ of all sets which are composed of taking unions of at most $2t$ disjoint sets as above (this is a standard fact; for details, see the proof of Lemma 2.8). Each such union has at most $2t$ intervals; therefore, the number of sets in $\mathcal{T}$ can be bounded by $|\mathcal{T}| \leq \sum_{i=0}^{2t} \binom{n}{2i} = (\frac{n}{t})^{O(t)}$. In other words, we replace step (1) of the above protocol by the following.

1′. The trusted dealer Q chooses a vector $\vec{z} = z_1, \ldots, z_{n-1}$ from a space which is immune to the sets in $\mathcal{T}$ (as constructed in the previous section). Since, by Definition 2.1, all the spaces we construct are of size which is a power of

2, then $O(\log|\mathcal{T}|) = O(t\log(\frac{n}{t}))$ random bits suffice for choosing a vector in the sample space. Q sends z_i to P_i $(1 \leq i \leq n-1)$ and $z_n = \sum_{i=1}^{n-1} z_i$ to P_n.

The same proofs of correctness and privacy remain valid. Hence, we have just proved the following theorem.

THEOREM 3.1. *There exists an n-party, t-private protocol in the trusted-dealer model to compute xor using $O(t\log(n/t))$ random bits.*

To transform the above protocol to the standard model, we let the players $P_1, \dots, P_t$ simulate the role of the trusted dealer Q. That is, we replace $(1')$ by the following.

$1''$. Each P_j $(1 \leq j \leq t)$ chooses, using $O(t\log(\frac{n}{t}))$ random bits, a vector $\vec{z}^j = z_1^j, \dots, z_{n-1}^j$ from a space which is immune to the sets in $\mathcal{T}$. Player P_j sends z_i^j to P_i $(1 \leq i \leq n-1)$ and $z_n^j = \sum_{i=1}^{n-1} z_i^j$ to P_n. Each P_i computes $z_i = \sum_{j=1}^{t} z_i^j$.

The key observation now is that if the coalition is $P_1, \dots, P_t$, then the coalition gets no messages (from noncoalition players) during the protocol and hence gets no additional information. For any other coalition, there exists at least one player P_m, $1 \leq m \leq t$, whose random string is not known to the coalition. The same proof above, regarding the distribution of communications, can be repeated using only the choices of P_m in the argument. To conclude, we have proved the following theorem.

THEOREM 3.2. *There exists an n-party, t-private protocol to compute xor using $O(t^2\log(n/t))$ random bits.*

Remark. Note that the size of the sets $T_{i,j}$ defined above may be very large (up to $n-1$); hence the sample spaces of [36] are not enough. It should also be clear that just by using a t-wise independent sample space the protocol fails. This is because large sets of variables may be dependent and hence their sum (i.e., the Y_{j_k}'s) is not equally distributed.

3.3. Lower bound. In this section we prove a lower bound on the number of random bits required for solving the privacy problem. We start with a simple combinatorial lemma to be used in the proof.

LEMMA 3.3. *Let S be a set of at most K distinct vectors of n coordinates (i.e., $S \subseteq \Sigma^n$, for some Σ). Then, there are $k = \lfloor \log K \rfloor$ coordinates and an assignment to these k coordinates such that there is a unique vector $\vec{s} \in S$ which is consistent with this assignment.*

Proof. The proof is by induction on K. It can be easily verified for small values of K (e.g., $K = 2, 3$). Now, given a set S, we find a coordinate i in which not all the vectors have the same value. Choose this coordinate, and take a value which appears in at most half of the vectors (but does appear at least once). After this we are left with at most $\frac{K}{2}$ vectors and, by induction hypothesis, we can fix $\lfloor \log(\frac{K}{2}) \rfloor = k-1$ more coordinates so that exactly one vector in S is consistent with the fixed bits. □

Again, for clarity of the presentation, we first consider the trusted-dealer model. Let us call the set of vectors S from which the dealer Q picks its vector the *support* of Q (and recall that each of these vectors is assumed to be picked with positive probability).

THEOREM 3.4. *Let $\mathcal{P}$ be a protocol for $n \geq 3$ players that allows computing xor t-privately, $t \leq n-2$, using a trusted dealer Q. Then, S, the support of Q, has at least 2^t vectors.*

Proof. By contradiction, assume that the trusted dealer has only $2^t - 1$ distinct vectors (in Σ^n for some Σ) in its support set. By Lemma 3.3, there are $t-1$ processors, $P_{i_1}, \dots, P_{i_{t-1}}$, such that if each processor P_{i_j} receives a certain value s_{i_j} from Q, the trusted dealer, then they uniquely determine that the entire vector used by Q is $\vec{s}$. Thus, in this event (which occurs with positive probability), the coalition of the $t-1$ players knows all the values that Q distributed. Intuitively, in such a case, the protocol becomes deterministic. The impossibility result follows from the fact that there is no deterministic 1-private protocol for three or more parties (see [32]). To be more formal, consider the protocol for $n-(t-1) \geq 3$ players that is obtained by running $\mathcal{P}$ while giving each P_{i_j} input 0 and each P_i (in this set and out of it) the corresponding coordinate of $\vec{s}$. It can be seen that the modified protocol is deterministic (the random choices were fixed), 1-private (using the t-privacy of $\mathcal{P}$) and correct. (A more detailed proof can be given, following the proof of Theorem 3.6 below.) □

The above theorem implies that in the trusted-dealer model t random bits are required by the trusted dealer to allow t-private computation of xor. This is quite close to the $O(t\log(\frac{n}{t}))$ upper bound proven for this model. The following theorem shows that the same lower bound holds for the standard model, where no such trusted dealer exists (note that the standard model is *not* a special case of the trusted-dealer model since in the trusted-dealer model all other players are *deterministic*). One of the difficulties in transforming the proof from the trusted-dealer model to the standard model is the possibility that in different executions different players toss the coins.

THEOREM 3.5. *Let $\mathcal{P}$ be a protocol for $n \geq 3$ players that allows computing xor t-privately, $t \leq n-2$. Then, $\mathcal{P}$ requires at least t random bits*; *that is, there exists an input $\vec{x}$ and an execution* (*i.e., an assignment for the random choices*) *in which a total of at least t bits are used.*

Theorem 3.5 follows from the following more general theorem. It claims that not only is there an input $\vec{x}$ for which at least t random bits are used but that this is true for *every* $\vec{x}$. Moreover, it claims that not only are at least t random-bits used but that there are at least t players who toss these bits.

THEOREM 3.6. *Let $\mathcal{P}$ be a protocol for $n \geq 3$ players that allows computing xor t-privately, $t \leq n-2$. Then, for every input $\vec{x}$ there exists an execution of $\mathcal{P}$ in which at least t players toss coins.*

Proof. Suppose, toward a contradiction, that for some input vector, during every execution of $\mathcal{P}$ at most $s \leq t-1$ players toss coins. Assume, without loss of generality, that $\vec{0}$ is such an input, and that in some execution only players $P_1,\ldots,P_s$ toss coins. Denote by $R_1,\ldots,R_s$ possible random tapes for these s players in such an execution.

We will construct a new protocol $\mathcal{P}'$ that computes xor for $n-s \geq 3$ players, $P_{s+1},\ldots,P_n$, 1-privately and deterministically. It is well known that such a protocol does not exist (see [32]), and hence we will get the desired contradiction. To do so, let $P_{s+1},\ldots,P_n$ (who wish to compute the xor of $\vec{x}' = (x_{s+1},\ldots,x_n)$) execute the protocol $\mathcal{P}$ on $\vec{x} = (0,\ldots,0,x_{s+1},\ldots,x_n)$; in addition, if any of these players needs to send a message to one of $P_1,\ldots,P_s$ he informs its value to everybody, and if he needs to receive a message from one of $P_1,\ldots,P_s$ he computes it himself by taking 0 as the input of each of them, the corresponding R_i (as fixed above), and the messages they received in previous rounds (which are known to all). First we will show that $\mathcal{P}'$ is deterministic. Clearly, on the input $\vec{0}$ protocol $\mathcal{P}'$ is deterministic, but we are using $\mathcal{P}'$ also on other inputs $\vec{x}' = (x_{s+1},\ldots,x_n)$ (by running $\mathcal{P}$ on inputs of the form $(0,\ldots,0,x_{s+1},\ldots,x_n)$). The idea is to use the privacy property of $\mathcal{P}$ to show that on all these inputs $\mathcal{P}'$ has to be deterministic as well, namely, that for each input there is only one possible execution.

CLAIM 3.7. *$\mathcal{P}'$ is deterministic.*

Proof. To prove that $\mathcal{P}'$ is deterministic we need to show that for all $\vec{x}'$ the players never reach a point in their code that they are required to flip coins and to use it (to be formalized below). Note that if $\vec{x}'$ is the all-0 vector, then so is $\vec{x}$ (though $\vec{x}$ has n entries while $\vec{x}'$ has only $n - s$ entries). In this case, by the assumptions, none of $P_{s+1}, \ldots, P_n$ flips coins, and the whole communication is fixed to some vector of values C. The problem is that it is not obvious that this is the case for all vectors $\vec{x}$ as above (we will prove that this is the case). Let $step_0$ be the first step in which for some input, some of $P_{s+1}, \ldots, P_n$ tosses a coin. In all previous steps the communication is deterministic (for all inputs), which implies that the messages sent are identical to those in C. To see this, suppose this is false and consider the first step for which on some input P_i sends to P_j the message 0 and on others he sends 1. Since this is the *first* such step, this implies that P_i behaves differently when the input $\vec{x}'$ is the all-0 vector and when it is a vector in which only P_i and P_k (where P_k is any player different than $P_i, P_j, P_1, \ldots, P_s$) have input 1. This implies that in the original protocol, $\mathcal{P}$, the coalition $P_1, \ldots, P_s$ and P_j (which is of size $s+1 \leq t$) could be able to distinguish between these two inputs (when $P_1, \ldots, P_s$ have randomness $R_1, \ldots, R_s$). But, because in both cases they have the same input and the output is the same, they should not be able to do that.

So we proved that up to step $step_0$ for all inputs the communication is identical to C. Now, by assumption, in this step some players toss coins and "use" them. That is, for some input $\vec{x}'$, some player P_i sends to some player P_j either a 0 or 1 both with positive probability (it can be seen that it must be that $x_i = 1$). Again, because so far the communication was the same for all inputs, this is true for the vector in which only P_i and P_k have 1, which again allows the coalition $P_1, \ldots, P_s$ and P_j to distinguish between $\vec{0}$ (on which the communication must be C) and this vector (on which, with positive probability, the communication is different from C), thus contradicting the t-privacy of the original protocol. This concludes the proof of the claim that $\mathcal{P}'$ is deterministic. □

Now, because we know that no player will be required to toss coins in $\mathcal{P}'$, we can claim that $\mathcal{P}'$ is correct; each execution of it on input $x_{s+1}, \ldots, x_n$ has a corresponding execution of the original protocol (on input $0, \ldots, 0, x_{s+1}, \ldots, x_n$ and with randomness $R_1, \ldots, R_s$ and where none of $P_{s+1}, \ldots, P_n$ tosses coins), which by assumption is correct. Note that the inputs given to $P_1, \ldots, P_s$ contribute nothing to the outcome (had we fixed for these players inputs which are different than 0's we may need to flip the outcome).

Finally, we claim that $\mathcal{P}'$ is 1-private; the (deterministic) view in $\mathcal{P}'$ of any single player P_j in $\{P_{s+1}, \ldots, P_n\}$ is the same as the view of $P_1, \ldots, P_s$ together with P_j in the original protocol (where $P_1, \ldots, P_s$ have inputs $0, \ldots, 0$ and randomness $R_1, \ldots, R_s$). Note that inputs for which we have the privacy requirement in $\mathcal{P}'$ remain so if we extend them with the inputs of $P_1, \ldots, P_s$. Hence, the 1-privacy of $\mathcal{P}'$ follows from the t-privacy of $\mathcal{P}$. This concludes the proof of the theorem. □

As we already remarked, the above theorem is stronger than what we actually need to prove by the definition. It is important to notice that we prove that at least t players need to toss coins, and that the proof does not depend on the actual probabilities of the random choices but only on the number of possibilities. Then, this lower bound still works in other models of randomness (such as the model which is considered in the Appendix).

Appendix. An alternative private protocol. In this section, we briefly describe another protocol with similar properties to the protocol presented in section 3.2. This construction does not use the notion of *immune spaces*. The construction was suggested by Ran Canetti and it appears here with his permission.

First, we need a slight modification to our model. Rather than assuming that the players have access only to random bits, we would assume that the players can choose uniformly at random an element from a set of size k using $\lceil \log k \rceil$ bits. (As mentioned in section 3.3, our lower bound applies to this generalized model as well.)

The protocol would be described for the trusted dealer model, and it is a variation on our protocol. Let p be a prime such that $p > n$. The trusted dealer chooses at random a polynomial $q(\cdot)$ over Z_p of degree $2t$. This is done by choosing $2t + 1$ coefficients each taken from the set $\{0, 1, \ldots, p-1\}$; hence, it requires $O(t \log p) = O(t \log n)$ random bits. The trusted dealer sends to player P_i, $1 \leq i \leq n-1$, the value of $z_i = q(i)$, and it sends player P_n the value of $z_n = \sum_{i=1}^{n-1} q(i) \bmod p$.

Similar to our protocol, each player P_i, when it receives a message m_{i-1}, sends to player P_{i+1} the message $m_i = m_{i-1} + z_i + x_i \bmod p$, where x_i is the input of player P_i. The last player P_n outputs the value $m_{n-1} + z_n + x_n \bmod p$.

The argument for correctness is simple and similar to the one of our protocol. The privacy argument uses the fact that the view of any coalition of t players has access to the t values of the polynomial $q(\cdot)$ they received from the trusted dealer, and to at most t messages they received. It is not hard to see that, due to the fact that q is of degree $2t$, the view of any such coalition is random and independent from the specific input to the other players. For example, assume that the coalition consists of t players, $P_{i_1}, \ldots, P_{i_t}$, not including P_n. We will use the values of the trusted dealer to show that any view of this coalition is equally likely. The values $z_{i_1}, \ldots, z_{i_t}$ are just the values of the polynomial q in certain t points. In addition, the view of the coalition contains messages m_{i_j-1} where P_{i_j-1} is not in the coalition. Consider the value z_{i_j-1}; the message m_{i_j-1} can be expressed as some value added with z_{i_j-1} modulo p. Therefore, by the appropriate choice of z_{i_j-1} we can set the value of m_{i_j-1} to any value we like. This implies, using the fact that a random degree-$2t$ polynomial gets every combination of at most $2t$ values with the same probability, that any setting of $z_{i_1}, \ldots, z_{i_t}$ and $m_{i_1-1}, \ldots, m_{i_t-1}$ has the same probability.

Acknowledgment. We wish to thank Noga Alon, Amos Beimel, Ran Canetti, Benny Chor, Oded Goldreich, and Adi Rosén for useful discussions concerning randomness and privacy and for very useful comments and suggestions. In particular, Ran Canetti came up with the construction that appears in the Appendix.

REFERENCES

[1] N. Alon, O. Goldreich, J. Håstad, and R. Peralta, *Simple constructions of almost k-wise independent random variables*, Random Structures Algorithms, 3 (1992), pp. 289–304. (Addendum: 4 (1993), pp. 119–120.)

[2] N. Alon and J. Spencer, *The Probabilistic Method*, John Wiley, New York, 1992.

[3] J. Bar-Ilan and D. Beaver, *Non-cryptographic fault-tolerant computing in a constant number of rounds*, in Proc. of 8th Annual ACM Symposium on Principles of Distributed Computing, ACM, New York, 1989, pp. 201–209.

[4] D. Beaver, *Perfect Privacy for Two-Party Protocols*, Technical Report TR-11-89, Harvard University, Cambridge, MA, 1989.

[5] M. Bellare, O. Goldreich, and S. Goldwasser, *Randomness in interactive proofs*, in Proc. of 31st Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1990, pp. 563–571.

[6] M. Ben-or, S. Goldwasser, and A. Wigderson, *Completeness theorems for non-cryptographic fault-tolerant distributed computation*, in Proc. of 20th Annual ACM Symposium on Theory of Computing, ACM, New York, 1988, pp. 1–10.

[7] M. Blum and S. Micali, *How to generate cryptographically strong sequences of pseudo-random bits*, SIAM J. Comput., 13 (1984), pp. 850–864.

[8] C. Blundo, A. Giorgio Gaggia, and D. R. Stinson, *On the dealer's randomness required in secret sharing schemes*, in Proc. of EuroCrypt94, Lecture Notes in Comput. Sci. 950, 1995, Springer-Verlag, New York, pp. 35–46.

[9] C. Blundo, A. De-Santis, and U. Vaccaro, *Randomness in distribution protocols*, in Proc. of 21st International Colloquium on Automata, Languages and Programming, Lecture Notes in Comput. Sci. 820, 1994, Springer-Verlag, New York, pp. 568–579.

[10] C. Blundo, A. De-Santis, G. Persiano, and U. Vaccaro, *On the number of random bits in totally private computations*, in Proc. of 22nd International Colloquium on Automata, Languages and Programming, Lecture Notes in Comput. Sci. 944, Springer-Verlag, New York, 1995, pp. 171–182.

[11] R. Canetti, U. Feige, O. Goldreich, and M. Naor, *Adaptively secure multi-party computation*, in Proc. of 28th Annual ACM Symposium on Theory of Computing, ACM, New York, 1996, pp. 639–648.

[12] R. Canetti and O. Goldreich, *Bounds on tradeoffs between randomness and communication complexity*, Comput. Complexity, 3 (1993), pp. 141–167.

[13] D. Chaum, C. Crepeau, and I. Damgard, *Multiparty unconditionally secure protocols*, in Proc. of 20th Annual ACM Symposium on Theory of Computing, ACM, New York, 1988, pp. 11–19.

[14] B. Chor and E. Kushilevitz, *A zero-one law for boolean privacy*, SIAM J. Discrete Math., 4 (1991), pp. 36–47.

[15] B. Chor and E. Kushilevitz, *A communication-privacy tradeoff for modular addition*, Inform. Process. Lett., 45 (1993), pp. 205–210.

[16] B. Chor, M. Geréb-Graus, and E. Kushilevitz, *Private computations over the integers*, SIAM J. Comput., 24 (1995), pp. 376–386.

[17] B. Chor, M. Geréb-Graus, and E. Kushilevitz, *On the structure of the privacy hierarchy*, J. Cryptology, 7 (1994), pp. 53–60.

[18] B. Chor and O. Goldreich, *Unbiased bits from sources of weak randomness and probabilistic communication complexity*, SIAM J. Comput., 17 (1988), pp. 230–261.

[19] A. Cohen and A. Wigderson, *Dispersers, deterministic amplification, and weak random sources*, in Proc. of 30th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1989, pp. 14–19.

[20] M. Franklin and M. Yung, *Communication complexity of secure computation*, in Proc. of 24th Annual ACM Symposium on Theory of Computing, ACM, New York, 1992, pp. 699–710.

[21] O. Goldreich, S. Micali, and A. Wigderson, *How to play any mental game*, in Proc. of 19th Annual ACM Symposium on Theory of Computing, ACM, New York, 1987, pp. 218–229.

[22] R. Impagliazzo and D. Zuckerman, *How to recycle random bits*, in Proc. of 30th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1989, pp. 248–253.

[23] D. Karger and D. Koller, *(De)randomized construction of small samples spaces in NC*, in Proc. of 35th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1994, pp. 252–263.

[24] D. Karger and R. Motwani, *Derandomization through approximation: An NC algorithm for minimum cuts*, in Proc. of 26th Annual ACM Symposium on Theory of Computing, ACM, New York, 1994, pp. 497–506.

[25] H. Karloff and Y. Mansour, *On construction of k-wise independent random variables*, in Proc. of 26th Annual ACM Symposium on Theory of Computing, ACM, New York, 1994, pp. 564–573.

[26] D. E. Knuth and A. C. Yao, *The complexity of non-uniform random number generation*, in Algorithms and Complexity, J. Traub, ed., Academic Press, New York, 1976, pp. 357–428.

[27] D. Koller and N. Megiddo, *Constructing small sample spaces satisfying given constraints*, SIAM J. Discrete Math., 7 (1994), pp. 260–274.

[28] D. Krizanc, D. Peleg, and E. Upfal, *A time-randomness tradeoff for oblivious routing*, in Proc. of 20th Annual ACM Symposium on Theory of Computing, ACM, New York, 1988, pp. 93–102.

[29] E. Kushilevitz, S. Micali, and R. Ostrovsky, *Reducibility and completeness in multi-party private computations*, in Proc. of 35th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1994, pp. 478–489.

[30] E. Kushilevitz, R. Ostrovsky, and A. Rosén, *Characterizing linear size circuits in terms of privacy*, in Proc. of 28th Annual ACM Symposium on Theory of Computing, ACM, New York, 1996, pp. 541–550.

[31] E. Kushilevitz, *Privacy and communication complexity*, SIAM J. Discrete Math., 5 (1992), pp. 273–284.

[32] E. Kushilevitz and A. Rosén, *A randomness-rounds tradeoff in private computation, advances in cryptology*, in Proc. of Crypto '94, Y. Desmedt, ed., Lecture Notes in Comput. Sci. 839, Springer-Verlag, New York, 1994, pp. 397-410; SIAM J. Discrete Math., 11 (1998), to appear.

[33] J. Naor and M. Naor, *Small-bias probability spaces: Efficient constructions and applications*, SIAM J. Comput., 22 (1993), pp. 838–856.

[34] N. Nisan, *Pseudorandom generator for space bounded computation*, in Proc. of 22nd Annual ACM Symposium on Theory of Computing, ACM, New York, 1990, pp. 204–212.

[35] P. Raghavan and M. Snir, *Memory vs. randomization in on-line algorithms*, in Proc. of 16th International Colloquium on Automata, Languages and Programming, Lecture Notes in Comput. Sci. 372, Springer-Verlag, New York, 1989, pp. 687–703.

[36] L. J. Schulman, *Sample spaces uniform on neighborhoods*, in Proc. of 24th Annual ACM Symposium on Theory of Computing, ACM, New York, 1992, pp. 17–25.

[37] U. Vazirani and V. Vazirani, *Random polynomial time is equal to slightly-random polynomial time*, in Proc. of 26th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1985, pp. 417–428.

[38] A. C. Yao, *Theory and applications of trapdoor functions*, in Proc. of 23rd Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1982, pp. 80–91.

[39] A. C. Yao, *How to generate and exchange secrets*, in Proc. of 27th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1986, pp. 162–167.

[40] D. Zuckerman, *Simulating BPP using a general weak random source*, in Proc. of 32nd Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1991, pp. 79–89.

SIAM J. Discrete Math.
Vol. 10, No. 4, pp. 662–687, November 1997

008

REALIZING INTERVAL GRAPHS WITH SIZE AND DISTANCE CONSTRAINTS[*]

ITSIK PE'ER[†] AND RON SHAMIR[†]

Abstract. We study the following problem: given an interval graph, does it have a realization which satisfies additional constraints on the distances between interval endpoints? This problem arises in numerous applications in which topological information on intersection of pairs of intervals is accompanied by additional metric information on their order, distance, or size. An important application is physical mapping, a central challenge in the human genome project. Our results are (1) a polynomial algorithm for the problem on interval graphs which admit a unique clique order (UCO graphs). This class of graphs properly contains all prime interval graphs. (2) In case all constraints are upper and lower bounds on individual interval lengths, the problem on UCO graphs is linearly equivalent to deciding if a system of difference inequalities is feasible. (3) Even if all the constraints are prescribed lengths of individual intervals, the problem is NP-complete. Hence, problems (1) and (2) are also NP-complete on arbitrary interval graphs.

Key words. graph algorithms, computational biology, NP-completeness, interval graphs, size constraints, distance constraints

AMS subject classifications. 05G85, 68Q25, 68R10

PII. S0895480196306373

1. Introduction.

A graph $G(V, E)$ is an *interval graph* if one can assign to each vertex v an interval I_v on the real line, so that two intervals have a nonempty intersection if and only if their vertices are adjacent. The set of intervals $\{I_v\}_{v \in V}$ is called a *realization* of G. The problems which we study here are concerned with the existence of an interval realization to a graph, subject to various types of *distance (or difference) constraints* on interval endpoints. These are inequalities of the form $x - y < C_{xy}$ or $x - y \leq C_{xy}$ for variables x, y and constant C_{xy}. Specifically, we study the following problems (we defer further definitions to section 2).

Distance-constrained interval graph ($DCIG$):
INSTANCE: A graph $G = (V, E)$ and a system S of distance constraints on the variables $\{l_v, r_v\}_{v \in V}$.
QUESTION: Does G have a closed interval realization whose endpoints satisfy S? That is, is there a set of intervals $\{[l_v, r_v] | v \in V\}$ which form a realization of G and their endpoints satisfy S?

A special case is $DCIG$ in which all constraints are lower and upper bounds on interval lengths:

Bounded interval graph recognition (BIG):
INSTANCE: A graph $G = (V, E)$ and functions $L : V \to \mathbb{N}$, $U : V \to \mathbb{N}$.
QUESTION: Is there a closed interval realization of G such that for each vertex v, $L(v) \leq |I_v| \leq U(v)$?

In the following problem, each interval must have a prescribed length

[*] Received by the editors June 13, 1996; accepted for publication (in revised form) October 9, 1996. Portions of this paper were presented at the Third European Symposium on Algorithms, Corfu, Greece, 1995. This research was supported in part by a grant from the Ministry of Science and the Arts, Israel.

http://www.siam.org/journals/sidma/10-4/30637.html

[†] Department of Computer Science, Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel (izik@math.tau.ac.il, shamir@math.tau.ac.il).

Measured interval graph recognition (MIG^*):
INSTANCE: A graph $G = (V, E)$ and a *length* function $L : V \to \mathbb{N}$.
QUESTION: Is there a closed interval realization of G in which for every $v \in V$, $|I_v| = L(v)$?

We shall prove here that even MIG^*, the most restricted problem of the three, is strongly NP-complete. Unlike the situation with interval graphs, the fact that the intervals must be closed causes some loss in generality. In contrast, we show that when the interval graph admits a unique consecutive clique order (up to complete reversal), $DCIG$ is polynomial, and, hence, so are the other two problems. The class of graphs satisfying this property (which we call UCO graphs) properly contains the class of prime interval graphs and is recognizable in linear time. Our solution is based on reducing the problem to a system of difference constraints. We also prove that we cannot do better, by showing that the problem of solving a system of difference constraints and the problem BIG on UCO graphs are linearly equivalent.

Interval graphs have been intensively studied due to their central role in many applications (cf. [33, 17, 11]). They arise in many practical problems which require the construction of a time line where each particular event or phenomenon corresponds to an interval representing its duration. Among the applications are planning [3], scheduling [22, 31], archaeology [26], temporal reasoning [2], medical diagnosis [29], and circuit design [36]. There are also nontemporal applications in genetics [6] and behavioral psychology [9]. In the human genome project, a central problem which bears directly on interval graphs is the physical mapping of DNA [8, 25]: it calls for the reconstruction of a map (a realization) for a collection of DNA segments based on information on the pairwise intersections of segments.

In the applications above, size and distance constraints on the intervals may occur naturally: the lengths of events (intervals) may be known precisely or may have upper and lower bounds. The order or distance between two events may be known. This is often the case in scheduling problems and temporal reasoning. In physical mapping, certain experiments provide information on the sizes of the DNA segments [21]. Our goal here is to study how to combine those additional constraints with precise intersection data.

Green and Xu (cf. Green and Green [20]) developed and implemented a program (called SEGMAP) for construction of physical maps of DNA, which utilizes intersection and size data. The intersection data is obtained by experimentally testing whether each of the segments contain a sequence of DNA (called STS) which appears in a unique, unknown location along the chromosome. Hence, two segments which contain a common STS must intersect. Their algorithm works in two phases: the first phase ignores the size data. It obtains a partition of the STSs into groups, and a linear order on the groups. The second phase uses the partial order of phase 1 together with the size data to obtain the map using linear programming algorithms. Our results in section 3 imply that faster algorithms (utilizing network flow techniques) can be used under certain conditions on the data. However, the results in section 5 imply that the general problem tackled by SEGMAP is intractable (unless P=NP) even with perfect data.

Recognizing interval graphs (i.e., deciding if a graph has an interval realization) can be done in linear time [7, 28, 23]. Surprisingly, much less is known about the realization problem when the input contains additional constraints on the realization. The special case of MIG where all intervals have equal (unit) length corresponds to recognizing *unit interval graphs* [33], which can be done in linear time [10]. The special case of $DCIG$, where all distance constraints have the form $r_v - l_u < 0$ or

$l_v - r_u \leq 0$, is the problem of *seriation with side constraints* [27, 19] (also called *interval graph with order constraints*), which can also be solved in linear time [32]. When $DCIG$ is further restricted to the special case where for each pair u, v, where $(u, v) \notin E$, we have either the constraint $r_v - l_u < 0$ or $r_u - l_v < 0$. The problem is equivalent to recognizing an interval order, which can be done in linear time [4]. Fishburn and Graham [12] discussed a special case of BIG where all intervals have the same pair p and q of upper and lower bounds. For each p and q, they characterized the resulting class of interval graphs (and interval orders) in terms of the family of minimal forbidden induced subgraphs (respectively, suborders). They proved that such a family is finite if and only if $\frac{p}{q}$ is rational. In this case, for integer p and q, their characterization yields an exponential time $n^{O(pq)}$ algorithm for identification of such graphs (orders), where n is the number of vertices. Isaak [24] studied a variant of BIG in which the input is an interval order, there are upper and lower integer bounds on individual interval lengths, and the question is whether there exist a realization in which all endpoints are integers. Using Bellman's notion of a distance graph, Isaak gave an $O(\min(n^3, n^{2\frac{1}{2}} \log nC))$ time algorithm for that problem, where C is the sum of bounds on lengths. He also posed the more general problem of BIG, which we answer here. We generalize distance graphs to handle both strict and weak inequalities on endpoints in order to solve $DCIG$ on a particular class of graphs.

There have been other studies on the realization of a set of intervals based on partial information on their intersection, length, and order. Those are different from our problems here inasmuch as the information on intersection is incomplete; i.e., the underlying interval graph is not completely known. Among these are studies on interval sandwich [18], interval satisfiability [19, 37, 32], on interval graphs and orders which have realizations with at most k different lengths [11, Chapter 9], on the smallest interval orders whose representation requires at least k different lengths [11, Chapter 10], and on the number of distinct interval graphs and orders on n vertices which have a realization with k given lengths [35].

The paper is organized as follows: section 2 contains some preliminaries and background. Section 3 studies problem $DCIG$ on UCO graphs and proves its linear equivalence to solving systems of difference constraints. This implies in particular an $O(\min(n^3, n^{2\frac{1}{2}} \log nC))$ time algorithm for all three problems on UCO graphs. In section 4 we sketch a simple proof that $DCIG$ is strongly NP-complete. Section 5 proves the stronger result that MIG^* is strongly NP-complete. The reduction (performed in two steps) is rather involved, but we feel it gives insight on the interplay between the topological side of the problem (i.e., intersection, open or closed intervals) and its metric aspect (i.e., the intervals' sizes).

2. Preliminaries.

A graph $G = (V, E)$ is called an *intersection graph* of a family of sets $S = \{I_v\}_{v \in V}$ if $I_v \cap I_u \neq \emptyset \Leftrightarrow vu \in E$. G is called an *interval graph* if it is an intersection graph of a family $S = \{I_v\}_{v \in V}$ of intervals on the real line. In that case, S is called a *realization* of G. Depending on the convention, each interval may be either closed or open, with no loss of generality. For simplicity, we sometimes use the same names for the intervals and for the corresponding vertices.

For an interval I denote its left and right endpoints by $l(I)$ and $r(I)$, respectively. The *length* of I, denoted $|I|$, is $r(I) - l(I)$. If G has a realization in which all the intervals are of equal length, then it is called a *unit interval graph.*

Let $C_1, \ldots, C_k$ be the maximal cliques in a graph $G = (V, E)$, where $V = \{v_1, \ldots, v_n\}$. The *clique matrix* of G is the $n \times k$ zero-one matrix $C(G) = (m_{ij})$, where $m_{ij} = 1$ if and only if $v_i \in C_j$. If the columns in $C(G)$ can be permuted so that the ones in each row are consecutive, then we say that $C(G)$ has the *consecutive ones* property, and we call such a permutation of the columns a *consecutive (clique) order*. According to Gilmore and Hoffman [16], G is an interval graph if and only if $C(G)$ has the consecutive ones property.

For two nonintersecting intervals x, y, where x is completely to the left of y, we write $x \prec y$ or, equivalently, $y \succ x$. Let $P = (V, <)$ be a partial order. Call $<$ an *interval order* if there exists a set of intervals $S = \{I_v\}_{v \in V}$ such that $v < u$ if and only if $I_v \prec I_u$. S is called a *realization* for P. Call $G = (V, E)$ the *incomparability graph* of P if for each $u, v \in V$, $uv \in E$ if and only if u and v are incomparable in P; i.e., $u \not< v$ and $v \not< u$. Hence, G is an interval graph if and only if it is the incomparability graph of some interval order. In this case we will say that the graph G *admits* the order $<$.

For a vertex $v \in V$ in the graph $G = (V, E)$, denote $N(v) = \{u \in V | uv \in E\}$ and $N[v] = N(v) \cup \{v\}$. For a vertex set $U \subseteq V$ denote $N[U] = \cup_{u \in U} N[u]$ and $N(U) = N[U] \setminus U$. A set $M \subseteq V$ is called a *module* in $G = (V, E)$ if for each $x, y \in M$, and for each $u \notin M$: $xu \in E \Leftrightarrow yu \in E$. Surely, V is a module, and for each $v \in V$, $\{v\}$ is a module. Such modules are called *trivial*. If all modules in G are trivial, then G is called *prime*. For a subset $X \subset V$ define $E[X] = \{uv \in E | u, v \in X\}$. For a module M in the graph G, create the graph $G' = (V', E')$, where $V' = (V \setminus M) \cup \{v\}$, and $E' = E[V \setminus M] \cup \{uv | u \in N(M)\}$. G' is said to be *obtained from G by contracting M to v*. We usually denote by n and m the number of vertices and edges, respectively, in the graph.

3. Distance constraints in UCO graphs.

We call an interval graph *uniquely clique-orderable* (UCO for short) if it has a unique consecutive clique order, up to complete reversal, in every realization. An interval graph G is UCO if and only if the only nontrivial modules in it are cliques [34]. Note that G is UCO if and only if the interval order admitted by G is unique, up to complete reversal, because an interval order of the vertices of G uniquely determines a linear order of the maximal cliques in G and vice versa. Denote this order by $\prec_G$. Note also that the class of UCO graphs properly contains the class of prime interval graphs. UCO graphs can be recognized in linear time by applying the PQ-tree algorithm of Booth and Lueker [7] and noting that G is UCO if and only if the final tree consists of a single internal Q-node and the leaves. This procedure also computes $\prec_G$ in $O(m + n)$ time.

In this section we study the problem *DCIG* when the input graph is UCO. We show how to reduce this problem, in linear time, to the problem of deciding whether a system of difference constraints is feasible. Hence, *DCIG*, *BIG*, and *MIG* are all polynomial on UCO graphs. We also prove that for *BIG* and *DCIG* we cannot do any better, since deciding the feasibility of a system of difference constraints can be reduced in linear time to an instance of *BIG* with a UCO graph.

3.1. A polynomial algorithm for *DCIG* on UCO graphs.

Let $P = (G, A)$ be an instance of *DCIG*, where $G = (V, E)$ is UCO and A is a set of difference inequalities on the interval endpoints. Construct two systems T and $\bar{T}$ of difference constraints on the variables $\{l_v, r_v\}_{v \in V}$, as follows: both systems include all inequalities in A. In addition, for each $x, y \in V$, if $x \prec_G y$ then T contains an inequality $r_x < l_y$, and $\bar{T}$ contains an inequality $r_y < l_x$. If $xy \in E$ then both T and $\bar{T}$ contain an inequality $r_x \geq l_y$ (and $r_y \leq l_x$). With these definitions we prove Lemma 3.1.

LEMMA 3.1. *P has a realization if and only if either T or $\bar{T}$ has a feasible solution.*

Proof. If $X = \{\tilde{l}_v, \tilde{r}_v\}_{v \in V}$ is a feasible solution to T or to $\bar{T}$, then X is a solution to A, and $\{[\tilde{l}_v, \tilde{r}_v]\}_{v \in V}$ realizes G. On the other hand, let $\{[\tilde{l}_v, \tilde{r}_v]\}_{v \in V}$ be a realization of G, whose endpoints satisfy A. Then the order of the intervals $\{[\tilde{l}_v, \tilde{r}_v]\}_{v \in V}$ on the real line is either $\prec_G$ or its reversal. Therefore, $\{\tilde{l}_v, \tilde{r}_v\}_{v \in V}$ is a feasible solution to either T or $\bar{T}$. □

Hence, we can solve our problem by deciding whether system T or $\bar{T}$ is feasible. We shall prove now that a system S of weak and strict difference constraints on n variables is reducible in linear time to a system S' which consists of weak difference constraints, with numbers only $O(n)$ times larger. (Standard transformation techniques [14] would give numbers $O(2^L)$ times larger for binary input length L.)

Assume all constants in S to be integral, and fix $\epsilon \leq \frac{1}{n}$. Define S' to include every weak inequality $x - y \leq c$ in S, and a weak inequality $x - y \leq c - \epsilon$ for every strict inequality $x - y < c$ in S. Note that the number of variables and number of inequalities in the two systems is the same, and the constants in S' (after multiplying by an appropriate factor to restore integrality) are larger than the constants in S by a factor of $\Theta(n)$.

Lemma 3.2. *S has a feasible solution if and only if S' has one.*

Proof. The "if" direction is trivial, since a feasible solution to S' also satisfies S.

To prove the "only if," we generalize the notion of a distance graph (cf. [1, p. 103]) to handle strict and weak inequalities: for a system T of difference constraints, construct a directed weighted graph $D(T) = (V, A)$, with arc weights and arc labels, as follows: for every constraint $x - y \leq C_{xy}$ or $x - y < C_{xy}$ add an arc (y, x) to $D(T)$ with weight C_{xy} and label the arc $\leq$ or $<$, respectively. $D(T)$ is called the *distance graph* of the system T. The *weight* of a path (or a cycle) in this graph is the sum of the weights of its arcs. Bellman has shown that when all inequalities in T are weak, T is feasible if and only if $D(T)$ contains no negative cycle ([5]; see also [1, p. 103]).

Suppose S' is not feasible. Then $D(S')$ must contain a negative-weight cycle c. Let $w(c)$ and $w'(c)$ be the total weight of c in $D(S)$ and $D(S')$, respectively. Distinguish two cases:

- All arcs in c have labels $\leq$. Then $w(c) = w'(c) < 0$. But

$$w(c) = \sum_{(y,x) \in c} C_{xy} \geq \sum_{(y,x) \in c} (x - y) = 0. \tag{1}$$

  Hence, S is infeasible.
- c contains an arc marked $<$. Since the weight of each arc in c differs from the weight of the corresponding arc in c' by no more than ϵ, we get

$$w(c) \leq w'(c) + n\epsilon \leq w'(c) + 1 < 1.$$

  Since the weights in $D(S)$ are integral, it follows that $w(c) \leq 0$. Since the cycle c in $D(S)$ contains an arc marked $<$, the inequality (1) is strict, namely, $w(c) > 0$, so S is infeasible. □

Corollary 3.3. *A system T is feasible if and only if the weight of every cycle in its distance graph $D(T)$ is either positive, or it is zero and the cycle consists of $\leq$ arcs only.*

We now show that addition of identical strict inequalities to the equivalent systems S and S' above maintains the equivalence between them. (We will need this property in section 5.3.) For constants $\{C_i\}_{i \in I_1 \cup I_2 \cup I_3}$, define the following systems S_1, S_2, S_2', and S_3 on the set of variables $X = \{x_i\}_{i=1}^{n}$:

$$(S_1) \qquad x_{j_i} - x_{k_i} \leq C_i, \qquad i \in I_1,$$

$$(S_2) \qquad x_{j_i} - x_{k_i} < C_i, \qquad i \in I_2,$$

$$(S_2') \qquad x_{j_i} - x_{k_i} \leq C_i - \epsilon, \qquad i \in I_2,$$

$$(S_3) \qquad x_{j_i} - x_{k_i} < C_i, \qquad i \in I_3.$$

LEMMA 3.4. *Let $S = S_1 \cup S_2 \cup S_3$ and $S' = S_1 \cup S_2' \cup S_3$, where all C_i's are integers and $\epsilon < \frac{1}{n}$. S has a feasible solution if and only if S' has one.*

Proof. The proof is by induction on the size of I_3. For $I_3 = \phi$, this is Lemma 3.2. Suppose both S and S' have feasible solutions and consider adding a single strict inequality E: $x - y < C$ to both systems. This implies adding an arc $e = yx$ labeled $<$ with $w(e) = C$ to both distance graphs $D(S)$ and $D(S')$. By Corollary 3.3, it suffices to prove that there exists a cycle of nonpositive weight passing through e in $D(S \cup E)$ if and only if such a cycle exists in $D(S' \cup E)$. But for every simple path p from x to y, $w_{S'}(p) = w_S(p) - k\epsilon$, where $k < n$ and $w_S(p)$ is an integer. Hence, $\lceil w_{S'}(p) \rceil = w_S(p)$, and since C is integral, $w_{S'}(p \cup e) \leq 0$ if and only if $w_S(p \cup e) \leq 0$. □

By Lemmas 3.1 and 3.2, solving an instance of *DCIG* linearly reduces into determining if at least one of two systems of difference constraints is feasible. Using the distance graph reformulation, the feasibility of such a system with M weak inequalities on N variables, with sum of absolute values of arc weights C, can be decided in $O(\min(NM, \sqrt{N}M \log NC))$ time [30, 13]. In our instance (G, A) there are n vertices, so $N = 2n, M = \Theta(n^2)$. Hence Corollary 3.5 follows.

COROLLARY 3.5. *Deciding if a UCO graph with difference constraints has a realization can be done in $O(\min(n^3, n^{2\frac{1}{2}} \log nC))$ time.*

Note that the algorithms of [30, 13] for deciding the feasibility of a system also produce a feasible solution if one exists. This enables construction of a realization (if one exists) in $O(\min(n^3, n^{2\frac{1}{2}} \log nC))$ time.

3.2. Reducing a system of difference constraints to *BIG* on UCO graphs.

Given a system of weak difference constraints, we shall show how to reduce it, in linear time, to an equivalent instance of *BIG*, in which the graph is UCO. According to Lemma 3.2, the assumption that all constraints are weak can be made without loss of generality.

Let P be the following system of weak difference constraints in the variables $X = \{x_1, \ldots, x_N\}$:

$$(\mathrm{P}) \qquad x_{j_i} - x_{k_i} \leq c_i, \qquad i = 1, \ldots, M.$$

Fix $C > 1 + \sum_{i=1}^{M} |c_i|$, and let $c_i' = c_i + (j_i - k_i)C$. Define a new system P' of difference constraints on the same variable set X:

$$(\mathrm{P}') \qquad x_{j_i} - x_{k_i} \leq c_i', \qquad i = 1, \ldots, M.$$

Note that the choice of C guarantees that $c_i' > 1$ (< -1) if and only if $j_i > k_i$ $(j_i < k_i)$, so P' can be rewritten as

$$(\mathrm{P}') \qquad \begin{array}{ll} x_{j_i} - x_{k_i} \leq c_i', & j_i > k_i, \\ x_{k_i} - x_{j_i} \geq -c_i', & j_i < k_i, \end{array}$$

where all right-hand-side terms are larger than one.

We call a solution $\{\tilde{x}_i\}_{i=1}^N$ to P' *monotone* if $\tilde{x}_i < \tilde{x}_{i+1} - 1$ for each $i = 1, \dots, N-1$.

Lemma 3.6. *P has a solution if and only if P' has a monotone solution. Moreover, if $\bar{X} = \{\bar{x}_i\}$ is a feasible solution to P for which $\Delta = \max\{\bar{x}_i\} - \min\{\bar{x}_i\}$ is minimal, then $\bar{X}' = \{\bar{x}'_i | \bar{x}'_i = \bar{x}_i + iC\}$ is a monotone feasible solution to P'.*

Proof. Suppose P' has a monotone solution $x'_1 < x'_2 < \cdots < x'_N$. Let $\tilde{x}_i = x'_i - iC$ for each $1 \le i \le N$. By P' we get, for each $1 \le i \le M$: $\tilde{x}_{j_i} - \tilde{x}_{k_i} = x'_{j_i} - x'_{k_i} - (j_i - k_i)C \le c'_i - (j_i - k_i)C = c_i$. Therefore, the ith inequality in P is satisfied by $\{\tilde{x}_i\}_{i=1}^N$. Hence, P has a feasible solution.

Let $\tilde{x}_1, \dots, \tilde{x}_N$ be a solution of P for which $\Delta = \max\{\tilde{x}_i\} - \min\{\tilde{x}_i\}$ is minimal. (P defines an intersection of closed half-spaces, which is a closed set; therefore, there is a solution attaining this minimal value.) By [5], Δ is the sum of arc weights along some simple path in the distance graph $D(P)$; hence, $\Delta < C - 1$. Let $x'_i = \tilde{x}_i + iC$, for each $1 \le i \le N$. By P, for each $1 \le i \le M$ we get the following: $x'_{j_i} - x'_{k_i} = \tilde{x}_{j_i} - \tilde{x}_{k_i} + (j_i - k_i)C \le c_i + (j_i - k_i)C = c'_i$. Hence, $X' = \{x'_i\}_{i=1}^k$ is a feasible solution of P'. $\tilde{x}_i - \tilde{x}_j \le \Delta < (C-1)(j-i)$ for each $1 \le i < j \le N$; hence, $x'_i - x'_j = \tilde{x}_i - \tilde{x}_j + (i-j)C < -1$, and X' is monotone. □

For the above system P, define $J = (G, U, L)$ to be the following *BIG* instance (compare Figure 1):

- G is the intersection graph of the set of intervals $A \cup B \cup W$, defined as follows:
  - $A = \{a_i\}_{i=0}^N$, where $a_i = [i, i+1]$;
  - $B = \{b_{i/2}\}_{i=1}^{2N+1}$, where $b_x = [x, x]$;
  - $W = \{w_{j_i k_i}\}_{i=1}^M$, where if $j_i > k_i$ then $w_{j_i k_i} = [k_i, j_i]$, and if $j_i < k_i$ then $w_{j_i k_i} = [j_i + \frac{1}{4}, k_i - \frac{1}{4}]$.
- The length constraints are as follows:
  - $U(a_i) = \infty, L(a_i) = 0$;
  - for integral i, $U(b_i) = L(b_i) = 0$ and $U(b_{i+\frac{1}{2}}) = L(b_{i+\frac{1}{2}}) = 1$;
  - if $j_i > k_i$, then $L(w_{j_i k_i}) = 0$, $U(w_{j_i k_i}) = c'_i$;
  - if $j_i < k_i$, then $L(w_{j_i k_i}) = -c'_i - \epsilon$, where $\epsilon < 1/N$, and $U(w_{j_i k_i}) = \infty$.

Lemma 3.7. *G is UCO.*

Proof. Let G' be the intersection graph of $A \cup B$. It is easy to see that G' is prime, and, hence, it has a unique clique order [34]. Moreover, G' has exactly $2N+1$ maximal cliques, each one containing (among other vertices) a unique and distinct b_i. The set of maximal cliques in G is $\{N[b_x]\}_{b_x \in B}$; namely, each clique is distinguished by a single b_i. Since G' is UCO, its unique clique order determines a unique linear order on $\{b_x | x \in B\}$ and, hence, also on the maximal cliques of G. Hence, G is UCO. □

Theorem 3.8. *P has a feasible solution if and only if J has a realization.*

Proof. **Only if:** Suppose $\{\tilde{x}_i\}_{i=1}^N$ is a feasible solution to P for which $\Delta = \max\{\tilde{x}_i\} - \min\{\tilde{x}_i\}$ is minimal. By Lemma 3.6, $\{x'_i\}_{i=1}^N$, where $x'_i = \tilde{x}_i + iC$ is a monotone solution to P'. Choose arbitrary $x'_0 < x'_1 - 1$, $x'_{N+1} > x'_N + 1$. Define the following set $R \cup T \cup S$ of intervals:

- $T = \{T_i\}_{i=0}^N$, where $T_i = [x'_i, x'_{i+1}]$;
- $R = \{R_{i/2}\}_{i=1}^{2N+1}$, where $R_i = [x'_i, x'_i]$ if i is integral, and $R_{i+\frac{1}{2}} = [\frac{x'_i + x'_{i+1} - 1}{2}, \frac{x'_i + x'_{i+1} + 1}{2}]$ otherwise;
- $S = \{S_i\}_{i=1}^M$, where $S_i = [x'_{k_i}, x'_{j_i}]$ if $j_i > k_i$, and $S_i = [x'_{j_i} + \frac{\epsilon}{2}, x'_{k_i} - \frac{\epsilon}{2}]$ if $j_i < k_i$.

Since X' is monotone, the intersection graph of $R \cup T \cup S$ is isomorphic to G.

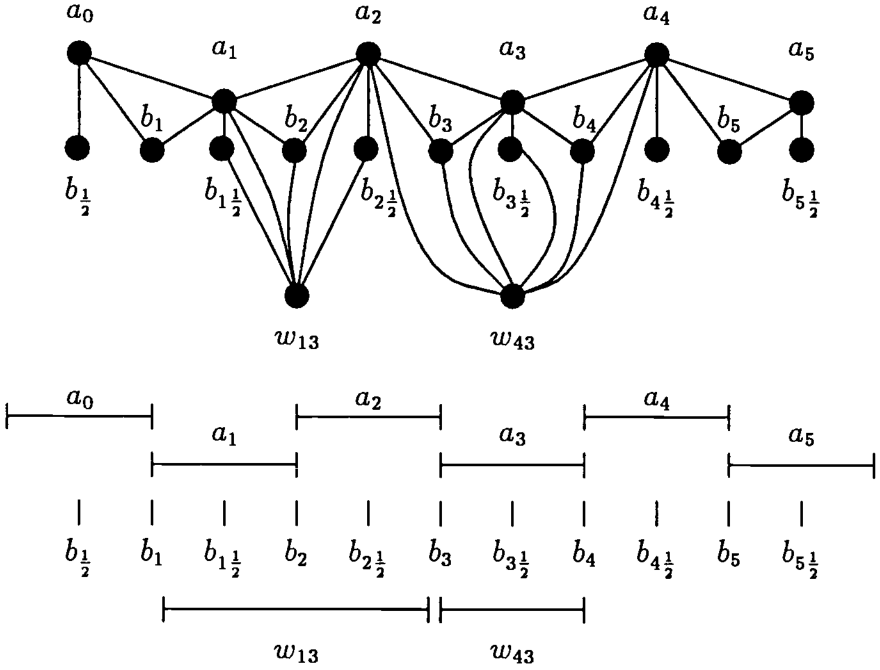


FIG. 1. *The graph G used in the reduction* (*top*) *and a realization for it* (*bottom*).

The length bounds on vertices of T and R are trivially satisfied. If $j_i > k_i$ then $|S_i| = x'_{j_i} - x'_{k_i} \leq c'_i$ as required. If $j_i < k_i$ then $x'_{k_i} - x'_{j_i} \geq -c'_i$, so indeed $|S_i| = x'_{k_i} - x'_{j_i} - \epsilon \geq -c'_i - \epsilon$, satisfying the length bounds on the vertices of S.

If: Suppose J has a realization. Let $\{y_i\}_{i=1}^N$ be the points in a realization of J which correspond to the intervals $\{b_i\}_{i=1}^N$ (which have length zero). Without loss of generality $y_N > y_1$, because otherwise we can reverse the realization. Since G is UCO, the order of the intervals in J is identical to the order of the intervals $A \cup B \cup W$ in the definition of G. Therefore $y_i < y_j$ if and only if $i < j$, and, due to the length constraint on $b_{i+\frac{1}{2}}$, $y_i < y_{i+1} - 1$ for $i = 1, \dots, N-1$. Let S_i be the interval corresponding to $w_{j_i k_i}$ in the realization. Define a system P'' of difference constraints as follows:

$$(\mathrm{P}'') \qquad \begin{array}{ll} x_{j_i} - x_{k_i} \leq c'_i, & j_i > k_i, \\ x_{k_i} - x_{j_i} > -c'_i - \epsilon, & j_i < k_i. \end{array}$$

If $j_i < k_i$ then $y_{k_i} - y_{j_i} > |S_i| \geq -c'_i - \epsilon$, and if $j_i > k_i$ then $y_{j_i} - y_{k_i} \leq |S_i| \leq c'_i$. It follows that $\{y_i\}_{i=1}^N$ is a monotone solution to P''. A proof similar to Lemma 3.2 implies that P' and P'' are equivalent, so P' is feasible. We would like to show that P' has a monotone solution. Let Q' be the system of constraints $x_i < x_{i+1} - 1$, $i = 1, \dots, N-1$. $P' \cup Q'$ and $P'' \cup Q'$ have only monotone solutions. According to Lemma 3.4, adding Q' to both P' and P'' maintains the equivalence between them. But a monotone solution of P'' realizes $P'' \cup Q'$; hence, P' has a monotone solution and according to Lemma 3.6 P is feasible. □

COROLLARY 3.9. *The problem of deciding whether there exists a feasible solution to a system of difference constraints is linearly reducible to the problem BIG on a UCO graph.*

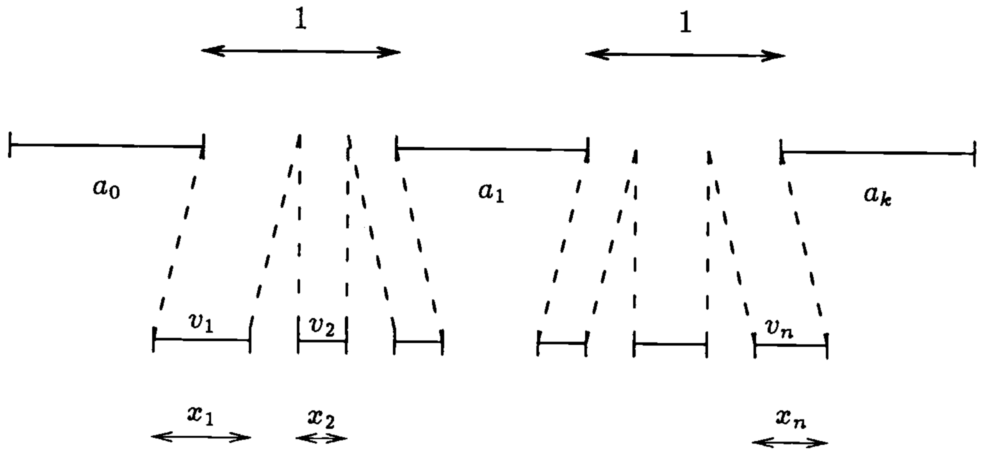


FIG. 2. *The v_i's can be squeezed between the a_i's if and only if a 3-partition exists.*

4. *DCIG* is NP-complete.

We will now show that although *DCIG* is polynomial when restricted to UCO graphs, it is NP-complete in general. A stronger result will be proven in the next section, but we include a sketch of this proof as it is much more transparent.

THEOREM 4.1. *DCIG is strongly NP-complete.*

Proof. We show a pseudo-polynomial reduction from the problem 3-PARTITION which is known to be strongly NP-complete (see, e.g., [15]).

An instance of 3-PARTITION is a set X of $n = 3k$ real numbers $x_1, \ldots, x_n \in (\frac{1}{4}, \frac{1}{2})$ such that $\sum_{i=1}^{n} x_i = k$. The question is to determine whether there exists a partition of X into k subsets (which have to be triplets) $X_1, \ldots, X_k$ so that for each $1 \leq j \leq k$, $\sum_{x \in X_j} x = 1$.

Let $X = \{x_1, \ldots, x_n\}$ be an instance of 3-PARTITION. Define an instance of *DCIG*, $I = (G, S)$ where G is the empty graph on the vertices $\{v_j\}_{j=1}^{n} \cup \{a_j\}_{j=0}^{k}$, and S consists of the following three types of constraints:

- $r(v_j) - l(v_j) \geq x_j$ for each $1 \leq j \leq n$;
- $l(a_{j+1}) - r(a_j) = 1$ for each $0 \leq j \leq k-1$;
- $r(a_0) \leq r(v_j) \leq l(a_k)$ for each $1 \leq j \leq n$.

We shall see that I is satisfiable if and only if X is a "yes" instance (see Figure 2). Assume for now that all intervals in X must be open.

Suppose there exists a partition $X_1, \ldots, X_k$ as required, where $X_j = \{x_j^i\}_{i=1}^{3}$. Define $s_j^i = \sum_{r=1}^{i} x_j^r$, $s_j^0 = 0$. Examine the set of intervals $T = \{I_{a_j}\}_{j=0}^{k} \cup \{I_{v_j^i}\}_{1 \leq j \leq k, 1 \leq i \leq 3}$ where $I_{a_j} = (2j-1, 2j)$ and $I_{v_j^i} = (2j + s_j^{i-1}, 2j + s_j^i)$. The intervals in T are disjoint, and their endpoints trivially satisfy S; hence, T is a realization of I.

Conversely, suppose $\{I_{a_i}\}_{i=0}^{k} \cup \{I_{v_i}\}_{i=1}^{n}$ is a realization of I. For each $1 \leq j \leq k$ define $I_j = (r(a_{j-1}), l(a_j))$, $X_j = \{x_i | I_{v_i} \subseteq I_j\}$. According to the constraints, $l(a_0) < r(a_0) < \cdots < l(a_k) < r(a_k)$, the I_j's do not intersect each other, and therefore the sets X_j are disjoint. Moreover, every x_i is a member of some X_j. Therefore, $X_1, \ldots, X_k$ is a partition of X. For each $1 \leq j \leq n$, since G is empty all the I_{v_i}'s are disjoint; hence, $\sum_{x_i \in X_j} x_i \leq |I_j| = 1$; and, hence, $X_1, \ldots, X_k$ form a 3-partition.

We assumed here that all intervals in the realization are open. To form a closed realization, it suffices to modify the reduction by allowing an interval of length $1 + \epsilon$

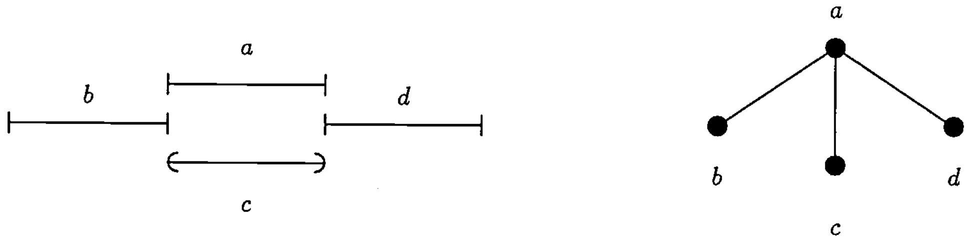


FIG. 3. *The $K_{1,3}$ graph shown (right) has a realization (left) if all intervals but c are closed.*

(instead of length 1) for each "gap" interval $[r(a_{i-1}), l(a_i)]$, where ϵ is sufficiently small. (If each $a_i = \frac{p_i}{q_i}$, where p_i,q_i are integers, then $\epsilon < \frac{1}{4}(\max_i q_i)^{-3}$ suffices.)

Since 3-partition is strongly NP-complete, and the reduction is pseudopolynomial, our problem is strongly NP-complete. □

5. Recognizing measured interval graphs is NP-complete.

In this section we prove the NP-completeness of the problem *MIG**, introduced in section 1. The main part in this proof is a hardness result for the following, slightly more general problem in which we specify in advance for each interval whether it should be closed or open:

> RECOGNIZING A MEASURED INTERVAL GRAPH WITH SPECIFIED ENDPOINTS (*MIG*):
> **INSTANCE:** A graph $G = (V, E)$, a nonnegative *length* $L(v)$ for every $v \in V$, and a function $\phi : V \to \{open, closed\}$.
> **QUESTION:** Is there a realization of G in which the length of I_v is exactly $L(v)$, and I_v is open if and only if $\phi(v) = open$?

We shall denote such an instance by $P = (G, L, \phi)$. When P is a "yes" instance, we say that P is a *measured interval graph* (with endpoint specification). We shall first prove that *MIG* is NP-complete and then reduce *MIG* to *MIG**.

The issue of endpoint specification seems unnatural at first sight. It is well known that for interval graphs in general the endpoint specification can be arbitrary; namely, a graph is interval if and only if it has a realization for any possible specification of endpoints. This is not the case in the presence of length constraints. For example, a $K_{1,3}$ graph with length 1 assigned to all vertices has no realization if all intervals are open (or all closed), but it has a realization precisely if the degree-3 vertex and two of the others are closed, as in Figure 3.

We shall often use the following implicit formulation for the problem by representing G and ϕ using intervals and using L to modify their length:

> *MIG*: IMPLICIT FORMULATION:
> **INSTANCE:** A pair (T, L) where $T = \{I_x\}_{x \in V}$ is a set of intervals and $L : V \to \mathbb{Q}^+$ is a length function.
> **QUESTION:** Is there a set of intervals $S = \{J_x\}_{x \in V}$ s.t. (i) $J_x \cap J_y \neq \phi$ if and only if $I_x \cap I_y \neq \phi$, (ii) $|J_x| = L(x)$ for all x, (iii) J_x is closed if and only if I_x is closed.

This formulation is sometimes more convenient as it suggests a possible realization. We need the following notation and definitions.

DEFINITION 5.1. *Let $P = (G, L, \phi)$ be a measured interval graph with endpoints specification, and let $U \subseteq V$ be a set of its vertices. Define the measured interval graph P_U* induced *by P on U to be (G_U, L_U, ϕ_U), where G_U is the subgraph of G induced* *on U, and L_U, ϕ_U are the restrictions of L and ϕ, respectively, to U.*

Call two MIG instances $P = (G, L, \phi)$ and $P' = (G', L', \phi')$ isomorphic *if there is a graph isomorphism f between G and G', and for each $v \in V(G)$ the following holds: $L(v) = L'(f(v))$ and $\phi(v) = \phi'(f(v))$; namely, the length and closure properties of intervals are preserved by f. In this case denote $P \cong P'$.*

Definition 5.2. *Let $P = (G, L, \phi)$ be an instance of MIG. Let $S = \{I_1, \ldots, I_{|V(G)|}\}$ be a realization of P. Define Length$(S) = \sup\{x \in I | I \in S\} - \inf\{x \in I | I \in S\}$. Define Length$(P) = \inf\{Length(S) | S$ is a realization of $P\}$.*

5.1. Basic structures. We now describe three “gadgets” which are building blocks in our NP-completeness construction and prove some of their properties. The structure of these gadgets assures us that their realization has very few degrees of freedom. To formalize this we introduce the following definition.

Definition 5.3. *Two realizations of the same interval graph are* isometric *if they are identical up to reversal and an additive shift. Namely, there exists a function $f(x) = s \cdot x + c$ where $s = \pm 1$ and $c \in \mathbb{R}$, and $f(I_j) = I'_j$ for all j. Let $P = (G, L, \phi)$ be an instance of MIG. We call $U \subseteq V(G)$* rigid *in P if in any two realizations of P, the sets of intervals realizing U are isometric. In particular, all endpoints are located at fixed distances from the leftmost endpoint, including the rightmost one. Thus in every realization U has the same length. If $V(G)$ is rigid in P, we call P* rigid.

Note that the fact that U is rigid in P does not imply that P_U is rigid. For example, the instance P defined implicitly by the intervals in Figure 3 is rigid, and in particular $\{b, c, d\}$ is rigid in P, but $P_{\{b,c,d\}}$ is not rigid.

5.1.1. The switch. We first define the switch, a gadget which will be used as a toggle in larger structures. For the parameter real value $a \geq 1$, define the *MIG* instance *Switch*$(a) = (G, L, \phi)$ as follows (compare Figure 4): G is the graph on the five vertices v_1, v_2, v_3, v_4, v_5, with edges $v_1v_2, v_2v_3, v_2v_4, v_3v_4, v_4v_5$. L assigns lengths $0, \frac{1}{2}, 1, \frac{1}{2}, a-1$ to $v_1, \ldots, v_5$, respectively, and $\phi(v)$ specifies v_3 to be *open* and all the other vertices to be *closed*.

A realization $\{I_1, I_2, I_3, I_4, I_5\}$ of a *Switch*(a) will be called *straight* if I_1 is to the left of I_5. Otherwise, it will be called *reversed*. We say that such a realization is *located* at I_3. For a straight realization U of a *Switch*(a) located at $(x, x+1)$, denote by $-U$ the reverse realization located at $(x+a-1, x+a)$. Hence, $-U$ is a “mirror image” of U, covering the same interval $[x, x+a]$ along the real line.

Lemma 5.4. *Switch(a) is rigid. In particular, Length$(Switch(a)) = a$.*

Proof. Let S be a straight realization, as in the top left of Figure 4. Suppose S' is another realization such that both leftmost endpoints I_1 and I'_1 are identical. The intersection graph of a *Switch*(a) is prime; hence, I_3 is between I_1 and I_5, and $l(I_5) \geq l(I_1) + 1$. But $l(I_5) \leq l(I_4) + \frac{1}{2} \leq l(I_2) + \frac{1}{2} + \frac{1}{2} \leq l(I_1) + \frac{1}{2} + \frac{1}{2}$; therefore, all inequalities hold as equalities. In particular, *Length*$(S) = a$, yielding *Length*$(Switch(a)) = a$. ☐

Note that Lemma 5.4 implies that a realization of a straight *Switch*(a) located at $(x, x+1)$ is unique. The same is true for a reversed *Switch*.

Lemma 5.5. *Let $P = (G, L, \phi)$ be an instance of MIG, and let $S = \{I(v)\}_{v \in V(G)}$ be a realization for it. Let $U \subseteq V$ be a module such that $P_U \cong Switch(a)$. Then for $x, y \in N(U)$, $|I(x) \cap I(y)| \geq 1$.*

Proof. Let $U = \{v_i\}_{i=1}^{5}$, with vertices numbered in the same order as in the definition of a *Switch*. According to Lemma 5.4, $I(v_1)$ and $I(v_5)$ are one unit apart, but both of them intersect $I(x)$ and $I(y)$. Therefore, $I(x) \cap I(y)$ contains the unit length interval between $I(v_1)$ and $I(v_5)$, yielding $|I(x) \cap I(y)| \geq 1$. ☐

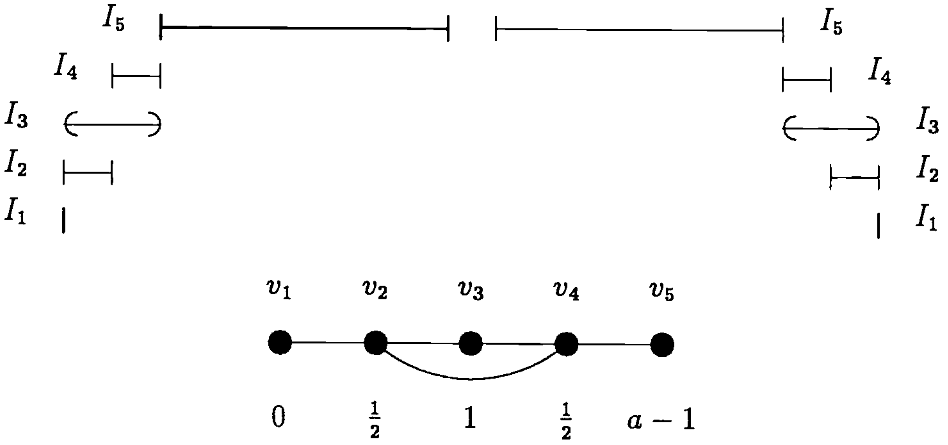


FIG. 4. *The Switch (bottom), a straight realization (top left), and a reversed realization (top right).*

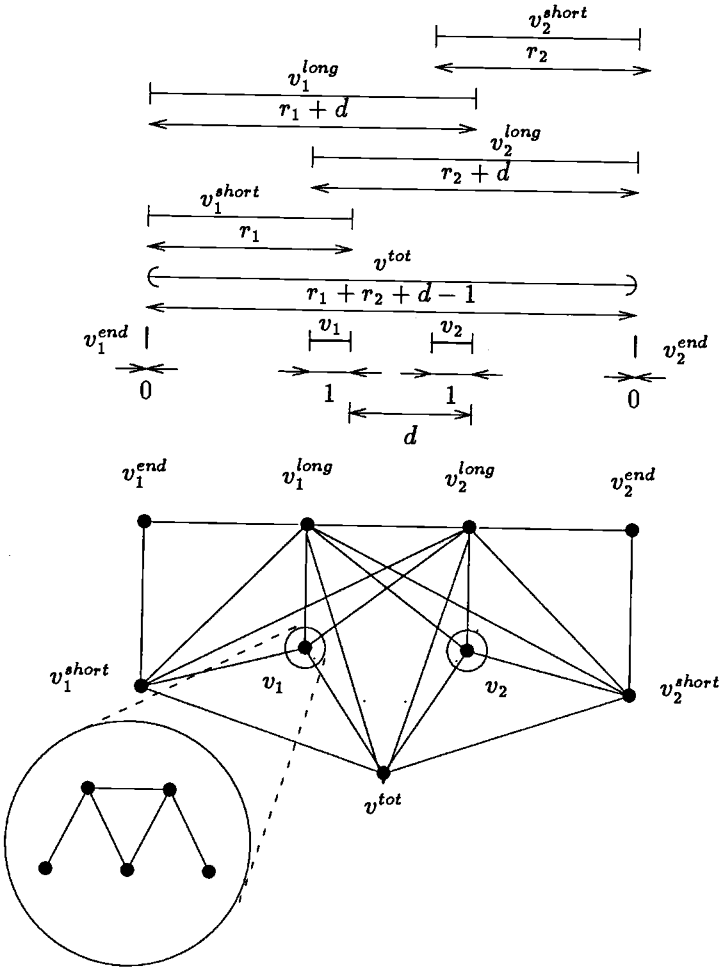


FIG. 5. *The Fetters (bottom) and a realization of it (top). The distance between the Switches* v_1 *and* v_2 *is fixed.*

5.1.2. The fetters. Our second gadget binds two *Switch*es and imposes a prescribed distance between them. For positive real parameters, d, r_1, r_2, and two sets of vertices, U_1 and U_2, define a five-parameter instance of MIG, $Fetters(d, r_1, r_2, U_1, U_2)$ or, in short, $Fetters = (G, L, \phi)$, as follows:

- U_1 and U_2 are modules in G, and each of them induces a *Switch*. More precisely, there exist constants a_1, a_2 such that $Fetters_{U_1} \cong Switch(a_1)$, $Fetters_{U_2} \cong Switch(a_2)$.
- The graph $\tilde{G} = (\tilde{V}, \tilde{E})$, constructed from G by contracting U_1, U_2 into vertices v_1, v_2, respectively, is as follows (compare Figure 5):
  - $\tilde{V} = \{v^{tot}\} \cup \cup_{i=1,2}\{v_i, v_i^{end}, v_i^{short}, v_i^{long}\}$;
  - $\tilde{E} = \hat{E} \cup E_1 \cup E_2$, where $\hat{E} = \{(v_2^{short}, v_1^{long}), (v_1^{short}, v_2^{long}), (v_1^{long}, v_2^{long}), (v_2^{long}, v_1), (v_1^{long}, v_2)\}$ and $E_i = \{(v_i^{short}, v_i^{long}), (v_i^{short}, v_i^{end}), (v_i^{end}, v_i^{long}), (v_i^{short}, v^{tot}), (v^{tot}, v_i^{long}), (v_i^{short}, v_i), (v_i, v_i^{long}), (v_i, v^{tot}), (v^{tot}, v_i)\}$ for $i = 1, 2$.
- The lengths for the remaining intervals are
  - $L(v^{tot}) = r_1 + r_2 + d - 1$;
  - $L(v_i^{short}) = r_i$ for $i = 1, 2$;
  - $L(v_i^{long}) = r_i + d$ for $i = 1, 2$;
  - $L(v_i^{end}) = 0$ for $i = 1, 2$.
- ϕ specifies v^{tot} to be *open*, and all the other intervals outside U_1, U_2 to be *closed.*

When there is no confusion, we shall use the vertex and the corresponding interval in the realization interchangeably. For example, $l(v_i^{short})$ is the position of the left endpoint of the interval corresponding to v_i^{short} in the realization, $|v_i^{short}|$ is its length, etc.

Call a realization of $Fetters$ *straight* if v_1 is to the left of v_2. Otherwise call the realization *reversed*. A realization of $Fetters$ is said to be *located* at the interval corresponding to v^{tot}. The $Fetters$ instance fixes the distance between its two *Switch*es. To formalize this notion we need the following definition.

DEFINITION 5.6. *Let $P = (G, L, \phi)$ be an MIG instance. Let $M, M' \subseteq V(G)$ be modules in G where $P_M \cong Switch(a)$ and $P_{M'} \cong Switch(a')$. For a realization of P, in which I and I' are the intervals corresponding to the middle vertices in M and M', respectively, define $Dist(M, M') = |l(I) - l(I')|$.*

LEMMA 5.7. *$\tilde{V} \backslash \{v_1, v_2\}$ is rigid in the $Fetters$. In particular, in every realization of the $Fetters$, $Dist(U_1, U_2) = d$.*

Proof. Recall that $\tilde{G}$ is the graph constructed from G by contracting U_1 and U_2 into v_1 and v_2, respectively. It is easy to see that $\tilde{G}$ is prime. Prime interval graphs have an interval order which is unique, up to complete reversal [34]. Hence, let us refer to the order in Figure 5, where w.l.o.g. $l(v_1^{end}) = 0$, and $l(v_2^{end}) > 0$. v^{tot} is between v_1^{end} and v_2^{end}, so $l(v_2^{end}) \geq r_1 + r_2 + d - 1$. Furthermore, according to the length constraints, $l(v_2^{end}) \leq r_2 + l(v_2^{short})$ and $r(v_1^{long}) \leq r_1 + d$.

By Lemma 5.5, $r(v_1^{long}) - l(v_2^{short}) = |v_1^{long} \cap v_2^{short}| \geq 1$, so $l(v_2^{end}) \leq r_2 + r(v_1^{long}) - 1 \leq r_1 + r_2 + d - 1$, yielding $l(v_2^{end}) = r_1 + r_2 + d - 1$, $v_1^{long} = [0, r_1 + d]$, $v_2^{short} = [r_1 + d - 1, r_1 + r_2 + d - 1]$, and $v_2 = [r_1 + d - 1, r_1 + d]$.

In a similar way we prove $v_1^{short} = [0, r_1]$, $v_2^{long} = [r_1 - 1, r_1 + r_2 + d - 1]$, and the result follows. ▯

By Lemma 5.7, for a realization of the $Fetters(d, r_1, r_2, U_1, U_2)$ which is straight (or reversed) and located at $(x, x + r_1 + r_2 + d - 1)$, the only degrees of freedom are reversals of the *Switch*es.

5.1.3. The frame. We now construct an element which divides an interval into subintervals of prescribed lengths. Each subinterval is characterized by a distinct set of intervals which contain it. This element will be used as a frame, into which the moving and toggling elements will fit, and will have the desired degrees of freedom.

Let $k = 2r - 1 \geq 3$, and let $x_1, x_2, \ldots, x_k$ be a sequence of real positive numbers whose sum is s. Define $Frame(x_1, x_2, \ldots, x_k)$ to be an instance $P = (G(V, E), L, \phi)$ (see Figure 6) as follows:

- $V = V^\alpha \cup V^\beta \cup V^\gamma$ consists of $3r + 3$ vertices, where $V^\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_k\}$, $V^\beta = \{\beta_1, \beta_3, \ldots, \beta_k\}$, and $V^\gamma = \{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}$.
- The edges in G are
  - $\gamma_2 v$ for each $v \in V$;
  - $\gamma_4 v$ for each $v \in V^\alpha \cup V^\beta$;
  - $\gamma_1\beta_1$ and $\gamma_3\beta_k$;
  - $\alpha_i\beta_j$ for each $1 \leq i, j \leq k$ such that j is odd and $|i - j| \leq 1$;
  - $\beta_i\beta_j$ for each $1 < i \neq j < k$ such that i, j are odd and $|i - j| \leq 2$.
- The lengths are
  - $L(\alpha_i) = x_i$ for each $1 \leq i \leq k$;
  - $L(\beta_i) = x_{i-1}+x_i+x_{i+1}$ for each odd i, $3 \leq i \leq k-2$, and $L(\beta_1) = x_1+x_2$, $L(\beta_k) = x_{k-1} + x_k$;
  - $L(\gamma_1) = L(\gamma_3) = 0$ and $L(\gamma_2) = L(\gamma_4) = s$.
- ϕ specifies α_i to be *open* if i is odd and γ_4 to be *open* but all other intervals to be *closed*.

A realization of a $Frame$ is said to be *straight* if γ_1 is to the left of γ_3. Otherwise it is called *reversed*. Such a realization is *located* at the interval corresponding to γ_4.

LEMMA 5.8. *$V \setminus \{\beta_1, \beta_k\}$ is rigid in a Frame.*

Proof. Let G' be the subgraph of G induced on $V^\alpha \cup V^\beta$. It is easy to see that G' is prime and, hence, has a unique clique order [34]. Moreover, G' has exactly k maximal cliques, each one containing (among other vertices) a unique and distinct α_i. The set of maximal cliques in G is $\{N[\alpha_i]\}_{i=1}^k$; namely, each clique is distinguished by a single α_i. Since G' is UCO, its unique clique order determines a unique linear order on V^α and, hence, also on the maximal cliques of G. Hence, G is UCO.

Let S and S' be two realizations of the same $Frame$. Suppose $\gamma_1 = \gamma'_1$ are their leftmost endpoints, respectively. The $Frame$ graph is UCO; hence, the order of the α-intervals is identical in both S and S'. Moreover, $\sum_{i=1}^k L(\alpha_i) = s$, and all the α-intervals are disjoint and must be between γ'_1 and γ'_3, which are at distance exactly s. Thus, the position of all α endpoints is uniquely determined. It is easy to see that also all β-intervals except β_1, β_k must have identical position in both realizations. □

By Lemma 5.8, for any straight (or reversed) realization of a $Frame(x_1, \ldots, x_k)$ located at $(x, x+s)$, the positions of all intervals except β_1, β_k are uniquely determined.

In the sequel, when we use a realization of such a $Frame$ to implicitly define an MIG instance, we shall assume that β_1 and β_k are contained in $[x, x + s]$, so the realization has the shortest possible length. In addition, when we use any gadget in the implicit definition, and we describe its intervals by saying that "the gadget is located at ...," we mean that "a straight realization of the gadget is located at ...."

5.2. The reduction. The realization of an MIG instance is a polynomial witness for a "yes" instance; hence, MIG is in NP. We describe a reduction from 3-COLORING, which is NP-complete (see, e.g., [15]). Let $G = (V, E)$ be an instance of 3-COLORING. We construct an instance $\mathbb{P} = (T, L)$ of MIG (in implicit form) and prove that $\mathbb{P}$ is a "yes" instance if and only if G is 3-colorable.

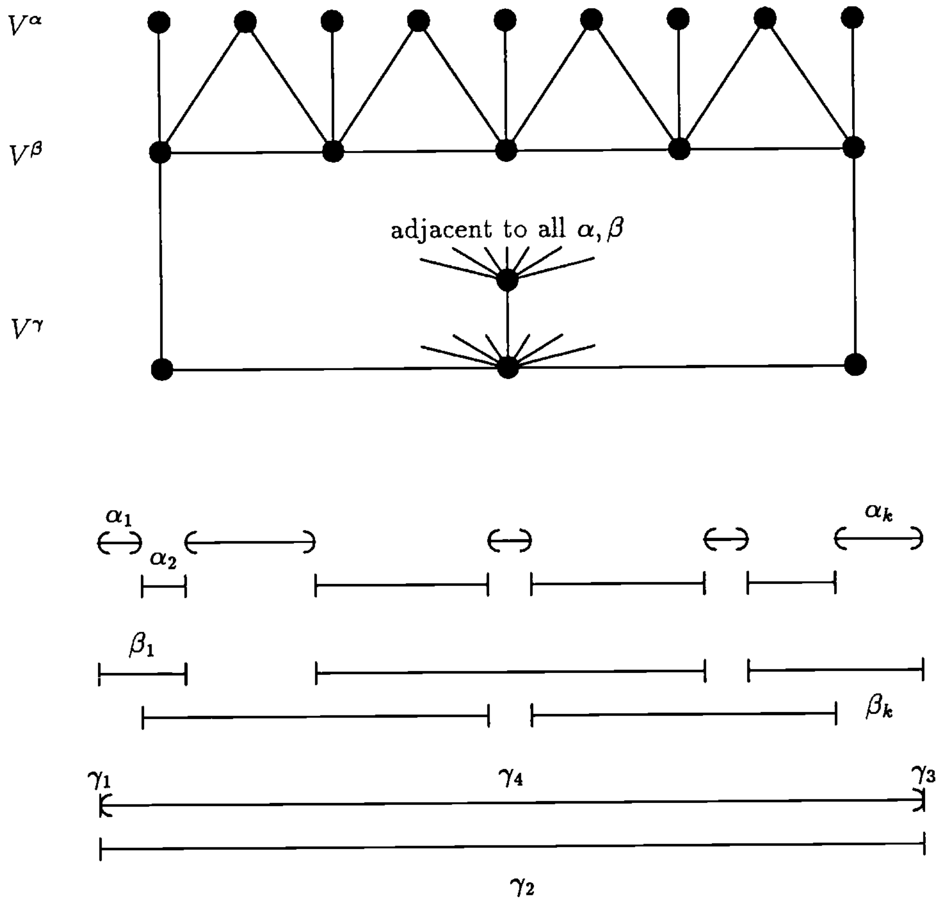


FIG. 6. *A graph of a Frame (top) and its realization (bottom). The Frame structure is rigid and divides an interval into smaller intervals of prescribed lengths and positions.*

The general plan is as follows: we construct measured interval subinstances for each vertex and for each edge of G. The subinstance of a vertex is designed so that it can be realized only in three possible ways, which will correspond to its color. The subinstance for each edge will prevent the vertices at its endpoints from having the same color.

5.2.1. The vertex subinstance. Let $n = |V|$, $m = |E|$, $M = 24m + 11n + 1$. Define the following set $S = \omega \cup \delta \cup \zeta$ of intervals (compare Figures 7 and 8):

- $\omega = \alpha \cup \beta \cup \gamma$ is a $Frame(1, 4, 1, 4, 1)$, located at $(0, 11)$.
- $\delta = \delta_1 \cup \delta_2$, where $\delta_j = \{\delta_j^i\}_{i=1}^5$, and each of δ_1 and δ_2 is a $Switch(3)$, located at $(1, 2)$ and $(7, 8)$, respectively. The superscripts match the vertex numbers in each $Switch$.
- $\zeta = \{\zeta^{tot}, \zeta_1^{end}, \zeta_1^{short}, \zeta_1^{long}, \zeta_2^{end}, \zeta_2^{short}, \zeta_2^{long}\}$ such that $\zeta \cup \delta_1 \cup \delta_2$ is a $Fetters(6, 2M + 2, 2M - 7, \delta_1, \delta_2)$ located at $(-2M, 2M)$.

Note that the intersection graph of $\omega \cup \zeta$ is prime.

For each interval $I \in S$ let $L(I) = |I|$. The subinstance of each vertex is isomorphic to (S, L), and the $Frame$s of the n vertices are laid out contiguously as follows: for an interval J and a real number x, denote $J + x = \{y + x | y \in J\}$.

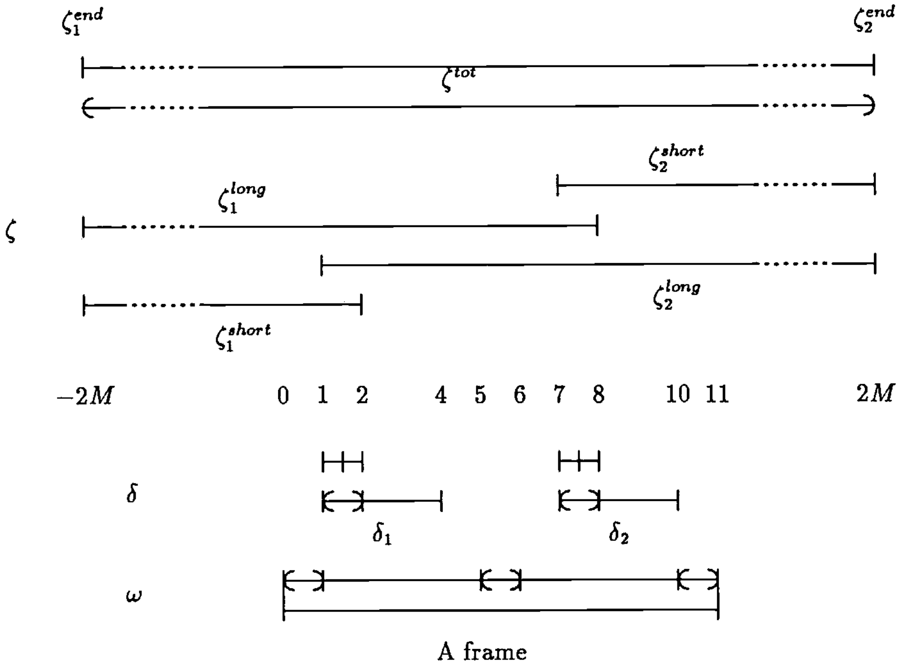


FIG. 7. *This is the set of intervals S. δ can be positioned in the frame. The distance between δ_1 and δ_2 is enforced by the ζ.*

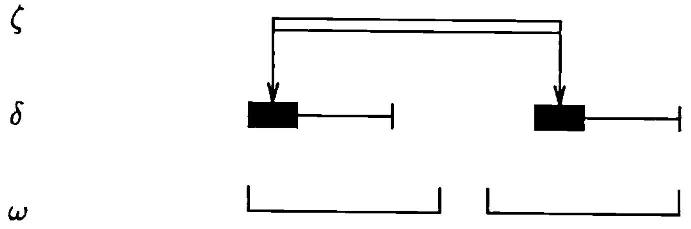


FIG. 8. *This is a sketch of the structure of a vertex. The Switches δ can be positioned in the frame ω. The distance between δ_1 and δ_2 is enforced by the ζ.*

Let $V = \{0, \ldots, n-1\}$. For each $i \in V$, $J \in S$ denote $J(i) = J + 11i$. Denote $S(i) = \{J(i)\}_{J \in S}$ and $\mathbb{S} = \cup_{i \in V} S(i)$. Let $\mathbb{P}^V$ be the measured interval graph defined implicitly by $\mathbb{S}$.

A realization of a vertex subinstance is called *straight* (respectively, reversed) if the realization of its ω is straight (respectively, reversed).

LEMMA 5.9. *Let $\cup_{i \in V} S(i)'$ be a realization of $\mathbb{P}^V$ with $S(i)' = \{J(i)'\}_{J(i) \in S(i)}$. Then either every $S(i)'$ is straight or every $S(i)'$ is reversed.*

Proof. It suffices to prove that $l(\gamma_1(i)') < l(\gamma_3(i)')$ if and only if $l(\gamma_1(i+1)') < l(\gamma_3(i+1)')$. This follows from the identity of the zero-length intersecting intervals $\gamma_1(i+1)'$ and $\gamma_3(i)'$, and the disjointness of $\gamma_1(i)'$ and $\gamma_3(i+1)'$, which are both, by Lemma 5.8, at distance 11 from the former pair, respectively. □

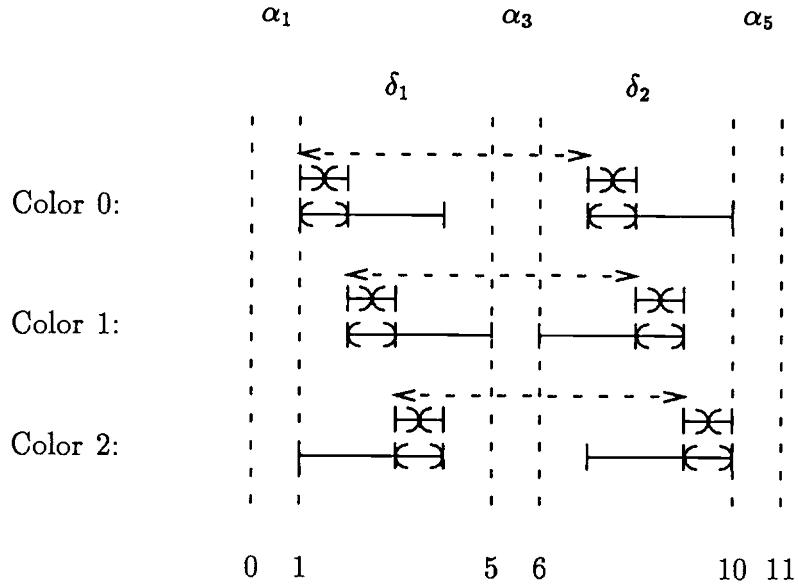


FIG. 9. *The three possible positions of δ_1,δ_2 in a vertex subinstance, which will correspond to the three possible colors of the vertex.*

Let $\mathbb{S}'$ be a straight realization of $\mathbb{P}^V$. Define the function $Col : V \longrightarrow R$, as follows:

$$Col(i) = l(\delta_1^3(i)') - l(\gamma_1(i)') - 1. \tag{2}$$

Call Col the *coloring* defined by $\mathbb{S}'$. We now show that each vertex subgraph can be realized in exactly three distinct colors. This is also demonstrated in Figure 9.

LEMMA 5.10. *For each $i \in V$: $Col(i) \in \{0, 1, 2\}$.*

Proof. According to Lemma 5.8, in each straight $S(i)'$ the positions of the intervals in $\alpha(i)'$ relative to $l(\gamma_1(i)')$ are fixed. Assume w.l.o.g. $l(\gamma_1(i)') = 0$. For each $J \in \delta_1(i)'$, $J \subseteq [1, 5]$, and for each $J \in \delta_2(i)'$, then $J \subseteq [6, 10]$. By Lemma 5.4, $Length(\delta_1(i)) = Length(\delta_2(i)) = 3$. Hence (compare Figure 9) the following hold:

$$\text{if } \delta_1(i)' \text{ is straight,} \quad \text{then } 1 \le l(\delta_1^3(i)') \le 2; \tag{3}$$
$$\text{if } \delta_1(i)' \text{ is reversed,} \quad \text{then } 3 \le l(\delta_1^3(i)') \le 4; \tag{4}$$
$$\text{if } \delta_2(i)' \text{ is straight,} \quad \text{then } 6 \le l(\delta_2^3(i)') \le 7; \tag{5}$$
$$\text{if } \delta_2(i)' \text{ is reversed,} \quad \text{then } 8 \le l(\delta_2^3(i)') \le 9. \tag{6}$$

But according to Lemma 5.7,

$$l(\delta_1^3(i)') - l(\delta_2^3(i)') = 6. \tag{7}$$

Therefore,

$$l(\delta_1^3(i)') \in \{1, 2, 3\} \tag{8}$$

and $Col(i) \in \{0, 1, 2\}$. □

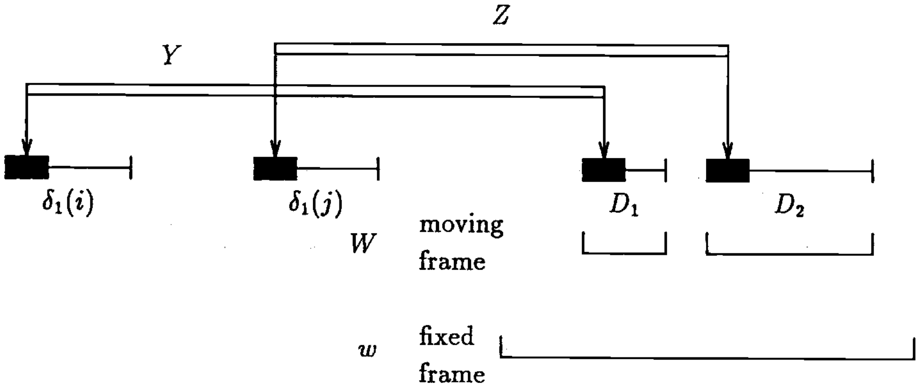


FIG. 10. *This is a functional sketch of the edge subinstance. The two Switches D can only reverse in their relative fixed positions inside the moving frame. Their distances from the corresponding Switches in the vertices i, j, respectively, are fixed. The moving frame itself can have different positions along the fixed frame.*

5.2.2. The edge subinstance. Let the edges of G be $E = \{e_0, \dots, e_{m-1}\}$, and let $e_k = (i, j)$ be an edge in E, where $i < j$. For each edge e_k we construct an edge subinstance that forces the colors of the vertices i and j to be different.

We first give an overview of this construction (compare Figure 10): each edge is assigned a fixed *Frame* w which contains two *Switch*es, D_1 and D_2, which are the heart of its subinstance. The *Frame*s of the edges are laid out contiguously to the right of the vertex *Frame*s. The subinstance is a collection of intervals $\{A_0\} \cup W \cup D \cup Y \cup Z \cup w$ designed so that the following hold:

1. D_1 and $\delta_1(i)$ are kept at a fixed distance (this is done by the Y intervals).
2. D_2 and $\delta_1(j)$ are kept at a fixed distance (this is done by Z).
3. D_1 and D_2 are restricted to be in one of four possible relative positions, allowing the four possible color differences between the vertices i and j (this is done by W).
4. D_1 and D_2 together can undergo a translation, allowing the six possible color combinations of the vertices i and j, as demonstrated in Figure 12 (this is done by A_0 and w).

We now describe the construction in detail (compare Figure 11): define the following set $X(k) = \{A_0(k)\} \cup W(k) \cup D(k) \cup Y(k) \cup Z(k) \cup w(k)$ of intervals. Let $Base(k) = 18k + 11n$. For readability, we omit the parameter k whenever possible.

- $w = a \cup b \cup c$ is a $Frame(5, 12, 1)$, located at $(0, 18) + Base(k)$.
- $A_0 = [3, 7] + Base(k)$.
- $W = A \cup B \cup C$ is a $Frame(1, 2, 1, 4, 1)$, located at $(7, 16) + Base(k)$.
- $D = D_1 \cup D_2$, where $D_j = \{D_j^i\}_{i=1}^5$, and
  - D_1 is a $Switch(2)$ located at $(8, 9) + Base(k)$,
  - D_2 is a $Switch(4)$ located at $(11, 12) + Base(k)$.
- Define $d_Y(k, i) = Base(k) + 8 - (11i + 2) = 11(n - i) + 18k + 6$.
  $Y = \{Y^{tot}, Y_1^{end}, Y_1^{short}, Y_1^{long}, Y_2^{end}, Y_2^{short}, Y_2^{long}\}$ such that $Y \cup D_1 \cup \delta_1(i)$ is the $Fetters(d_Y(k, i) + 1, 6k + 5 + 11i + 2, M - 6k + 4 - (11n + 18k + 8), \delta_1(i), D_1)$ located at $(-6k - 5, M - 6k + 4)$.
- Define $d_Z(k, j) = Base(k) + 11 - (11j + 1) = 11(n - j) + 18k + 10$.
  $Z = \{Z^{tot}, Z_1^{end}, Z_1^{short}, Z_1^{long}, Z_2^{end}, Z_2^{short}, Z_2^{long}\}$ such that $Z \cup D_2 \cup \delta_1(j)$ is the $Fetters(d_Z(k, j), 6k + 3 + 11j + 1 + 1, M - 6k - (11n + 18k + 11), \delta_1(j), D_2)$

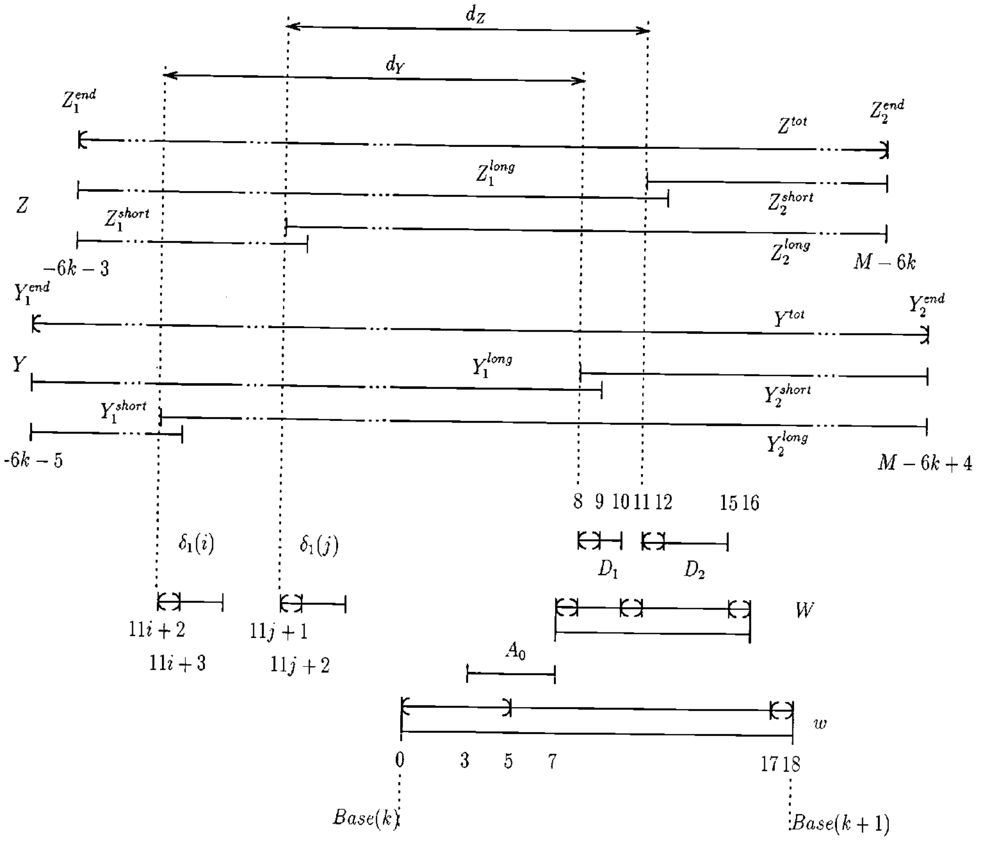


FIG. 11. *The edge subinstance: a moving frame can be positioned inside the fixed frame. The Switches D_1 and D_2 are positioned inside the moving frame. Each of D_1 and D_2 is connected to its vertex subinstance via Fetters.*

located at $(-6k-3, M-6k)$.

The length function on the subinstance (X, L) is defined so that $L(I) = |I|$ for all intervals except the Y's in which we set

$$L(Y_1^{short}) = |Y_1^{short}| + 1, \qquad L(Y_2^{long}) = |Y_2^{long}| - 1.$$

This change, together with the $+1$ in the first parameter of the *Fetters* of Y, forces a $+1$ shift on the location of $\delta_1(i)$. This shift will be crucial in forcing the vertices i and j to have different colors.

Note that the intersection graph of $X(k) - D(k)$ is prime. Note also that the left and right ends of the *Fetters* subinstances Y and Z are positioned way beyond the contiguous *Frame*s in all edge subinstances and vertex subinstances. This allows every *Fetters* to move independently, and no vertex or edge subgraph is a module.

Denote $\mathbb{X} = \cup_{e_i \in E} X(i)$. Let $\mathbb{X}' = \cup_{J \in \mathbb{X}} J'$ be a set of intervals with the same intersection graph as of $\mathbb{X}$, which satisfy the corrected length constraints, where $X(i)' = \{J(i)'\}_{J(i) \in X(i)}$. Call $X(i)'$ a *straight* (respectively, *reversed*) realization if the frame $w(i)'$ is straight (respectively, reversed). A proof similar to that of Lemma 5.9 implies Lemma 5.11.

LEMMA 5.11. *For each $1 \leq i < j \leq m$, $X(i)'$ is straight if and only if $X(j)'$ is* *straight.* □

The complete constructed instance is $\mathbb{P} = (\mathbb{S} \cup \mathbb{X}, L)$, where the interval lengths L are implicit in each of the two types of subgraphs, and the only exception is the corrected length in the $Y(k)$'s. Due to this exception, simple superimposition of $\mathbb{S}$ and $\mathbb{X}$ does not give a realization.

LEMMA 5.12. *If $\mathbb{S}' \cup \mathbb{X}'$ is a realization of $\mathbb{P}$ for which $\mathbb{S}'$ is straight, then $\mathbb{X}'$ is straight.*

Proof. Recall that $c_1(1)$ and $\gamma_3(n)$ are the leftmost and the rightmost zero length intervals in the leftmost edge subinstance and the rightmost vertex subinstance, respectively. Suppose, to the contrary, that $\mathbb{X}'$ is not straight. The zero-length intersecting intervals $c_1(1)'$ and $\gamma_3(n)'$ must be identical. Without loss of generality, $c_1(1)' = \gamma_3(n)' = [0, 0]$. Then $c_2(1)' = [-18, 0]$, $\gamma_1(n)' = [-11, -11]$, and these two intervals intersect, in contradiction to our constructed interval graph. □

LEMMA 5.13. *If $e_k = (i, j)$, then for every realization $\mathbb{S}' \cup \mathbb{X}'$ of $\mathbb{P}$, $Col(i) \neq Col(j)$.*

Proof. Assume w.l.o.g. that the realization is straight, and that $l(c_1(k)') = Base(k)$. Again, we omit the parameter k whenever possible. Surely

$$l({C_1}') \leq r({A_0}') < r({a_1}') + 4 = 9 + Base(k). \tag{9}$$

The first inequality follows since C_1' and A_0' must intersect, the second inequality follows since A_0' and a_1' should intersect, and the last equality follows since w' is rigid (Lemma 5.8). Since $Length(W) = |W'| = 9$, we conclude that $W' = A' \cup B' \cup C'$ is straight.

According to Lemma 5.8 the relative positions of all the intervals in the $Frame$ (except B_1' and B_5') are fixed relative to $l(C_1')$. The realization for each of D_1 and D_2 can be either straight or reversed, giving rise to four possible combinations of positions. (Any of these combinations fixes the positions of D' with respect to $l(C_1')$.) In particular,

$$l({D_2^3}') - l({D_1^3}') \in \{2, 3, 5, 6\}. \tag{10}$$

Due to Lemma 5.7 and the realization being straight,

$$l(\delta_1^3(j)') - l(\delta_1^3(i)') \tag{11}$$

$$= (l({D_2^3}') - d_Z(k, j)) - (l({D_1^3}') - d_Y(k, i)) \tag{12}$$

$$= (l({D_2^3}') - (11(n - j) + 18k + 10)) - (l({D_1^3}') - (11(n - i) + 18k + 6)) \tag{13}$$

$$= l({D_1^3}') - l({D_1^3}') - 4 + 11(j - i) \in \{11(j - i) \pm 1, 11(j - i) \pm 2\}. \tag{14}$$

Therefore,

$$Col(j) - Col(i) \tag{15}$$

$$= (l(\delta_1^3(j)') - l(\gamma_1(j))' - 1) - (l(\delta_1^3(i)') - l(\gamma_1(i))' - 1) \tag{16}$$

$$= l(\delta_1^3(j)') - l(\delta_1^3(i)') - 11j + 11i \in \{\pm 1, \pm 2\}. \tag{17}$$

□

COROLLARY 5.14. *If $\mathbb{P}$ is a "yes" instance, then G is 3-colorable.*

Proof. If $\mathbb{P}$ is a measured interval graph, then it has a straight realization (since the realization can be reversed completely). Define the coloring $Col : V \longrightarrow \{0, 1, 2\}$ as described in (2). By Lemma 5.13, Col is a proper 3-coloring of G. □

Let us now prove the converse.

LEMMA 5.15. *If G is 3-colorable, then $\mathbb{P}$ admits a realization.*

Proof. Let $Col : V \to \{0, 1, 2\}$ be a proper 3-coloring of G. We build a realization $\mathbb{S}' \cup \mathbb{X}'$ for the instance $\mathbb{P}$ as follows:

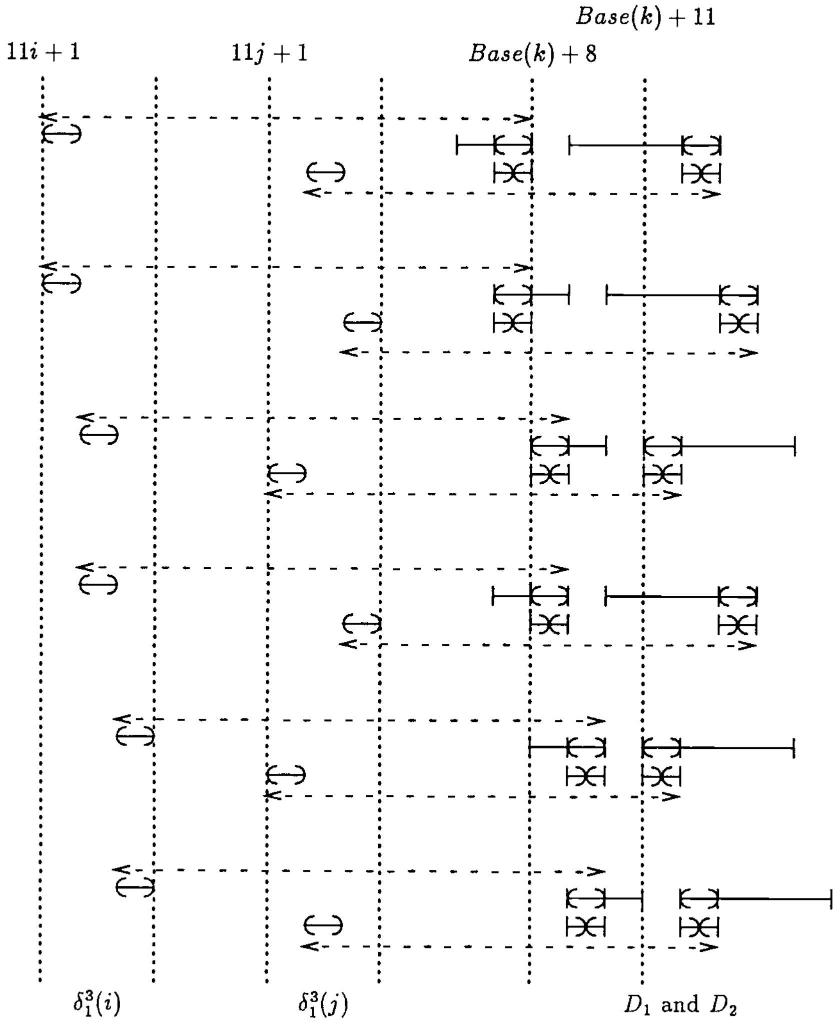


FIG. 12. *The relative position of D_1^3 and D_2^3 forces the colors of the vertices to be different.*

1. For the vertex $i \in V$ with $Col(i) = x$, position its *Switch*es δ_1', δ_2' as follows (compare Figure 9):
   - if $x = 0$ then $\delta_1' = \delta_1$, $\delta_2' = \delta_2$;
   - if $x = 1$ then $\delta_1' = \delta_1 + 1$, $\delta_2' = -\delta_2 - 1$;
   - if $x = 2$ then $\delta_1' = -\delta_1$, $\delta_2' = -\delta_2$.

   The rest of the intervals in the vertex subgraph are positioned accordingly (cf. Lemma 5.10).
2. For the edge $e_k = (i, j)$ with $y = Col(j) - Col(i)$, the directions of the *Switch*es $D_1(k)'$ and $D_2(k)'$ in the realization are determined by y, thus fixing the distance between $D_1^3(k)'$ and $D_1^3(k)'$. The absolute position of these *Switch*es is determined according to $Col(i)$ and $Col(j)$ as follows in Table 1 (compare Figure 12):

$\mathbb{S}' \cup \mathbb{X}'$ and $\mathbb{S} \cup \mathbb{X}$ have the same intersection graph, all interval lengths match the prescribed lengths, and their endpoints meet the specification. □

TABLE 1

$Col(i)$	$Col(j)$	$D_1(k)'$	$D_2(k)'$
0	1	$-D_1(k)'-2$	$-D_2(k)'-2$
0	2	$D_1(k)'-1$	$-D_2(k)'-1$
1	0	$D_1(k)'$	$D_2(k)'$
1	2	$-D_1(k)'-1$	$-D_2(k)'-1$
2	0	$-D_1(k)'$	$D_2(k)'$
2	1	$D_1(k)'+1$	$D_2(k)'+1$

From Lemma 5.15 and Corollary 5.14 we can finally conclude Theorem 5.16.

THEOREM 5.16. *MIG is NP-complete.* □

In fact, the same reduction implies strong NP-completeness, because 3-coloring is strongly NP-complete, and the reduction is also pseudopolynomial.

5.3. Closing the open intervals. We have proved that recognizing a measured interval graph with specified endpoints is NP-complete. We now show that this problem is hard even where all the intervals are closed. Given an instance $P = (G, L, \phi)$ of MIG, define a new instance $P' = (G, L')$ of MIG^* (in which all intervals are closed), as follows: let $n = |V(G)|$, and fix $\epsilon < \frac{1}{20n^2}$.

$$L'(v) = \begin{cases} L(v) & \text{if } v \text{ is closed,} \\ L(v) - 2\epsilon & \text{if } v \text{ is open.} \end{cases} \tag{18}$$

Let $\mathbb{P}$ be an instance generated by the reduction in section 5.2. We shall prove that $\mathbb{P}$ has a realization if and only if $\mathbb{P}'$ has one.

First, we observe that the construction introduced in the proof of Theorem 5.16 has a special property: let S be a realization in which the shortest nonzero length of an interval is C. S is called *discrete* if all the endpoints of its intervals are integer multiples of C. In that case, C is called the *grid size* of S.

REMARK 5.17. *By the proofs of Lemma* 5.15 *and Corollary* 5.14, $\mathbb{P}$ *has a realization if and only if it has a discrete realization, with grid size* $\frac{1}{2}$.

LEMMA 5.18. *If* $\mathbb{P}$ *has a realization, then* $\mathbb{P}'$ *has one.*

Proof. If $\mathbb{P}$ has a realization, then by Remark 5.17 it has a discrete realization $\{I_v\}_{v \in V(G)}$ with grid size $\frac{1}{2}$. Construct the set of closed intervals $\{I'_v\}_{v \in V(G)}$ defined as follows: if I_v is closed, let $I'_v = I_v$. If I_v is open, let $I'_v = [l(I_v) + \epsilon, r(I_v) - \epsilon]$.

We assume that this set is a realization of $\mathbb{P}'$: since $\mathbb{P}$ is discrete, the intervals I_v and I_u intersect if and only if I'_v and I'_u intersect, since if one (or both) of I_v, I_u is open, then their overlap is at least $\frac{1}{2}$. Furthermore, clearly $|I'_v| = L'(v)$ for every $v \in V$. □

Unfortunately, the converse of Lemma 5.18 does not always hold for arbitrary MIG instances, as demonstrated in Figure 13. We shall prove that the converse does hold for instances generated by the reduction in section 5.2.

Define the following order-oriented analogue of MIG and MIG^*, respectively.

RECOGNIZING A MEASURED INTERVAL ORDER WITH SPECIFIED ENDPOINTS (MIO):

INSTANCE: A partial order $\prec$ on a set V, a nonnegative *length* $L(v)$ for every $v \in V$, and a function $\phi : V \to \{open, closed\}$.

QUESTION: Is there an interval realization of $(V, \prec)$ in which the length of I_v is exactly $L(v)$, and I_v is open if and only if $\phi(v) = open$?

MIO^* is the restriction of MIO to instances with all intervals closed. MIO^* can be solved in polynomial time [19, 24, 32], and that solution can be generalized to deal with open intervals and solve MIO as well.

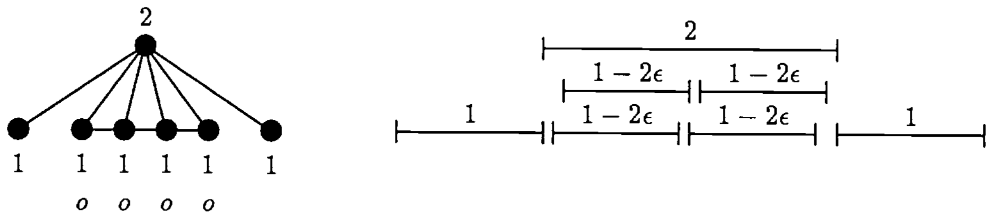


FIG. 13. *In the MIG instance P on the left, the numbers denote lengths, and the four intervals corresponding to the vertices marked "o" should be open. P has no realization, but P' has one, as shown on the right.*

We need to generalize the notion of rigidness in the following manner.

DEFINITION 5.19. *For a real $p \geq 0$, two realizations $\{I_j\}$ and $\{I'_j\}$ of the same interval graph are* p-isometric *if there exists a function $f(x) = s \cdot x + c$ where $s = \pm 1, c \in \mathbb{R}$, and constants c_j, $|c_j| < p$ such that $f(I_j) + c_j = I'_j$ for all j. We call $U \subseteq V(G)$* p-rigid *in an MIO instance if in any two realizations of the instance, the sets of intervals realizing U are p-isometric. Note that in this case all endpoints of U are located at fixed distances from the leftmost endpoint, up to $\pm p$. Hence, every realization has the same length, up to $\pm p$.*

For an instance Q of MIO, define the following system of inequalities $S(Q)$ on the variables $\{l_v\}_{v \in V}$:

- If $x \prec y$, and both x, y are closed, $l_x + L(x) < l_y$.
- If $x \prec y$, and at least one of x, y is open, $l_x + L(x) \leq l_y$.
- If $x \cap y \neq \phi$, and both x, y are closed, $l_x + L(x) \geq l_y$.
- If $x \cap y \neq \phi$, and at least one of x, y is open, $l_x + L(x) > l_y$.

Q has a realization if and only if $S(Q)$ has a feasible solution, since the left endpoints of the realization satisfy $S(Q)$, and vice versa. Recall that $D(S(Q))$ is the distance graph of $S(Q)$, as in the proof of Lemma 3.2, and denote it $D(Q)$ for short.

LEMMA 5.20. *Let Q be an instance of MIO. If U is a strongly connected component in the union of all zero-weight cycles in $D(Q)$, then U is rigid in Q.*

Proof. For vertices $x, y \in U$, there is a zero-weight cycle c in $D(Q)$ passing through both x and y. Let d (resp., $-d$) be the weight of the path from x to y (resp., y to x) along c. Summing the inequalities in $S(Q)$ along the two paths we get $l_y \leq l_x + d$ and $l_x \leq l_y - d$, respectively, implying $l_y - l_x = d$. Since every realization must satisfy $S(Q)$ for every realization $l_y - l_x = d$, then U is rigid. □

The converse holds as well.

LEMMA 5.21. *Let Q be a realizable instance of MIO, with U rigid in Q. Then for each $x, y \in U$ there is a zero-weight cycle in $D(Q)$ containing both x and y.*

Proof. Suppose to the contrary $x, y \in U$ and there is no zero-weight cycle in $D(Q)$ containing both x and y. Either there is a cycle in $D(Q)$ through x and y or there is no such cycle.

If there is no cycle in $D(Q)$ through x and y, then w.l.o.g. there is no path in $D(Q)$ from x to y. Let $W \subseteq V \setminus \{x\}$ be the set of all vertices in V to which there is a path from y in $D(Q)$ (including y itself). Then Q does not contain any inequalities $v < w + C$ or $v \leq w + C$ for $w \in W$, $v \in V \setminus W$. Let $\{I_v\}_{v \in V}$ be a realization for Q. Then $\{I_w - 1\}_{w \in W} \cup \{I_v\}_{v \in V \setminus W}$ also realizes Q, with a different distance between the intervals corresponding to x and to y, contradicting the rigidness of U in Q.

If there exists a cycle in $D(Q)$ through x and y, then let c be such a cycle of minimum weight, and let $l = w(c)$. By assumption $l \neq 0$, and since Q has a realization, by Corollary 3.3 $l > 0$. Let d be the weight of the path from x to y along c. For every Δ, $d - l < \Delta < d$, consider a new directed graph $D'(\Delta)$ obtained from $D(Q)$ by adding two arcs xy and yx, both labeled $\leq$, with weights Δ and $-\Delta$, respectively. Observe that adding the two arcs does not introduce into the graph any cycles of negative-weight or zero-weight cycles with strict arcs. Hence, the augmented system corresponding to $D'(\Delta)$ has a realization. Moreover, in every realization of $D'(\Delta)$, the distance between the left endpoints of x and y is Δ. By choosing different values of Δ, we contradict the rigidness of U. □

We can now generalize Lemma 5.20.

Lemma 5.22. *Let Q be a realizable instance of MIO. For a nonnegative p, let C be a union of cycles in $D(Q)$, each of weight less than p. If U is a strongly connected component in C, then U is $|U|p$-rigid in Q.*

Proof. For vertices $x, y \in U$, by the definition of U, there is a simple path $P_0 : x = x_1, x_2, \ldots, x_k = y$ in C. Every edge $x_i x_{i+1}$ is in C; therefore, there exists a path P_i in C from x_{i+1} to x_i s.t. $x_i x_{i+1}$ and P_i form a cycle in C. The concatenation of $P_{k-1}, P_{k-2}, \ldots, P_1$ is a path P from y to x in C. Moreover, the concatenation of P_0 and P is a cycle c in C (not necessarily simple) of weight at most $(k-1)p$. Let $d = w(P_0)$, $d' = w(P)$, and $q = |U|p$. Since $k \leq |U|$, we have established that $w(c) = w(P_0) + w(P) = d + d' \leq q$. Since Q is realizable, $d + d' \geq 0$. Hence, $-d \leq d' \leq q - d$. Summing the inequalities in $S(Q)$ along the two paths P_0 and P we get $l_y \leq l_x + d$ and $l_x \leq l_y + d'$, respectively. Any realization $\{l_x\}_{x \in V}$ of Q satisfies $S(Q)$, so $-d \leq l_x - l_y \leq d' \leq -d + q$. Hence, any two realizations are q-isometric. □

Lemma 5.23. *Let $Q = (\prec, L, \phi)$ be an MIO instance, and let $Q' = (\prec, L')$ be the corresponding MIO* instance (obtained by the transformation in (18)). Suppose $U \subseteq V(\prec)$ is rigid in Q, and let $n = |V(\prec)|$. If $n^2\epsilon < \frac{1}{2}$ then U is $2n^2\epsilon$-rigid in Q'.*

Proof. The weight of each arc in $D(Q')$ changes by no more than 2ϵ, compared with $D(Q)$. Hence, the weight of every simple cycle changes by at most $2n\epsilon$. U is rigid in Q; hence, by Lemma 5.21 it is contained in a strongly connected component W of a union of zero-weight cycles in $D(Q)$. W is also a union of simple zero-weight cycles in $D(Q)$. The weight of each such cycle in $D(Q')$ is at most $2n\epsilon$. Hence, by Lemma 5.22, W is $2|W|n\epsilon$-rigid in Q', and so is U. □

We now return to the instance $\mathbb{P}$ generated by the reduction in the proof of Theorem 5.16. Recall that $\mathbb{P}'$ is the instance obtained from $\mathbb{P}$ by the transformation (18). Suppose $\mathbb{P}'$ has a realization. Let $\prec$ be the corresponding interval order, and let $\mathbb{Q} = (\prec, L, \phi)$ and $\mathbb{Q}' = (\prec, L')$. Consider each of our gadgets: By Lemma 5.4, every *Switch* is rigid in $\mathbb{Q}$. A slight modification of Lemma 5.7 shows that every *Fetters* must be rigid in $\mathbb{Q}$ (since the directions of the *Switch*es are set). By Lemma 5.8 every *Frame* is rigid in $\mathbb{Q}$, with the exception of its end β-intervals. Hence, each of these gadgets is $2n^2\epsilon$-rigid in $\mathbb{Q}'$ by Lemma 5.23. This imposes, up to small additive shifts, the relative positions of the intervals in each vertex (or edge) subinstance. Define the function *Col* as in (2). We shall show that the choice of ϵ makes these shifts sufficiently small so that the properties of the coloring are preserved.

Lemma 5.24. *For each $i \in V$ there exist C_i, $|C_i| < 4n^2\epsilon$ s.t. $Col(i) + C_i \in \{0, 1, 2\}$.*

Proof. The proof is analogous to Lemma 5.10: relations (3)–(7) hold up to $\pm 2n^2\epsilon$. Hence, (8) holds up to $\pm 4n^2\epsilon$. □

Lemma 5.25. *For every edge (i, j), $|Col(i) - Col(j)| \geq 1 - 8n^2\epsilon$.*

Proof. The proof is analogous to Lemma 5.13: relations (9)–(14) hold up to $\pm 2n^2\epsilon$. Relations (15)–(17) hold up to $\pm 8n^2\epsilon$, because they involve up to four differences of endpoint distances. □

Let $round(x)$ be the integer closest to x. Recall that $\epsilon < \frac{1}{20n^2}$. By Lemma 5.24 $|Col(i) - round(Col(i))| < 0.2$. By Lemma 5.25, for every edge (i,j), $|Col(i) - Col(j)| \geq 0.6$. Hence, $round(Col(i)) \neq round(Col(j))$. This proves that if there exists a realization to $\mathbb{P}'$, by rounding the colors to the nearest integer we obtain a proper 3-coloring. By Lemma 5.15 this implies the existence of a realization to $\mathbb{P}$. Thus, $\mathbb{P}$ has a realization if and only if $\mathbb{P}'$ has one. Since the transformation described in (18) is polynomial, we conclude the following theorem.

THEOREM 5.26. *MIG^* is NP-complete.* □

5.4. Related problems. In section 1 we introduced the recognition problem of interval graph with individual lower and upper bounds on interval lengths (the *BIG* problem). Since MIG^* is a restriction of *BIG* and *DCIG*, Corollary 5.27 holds.

COROLLARY 5.27. *BIG and DCIG are NP-complete.*

When restricted to interval graphs with depth 0 decomposition trees (see [23] for a definition of the decomposition tree), i.e., to prime interval graphs, the *MIG* problem can be solved in polynomial time, using the algorithm devised in section 3 for UCO graphs. This depth bound is indeed tight; namely, when allowing deeper decomposition trees the problem is NP-complete.

PROPOSITION 5.28. *MIG is NP-complete even when restricted to interval graphs with decomposition tree of depth* 1.

Proof. We shall see that besides the *Switch*es $\delta_i(j)$ and $D_i(k)$, and the K_2 modules $\{c_3(k), c_1(k+1)\}$, $\{\gamma_3(i), \gamma_1(i+1)\}$, and $\{\gamma_3(n-1), c_1(0)\}$ there are no nontrivial modules in the interval graph constructed by the reduction in the proof of Theorem 5.26: let H be the graph obtained by contraction of the above modules. Suppose to the contrary that H contains a nontrivial module M, and suppose $v, u \in M$. If v, u are in the same vertex subgraph (or in the same edge subgraph) H_U, then $M \cap U$ is a nontrivial module in H_U, contradicting the primality of the vertex subgraph (and the edge subgraph). Hence, u, v are in different vertex/edge subgraphs. In this case, there are intervals in these subgraphs, which intersect only one out of u, v, in contradiction to M being a module. □

Acknowledgments. We thank Phil Green for illuminating conversations on the physical mapping problems which motivated this work and thank Garth Isaak for helpful discussions. We further thank the referees for their useful remarks.

REFERENCES

[1] R. K. Ahuja, T. A. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[2] J. F. Allen, *Maintaining knowledge about temporal intervals*, Comm. ACM, 26 (1983), pp. 832–843.

[3] J. F. Allen, *Reasoning about Plans*, Morgan Kaufman, San Mateo, CA, 1991.

[4] P. Baldy and M. Morvan, *A linear time and space algorithm to recognize interval orders*, Discrete Appl. Math., 46 (1993), pp. 173–178.

[5] R. Bellman, *On a routing problem*, Quart. Appl. Math., 16 (1958), pp. 87–90.

[6] S. Benzer, *On the topology of the genetic fine structure*, in Proc. Nat. Acad. Sci. USA, 45 (1959), pp. 1607–1620.

[7] K. S. Booth and G. S. Lueker, *Testing for the consecutive ones property, interval graphs, and planarity using PQ-tree algorithms*, J. Comput. System Sci., 13 (1976), pp. 335–379.

[8] A. V. Carrano, *Establishing the order of human chromosome-specific DNA fragments*, in Biotechnology and the Human Genome, A. D. Woodhead and B. J. Barnhart, eds., Plenum Press, New York, 1988, pp. 37–50.

[9] C. H. Coombs and J. E. K. Smith, *On the detection of structures in attitudes and developmental processes*, Psych. Rev., 80 (1973), pp. 337–351.

[10] X. Deng, P. Hell, and J. Huang, *Linear Time Representation Algorithms for Proper Circular Arc Graphs and Proper Interval Graphs*, Technical report, School of Computing Science, Simon Fraser University, B.C., Canada, 1993.

[11] P. Fishburn, *Interval Orders and Interval Graphs*, Wiley, New York, 1985.

[12] P. Fishburn and R. L. Graham, *Classes of interval graphs under expanding length restrictions*, J. Graph Theory, 9 (1985), pp. 459–472.

[13] H. Gabow and R. E. Tarjan, *Faster scaling algorithms for network problems*, SIAM J. Comput., 18 (1989), pp. 1013–1036.

[14] P. Gács and L. Lovász, *Khachiyan's algorithm for linear programming*, Math. Prog. Study, 14 (1981), pp. 61–68.

[15] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Co., San Francisco, CA, 1979.

[16] P. C. Gilmore and A. J. Hoffman, *A characterization of comparability graphs and of interval graphs*, Canad. J. Math., 16 (1964), pp. 539–548.

[17] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[18] M. C. Golumbic, H. Kaplan, and R. Shamir, *Graph sandwich problems*, J. Algorithms, 19 (1995), pp. 449–473.

[19] M. C. Golumbic and R. Shamir, *Complexity and algorithms for reasoning about time: A graph-theoretic approach*, J. ACM, 40 (1993), pp. 1108–1133.

[20] E. D. Green and P. Green, *Sequence-tagged site (STS) content mapping of human chromosomes: Theoretical considerations and early experiences*, PCR Methods and Applications, 1 (1991), pp. 77–90.

[21] E. D. Green and M. V. Olson, *Chromosomal region of the Cystic Fibrosis gene in yeast artificial chromosomes: a model for human genome mapping*, Science, 250 (1990), pp. 94–98.

[22] G. Hajös, *Über eine art von graphen*, Problem 65, Internat. Math. Nachr., 11 (1957).

[23] W-L. Hsu and T-H Ma, *Substitution decomposition on chordal graphs and applications*, in Proc. 2nd Int. Symp. on Algorithms (ISA '91), W L. Hsu and R. C. T. Lee, eds., Lecture Notes in Comput. Sci. 557, Springer-Verlag, New York, 1991, pp. 52–60.

[24] G. Isaak, *Discrete interval graphs with bounded representation*, Discrete Appl. Math., 33 (1993), pp. 157–183.

[25] R. M. Karp, *Mapping the genome: Some combinatorial problems arising in molecular biology*, in Proc. 25th Annual ACM Symposium on the Theory of Computing, ACM Press, New York, 1993, pp. 278–285.

[26] D. G. Kendall, *Incidence matrices, interval graphs, and seriation in archaeology*, Pacific J. Math., 28 (1969), pp. 565–570.

[27] N. Korte and R. H. Möhring, *Transitive orientation of graphs with side constraints*, in Proc. of the International Workshop on Graphtheoretic concepts in Computer Science (WG '85), H. Noltemeier, ed., Universitätsverlag Rudolf Trauner, Linz, 1985, pp. 143–160.

[28] N. Korte and R. H. Möhring, *An incremental linear time algorithm for recognizing interval graphs*, SIAM J. Comput., 18 (1989), pp. 68–81.

[29] K. Nökel, *Temporally Distributed Symptoms in Technical Diagnosis*, Lecture Notes in Artificial Intelligence 517, Springer-Verlag, New York, 1991.

[30] J. B. Orlin and R. K. Ahuja, *New scaling algorithms for the assignment and minimum cycle mean problems*, Math. Programming, 54 (1992), pp. 41–56.

[31] C. Papadimitriou and M. Yannakakis, *Scheduling interval ordered tasks*, SIAM J. Comput., 8 (1979), pp. 405–409.

[32] I. Pe'er and R. Shamir, *Satisfiability problems on intervals and unit intervals*, Theoret. Comput. Sci., 175 (1997), pp. 349–372.

[33] F. S. Roberts, *Discrete Mathematical Models, with Applications to Social, Biological and Environmental Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1976.

[34] L. N. Shevrin and N. D. Filippov, *Partially ordered sets and their comparability graphs*, Siberian Math. J., 11 (1970), pp. 497–509.

[35] R. Stanley, *Hyperplane arrangements, interval orders, and trees*, Proc. Nat. Acad. Sci., 93 (1996), pp. 2620–2625.

[36] S. A. Ward and R. H. Halstead, *Computation Structures*, MIT Press, Cambridge, MA, 1990.

[37] A. B. Webber, *Proof of an Interval Satisfiability Conjecture*, Western Illinois University, Macomb, IL, 1994, manuscript.