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

CODES CLOSED UNDER ARBITRARY ABELIAN GROUP OF PERMUTATIONS*

BIKASH KUMAR DEY† AND B. SUNDAR RAJAN‡

Abstract. Algebraic structure of codes over F_q, closed under arbitrary abelian group G of permutations with exponent relatively prime to q, called G-invariant codes, is investigated using a transform domain approach. In particular, this general approach unveils algebraic structure of quasi-cyclic codes, abelian codes, cyclic codes, and quasi-abelian codes with restriction on G to appropriate special cases. Dual codes of G-invariant codes and self-dual G-invariant codes are characterized. The number of G-invariant self-dual codes for any abelian group G is found. In particular, this gives the number of self-dual l-quasi-cyclic codes of length ml over F_q when $(m, q) = 1$. We extend Tanner's approach for getting a bound on the minimum distance from a set of parity check equations over an extension field and outline how it can be used to get a minimum distance bound for a G-invariant code. Karlin's decoding algorithm for a systematic quasi-cyclic code with a single row of circulants in the generator matrix is extended to the case of systematic quasi-abelian codes. In particular, this can be used to decode systematic quasi-cyclic codes with columns of parity circulants in the generator matrix.

Key words. quasi-cyclic codes, permutation group of codes, discrete Fourier transform, self-dual codes

AMS subject classifications. 94B60, 11T71

DOI. 10.1137/S0895480102416192

1. Introduction.

Codes with rich algebraic structure are of strong interest to coding theorists because such codes are easy to design and decode. Classical families of cyclic codes, such as Bose–Chaudhuri–Hocquenghem (BCH) codes and Reed–Muller codes, were the center of attention for a long time. For a cyclic code, the code's permutation group contains a cyclic subgroup generated by the cyclic permutation. A cyclic code can also be viewed as an ideal of the group algebra on the cyclic group of order n (length of the code). More generally, ideals of group algebras on abelian groups are known as abelian codes.

A different direction of generalization gives another class of codes: quasi-cyclic codes. A code of length n is said to be l-quasi-cyclic for some $l|n$ if every l times cyclic shift of a codeword is also a codeword. Thus an l-quasi-cyclic code can be viewed as a submodule of the l-dimensional free module $(F_qC_{\frac{n}{l}})^l$ over the group algebra $F_qC_{\frac{n}{l}}$, where $C_{\frac{n}{l}}$ is a cyclic group of order $\frac{n}{l}$.

A more general, but less popular, class of codes is the class of quasi-abelian codes [15]. For a finite abelian group G and its subgroup H, an F_qH-submodule of F_qG is called a $G - H$ quasi-abelian code. In fact, for an abelian group H and any positive integer t, any submodule of $(F_qH)^t$ can be considered a quasi-abelian code. In that case, any abelian $G \supseteq H$ with $|G| = t|H|$ can be used to define quasi-abelian codes, as in [15]. Thus, such codes will be called H-quasi-abelian codes. When $t = 1$, this class

*Received by the editors October 17, 2002; accepted for publication (in revised form) December 12, 2003; published electronically July 2, 2004. Part of this work was presented at the International Symposium on Information Theory (ISIT), June 30–July 5, 2002, Lausanne, Switzerland. An extended abstract appeared in the *Proceedings of ISIT*, IEEE, Piscataway, NJ, 2002, p. 201.
http://www.siam.org/journals/sidma/18-1/41619.html

†International Institute of Information Technology, Hyderabad 500019, India (bikash@iiit.net).
‡Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore 560012, India (bsrajan@ece.iisc.ernet.in).

specializes to abelian codes and, when H is a cyclic group, specializes to the class of quasi-cyclic codes.

Transform techniques for cyclic codes and abelian codes are well known [1, 13]. Transform techniques for repeated root cyclic codes were discussed in [10]. Recently, quasi-cyclic codes were studied in the transform domain [5, 9]. Tanner [14] introduced ways to transform a group invariant parity check matrix into a parity check matrix over an extension field, and he used this technique to get a lower bound on the minimum distance of group invariant codes.

In this paper, the algebraic structure of codes closed under any arbitrary abelian subgroup G of S_n (the group of permutations of n elements) is investigated. We call this class G-invariant codes. When special types of G are taken, G-invariant codes coincide with the class of quasi-abelian codes, and thus with the classes of quasi-cyclic codes and abelian codes. Figure 1 shows the relation between different classes of codes.

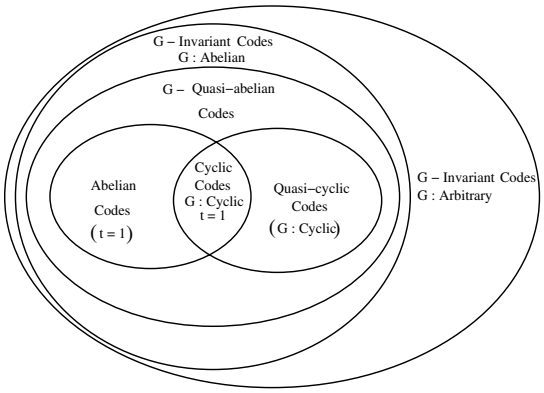


FIG. 1. *Different families of codes and their defining groups of permutations.*

Following are a few examples of some types of permutation groups G shown in Figure 1.

Example 1.1. For any $a, b \in F_q$, $a \neq 0$, let $\sigma_{a,b}$ denote the permutation $\sigma_{a,b} : x \mapsto ax + b$. Then $G = \{\sigma_{a,b} | a \in F_q^*, b \in F_q\}$ is a subgroup of S_q and is called the group of affine permutations. For $q > 2$, G is nonabelian and the G-invariant codes are known as affine invariant codes.

Example 1.2. Figure 2 (ignore the solid, dashed, and dotted boxes for now) shows the cycle structure of the generator σ of a permutation group $G = \langle \sigma \rangle \subseteq S_{16}$. Here G is abelian, and G-invariant codes cannot be seen as G-quasi-abelian codes.

Example 1.3. Consider a permutation group $G = \langle \sigma_1, \sigma_2 \rangle \subseteq S_{54}$. Figure 3 shows the cycles of σ_1 with solid lines with arrows and the cycles of σ_2 with dashed lines with arrows. Here G is abelian, and G-invariant codes are the same as G-quasi-abelian codes of length 54.

All abelian codes on an abelian group G are decomposable as a direct sum of minimal abelian codes if and only if the exponent of G is relatively prime to q. The same is true for l-quasi-cyclic codes if and only if $\frac{n}{l}$ is relatively prime to q [2]. It will be shown that this is true for any G-invariant code (G abelian); i.e., for an abelian subgroup $G \subseteq S_n$, any G-invariant code of length n can be decomposed as a direct sum of minimal G-invariant codes if and only if the exponent of G is relatively prime to q.

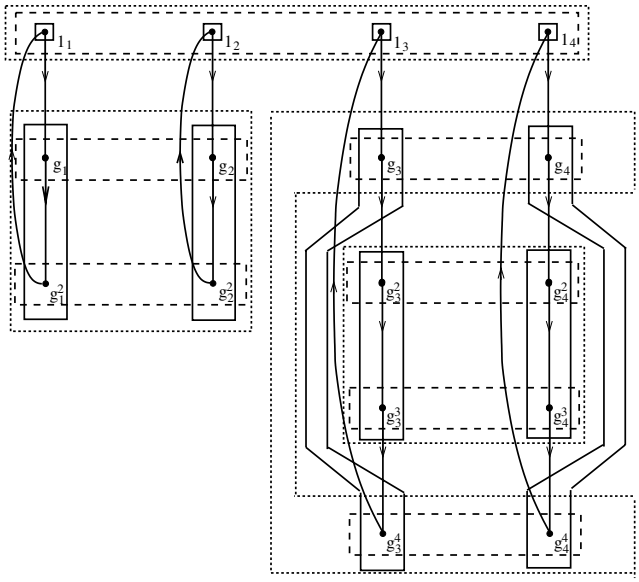


FIG. 2. *Cycle structure of the generator of G in Example* 1.2.

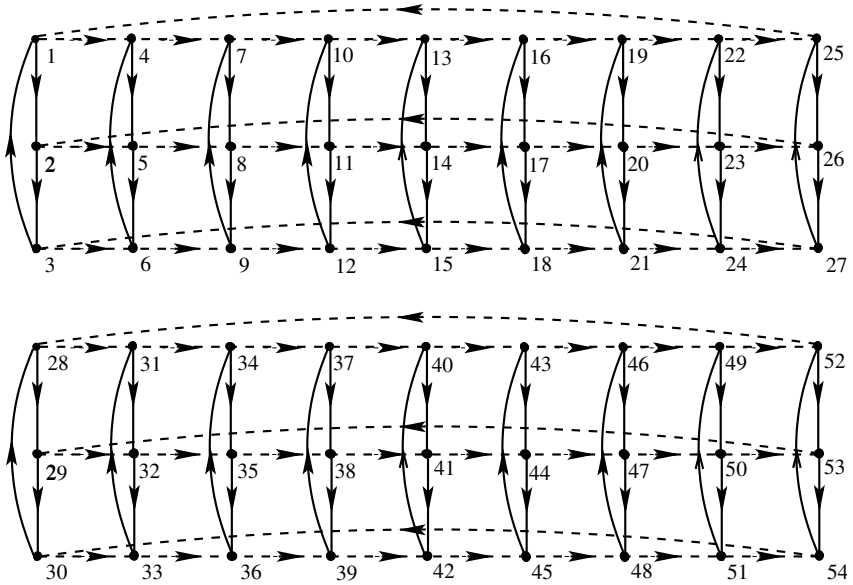


FIG. 3. *Cycle structure of the generators of G in Example* 1.3.

Karlin [7] showed a way to decode a class of one-generator quasi-cyclic codes. Heijnen and van Tilborg [6] proposed another decoding technique for the class of one-generator quasi-cyclic codes, which uses the same basic idea as Karlin's technique but achieves some computational advantages by better usage of the quasi-cyclic property of the code. In this paper, Karlin's approach is extended to a class of quasi-cyclic codes, not necessarily one-generator. When restricted to one-generator quasi-cyclic codes, this method reduces to Karlin's method. Moreover, this method also applies

to a class of quasi-abelian codes specified in subsection 7.1.

In section 2, the DFT on abelian group is reviewed, and in section 3 is used to define a DFT for G-invariant codes for any abelian group G of permutations with exponent relatively prime to q. Such G-invariant codes are characterized in the transform domain, and their structural properties are investigated in section 4. Dual codes of G-invariant codes and self-dual G-invariant codes are characterized in section 5. The number of G-invariant self-dual codes for any abelian group G is also found. In section 6, we extend Tanner's approach for getting a bound on the minimum distance from a set of parity check equations over an extension field and outline how it can be used to get a minimum distance bound for G-invariant codes. Quasi-abelian codes are discussed in section 7, and Karlin's approach [7] for decoding systematic quasi-cyclic codes with parity circulants in a single row is extended to the case of systematic quasi-abelian codes. In particular, this approach can be used to decode systematic quasi-cyclic codes which are not necessarily one-generator, which was the case left open by Karlin.

2. Review of the DFT for abelian codes. Let G be an abelian group with exponent ν such that $(\nu, q) = 1$. Let r be the smallest positive integer such that $\nu|(q^r - 1)$. Then the group of all distinct F_{q^r}-characters of G is isomorphic to G. In fact, an isomorphism $x \mapsto \psi_{(x)}$ can be chosen (see, for example, [3] and the references therein) such that $\psi_{(x)}(y) = \psi_{(y)}(x)$. We denote $\psi_{(x)}(y)$ as $\psi(x, y)$, considering it a map $\psi : G \times G \to F_{q^r}$. It satisfies the following properties:

$$\psi(x, yz) = \psi(x, y)\psi(x, z), \tag{1a}$$

$$\psi(x, y) = \psi(y, x), \tag{1b}$$

$$(\psi(x, y) = \psi(x', y)\ \forall y \in G) \iff x = x', \tag{1c}$$

$$\sum_{x \in G} \psi(x, y) = \begin{cases} |G| & \text{if } y = 1, \\ 0 & \text{if } y \neq 1, \end{cases} \tag{1d}$$

where $|G|$ and 1 denote, respectively, the cardinality of G and the identity element in G.

The DFT of any element $\mathbf{a} = \sum_{x \in G} a_x x \in F_q G$ is defined as $\mathbf{A} = \sum_{x \in G} A_x x \in F_{q^r} G$ such that $A_x = \sum_{y \in G} \psi(x, y) a_y$. The inverse DFT is obtained as $a_x = |G|^{-1} \sum_{y \in G} \psi(x, y)^{-1} A_y$.

3. DFT for G-invariant codes. We consider codes of length n over F_q with components indexed by a set I. Let $G \subseteq Perm(I)$ be an abelian subgroup of the group of permutations of I. Let the characteristic of F_q be p.

Suppose $I_1, \ldots, I_t$ are the orbits of I under the action of G. Let us denote $G_k = \{g^{(k)} | g \in G\}$ for $k = 1, \ldots, t$, where $g^{(k)} \stackrel{\triangle}{=} g|_{I_k} \in Perm(I_k)$ is the permutation g restricted to I_k. Since G_k is abelian and acts on I_k faithfully and transitively, the stabilizer of any $i \in I_k$ is $\{1_k\}$ (1_k denotes the identity element of G_k). Thus, for any $i_1 \in I_k$, there is a unique $g \in G_k$, such that $i_1 = g(i)$. This defines a one-to-one correspondence between G_k and I_k. Using this, the symbols can be indexed by the elements of G_k instead of I_k by first associating a fixed element $i \in I_k$ with the identity element 1_k. Hence, the code symbols are indexed by $\mathcal{G} \stackrel{\triangle}{=} \cup_{i=1}^{t} G_i$ instead of I. Then the element g of G acts on $\mathcal{G}$ as $x \stackrel{g}{\mapsto} g^{(k)} x$ when $x \in G_k$. For any $\mathbf{a} = (a_x)_{x \in \mathcal{G}} \in F_q^{\mathcal{G}}$, $g \in G$ acts on $\mathbf{a}$ as $\mathbf{a} \stackrel{g}{\mapsto} \mathbf{b} = g(\mathbf{a})$ such that $b_x = a_{g^{(k)-1}x}$ when $x \in G_k$. Henceforth, we'll use the letters $f, g,$ and h, possibly with subscripts, to denote elements of G, and use the letters x, y, and z to denote elements of $\mathcal{G}$.

Let the exponent of G, $\exp(G) = lcm\,(\{\exp(G_k)|k = 1, \ldots, t\})$ be relatively prime to q, and let r be the smallest positive integer such that $\exp(G)$ divides $(q^r - 1)$. Then on each orbit, DFT is defined as discussed in the last section; i.e., the DFT of $\mathbf{a} \in F_q^{\mathcal{G}}$ is defined as $\mathbf{A} = (A_x)_{x\in\mathcal{G}} \in F_{q^r}^{\mathcal{G}}$, where

$$A_x = \sum_{y\in G_k} \psi_k(x, y)a_y \qquad \forall x \in G_k,\ \forall k.$$

Here ψ_k is as defined in the last section for G_k. For any two $x, y \in \mathcal{G}$, define

$$\Psi(x, y) = \begin{cases} \psi_k(x, y) & \text{when } x, y \in G_k \text{ for some } k, \\ 0 & \text{when } x \in G_{k_1} \text{ and } y \in G_{k_2}, \text{ s.t. } k_1 \neq k_2. \end{cases}$$

With this notation, the DFT can be rewritten as $A_x = \sum_{y\in\mathcal{G}} \Psi(x, y)a_y \forall x \in \mathcal{G}$. Clearly, $\mathbf{A}$ satisfies $A_{x^q} = A_x^q\ \forall x \in \mathcal{G}$. For any $h \in G$ and $x \in \mathcal{G}$, we define the symbol

$$\langle h, x\rangle \stackrel{\triangle}{=} \psi_k(h^{(k)}, x) \quad \text{when } x \in G_k. \tag{2}$$

It follows from this definition that the DFT of $\mathbf{b} = h(\mathbf{a})$ is given by $B_x = \langle h, x\rangle A_x$. Suppose $h_1, h_2 \in G_k$. Then using (1a) and (1c), we have $\langle g, h_1\rangle^l = \langle g, h_2\rangle\ \forall g \in G$ if and only if $h_1^l = h_2$.

For any element $x \in \mathcal{G}$, it is in G_k for some k, and thus a *cyclotomic coset* of x is defined as $[x]^q \stackrel{\triangle}{=} \{y \in G_k|y = x^{q^t} \text{ for some nonnegative } t\}$. Cardinality of $[x]^q$ will be denoted as r_x. For any subset $S \subseteq \mathcal{G}$, we define $[S]^q \stackrel{\triangle}{=} \cup_{s\in S}[s]^q$.

Corollary 3.1. *For any $x \in \mathcal{G}$, r_x is the smallest positive integer such that $\langle g, x\rangle^{q^{r_x}} = \langle g, x\rangle\ \forall g \in G$. Thus, r_x is the least common multiple (lcm) of the lengths of the conjugacy classes of $\langle g, x\rangle\ \forall g \in G$.*

The *residue class* of $x \in \mathcal{G}$ is defined as $\widetilde{x} \stackrel{\triangle}{=} \{x_1 \in \mathcal{G}|\langle g, x_1\rangle = \langle g, x\rangle \text{ for each } g \in G\}$. Cardinality of $\widetilde{x}$ will be denoted by e_x. For any subset $X = \{x_1, x_2, \ldots, x_k\} \subseteq \mathcal{G}$, A_X denotes the ordered tuple $(A_{x_1}, A_{x_2}, \ldots, A_{x_k})$ with an arbitrary fixed order in X. In particular, for any residue class $\widetilde{y_1} = \{y_1, y_2, \ldots, y_l\}$, we denote by $A_{\widetilde{y}}$ the ordered l-tuple $(A_{y_1}, A_{y_2}, \ldots, A_{y_l})$ with an arbitrarily chosen fixed order on $\widetilde{y}$. For some ordered tuples $T_1 = (t_{1,1}, \ldots, t_{1,j_1}), \ldots, T_l = (t_{l,1}, \ldots, t_{l,j_l})$ the concatenated tuple $(t_{1,1}, \ldots, t_{1,j_1}, \ldots, t_{l,1}, \ldots, t_{l,j_l})$ is denoted $(T_1, \ldots, T_l)$.

The *cyclotomic residue class* of $x \in \mathcal{G}$ is defined as $(x)^q \stackrel{\triangle}{=} \{x_1 \in \mathcal{G}| \text{ for some nonnegative } t, \langle g, x_1\rangle^{q^t} = \langle g, x\rangle\ \forall g \in G\} = [\widetilde{x}]^q$. Figure 4 shows the relation between a cyclotomic residue class and the cyclotomic cosets and residue classes in it. By the conjugacy constraint, the values of the DFT components in one residue class determine the values of the other transform components in the same cyclotomic residue class. To be specific, $A_{\widetilde{x^{q^i}}} = A_{\widetilde{x}}^{q^i}$ for any $\mathbf{a} \in F_q^{\mathcal{G}}$, where the power of the vector $A_{\widetilde{x}}$ is taken componentwise. Thus, the values of the transform components in one representative residue class from each cyclotomic residue class specify a vector completely.

Example 3.1 (continuation of Example 1.2). The index set has four orbits under the action of G and $G_1 \simeq G_2 \simeq \mathbb{Z}_3$, and $G_3 \simeq G_4 \simeq \mathbb{Z}_5$. Let a set of generators of the groups G_1, G_2, G_3, and G_4 be g_1, g_2, g_3, and g_4, respectively. If $\alpha \in F_{q^r}$ is an element of order 15, then we define DFT in $F_q^{16} \simeq F_q^{\mathcal{G}}$ with respect to the maps ψ_k defined by $\psi_1(g_1, g_1) = \psi_2(g_2, g_2) = \alpha^5,\ \psi_3(g_3, g_3) = \psi_4(g_4, g_4) = \alpha^3$. The residue classes in $\mathcal{G}$ are shown in Figure 2 with dashed boxes. The figure shows the cyclotomic cosets with solid boxes and the cyclotomic residue classes with dotted boxes for $q \equiv 2 \bmod 3$, $q \equiv 4 \bmod 5$ (e.g., $q = 29, 59$).

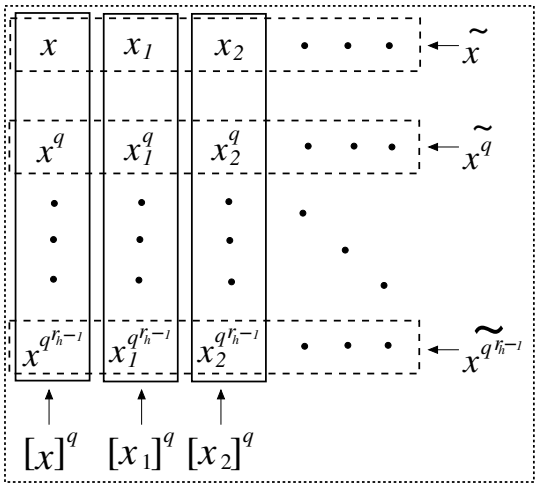


FIG. 4. *A generic cyclotomic residue class* $(x)^q$.

4. Transform domain characterization of G-invariant codes. A linear code $\mathcal{C} \subseteq F_q^{\mathcal{G}}$ is G invariant if for every codeword $\mathbf{a} \in \mathcal{C}$ and $h \in G$, $h(\mathbf{a}) \in \mathcal{C}$. The equivalent condition in the transform domain is that for any $h \in G$, $\mathbf{A} = DFT(\mathbf{a})$ for some $\mathbf{a} \in \mathcal{C}$ and $\mathbf{B} \in F_{q^r}^{\mathcal{G}}$ with $B_x = \langle h, x\rangle A_x \ \forall x \in \mathcal{G} \Rightarrow \mathbf{B} = DFT(\mathbf{b})$ for some $\mathbf{b} \in \mathcal{C}$.

For any ordered tuple $(x_1, x_2, \ldots, x_l)$ on $\mathcal{G}$, we say $(A_{x_1}, A_{x_2}, \ldots, A_{x_l})$ takes values from $\{(A_{x_1}, A_{x_2}, \ldots, A_{x_l}) \,|\mathbf{a} \in \mathcal{C}\}$ for $\mathcal{C}$. If for $\mathcal{C}$, $(A_{x_1}, A_{x_2}, \ldots, A_{x_l})$ takes values from $V \subseteq F_{q^r}^l$ and $U \subseteq V$, then the subcode $\{\mathbf{a} \in \mathcal{C} |\, (A_{x_1}, A_{x_2}, \ldots, A_{x_l}) \in U\}$ will be referred to as the subcode obtained from $\mathcal{C}$ by restricting $(A_{x_1}, A_{x_2}, \ldots, A_{x_l})$ to U.

LEMMA 4.1. *For any G-invariant code $\mathcal{C}$ and $x \in \mathcal{G}$, $A_{\widetilde{x}}$ takes values from a subspace of $F_{q^{r_x}}^{e_x}$.*

Proof. Suppose $A_{\widetilde{x}}$ takes values from an F_q-subspace (since the code is linear) $V \subseteq F_{q^{r_x}}^{e_x}$ for $\mathcal{C}$. When any element $g \in G$ acts on a codeword $\mathbf{a}$, $A_{\widetilde{x}}$ is multiplied by $\langle g, x\rangle$. Since the code is G-invariant, $\langle g, x\rangle v \in V$ for each $g \in G$ and $v \in V$. Thus, V is closed under multiplication by elements of $Span_{F_q}(\langle\{\langle g, x\rangle | g \in G\}\rangle) = F_q\left[\{\langle g, x\rangle | g \in G\}\right] = F_{q^{r_x}}$. □

For any G-invariant code $\mathcal{C}$ and $x \in \mathcal{G}$, suppose $A_{\widetilde{x}}$ takes values from a subspace $V \subseteq F_{q^{r_x}}^{e_x}$. Then for any subspace $U \subseteq V$, the subcode obtained by restricting $A_{\widetilde{x}}$ to U is also G-invariant. For a linear code $\mathcal{C}$, suppose, $A_{\widetilde{x}}$ takes values from a subspace $V \subseteq F_{q^{r_x}}^{e_x}$, and $V = V_1 + V_2$. If the subcodes obtained by restricting $A_{\widetilde{x}}$ to V_1 and V_2 are, respectively, $\mathcal{C}_1$ and $\mathcal{C}_2$, then $\mathcal{C} = \mathcal{C}_1 + \mathcal{C}_2$.

DEFINITION 4.2. *Let $X_1, X_2, \ldots, X_l$ be some disjoint subsets of $\mathcal{G}$ and suppose $R_{X_j} = \{A_{X_j} | \mathbf{a} \in \mathcal{C}\}$ for $j = 1, 2, \ldots, l$. The sets of transform components $\{A_x | x \in X_j\}$, $1 \leq j \leq l$, are said to be* unrelated *in $\mathcal{C}$ if $\{(A_{X_1}, A_{X_2}, \ldots, A_{X_l}) \,|\mathbf{a} \in \mathcal{C}\} = R_{X_1} \times R_{X_2} \times \cdots \times R_{X_l}$. They are said to be* related *if they are not unrelated.*

Let $\widetilde{x_1}, \widetilde{x_2}, \ldots, \widetilde{x_l}$ be a set of representative residue classes of all the distinct cyclotomic residue classes. Suppose we fix arbitrary subspaces V_i, $i = 1, 2, \ldots, l$, of $F_{q^{r_{x_i}}}^{e_{x_i}}$, $i = 1, 2, \ldots, l$, respectively, and consider the code $\mathcal{C} = \left\{\mathbf{a} \in F_q^{\mathcal{G}} | A_{\widetilde{x_i}} \in V_i \text{ for } i = 1, 2, \ldots, l\right\}$. Clearly, the code is G-invariant. But it is not clear whether any G-invariant code can be obtained this way by choosing suitable V_i, $i = 1, 2, \ldots, l$. That is, are $A_{\widetilde{x_i}}$, $i = 1, \ldots, l$, unrelated for any G-invariant code? Theorem 4.6 will

answer this question in the affirmative.

If, in a G-invariant code, two transform components A_x and A_y are unrelated, then consider the subcodes $\mathcal{C}_1$ and $\mathcal{C}_2$ obtained by restricting, respectively, A_x and A_y to $\{0\}$. Clearly, the original code is the sum of the codes $\mathcal{C}_1$ and $\mathcal{C}_2$. Suppose $S_1, \ldots, S_l$ are some disjoint subsets of the index set such that $x, y \in \cup_{i=1}^{l} S_i$. Then the transform components in $S_1, \ldots, S_l$ are unrelated in $\mathcal{C}$ if and only if they are unrelated in $\mathcal{C}_1$ and $\mathcal{C}_2$. This process can be continued on $\mathcal{C}_1$ and $\mathcal{C}_2$ and repeated on the resulting subcodes to get a set of subcodes whose sum is $\mathcal{C}$ and in each of which either there is only one nonzero transform component or any pair of nonzero transform components is related. So, if the transform components in $S_1, \ldots, S_l$ are related in $\mathcal{C}$, then there is a G-invariant subcode of $\mathcal{C}$, where two transform components $A_x, A_y, x \in S_i, y \in S_j, i \neq j$, are related.

Suppose, in a G-invariant code, two transform components A_x and A_y are related. Then they must take values from $F_{q^{r_x}}$ and $F_{q^{r_y}}$, respectively. The relation must be by a bijection (so $r_x = r_y$) $\sigma : F_{q^{r_x}} \to F_{q^{r_x}}$ since the subcode obtained by restricting A_x or A_y to $\{0\}$ is G-invariant. Since the code is linear G-invariant, σ must be an F_q-linear isomorphism satisfying

$$\sigma(\langle g, x\rangle v) = \langle g, y\rangle \sigma(v) \ \ \forall g \in G, \ \forall v \in F_{q^{r_x}}. \tag{3}$$

For a map σ of a finite field, we denote by $f_\sigma(X)$ a polynomial which induces σ, that is, $\sigma(a) = f_\sigma(a)$.

LEMMA 4.3. *Let $\alpha, \beta \in F_{q^l}$ be such that the length of the F_q-conjugacy class of α is l_1. Suppose $a \in F_{q^l}^*$ and $\sigma : aF_{q^{l_1}} \longrightarrow F_{q^l}$ is an F_q-linear nonzero map. Then σ satisfies $\sigma(\alpha b) = \beta\sigma(b)\ \forall b \in aF_{q^{l_1}}$ if and only if $\beta = \alpha^{q^j}$ and $f_\sigma(X) = cX^{q^j}$ for some unique $c \in F_{q^l}$ and $j < l_1$.*

Proof. The reverse implication is obvious. For the forward implication, let us consider the F_q-linear map $\sigma' : F_{q^{l_1}} \to F_{q^l}$; $\sigma' : x \mapsto \frac{\sigma(ax)}{\sigma(a)}$. Clearly, $\sigma'(\alpha^i) = \beta^i$ for $i \geq 0$. Thus, σ' is a field isomorphism of $F_q[\alpha]$ onto $F_q[\beta]$. So for some j, $\sigma'(x) = x^{q^j}$ $\forall x \in F_q[\alpha] = F_{q^{l_1}}$. Therefore,

$$\sigma(x) = \sigma(a)\sigma'\left(\frac{x}{a}\right) = \sigma(a)a^{-q^j}x^{q^j} \text{ for any } x \in aF_{q^{l_1}}. \qquad \square$$

LEMMA 4.4. *Let α, β, and l_1 be as in Lemma 4.3 and V be an h-dimensional $F_{q^{l_1}}$-subspace of F_{q^l}. Suppose $\sigma : V \longrightarrow F_{q^l}$ is a nonzero F_q-linear map. If σ satisfies $\sigma(\alpha b) = \beta\sigma(b)\ \forall b \in V$, then $\beta = \alpha^{q^j}$ and $f_\sigma(X) = \sum_{i=0}^{h-1} c_i X^{q^{il_1+j}}$ for some unique $c_i \in F_{q^l}$ for $0 \leq i \leq h-1$.*

Proof. Suppose $V = \oplus_{i=0}^{h-1} V_i$, where $V_i = s_i F_{q^{l_1}}$. Since σ is nonzero, its restriction on at least one of $V_i, 0 \leq i \leq h-1$, is nonzero, and thus by Lemma 4.3, the first statement follows. Suppose $\sigma_i = \sigma|_{V_i}$. Then, $f_{\sigma_i}(X) = c_i' X^{q^j}$ for some unique c_i'. Thus,

$$\begin{aligned}
f_\sigma(X) &= \sum_{w=0}^{h-1} c_w X^{q^{wl_1+j}} \\
&\Leftrightarrow c_i'(s_i a)^{q^j} = \sum_{w=0}^{h-1} c_w (s_i a)^{q^{wl_1+j}} \ \ \forall a \in F_{q^{l_1}}, \ \forall i \in [0, h-1] \\
&\Leftrightarrow c_i' s_i' = \sum_{w=0}^{h-1} c_w \left(s_i'\right)^{q^{wl_1}} \ \ \forall i \in [0, h-1], \quad \text{where } s_i' = (s_i)^{q^j}
\end{aligned}$$

$$(4) \qquad \Leftrightarrow \begin{pmatrix} s'_0 & s'^{q^{l_1}}_0 & s'^{q^{2l_1}}_0 & \cdots & s'^{q^{(h-1)l_1}}_0 \\ s'_1 & s'^{q^{l_1}}_1 & s'^{q^{2l_1}}_1 & \cdots & s'^{q^{(h-1)l_1}}_1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s'_{h-1} & s'^{q^{l_1}}_{h-1} & s'^{q^{2l_1}}_{h-1} & \cdots & s'^{q^{(h-1)l_1}}_{h-1} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{h-1} \end{pmatrix} = \begin{pmatrix} c'_0 s'_0 \\ c'_1 s'_0 \\ \vdots \\ c'_{h-1} s'_{h-1} \end{pmatrix}.$$

Now, $\{s_0, s_1, s_2, \ldots, s_{h-1}\}$ are linearly independent over $F_{q^{l_1}}$ since $V_j = \oplus_{i=0}^{h-1} s_i F_{q^{l_1}}$. Thus, $\{s'_0, s'_1, s'_2, \ldots, s'_{h-1}\}$ are also linearly independent over $F_{q^{l_1}} \Rightarrow$ the $h \times h$ matrix in (4) is nonsingular, and thus there exists a unique solution of (4) for $c_0, c_1, \ldots, c_{h-1}$. □

LEMMA 4.5. *Let α_i, $1 \leq i \leq k$, be some elements of F_{q^l} with length of conjugacy classes l_i, $i = 1, \ldots, k$, respectively. Suppose $l' = lcm(l_1, \ldots, l_k)$ and $\sigma : F_{q^{l'}} \longrightarrow F_{q^l}$ is a nonzero F_q-linear map. If σ satisfies $\sigma(\alpha_i b) = \beta_i \sigma(b)\ \forall b \in F_{q^{l'}}$ for some $\beta_i \in F_{q^l}, i = 1, \ldots, k$, then there exists an integer $j \geq 0$ such that $\beta_i = \alpha_i^{q^j}$ for $i = 1, \ldots k$, and $f_\sigma(X) = cX^{q^j}$ for some unique $c \in F_{q^l}$.*

Proof. Suppose $l'_i = \frac{l'}{l_i}$, $i = 1, \ldots, k$. By Lemma 4.4, $\beta_i = \alpha_i^{q^{j_i}}$ for some nonnegative j_i, $i = 1, \ldots, k$. Now, $\exists$ a unique polynomial $f_\sigma(X)$ of degree $< q^{l'}$. Applying Lemma 4.4 for each i, we see that σ is induced by $f_i(X) = \sum_{h_i=0}^{l'_i - 1} c_{i,h_i} X^{q^{h_i l_i + j_i}}$, where $c_{h_i}, 0 \leq h_i \leq l'_i - 1$, are some unique constants. Since all the polynomials $f_i(X)$ are of degree $< q^{l'}$, they have to be the same. In particular, their smallest degree terms are the same, and that means, say, $j = h_1 l_1 + j_1 = \cdots = h_k l_k + j_k$. Now, if there is any nonzero monomial other than X^{q^j}, then such a monomial is of degree, say, $j' = h'_1 l_1 + j_1 = \cdots = h'_k l_k + j_k$. Thus,

$$\begin{aligned} (h'_1 - h_1) l_1 = \cdots = (h'_k - h_k) l_k \\ \Rightarrow l' = lcm(l_1, \ldots, l_k) | (h'_1 - h_1) l_1. \end{aligned}$$

This contradicts the fact that $(h'_1 - h_1) < l'_1 = \frac{l'}{l_1}$. Thus, $f_\sigma(X) = cX^{q^j}$ for some unique constant c and $\alpha_i = \beta_i^{q^j}, i = 1, \ldots, k$. □

By (3) and Lemma 4.5, for a linear G-invariant code, two transform components cannot be related unless they are in the same cyclotomic residue class. Thus, we have the following theorem.

THEOREM 4.6. *Let $(x_i)^q$, $i = 1, 2, \ldots, k$, be the distinct cyclotomic residue classes. Then for any linear G-invariant code, $\{A_x | x \in (x_i)^q\}$, $i = 1, 2, \ldots, k$, are unrelated.*

COROLLARY 4.7. *Let $(x_i)^q$, $i = 1, 2, \ldots, k$, be the distinct cyclotomic residue classes. Then, any linear G-invariant code $\mathcal{C}$ is*

$$(5) \qquad \mathcal{C} = \bigoplus_{i=1}^{k} \mathcal{C}_{(x_i)^q},$$

where $\mathcal{C}_{(x_i)^q}$ denotes the subcode of $\mathcal{C}$ obtained by restricting all the transform components outside $(x_i)^q$ to zero.

For quasi-cyclic codes, this gives the primary components of a code [8], and for cyclic and abelian codes, these subcodes, when nonzero, are minimal cyclic and abelian codes, respectively.

A nonzero linear G-invariant code is called minimal if it does not have any nontrivial linear G-invariant subcode. For a minimal G-invariant code, transform components in only one cyclotomic residue class $(x)^q$ are nonzero and $A_{\underset{\sim}{x}}$ takes values

from a one-dimensional subspace of $F_{q^{r_x}}^{e_x}$. Since any vector space is a direct sum of one-dimensional vector spaces, we have the following theorem.

THEOREM 4.8. *Any G-invariant code is a direct sum of minimal G-invariant codes.*

However, the decomposition of a G-invariant code in terms of some minimal G-invariant codes is not unique, though for the special case of abelian codes, such a decomposition (as a direct sum of minimal abelian codes) is unique.

It is known that if $(\exp(G), q) \neq 1$, then there are abelian codes on that group, which cannot be decomposed as a direct sum of minimal abelian codes. If $(\exp(G), q) \neq 1$, then for some k, $(\exp(G_k), q) \neq 1$. Then we can take an abelian code on G_k, which cannot be decomposed as a direct sum of minimal abelian codes. That code can be padded with zeros in all other orbits to get a G-invariant code, which is not decomposable as a direct sum of minimal G-invariant codes.

THEOREM 4.9 (transform domain characterization). *Let G be an abelian group of permutations with order relatively prime to q. Then a code is G-invariant if and only if the following hold:*

(i) *For any $x \in \mathcal{G}$, $A_{\widetilde{x}}$ takes values from a subspace of $F_{q^{r_x}}^{e_x}$.*

(ii) *If $x_1, \ldots, x_k$ are representatives of the distinct cyclotomic residue classes of $\mathcal{G}$, then $A_{\widetilde{x_1}}, \ldots, A_{\widetilde{x_k}}$ are unrelated.*

5. Duals of G-invariant codes. To characterize duals of G-invariant codes, some generalizations of Euclidean and Hermitian dual codes are needed. Let $\mathbf{v} = (v_1, \ldots, v_l) \subseteq F_q^l$ be a vector with each component nonzero. For any two vectors $\mathbf{a}, \mathbf{b} \in F_q^l$, the $\mathbf{v}$-weighted Euclidean inner product (or $E_{\mathbf{v}}$-inner product) of $\mathbf{a}$ and $\mathbf{b}$ is defined as

$$E_{\mathbf{v}}(\mathbf{a}, \mathbf{b}) = \sum_{x=1}^{l} v_x a_x b_x. \tag{6}$$

Similarly, for any $\mathbf{v} \in F_q^l$, the $\mathbf{v}$-weighted Hermitian inner product, or $H_{\mathbf{v}}$-inner product, of $\mathbf{a} \in F_{q^2}^l$ and $\mathbf{b} \in F_{q^2}^l$ is defined as

$$H_{\mathbf{v}}(\mathbf{a}, \mathbf{b}) = \sum_{x=1}^{l} v_x a_x b_x^q. \tag{7}$$

When $\mathbf{v}$ is an "all-ones" vector, the $\mathbf{v}$-weighted Euclidean inner product and $\mathbf{v}$-weighted Hermitian inner product reduce to the usual Euclidean and Hermitian inner products, respectively.

Two vectors are called orthogonal w.r.t. an inner product if the inner product of the vectors is zero. Two linear codes $\mathcal{C}_1$ and $\mathcal{C}_2$ are called the dual of each other with respect to an inner product if $\mathcal{C}_2$ is the set of all the vectors which are orthogonal to every vector in $\mathcal{C}_1$. When no inner product is specified, it is assumed to be a Euclidean inner product. A code is called self-dual when it is the dual of itself.

For any $x \in \mathcal{G}$, τ_x will denote the cardinality of the orbit containing x. For any residue class $\widetilde{x}$, $\tau_{\widetilde{x}}$ will denote the e_x-tuple with components τ_y, $y \in \widetilde{x}$, in the same order as A_y in $A_{\widetilde{x}}$. With abuse of notation, $\tau_{\widetilde{x}}^{-1}$ will denote the componentwise inverse (in $F_p \subseteq F_q$) of $\tau_{\widetilde{x}}$.

THEOREM 5.1. *For a G-invariant code $\mathcal{C}$, a vector $\mathbf{b} \in F_q^{\mathcal{G}}$ is orthogonal to $\mathcal{C}$ if*

and only if $\forall \mathbf{a} \in \mathcal{C}$,

$$\sum_{y \in \widetilde{x}} \tau_y^{-1} A_y B_{y^{-1}} = 0 \quad \forall \text{ cyclotomic residue classes } (x)^q. \tag{8}$$

Proof. Clearly, $\mathbf{b}$ is orthogonal to $\mathcal{C}$ if and only if

$$\begin{aligned}
\mathbf{a} \perp \mathbf{b} \forall \mathbf{a} \in \mathcal{C} &\iff \sum_{y \in \mathcal{G}} a_y b_y = 0 && \forall \mathbf{a} \in \mathcal{C} \\
&\iff \sum_{y \in \mathcal{G}} \tau_y^{-1} A_y B_{y^{-1}} = 0 && \forall \mathbf{a} \in \mathcal{C} \\
&\iff \sum_{i=0}^{r_x - 1} \sum_{y \in \widetilde{x}} \tau_y^{-1} A_{y^{q^i}} B_{\left(y^{q^i}\right)^{-1}} = 0 && \text{for each } (x)^q,\ \forall \mathbf{a} \in \mathcal{C} \qquad (9) \\
&\iff \sum_{i=0}^{r_x - 1} \left(\sum_{y \in \widetilde{x}} \tau_y^{-1} A_y B_{y^{-1}} \right)^{q^i} = 0 && " \\
&\iff Tr_{F_{q^{r_x}}/F_q} \left(\sum_{y \in \widetilde{x}} \tau_y^{-1} A_y B_{y^{-1}} \right) = 0 && " \\
&\iff \sum_{y \in \widetilde{x}} \tau_y^{-1} A_y B_{y^{-1}} = 0 && " \ . \qquad (10)
\end{aligned}$$

To get (9), we use the fact that the transform components in different cyclotomic residue classes are unrelated for a G-invariant code, and to obtain (10) we use the fact that $A_{\widetilde{x}}$ takes values from a subspace of $F_{q^{r_x}}^{e_x}$. ∎

Note that if (8) is satisfied for a residue class $\widetilde{x}$, then it is also satisfied for any other residue class in the same cyclotomic residue class. Thus, it is sufficient to consider only one representative residue class in each cyclotomic residue class. When two residue classes $\widetilde{x}$ and $\widetilde{x^{-1}}$ are considered, compatible orders are taken in them; i.e., if we take

$$A_{\widetilde{x}} = \left(A_x, A_{x_1}, \ldots, A_{x_{e_x - 1}} \right),$$

then we also take

$$A_{\widetilde{x^{-1}}} = (A_{x^{-1}}, A_{x_1^{-1}}, \ldots, A_{x_{e_x - 1}^{-1}}).$$

Let $\{x_1, x_2, \ldots, x_l\}$ be a set of representatives of the distinct cyclotomic residue classes of $\mathcal{G}$. Suppose, for the codes $\mathcal{C}_1$ and $\mathcal{C}_2$, $A_{\widetilde{x}}$ takes values from V_x and U_x, respectively. Then V_x and U_x can also be considered linear codes of length e_x over $F_{q^{r_x}}$. Using Theorem 5.1, the dual code of a G-invariant code can be characterized as follows.

THEOREM 5.2. *Two G-invariant codes $\mathcal{C}_1$ and $\mathcal{C}_2$ are the dual of each other if and only if for each x_i, $i = 1, 2, \ldots, l$, V_{x_i} and $U_{x_i^{-1}}$ are the $E_{\tau_{\widetilde{x}_i}^{-1}}$-dual of each other.*

5.1. Self-dual G-invariant codes. Let us classify the cyclotomic residue classes into the following three types:

1. Type A: Self-inverse cyclotomic residue classes $(x)^q$ with $x = x^{-1}$. In this case, suppose $x = x^{-1} \in G_k$, i.e., $x^2 = 1_k$. Then either $x = 1_k$ or order of G_k is even $\Rightarrow q$ is odd (since $(q, |G_k|) = 1) \Rightarrow x^q = x \Rightarrow r_x = 1$.

2. Type B: Self-inverse cyclotomic residue classes $(x)^q$ with $x \neq x^{-1}$. In this case,

$$x^{-1} = x^{q^i} \text{ for some } i < r_x, \quad i \neq 0.$$

Thus,

$$x = \left(x^{-1}\right)^{-1} = \left(x^{q^i}\right)^{-1} = \left(x^{-1}\right)^{q^i} = x^{q^{2i}} \Rightarrow r_x | 2i \Rightarrow 2 | r_x \text{ and } i = \frac{r_x}{2}.$$

3. Type C: Cyclotomic residue classes $(x)^q$ which are not self-inverse, i.e., $x^{-1} \notin (x)^q$.

The cyclotomic cosets are also assigned a "type" based on the type of cyclotomic residue classes they are in. Let us denote the distinct cyclotomic residue classes as

$$\begin{aligned}
&\text{Type A: } (x_1)^q, \ldots, (x_{i_1})^q, \\
&\text{Type B: } (y_1)^q, \ldots, (y_{i_2})^q, \\
&\text{Type C: } (z_1)^q, (z_1^{-1})^q \ldots, (z_{i_3})^q, (z_{i_3}^{-1})^q.
\end{aligned}$$

THEOREM 5.3. *Let $\mathcal{C}$ be a G-invariant code, where $A_{\underset{\sim}{x_i}}$, $A_{\underset{\sim}{y_j}}$, $A_{\underset{\sim}{z_k}}$, and $A_{\underset{\sim}{z_k^{-1}}}$ take values from the subspaces V_{x_i}, V_{y_j}, V_{z_k}, and $V_{z_k^{-1}}$, respectively, for $i = 1, \ldots, i_1$, $j = 1, \ldots, i_2$, $k = 1, \ldots, i_3$. The code is self-dual if and only if*

(i) *V_{x_i} is an $E_{\underset{\sim}{\tau^{-1}}_{x_i}}$-self-dual code for $i = 1, \ldots, i_1$.*

(ii) *V_{y_j} is an $H_{\underset{\sim}{\tau^{-1}}_{y_j}}$-self-dual code for $j = 1, \ldots, i_2$.*

(iii) *V_{z_k} is the $E_{\underset{\sim}{\tau^{-1}}_{z_k}}$-dual code of $V_{z_k^{-1}}$ for $k = 1, \ldots, i_3$.*

Proof. If the code is self-dual, then by Theorem 5.2, V_{y_j} is the $E_{\underset{\sim}{\tau^{-1}}_{x_i}}$-dual of $V_{y_j^{-1}}$. Now,

$$V_{y_j} \text{ is } E_{\underset{\sim}{\tau^{-1}}_{x_i}}\text{-dual of } V_{y_j^{-1}} \iff V_{y_j} = \left\{ \mathbf{v} \in F_{q^{r_{y_j}}}^{e_{y_j}} | E_{\underset{\sim}{\tau^{-1}}_{x_i}}(\mathbf{v}, \mathbf{u}) = 0 \ \forall \mathbf{u} \in V_{y_j^{-1}} \right\}.$$

But,

$$V_{y_j^{-1}} = \left\{ \left(u_1^{q^{\frac{r_{y_j}}{2}}}, \ldots, u_{e_{y_j}}^{q^{\frac{r_{y_j}}{2}}} \right) \middle| \mathbf{u} \in V_{y_j} \right\}.$$

Thus,

$$\begin{aligned}
V_{y_j} \text{ is } E_{\underset{\sim}{\tau^{-1}}_{x_i}}\text{-dual of } V_{y_j^{-1}} &\iff V_{y_j} = \left\{ \mathbf{v} \in F_{q^{r_{y_j}}}^{e_{y_j}} | H_{\underset{\sim}{\tau^{-1}}_{x_i}}(\mathbf{v}, \mathbf{u}) = 0 \ \forall \mathbf{u} \in V_{y_j} \right\} \\
&\iff V_{y_j} \text{ is } H_{\underset{\sim}{\tau^{-1}}_{y_j}} \text{ self-dual.}
\end{aligned}$$

The rest of the proof follows directly from Theorem 5.2. ∎

Let $N_{E_{\mathbf{v}}}(q,l)$ and $N_{H_{\mathbf{v}}}(q,l)$ denote the number of, respectively, $E_{\mathbf{v}}$-self-dual codes and $H_{\mathbf{v}}$-self-dual codes of length l over F_q. Also, let $N(q,l)$ denote the number of subspaces of F_q^l. All these numbers are known [11, 12] when $\mathbf{v}$ is all-ones and the values are as given below.

$$N(q,l) = \sum_{i=0}^{l} \prod_{j=0}^{i-1} \frac{q^l - q^j}{q^i - q^j}, \tag{11}$$

$$N_{E_{\mathbf{1}}}(q,l) = \begin{cases} \prod_{i=1}^{\frac{l}{2}-1}(q^i+1) & \text{for } q \text{ and } l \text{ even}, \\ 2\prod_{i=1}^{\frac{l}{2}-1}(q^i+1) & \text{for } q \equiv 1 \bmod 4,\ l \text{ even}, \\ 2\prod_{i=1}^{\frac{l}{2}-1}(q^i+1) & \text{for } q \equiv 3 \bmod 4,\ l \text{ is divisible by 4}, \\ 0 & \text{otherwise}, \end{cases} \tag{12}$$

$$N_{H_{\mathbf{1}}}(q,l) = \begin{cases} \prod_{i=0}^{\frac{l}{2}-1}(q^{i+\frac{1}{2}}+1), & \text{when } l \text{ is even}, \\ 0, & \text{otherwise}. \end{cases} \tag{13}$$

Theorem 5.3 directly gives Theorem 5.4.

THEOREM 5.4. *The number of self-dual G-invariant codes over F_q is*

$$\prod_{i=1}^{i_1} N_{E_{\underset{\sim}{\tau}^{-1}_{x_i}}}(q^{r_{x_i}}, e_{x_i}) \prod_{j=1}^{i_2} N_{H_{\underset{\sim}{\tau}^{-1}_{y_j}}}(q^{r_{y_j}}, e_{y_j}) \prod_{k=1}^{i_3} N(q^{r_{z_k}}, e_{z_k}),$$

where the empty product is 1 *by convention.*

When $|G_1| \equiv |G_2| \equiv \cdots \equiv |G_t| \bmod p$, the $E_{\underset{\sim}{\tau}^{-1}_{x_i}}$-duality and $H_{\underset{\sim}{\tau}^{-1}_{y_j}}$-duality are the same as the Euclidean and Hermitian dualities, respectively. So in that case,

$$N_{E_{\underset{\sim}{\tau}^{-1}_{x_i}}}(q^{r_{x_i}}, e_{x_i}) = N_{E_{\mathbf{1}}}(q^{r_{y_j}}, e_{x_i}),$$
$$N_{H_{\underset{\sim}{\tau}^{-1}_{x_i}}}(q^{r_{y_j}}, e_{y_j}) = N_{H_{\mathbf{1}}}(q^{r_{y_j}}, e_{y_j}).$$

Example 5.1 (continuation of Example 3.1). In the following, the number of self-dual G-invariant codes is found for different q s.t. $|G_1| \equiv |G_2| \equiv \cdots \equiv |G_t| \bmod p$.

$q \equiv 1 \bmod 3$, $q \equiv 4 \bmod 5$, and $3 \equiv 5 \bmod p$ (e.g., $q = 4$): Different types of cyclotomic residue classes are Type A $\{1_1, 1_2, 1_3, 1_4\}$; Type B $\{g_3^2, g_4^2, g_3^3, g_4^3\}$, $\{g_3, g_4, g_3^4, g_4^4\}$; and Type C $\{g_1, g_2\}$, $\{g_1^2, g_2^2\}$. So the number of self-dual G-invariant codes over F_q is $N_E(q,4)N(q,2)(N_H(q^2,2))^2$.

The number of self-dual G-invariant codes over F_q for other values of q can be calculated similarly as follows.

$q \equiv 1 \bmod 3$, $q \equiv 1 \bmod 5$, and $3 \equiv 5 \bmod p$ (e.g., $q = 16$): $N_E(q,4)\,(N(q,2))^3$.

$q \equiv 2 \bmod 3$, $q \equiv 2$ or $3 \bmod 5$, and $3 \equiv 5 \bmod p$ (e.g., $q = 2, 8$): $N_E(q,4)N_H(q^2,2)$ $N_H(q^4,2)$.

The values of $N_{E_{\mathbf{v}}}(q,l)$ and $N_{H_{\mathbf{v}}}(q^2,l)$ are not known for arbitrary $\mathbf{v}$. The following theorem allows computation of these quantities for certain cases.

THEOREM 5.5. *If either all components of $\mathbf{v} \in F_q^l$ are quadratic residues in F_q or all components are quadratic nonresidues in F_q, then* (1) $N_{E_{\mathbf{v}}}(q,l) = N_E(q,l)$ *and* (2) $N_{H_{\mathbf{v}}}(q^2,l) = N_H(q^2,l)$.

Proof. If all the components of $\mathbf{v}$ are quadratic nonresidues in F_q, then this vector can be divided by one of its components to get a scalar multiple of the vector, in which each component is a quadratic residue. So, it is sufficient to assume that the components of $\mathbf{v}$ are quadratic residues. Suppose $\mathbf{v} = (v_1, \ldots, v_l) = (s_1^2, \ldots, s_l^2)$.

We shall give a one-to-one correspondence between the $E_{\mathbf{v}}$-self-dual codes and the Euclidean self-dual codes to prove the first part of the result. Let $U \subseteq F_q^l$ be an $E_{\mathbf{v}}$-self-dual code of length l over F_q. Then it will be shown that the subspace $W \stackrel{\triangle}{=} \{(s_1a_1, \ldots, s_la_l) | \mathbf{a} = (a_1, \ldots, a_l) \in V\}$ is a Euclidean self-dual code. Suppose $(s_1a_1, \ldots, s_la_l), (s_1b_1, \ldots, s_lb_l) \in W$. Then, $\sum_{i=1}^{l} v_i a_i b_i = 0 \Rightarrow \sum_{i=1}^{l} (s_i a_i)(s_i b_i) = 0$. Thus, any two vectors in W are orthogonal w.r.t. the Euclidean inner product, and since the dimension of W is the same as the dimension of V, which is $\frac{l}{2}$, W is a Euclidean self-dual code. The second part follows similarly. □

COROLLARY 5.6. *If G is such that $|G_1| \equiv \cdots \equiv |G_t| \bmod p$ and there is a self-inverse cyclotomic coset $[x]^q \subseteq \mathcal{G}$ with e_x odd, then there is no self-dual G-invariant code over F_q.*

Proof. Both $N_{E_1}(q^{r_x}, e_x)$ and $N_{H_1}(q^{r_x}, e_x)$ are 0 when e_x is odd, and thus the result follows. □

COROLLARY 5.7. *If G is such that $|G_1| \equiv \cdots \equiv |G_t| \bmod p$ and the number t of orbits is odd, then there is no self-dual G-invariant code.*

Proof. The result follows by applying Corollary 5.6 to the cyclotomic residue class $\{0_j | j = 1, \ldots, t\}$. □

6. Minimum distance of G-invariant codes. Tanner used a BCH-like argument [14] to estimate minimum distance bounds from the parity check equations over an extension field. The same concept was used to get minimum distance bounds for quasi-cyclic codes from the transform domain description of F_q-linear cyclic codes over F_{q^m} [4]. A natural generalization of the results is given here. This can be used to guarantee some minimum distance by viewing the code as a shortened code of an abelian code. For s vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_s$ over F_{q^r} of lengths $n_1, n_2, \ldots, n_s$, respectively, let $\mathbf{v}_1 \boxtimes \mathbf{v}_2 \boxtimes \ldots \boxtimes \mathbf{v}_s$ denote the $n_1 \times n_2 \times \cdots \times n_s$ array, known as the Kronecker product of $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_s$, with $(i_1, i_2, \ldots, i_s)$th element $v_{1,i_1} v_{2,i_2} \ldots v_{s,i_s}$. The following theorem is available in [4] for the special case of $s = 1$. Here, *power of a vector* will mean the componentwise power, and I_l will denote the set $\{0, 1, \ldots, l-1\}$.

THEOREM 6.1. *Let r be an arbitrary positive integer and the components of each of the vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_s$ of lengths $n_1, n_2, \ldots, n_s$, respectively, be nonzero and distinct. If the components of a code $\mathcal{C}$ can be arranged in an $n_1 \times n_2 \times \cdots \times n_s$ array, and if S is a subset of $I_{q^r-1}^s$ such that for each $\mathbf{k} = (k_1, \ldots, k_s) \in S$, the array $\mathbf{v}_1^{k_1} \boxtimes \mathbf{v}_2^{k_2} \boxtimes \ldots \boxtimes \mathbf{v}_s^{k_s}$ is in the span of a set of parity check equations over F_{q^r}, then the minimum distance of the code is at least that of the s-dimensional cyclic code*

$$\mathcal{C}_c = \left\{ f(X_1, \ldots, X_s) \in \frac{F_{q^r}[X_1, \ldots, X_s]}{((X_1^{q^r-1} - 1), \ldots, (X_s^{q^r-1} - 1))} | f(\beta^{k_1}, \ldots, \beta^{k_s}) = 0 \right.$$
$$\left. \forall (k_1, \ldots, k_s) \in S \right\},$$

where β is a primitive element of F_{q^r}.

Proof. Suppose $\mathbf{v}_l = (v_{l,0}, v_{l,1}, \ldots, v_{l,n_l-1})$ with $v_{l,i} = \beta^{\lambda_{l,i}}$, where $\lambda_{l,i} \neq \lambda_{l,j}$ for $i \neq j$, $\forall l$. For any $\mathbf{a} \in \mathcal{C}$ with weight $\omega_H(\mathbf{a}) = d$, we construct

$$\mathbf{a}' = \sum_{(j_1,\ldots,j_s)\in I^s_{q^r-1}} a_{j_1,\ldots,j_s} X_1^{j_1} \ldots X_s^{j_s} \in \mathcal{C}_c$$

as

$$a'_{\lambda_{1,i_1},\ldots,\lambda_{s,i_s}} = a_{i_1,\ldots,i_s} \text{ for } (i_1,\ldots,i_s) \in I_{n_1} \times I_{n_2}, \times \cdots \times I_{n_s},$$
$$a'_{j_1,\ldots,j_s} = 0 \text{ when } (j_1,\ldots,j_s) \neq (\lambda_{1,i_1},\ldots,\lambda_{s,i_s}) \ \forall (i_1,\ldots,i_s) \in I_{n_1} \times I_{n_2}, \times \cdots \times I_{n_s}.$$

Clearly, $\omega_H(\mathbf{a}') = d$. Now,

$$\begin{aligned}
\mathbf{a} \in \mathcal{C} &\Rightarrow \sum_{i_1=0}^{n_1-1} \cdots \sum_{i_s=0}^{n_s-1} a_{i_1,\ldots,i_s} v_{1,i_1}^{k_1} \ldots v_{s,i_s}^{k_s} = 0 \ \forall \ (k_1,\ldots,k_s) \in S \\
&\Rightarrow \sum_{j_1=0}^{q^r-1} \cdots \sum_{j_s=0}^{q^r-1} a'_{j_1,\ldots,j_s} \beta^{j_1 k_1} \ldots \beta^{i_s k_s} = 0 \qquad \text{”} \\
&\Rightarrow \mathbf{a}' \in \mathcal{C}_c. \qquad \square
\end{aligned}$$

If $(x_1)^q, \ldots, (x_k)^q$ denote the distinct cyclotomic residue classes, then we know that any G-invariant code $\mathcal{C}$ is specified by the subspaces $V_{x_1}, \ldots, V_{x_k}$ of

$$F_{q^{r_{x_1}}}^{e_{x_1}}, \ldots, F_{q^{r_{x_k}}}^{e_{x_k}},$$

respectively, from which $A_{\widetilde{x_1}}, \ldots, A_{\widetilde{x_k}}$ take values. Now, each V_x , $x = x_1, \ldots, x_k$, can be considered a linear code over $F_{q^{r_x}}$ of length e_x. Thus, V_x is determined by a set of parity check equations. Suppose $\widetilde{x} = \{y_1, \ldots, y_l\}$, where $x = y_i$ for some i and $l = e_x$. Let $\sum_{i=1}^{l} c_i A_{y_i} = 0$ be a parity check equation of V_x. Then,

$$\sum_{y\in\mathcal{G}} \left(\sum_{i=1}^{l} c_i \Psi(y, y_i) \right) a_y = 0.$$

Clearly, this gives a parity check equation of $\mathcal{C}$ over $F_{q^{r_x}}$. The componentwise conjugate vectors of the parity check vectors obtained this way and the vectors in their span are also parity check vectors of the code.

Although Theorem 6.1 gives a way to get a minimum distance bound of any linear code, for which a set of parity check equations over an extension field is known, it is very difficult to know which arrangement of the code components, in how many dimensions, and what choice of $\mathbf{v}_l$ will give the maximum bound on the minimum distance. Even for the one-dimensional ($s = 1$) case it is very difficult to choose the best $\mathbf{v}_1$ and arrangement of code components because of the huge number of choices.

7. Quasi-abelian codes. For any abelian group G, the G-quasi-abelian codes of length $t|G|$ (which are submodules of $(F_qG)^t$) are closed under the action of G on the coordinates. So such codes are invariant under the coordinate permutations induced by the elements of G. However, this case has a more organized structure in that all the orbits of the coordinates under the action of G are of the same size $|G|$, and there are t such orbits. This raises the following natural reverse question: For a given abelian group G of permutations on code coordinates, when can we view

the G-invariant codes as G-quasi-abelian codes? The following theorem answers this question.

THEOREM 7.1. *The G-invariant codes are G-quasi-abelian codes, i.e., they can be viewed as submodules of $(F_qG)^t$ for some t if and only if $|G| = |G_k| \forall k$.*

Proof. The forward implication is obvious. If $|G| = |G_k|$, then $g \mapsto g^{(k)}$ is an isomorphism of G onto G_k. Thus, any G-invariant code can be viewed as a submodule of $(F_qG)^t$. □

Note that to see the G-invariant codes as G-quasi-abelian codes, $G_{k_1} \simeq G_{k_2} \forall k_1$, $k_2 \in I_t$, is not sufficient.

Example 7.1. Consider the group of permutations $G = \langle\{\sigma_1, \sigma_2\}\rangle$ of $\{1, 2, \ldots, 54\}$, where σ_1 and σ_2 are as shown in Figure 5. The solid lines with arrows represent the cycles of σ_1 and the dashed lines with arrows represent the cycles of σ_2. The order of the group G is 81, whereas the two groups G_1 and G_2 of restricted permutations are isomorphic to each other and of order 27. So, G-invariant codes cannot be seen as G-quasi-abelian codes in this case.

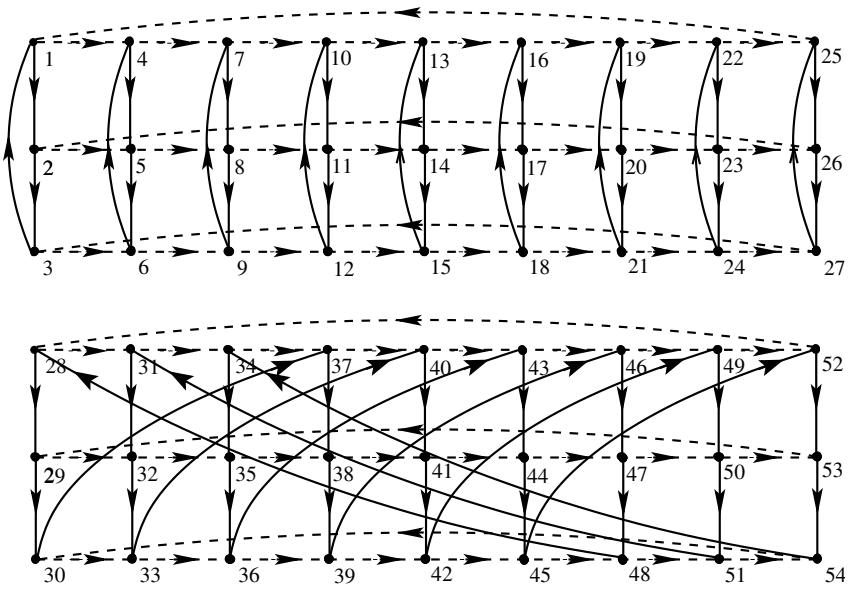


FIG. 5. *Cycle structures of σ_1 and σ_2 of Example* 7.1.

For G-quasi-abelian codes, we can index the coordinates in different orbits by copies $G_1, \ldots, G_t$ of the same group G. Thus, for any element $g \in G$, we have an element $g^{(i)} \in G_i$ for each i. So every residue class is of the form $\{g^{(1)}, \ldots, g^{(t)}\}$. We'll denote it by $\widetilde{g}$ instead of $\widetilde{g^{(i)}}$.

If, for a G-quasi-abelian code, symbols in some orbits form a set of information symbols and the symbols in the other orbits are the parity check symbols, then the code is called a *systematic G-quasi-abelian code*. For a systematic G-quasi-abelian code $\mathcal{C} \subseteq (F_qG)^t$ of dimension $k|G|$ $(k \leq t)$, without loss of generality we can assume that the first k orbits are information symbols and the rest are parity check symbols. Then there exist some $\mathbf{c}_{l,j} \in F_qG, l = 1, \ldots, t-k, , j = 1, \ldots, k$, such that each

codeword is of the form

$$\left(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k, \sum_{j=1}^{k} \mathbf{a}_j \mathbf{c}_{1,j}, \sum_{j=1}^{k} \mathbf{a}_j \mathbf{c}_{2,j}, \ldots, \sum_{j=1}^{k} \mathbf{a}_j \mathbf{c}_{t-k,j}\right) \in (F_q G)^t .$$

If the DFTs of $\mathbf{a}_j$ and $\mathbf{c}_{i,j}$ are denoted by $\mathbf{A}_j$ and $\mathbf{C}_{i,j}$, respectively, then each codeword in the transform domain is of the form

$$\left(\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_k, \sum_{j=1}^{k} \mathbf{A}_j \odot \mathbf{C}_{1,j}, \sum_{j=1}^{k} \mathbf{A}_j \odot \mathbf{C}_{2,j}, \ldots, \sum_{j=1}^{k} \mathbf{A}_j \odot \mathbf{C}_{t-k,j}\right) \in (F_q G)^t ,$$

where $\odot$ represents the componentwise product.

7.1. Decoding of systematic quasi-abelian codes. For a systematic G-quasi-abelian code with one information orbit, there are $\mathbf{c}_j \in F_q G$, $j = 1, \ldots, t-1$, such that every codeword is of the form $(\mathbf{a}, \mathbf{c}_1\mathbf{a}, \mathbf{c}_2\mathbf{a}, \ldots, \mathbf{c}_{t-1}\mathbf{a})$. For quasi-cyclic codes, i.e., for cyclic G and when $\mathbf{c}_j$ is a unit in $F_q G$ for $j = 1, \ldots, t-1$, Karlin [7] used alternate syndromes based on $\mathbf{c}_j$, $j = 1, \ldots, t-1$, and their inverses to gain considerable reduction in decoding operations. In the following, Karlin's approach is extended for systematic G-quasi-abelian codes with multiple information orbits. This is a two-step generalization of Karlin's algorithm: from quasi-cyclic codes to quasi-abelian codes and from one information orbit, i.e., one-generator codes to multiple generator codes.

For a systematic G-quasi-abelian code $\mathcal{C} \subseteq (F_q G)^t$ of dimension $k|G|$ $(k \leq t)$, there exist some $\mathbf{c}_{l,j} \in F_q G, l = 1, \ldots, t-k, j = 1, \ldots, k$, such that each codeword is of the form $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k, \mathbf{a}_{k+1}, \ldots, \mathbf{a}_t) \in (F_q G)^t$, where $\mathbf{a}_{k+i} = \sum_{j=1}^{k} \mathbf{a}_j \mathbf{c}_{i,j}$. We restrict our attention to the case where $\mathbf{c}_{i,j}, i = 1, \ldots, t-k, j = 1, \ldots, k$, are such that any $k \times k$ submatrix of the transposed generator matrix

$$M = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ \mathbf{c}_{1,1} & \mathbf{c}_{1,2} & \cdots & \mathbf{c}_{1,k} \\ \mathbf{c}_{2,1} & \mathbf{c}_{2,2} & \cdots & \mathbf{c}_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{c}_{t-k,1} & \mathbf{c}_{t-k,2} & \cdots & \mathbf{c}_{t-k,k} \end{pmatrix}$$

is invertible over $F_q G$. That is, any k orbits form a set of information symbols. For any subset $X \subseteq [1, t]$, the $|X| \times k$ submatrix comprising the corresponding rows of M is denoted by M_X. Similarly, $\mathbf{a}_X$ will denote the vector of length $|X|$ comprising the components $\mathbf{a}_i \in F_q G$, $i \in X$. We denote the complement $[1, t] \setminus X$ by $\bar{X}$. Thus, if we know k components of a codeword $\mathbf{a}$, i.e., $\mathbf{a}_X$ for some X of size k, then we can solve uniquely for the others as $\mathbf{a}_{\bar{X}} = M_{\bar{X}} M_X^{-1} \mathbf{a}_X$.

Suppose $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_t)$ is the transmitted codeword and the received vector is $\mathbf{a}' = (\mathbf{a}'_1, \mathbf{a}'_2, \ldots, \mathbf{a}'_t)$. Let $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_t) = \mathbf{a}' - \mathbf{a}$ denote the error vector. Suppose the code's known minimum distance is $2l + 1$ and a vector is received with at most l errors, that is, the Hamming weight of the error, $\sum_{i=1}^{t} wt_H(\mathbf{e}_i) \leq l$. Then

the transmitted vector is the only vector of the form

$$\left(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k, \sum_{j=1}^{k} \mathbf{a}_j \mathbf{c}_{1,j}, \sum_{j=1}^{k} \mathbf{a}_j \mathbf{c}_{2,j}, \ldots, \sum_{j=1}^{k} \mathbf{a}_j \mathbf{c}_{t-k,j}\right)$$

having distance from the received vector $\leq l$.

Given a received vector $\mathbf{a}'$, for each $X \subseteq [1,t]$ of size k a syndrome $S_X = M_{\bar{X}} M_X^{-1} \mathbf{a}'_X + \mathbf{a}'_{\bar{X}} = M_{\bar{X}} M_X^{-1} (\mathbf{a}_X + \mathbf{e}_X) + \mathbf{a}_{\bar{X}} + \mathbf{e}_{\bar{X}} = M_{\bar{X}} M_X^{-1} \mathbf{e}_X + \mathbf{e}_{\bar{X}}$ can be computed. Thus, given $\mathbf{e}_X$, $\mathbf{e}_{\bar{X}}$ can be calculated as $\mathbf{e}_{\bar{X}} = S_X - M_{\bar{X}} M_X^{-1} \mathbf{e}_X$. Now, if the error is of weight less than l, then there is at least one subset X of size k such that the weight of $\mathbf{e}_X$ is at most $\lfloor \frac{kl}{t} \rfloor$. Thus, if we presume an $\mathbf{e}_X$ of weight at most $\lfloor \frac{kl}{t} \rfloor$, and $wt_H\left(\mathbf{e}_X, S_X - M_{\bar{X}} M_X^{-1} \mathbf{e}_X\right) \leq l$, then $\mathbf{e}_X$ and $\mathbf{e}_{\bar{X}} = S_X - M_{\bar{X}} M_X^{-1} \mathbf{e}_X$ give the actual error.

Now, any $\mathbf{e}_X \in (F_q G)^{|X|}$ can be considered as a vector of length $|X||G|$ over F_q. If $\mathbf{e}_X^{(1)}, \mathbf{e}_X^{(2)} \in (F_q G)^{|X|}$ are such that $\mathbf{e}_X^{(1)} = \mathbf{e}_X^{(2)} g$ for some $g \in G$, then we call them equivalent. Let us call the equivalence classes the G-quasi-abelian equivalence classes. All the elements of an equivalence class have the same Hamming weight. If we compute $M_{\bar{X}} M_X^{-1} \mathbf{e}_X$ for one representative of an equivalence class, then for any $\mathbf{e}'_X = \mathbf{e}_X g$ in the same equivalence class, $M_{\bar{X}} M_X^{-1} \mathbf{e}'_X = g M_{\bar{X}} M_X^{-1} \mathbf{e}_X$ can be computed from $M_{\bar{X}} M_X^{-1} \mathbf{e}_X$ just by permuting its components.

Using these concepts, the decoding algorithm can be performed as follows.

1. For each subset $X \subseteq [1,t]$ of size k calculate S_X.
2. For $i = 0$ to $\lfloor \frac{kl}{t} \rfloor$
3. For each subset $X \subseteq [1,t]$ of size k
4. For each G-quasi-abelian equivalence class of Hamming weight i, take a representative $\mathbf{e}_X$. Compute $M_{\bar{X}} M_X^{-1} \mathbf{e}_X$.
5. For each $g \in G$
6. Compute $\mathbf{e}_{\bar{X}} = S_X - g M_{\bar{X}} M_X^{-1} \mathbf{e}_X$
7. Check if Hamming weight of $\mathbf{e}_{\bar{X}}$ is less than or equal to $t - i$. If so, take $(\mathbf{e}_X, \mathbf{e}_{\bar{X}})$ as the error and quit. Otherwise, continue with the loops.

The number of syndromes (in $(F_q G)^{t-k}$) calculated by this algorithm is $\binom{t}{k}$. If $k = 1$ and G is cyclic, then it specializes to the algorithm proposed by Karlin [7] and Heijnen and van Tilborg [6] for decoding systematic quasi-cyclic codes with a single row of circulants in the generator matrix, i.e., one-generator systematic quasi-cyclic codes. For $t = 2$, it further specializes to the single parity circulant case.

8. Discussion. The class of codes considered in this paper is a generalization of cyclic codes, quasi-cyclic codes, abelian codes, and quasi-abelian codes. All these special families of codes are defined as codes closed under one or more permutations of the code components. The algebraic structures of these special families of codes were investigated by different authors and, in all the cases, there seemed to exist some common structure. It is shown in this paper that such structures are not specific to those codes, but these structures are present in the family of G-invariant codes for any abelian group G of permutations with order of G relatively prime to q.

Also, a twofold extension of Karlin's decoding algorithm for quasi-cyclic codes is given. It is an extension from the case of one-generator systematic quasi-cyclic codes to arbitrary systematic quasi-cyclic codes and also from the case of quasi-cyclic codes to quasi-abelian codes. However, since the algebraic structure of G-invariant codes for any arbitrary abelian G (with order relatively prime to q) is only as complex as that

of quasi-cyclic codes and quasi-abelian codes, it would be interesting to see whether this decoding algorithm can be extended to cover this general class of codes.

The results of section 5 give as special cases all the results of [9] regarding existence and enumeration of self-dual quasi-cyclic codes. Theorem 5.4 gives the number of self-dual G-invariant codes in terms of the number of weighted self-dual codes and weighted Hermitian self-dual codes. Theorem 5.5 enables computation of these numbers in terms of the known numbers for some special cases of weight vectors. It remains an open problem to compute the values of $N_{E_{\mathbf{v}}}(q,l)$ and $N_{H_{\mathbf{v}}}(q,l)$ for arbitrary weight vector $\mathbf{v}$, and thus enable computation of the number of self-dual G-invariant codes for arbitrary abelian group G of permutations.

Acknowledgments. The authors are very grateful to the anonymous referees for their very careful reading of the manuscript and for their constructive comments towards improving the final version.

REFERENCES

[1] R. E. Blahut, *Algebraic Codes for Data Transmission*, Cambridge University Press, Cambridge, UK, 2003.
[2] J. Conan and G. Seguin, *Structural properties and enumeration of quasi-cyclic codes*, Appl. Algebra Engrg. Comm. Comput., 4 (1993), pp. 25–39.
[3] P. Delsarte, *Automorphisms of abelian codes*, Philips Res. Rep., 25 (1970), pp. 389–403.
[4] B. K. Dey and B. S. Rajan, *F_q-linear cyclic codes over F_{q^m}: DFT approach*, Des. Codes Cryptogr. to appear.
[5] B. K. Dey and B. S. Rajan, *DFT domain characterization of quasi-cyclic codes*, Appl. Algebra Engrg. Comm. Comput., 13 (2003), pp. 453–474.
[6] P. Heijnen and H. C. A. van Tilborg, *The decoding of binary quasi-cyclic codes*, in Communications and Coding, M. Darnell and B. Honary, eds., Research Studies Press, Taunton, UK, 1998, pp. 146–159.
[7] M. Karlin, *Decoding of circulant codes*, IEEE Trans. Inform. Theory, 16 (1970), pp. 797–802.
[8] K. Lally and P. Fitzpatrick, *Algebraic structure of quasi-cyclic codes*, Discrete Appl. Math., 111 (2001), pp. 157–175.
[9] S. Ling and P. Solé, *On the algebraic structure of quasi-cyclic codes* I*: Finite fields*, IEEE Trans. Inform. Theory, 47 (2001), pp. 2751–2760.
[10] P. Mathys, *Frequency domain description of repeated-root (cyclic) codes*, in Proceedings of the IEEE International Symposium on Information Theory, Trondheim, Norway, 1994, p. 47.
[11] V. Pless, *On the uniqueness of the Golay codes*, J. Combin. Theory Ser. A, 5 (1968), pp. 215–228.
[12] E. M. Rains and N. J. A. Sloane, *Self-dual codes*, in Handbook of Coding Theory, V. S. Pless and W. C. Huffman, eds., Elsevier Science, New York, 1998, pp. 177–294.
[13] B. S. Rajan and M. U. Siddiqi, *Transform domain characterization of Abelian codes*, IEEE Trans. Inform. Theory, 38 (1992), pp. 1817–1821.
[14] R. M. Tanner, *A transform theory for a class of group-invariant codes*, IEEE Trans. Inform. Theory, 34 (1988), pp. 752–775.
[15] S. K. Wasan, *Quasi Abelian codes*, Publ. Inst. Math. (Beograd) N.S., 21 (1977), pp. 201–206.

SIAM J. DISCRETE MATH.
Vol. 18, No. 1, pp. 19–29

IMPROVED COMPACT VISIBILITY REPRESENTATION OF PLANAR GRAPH VIA SCHNYDER'S REALIZER*

CHING-CHI LIN†, HSUEH-I LU†, AND I-FAN SUN†

Abstract. Let G be an n-node planar graph. In a visibility representation of G, each node of G is represented by a horizontal line segment such that the line segments representing any two adjacent nodes of G are vertically visible to each other. In the present paper we give the best known compact visibility representation of G. Given a canonical ordering of the triangulated G, our algorithm draws the graph incrementally in a greedy manner. We show that one of three canonical orderings obtained from Schnyder's realizer for the triangulated G yields a visibility representation of G no wider than $\lfloor \frac{22n-40}{15} \rfloor$. Our easy-to-implement $O(n)$-time algorithm bypasses the complicated subroutines for four-connected components and four-block trees required by the best previously known algorithm of Kant. Our result provides a negative answer to Kant's open question about whether $\lfloor \frac{3n-6}{2} \rfloor$ is a worst-case lower bound on the required width. Also, if G has no degree-three (respectively, degree-five) internal node, then our visibility representation for G is no wider than $\lfloor \frac{4n-9}{3} \rfloor$ (respectively, $\lfloor \frac{4n-7}{3} \rfloor$). Moreover, if G is four-connected, then our visibility representation for G is no wider than $n-1$, matching the best known result of Kant and He. As a by-product, we give a much simpler proof for a corollary of Wagner's theorem on realizers due to Bonichon, Le Saëc, and Mosbah.

Key words. visibility representation, planar graph algorithm, graph drawing, realizer, canonical ordering

AMS subject classifications. 05C62, 05C85, 68W35, 68U05, 68R10, 94C15

DOI. 10.1137/S0895480103420744

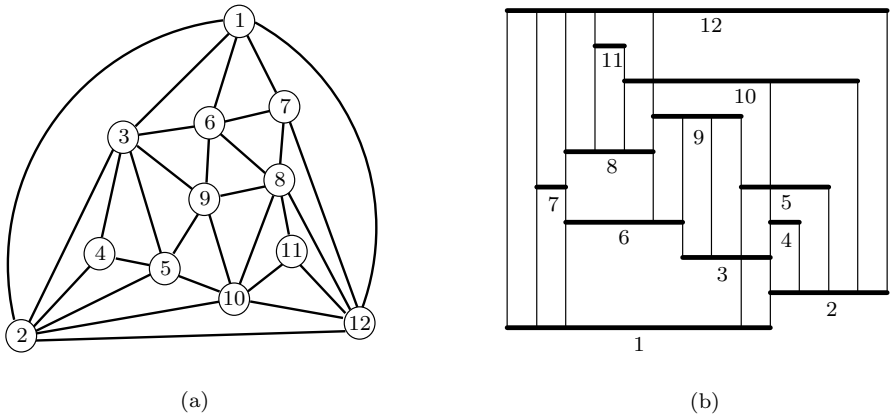
1. Introduction.

In a *visibility representation* of a planar graph G, the nodes of G are represented by nonoverlapping horizontal line segments, called *node segments*, such that the node segments representing any two adjacent nodes of G are vertically visible to each other. (See Figure 1.1.) Computing compact visibility representations of planar graphs is not only fundamental in algorithmic graph theory [31, 9] but also practically important in VLSI layout design [27].

Without loss of generality the input G can be assumed to be an n-node plane triangulation. Following the convention of placing the endpoints of node segments on the grid points, one can easily see that any visibility representation of G can be made no higher than $n-1$. Otten and van Wijk [25] gave the first known algorithm for visibility representations of planar graphs, but no width bound was provided for the output. Rosenstiehl and Tarjan [26], Tamassia and Tollis [30], and Nummenmaa [24] independently proposed $O(n)$-time algorithms whose outputs are no wider than $2n-5$. Kant [15, 17] improved the required width to at most $\lfloor \frac{3n-6}{2} \rfloor$ by decomposing G into its four-connected components and then combining the visibility representations of the four-connected components into a visibility representation of G. Kant left open the question of whether the upper bound $\lfloor \frac{3n-6}{2} \rfloor$ on the width is also a worst-case lower bound. In the present paper we provide a negative answer to Kant's question by

*Received by the editors January 4, 2003; accepted for publication (in revised form) November 22, 2003; published electronically July 2, 2004. A preliminary version of this paper appeared in *Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science*, H. Alt and M. Habib, eds., Lecture Notes in Comput. Sci. 2607, Springer-Verlag, Berlin, 2003, pp. 14–25.
http://www.siam.org/journals/sidma/18-1/42074.html

†Institute of Information Science, Academia Sinica, Taiwan, Republic of China (hil@iis.sinica.edu.tw, www.iis.sinica.edu.tw/~hil/). The research of the second author was supported in part by NSC grants NSC-91-2213-E-001-028 and NSC-92-2213-E-001-006.



(a) (b)

FIG. 1.1. *A plane triangulation and one of its visibility representations.*

presenting an algorithm that always produces a visibility representation for G whose width is at most $\lfloor \frac{22n-40}{15} \rfloor$.

Our algorithm, just like that of Nummenmaa [24], is based upon the concept of canonical ordering for plane triangulations. Specifically, our algorithm draws G incrementally in a greedy manner according to any given canonical ordering of G. An arbitrary canonical ordering of G may yield a visibility representation with width $2n - O(1)$. Rosenstiehl and Tarjan [26] even conjectured that selecting a node ordering to minimize the area of the corresponding visibility representation is NP-hard. We show that the required width can be bounded by $\lfloor \frac{22n-40}{15} \rfloor$ using the best one out of the three canonical orderings obtained from Schnyder's realizer [29, 28] for G. Our algorithm can easily be implemented to run in $O(n)$ time, bypassing the complicated subroutines of finding four-connected components and four-block trees [14] required by the best previously known algorithm of Kant [15, 17]. Also, for the case that G has no degree-three (respectively, degree-five) internal node, the output visibility representation of our algorithm is no wider than $\lfloor \frac{4n-9}{3} \rfloor$ (respectively, $\lfloor \frac{4n-7}{3} \rfloor$). Moreover, for the case that G is four-connected, the output visibility representation of our algorithm is no wider than $n - 1$, matching the best known result due to Kant and He [18, 19].

Schnyder's realizer [29, 28] for plane triangulation was invented for obtaining compact straight-line drawing of plane graphs. Researchers [5, 7, 8, 11, 12, 13, 16, 19] also obtained similar and other graph-drawing results using the concept of canonical ordering for triconnected plane graphs. Nakano [23] attempted to explain the hidden relation between these two concepts. Recently, Chiang, Lin, and Lu [4] presented a new algorithmic tool, the *orderly spanning tree*, that extends the concept of *st*-ordering [10] (respectively, canonical ordering and realizer) for plane graphs not required to be biconnected (respectively, triconnected and triangulated). The orderly spanning tree has been successfully applied to obtain improved results in compact graph drawing [4, 20, 3], succinct graph encoding with query support [4, 6], and design of compact routing tables [22]. Recently, Bonichon, Gavoille, and Hanusse [1] obtained the best known upper bounds on the numbers of distinct labeled and unlabeled planar graphs based on the *well orderly spanning tree*, a special case of the orderly spanning tree. As a matter of fact, we first successfully obtained the results of this paper using the orderly spanning tree and then found out that Schnyder's realizer suffices.

Our analysis requires an equality (see Lemma 2.3) relating the number of internal nodes in the three trees of a realizer R of G and the number of faces of G intersecting with all three trees of R. The equality was proved very recently by Bonichon, Le Saëc, and Mosbah [2] as a corollary of the so-called Wagner's theorem [32] on Schnyder's realizers. Their proof requires a careful case analysis for 32 different configurations. As a by-product, we give a much simpler proof for the equality without relying on Wagner's theorem on realizers.

The remainder of the paper is organized as follows. Section 2 gives the preliminaries. Section 3 describes and analyzes our algorithm. Section 4 discusses the tightness of our analysis. Section 5 concludes the paper with an open question.

2. Preliminaries. Let G be the input n-node *plane triangulation*, a planar graph equipped with a fixed planar embedding such that the boundary of each face is a triangle. Clearly, G has $2n-5$ internal faces. Let I consist of the internal nodes of G. Let $R=(T_1,T_2,T_3)$ be a *realizer* of G, which is obtainable in $O(n)$ time [28, 29]. That is, the following properties hold for R:

- The internal edges of G are partitioned into three edge-disjoint trees T_1, T_2, and T_3, each rooted at a distinct external node of G.
- The neighbors of each node v in I form six blocks U_1, D_3, U_2, D_1, U_3, and D_2 in counterclockwise order around v, where U_j (respectively, D_j) consists of the parent (respectively, children) of v in T_j for each $j\in\{1,2,3\}$.

For each index $i\in\{1,2,3\}$, let ℓ_i be the node labeling of G obtained from the counterclockwise preordering of the spanning tree $\bar{T}_i$ of G consisting of T_i plus the two external edges of G that are incident to the root of T_i. (Each $\bar{T}_i$ is, as a matter of fact, an orderly spanning tree [4] of G.) Let $\ell_i(v)$ be the label of v with respect to ℓ_i. For example, Figure 2.1 shows a realizer of the plane triangulation shown in Figure 1.1(a) with labeling ℓ_1. The counterclockwise preordering of $\bar{T}_2$ is $2,12,10,11,5,9,4,3,6,8,7,1$; and that of $\bar{T}_3$ is $12,1,7,8,11,10,9,6,3,5,4,2$.

LEMMA 2.1 (see, e.g., [4, 24, 6]). *The following properties hold for each index $i\in\{1,2,3\}$, where u_1 and u_2 are the nodes with $\ell_i(u_1)=1$ and $\ell_i(u_2)=2$.*

1. *The subgraph G_k of G induced by the nodes v with $1\le\ell_i(v)\le k$ is biconnected. The boundary of G_k's external face is a cycle C_k containing u_1 and u_2.*
2. *If v is the node with $\ell_i(v)=k$, then v is on C_k, and the neighbors of v in G_{k-1} form an interval with at least two nodes on the path $C_{k-1}-\{(u_1,u_2)\}$.*
3. *The neighbors of v in G form the following four blocks in counterclockwise order around v: (1) the parent of v in T_i, (2) the node set consisting of the neighbors u in $G-T_i$ with $\ell_i(u)<\ell_i(v)$, (3) the children of v in T_i, and (4) the node set consisting of the neighbors u in $G-T_i$ with $\ell_i(u)>\ell_i(v)$.*

A labeling ℓ of G that labels the external nodes by 1, 2, and n and satisfies Lemmas 2.1(1) and (2) is a *canonical ordering* of G (e.g., see [24, 16, 8]). Therefore, ℓ_1, ℓ_2, and ℓ_3 are all canonical orderings of G.

For each node v of G, let $\deg(v)$ denote the degree of v, i.e., the number of neighbors of v in G. For each index $i\in\{1,2,3\}$, let $\deg_i^-(v)$ (respectively, $\deg_i^+(v)$) be the number of neighbors u of v in G with $\ell_i(u)<\ell_i(v)$ (respectively, $\ell_i(u)>\ell_i(v)$). Clearly, we have $\deg(v)=\deg_i^-(v)+\deg_i^+(v)$. For each node v in I, let

$$\begin{aligned}\mathrm{score}_i(v)&=\min\{\deg_i^+(v),\deg_i^-(v)\};\\ \mathrm{score}(v)&=\mathrm{score}_1(v)+\mathrm{score}_2(v)+\mathrm{score}_3(v).\end{aligned}$$

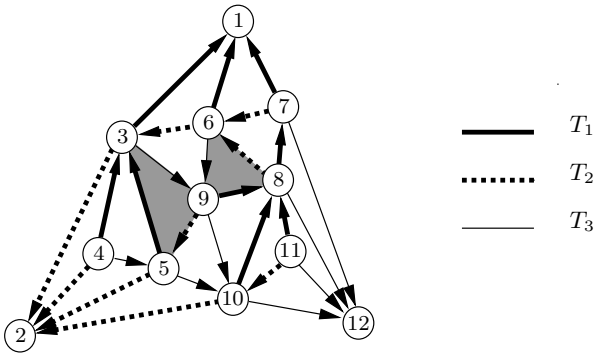


FIG. 2.1. *A realizer for the plane triangulation shown in Figure* 1.1(a)*, where* $(3, 5, 9)$ *and* $(6, 9, 8)$ *are the only two cyclic faces with respect to this realizer. The orientation of each edge is from a child to its parent in the corresponding tree.*

For example, if ℓ_1 is the labeling obtained from the tree T_1 consisting of the thick edges shown in Figure 2.1, then we have $\text{score}_1(v_8) = 2$, $\text{score}_1(v_9) = 1$, $\text{score}_1(v_{10}) = 2$, and $\text{score}_1(v_{11}) = 1$. Let

$$\text{score}_i = \sum_{v \in I} \text{score}_i(v).$$

Let $[\pi]$ be 1 (respectively, 0) if condition π is true (respectively, false). Let L_i consist of the leaves of T_i. For each node $v \in I$, let

$$\text{int}(v) = \sum_{i=1}^{3} [v \notin L_i].$$

Let B consist of the internal nodes v of G with $\text{int}(v) = 2$ and $\deg(v) = 5$. We have the following lemma.

LEMMA 2.2. *For each node* v *in* I*, we have* $\text{score}(v) \geq 3 + 2 \cdot \text{int}(v) - [v \in B]$.

Proof. By the definition of realizer and Lemma 2.1(3), it is clear that

$$\begin{aligned}
\text{score}_1(v) &= \min\{|D_3| + 2, |D_1| + |D_2| + 1\};\\
\text{score}_2(v) &= \min\{|D_1| + 2, |D_2| + |D_3| + 1\};\\
\text{score}_3(v) &= \min\{|D_2| + 2, |D_1| + |D_3| + 1\}.
\end{aligned}$$

We may assume without loss of generality that $|D_1| \geq |D_2| \geq |D_3| \geq 0$. One can verify the lemma by examining the inequality for all possible values $\{0, 1, 2, 3\}$ of $\text{int}(v)$. For example, if $v \in B$, then we know $\text{score}(v) = 2 + 2 + 2 = 6$ by $|D_1| = |D_2| = 1$ and $|D_3| = 0$. Also, if $\text{int}(v) = 2$ and $v \notin B$, then we have $\text{score}(v) \geq 2 + 2 + 3 = 7$ by observing $|D_1| \geq 2$, $|D_2| \geq 1$, and $|D_3| = 0$. The other cases can be verified similarly. □

An internal face of G is *cyclic* if its boundary intersects with all three trees T_1, T_2, and T_3. An internal face of G is *acyclic* if it is not cyclic. For example, in Figure 2.1, faces $(3, 5, 9)$ and $(6, 9, 8)$ are cyclic; all the other internal faces are acyclic. Let c be the number of cyclic faces of G. The following lemma was recently proved by Bonichon, Le Saëc, and Mosbah [2] in an equivalent form. Our alternative proof is much simpler.

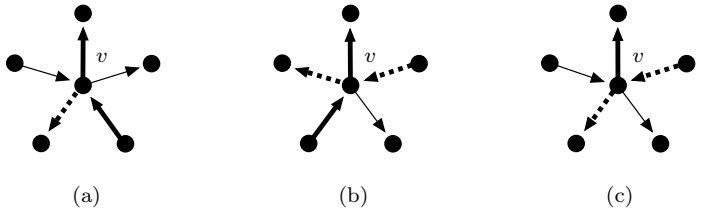


FIG. 2.2. *Three different kinds of nodes v in B.*

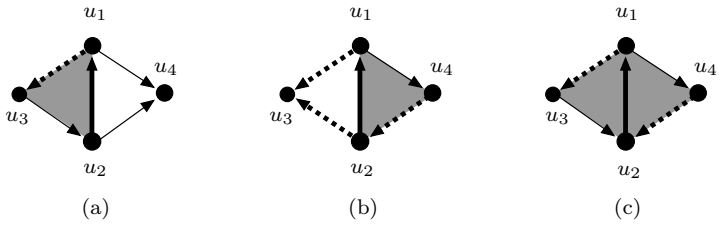


FIG. 2.3. *If u_1 and u_2 are two nodes of B that are adjacent in G, then at least one of faces (u_1, u_2, u_3) and (u_1, u_2, u_4) is cyclic.*

LEMMA 2.3 (see [2]). $\sum_{v \in I} \text{int}(v) = n + c - 4$.

Proof. For each index $i \in \{1, 2, 3\}$, let int_i be the number of internal nodes in T_i. Clearly, $\sum_{v \in I} \text{int}(v) = \sum_{i=1}^{3} \text{int}_i - 3$. For each node $v \in I$, let $p_i(v)$ denote the parent of v in T_i. For each $v \in L_i$, one can verify that $F_i(v) = (v, p_j(v), p_k(v))$ is an acyclic face of G, where $\{i, j, k\} = \{1, 2, 3\}$. By the orientations of the three edges, one can see that for each acyclic face F, there is exactly one pair (i, v) such that $v \in L_i$ and $F = F_i(v)$. Since G has $2n-5$ internal faces, the bijection between leaves and acyclic faces shows $\sum_{i=1}^{3} |L_i| = 2n - c - 5$. Therefore, $\sum_{i=1}^{3} \text{int}_i = 3(n-2) - (2n - c - 5) = n + c - 1$. □

LEMMA 2.4.

1. *If G has no degree-three internal nodes, then $\sum_{i=1}^{3} \text{score}_i \geq 5n - 15$.*
2. *If G has no degree-five internal nodes, then $\sum_{i=1}^{3} \text{score}_i \geq 5n - 17$.*
3. *If G is unrestricted, then $\sum_{i=1}^{3} \text{score}_i \geq \frac{23n}{5} - 16$.*

Proof. By Lemma 2.2 we know that if node v in I has degree more than 3, then $\text{score}(v) \geq 5$. By $|I| = n - 3$, statement 2.4 holds. It follows from Lemmas 2.2 and 2.3 that $\sum_{i=1}^{3} \text{score}_i = \sum_{v \in I} \text{score}(v) \geq \sum_{v \in I} 3 + 2 \cdot \text{int}(v) - [v \in B] = 3(n-3) + 2(n + c - 4) - |B|$. Therefore, $\sum_{i=1}^{3} \text{score}_i \geq 5n + 2c - |B| - 17$, which implies that (a) statement 2.4 holds (by observing that each node of B has degree five in G), and (b) statement 2.4 can be proved by ensuring $|B| - 2c \leq \frac{2n}{5} - 1$ as follows.

Let k be the number of connected components in the subgraph $G[B]$ of G induced by B. Since each of those $2n-5$ internal faces of G is incident to at most one connected component $G[B]$, and each connected component in $G[B]$ is incident to at least five internal faces of G, we have $5k \leq 2n - 5$.

Let u_1 and u_2 be two adjacent nodes of B such that (u_1, u_2) is an incoming edge of u_1. (That is, u_1 is the parent of u_2 in some tree T_i of R.) Let (u_3, u_1, u_2) and (u_4, u_1, u_2) be the two faces of G that contain edge (u_1, u_2). One can see that at least one of faces (u_1, u_2, u_3) and (u_1, u_2, u_4) is cyclic by verifying, with the assistance of Figure 2.2, that (a) both edges (u_1, u_3) and (u_1, u_4) have to be outgoing from u_1, and (b) at least one of edges (u_3, u_2) and (u_4, u_2) is incoming to u_2, as illustrated by Figure 2.3. Let F be an arbitrary spanning forest of $G[B]$, which clearly has $|B| - k$

edges. Each cyclic face contains at most two edges of F, and each edge of F is incident to at least one cyclic face. Thus, we have $|B| - k \leq 2c$. □

3. Our algorithm. Let ℓ_i be a given canonical ordering of the input n-node plane triangulation G. For each $k = 1, 2, \ldots, n$, let v_k be the node with $\ell_i(v_k) = k$ and let G_k be the subgraph of G induced by $v_1, v_2, \ldots, v_k$. Clearly, G_3 is a triangle and v_1, v_2, and v_n are the external nodes of G. Our algorithm initially produces a visibility representation of G_3, as shown in Figure 3.1, and then extends that into a visibility representation of $G = G_n$ in $n - 3$ iterations as follows: For each $k = 4, 5, \ldots, n$, the $(k-3)$rd iteration obtains a visibility representation of G_k from that of G_{k-1} by

1. extending the visibility representation of G_{k-1} in a greedy manner until the node segment of each neighbor of v_k is visible from above, and then
2. placing the shortest possible node segment representing v_k from above that yields a visibility representation of G_k.

For example, if G is as shown in Figure 1.1(a) and ℓ_i is as specified by the node labels, then the visibility representations for $G_3, G_4, \ldots, G_{11}$ are as shown in Figure 3.1, and the resulting visibility representation of $G = G_{12}$ is as shown in Figure 1.1(b). The correctness of our algorithm follows from the fact that ℓ_i is a canonical ordering of G, which therefore satisfies Lemma 2.1(1)).

A naive implementation of our algorithm takes $O(n^2)$ time. However, it is not difficult to implement our algorithm to run in $O(n)$ time using basic data structures like doubly linked lists to support $O(1)$-time operations such as determining whether a node segment is visible from above and inserting a new column of grid points. More specifically, one can represent each column of grid points by an object in the data structure. A sequence of consecutive columns can then be linked together through a doubly linked list such that the column to the right (respectively, left) of column c can be accessed by right_col(c) (respectively, left_col(c)). Clearly, one easily can insert a new column of grid points between columns c and right_col(c) in $O(1)$ time by calling subroutine INSERT_COLUMN(c). For each node v_i of G, we use left_end(i) (respectively, right_end(i)) to specify the column that contains the left (respectively, right) end of the node segment for v_i. We also use left_cover(i) (respectively, right_cover(i)) to specify how far from the left (respectively, right) the node segment for v_i is covered by other node segments from above. Clearly, whether the node segment for v_i is visible from above can be determined in $O(1)$ time by the condition left_cover(i) $\neq$ left_col(right_cover(i)). Let leftmost_nbr(i) (respectively, rightmost_nbr(i)) denote the leftmost (respectively, rightmost) neighbor of node v_i on the boundary of the external face of G_{i-1}. A linear-time implementation of the greedy algorithm is as shown in Figure 3.2.

THEOREM 3.1. *Any n-node plane triangulation G with $n > 3$ has an $O(n)$-time obtainable visibility representation whose width is at most*

1. $\lfloor \frac{4n-9}{3} \rfloor$ *if G has no degree-three internal nodes;*
2. $\lfloor \frac{4n-7}{3} \rfloor$ *if G has no degree-five internal nodes; or*
3. $\lfloor \frac{22n-40}{15} \rfloor$ *if G is unrestricted.*

Proof. By Lemma 2.4 and the fact that a realizer is obtainable in linear time, it suffices to show that the width of the output visibility representation by our algorithm is at most $3n - 8 - \sum_{v \in I} \text{score}_i(v)$. For each $k = 4, 5, \ldots, n$, consider the iteration that produces the visibility representation of G_k. Let v_j be any neighbor of v_k in G_k. In the first half of the iteration, if the node segment of v_j does not contain any grid point that is visible from above, then a new column of grid points is inserted

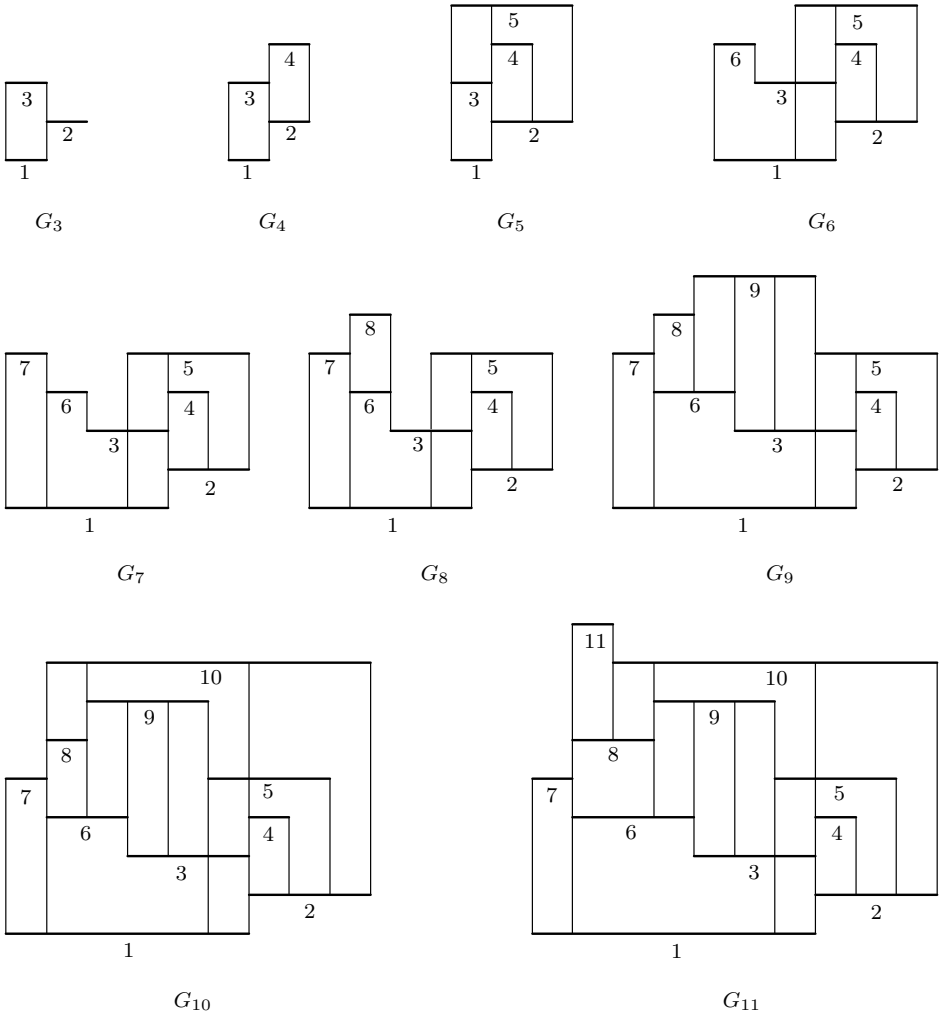


FIG. 3.1. *The intermediate steps of our algorithm for obtaining the visibility representation for the plane triangulation shown in Figure* 1.1(a) *with respect to the canonical ordering specified by its node labels.*

to ensure that the node segment for v_j is visible from above; otherwise, the number of grid points on the node segment of v_j that are visible from above stays the same. In the second half of the iteration, if v_k is the neighbor of v_j with the largest index, then the node segment of v_j can no longer be visible from above for the remaining iterations of our algorithm; otherwise, the number of grid points on the node segment of v_j that are visible from above decreases by exactly one. Moreover, the node segment of v_k contains at least $\deg_i^-(v_k)$ grid points that are visible from above in the resulting visibility representation of G_k. Therefore, in the first half of those $\deg_i^+(v_k)$ iterations, one for each neighbor v_ℓ of v_k with $\ell > k$, at most $\deg_i^+(v_k) - \text{score}_i(v_k)$ new columns of grid points are inserted. Note that $n > 3$ implies $\text{score}_i(v_3) \geq 1$. It follows that the resulting visibility representation for $G_n = G$ has width at most $2 + (-3 + \sum_{k=1}^{n-1} \deg_i^+(v_k)) - (1 + \sum_{k=3}^{n-1} \text{score}_i(v_k)) \leq 3n - 8 - \sum_{v \in I} \text{score}_i(v)$. □

```
    algorithm GREEDY
1       INITIALIZATION;
2       for i = 3 to n do {
3           for each neighbor v_j of v_i in G_i do
4               if left_cover(j) = left_col(right_cover(j)) then
5                   INSERT_COLUMN(left_cover(j));
6           let left_end(i) = left_col(right_cover(leftmost_nbr(i)));
7           let right_end(i) = right_col(left_cover(rightmost_nbr(i)));
8           let left_cover(i) = left_col(left_end(i));
9           let right_cover(i) = right_col(right_end(i));
10          let right_cover(leftmost_nbr(i)) = left_col(right_cover(leftmost_nbr(i)));
11          let left_cover(rightmost_nbr(i)) = right_col(left_cover(rightmost_nbr(i)));
12      }
13      ASSIGN_COORDINATES;

    subroutine INITIALIZATION
1       create two columns leftmost_col and rightmost_col with
            left_col(rightmost_col) = leftmost_col and
            right_col(leftmost_col) = rightmost_col;
2       call INSERT_COLUMN(leftmost_col) three times;
3       let temp_col = left_cover(1) = leftmost_col;
4       let temp_col = right_col(temp_col);
5       let left_end(1) = left_cover(2) = temp_col;
6       let temp_col = right_col(temp_col);
7       let right_end(1) = right_cover(1) = left_end(2) = temp_col;
8       let temp_col = right_col(temp_col);
9       let right_end(2) = temp_col;
10      let right_cover(2) = rightmost_col;

    subroutine ASSIGN_COORDINATES
1       let x(leftmost_col) = 0;
2       let temp_col = right_col(leftmost_col);
3       while temp_col ≠ rightmost_col do {
4           let x(temp_col) = x(left_col(temp_col)) + 1;
5           let temp_col = right_col(temp_col);
6       }
7       for i = 1 to n do
8           draw v_i between grid points (x(left_end(i)), i) and (x(right_end(i)), i);
```

FIG. 3.2. *A linear-time implementation of the greedy algorithm.*

Remark. As pointed out by an anonymous reviewer, an alternate and possibly quicker way to see the last inequality in the proof of Theorem 3.1 is by the fact that the output of our greedy algorithm is no wider than some constant plus

$$\begin{aligned}\sum_{v\in I}\max\left(\deg_i^+(v)-\deg_i^-(v),0\right) &= \sum_{v\in I}\left(\deg_i^+(v)-\min\left(\deg_i^-(v),\deg_i^+(v)\right)\right)\\ &= \sum_{v\in I}\left(\deg_i^+(v)-\mathrm{score}_i(v)\right).\end{aligned}$$

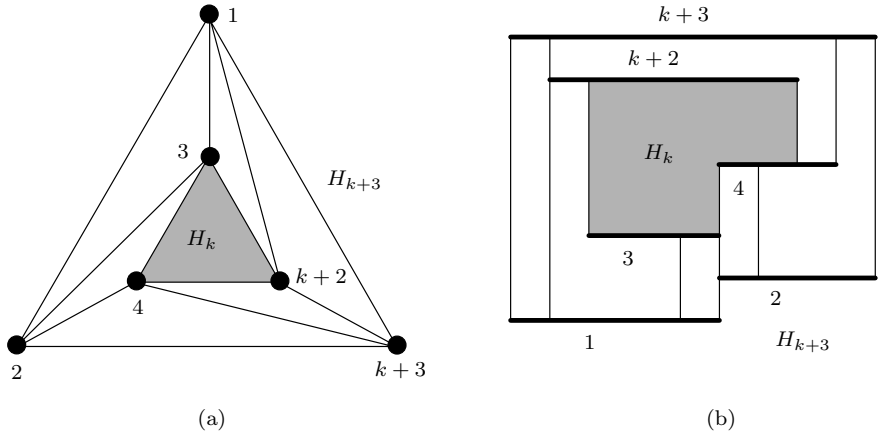


FIG. 3.3. *An example showing that our analysis on the required width is almost tight.*

The following result was first obtained by Kant and He [18, 19] and is based upon their linear-time algorithm for obtaining a canonical ordering ℓ_i for any n-node four-connected plane triangulation such that $\deg_i^+(v) \geq 2$ and $\deg_i^-(v) \geq 2$ hold for $n-4$ out of the $n-3$ internal nodes v of G. We can alternately prove the theorem in a much simpler way as follows: According to the proof of Theorem 3.1, the width of the output visibility representation by our algorithm is at most $3n-8-\sum_{v\in I}\text{score}_i(v) \leq n-1$.

THEOREM 3.2 (see [18, 19]). *If G is an n-node four-connected plane triangulation, then there is an $O(n)$-time obtainable visibility representation for G whose width is at most $n-1$.*

4. Near tightness of our analysis. The following lemma shows that our analysis on the required width is almost tight.

LEMMA 4.1. *For any $n > 3$, there exists an n-node plane triangulation H_n such that any visibility representation of H_n obtained by our algorithm with respect to any canonical ordering of H_n has width at least $\lfloor \frac{4n-9}{3} \rfloor$.*

Proof. We prove the lemma by induction on n. Let H_3 (respectively, H_4 and H_5) be a plane triangulation with 3 (respectively, 4 and 5) nodes. Clearly, any visibility representation of H_3 (respectively, H_4 and H_5) has width at least 2 (respectively, 3 and 4), so the lemma holds for $n = 3, 4, 5$. For each index $k \geq 3$, let H_{k+3} be the $(k+3)$-node plane triangulation obtained from H_k by adding three new external nodes and triangulating the faces as shown in Figure 3.3(a). By Lemma 2.1(1), if ℓ is a canonical ordering of H_{k+3}, then the ordering ℓ' with $\ell'(v) = \ell(v) - 2$ for each node v of H_k remains a canonical ordering of H_k. As illustrated in Figure 3.3(b), it is not difficult to see that the visibility representation for H_{k+3} produced by our algorithm with respect to any canonical ordering of H_{k+3} is at least four units wider than that of H_k produced by our algorithm with respect to any canonical ordering of H_k. Therefore, the lemma is proved. □

5. Concluding remarks. Very recently, Zhang and He [33] showed a linear-time algorithm that produces a visibility representation with height no more than $\lceil \frac{15n}{16} \rceil$. It would be interesting to see if combining their techniques and ours could reduce the worst-case area of a visibility representation to significantly less than $\frac{22n^2}{15} - \Theta(n)$.

Whether our upper bound $\frac{22n}{15} - \Theta(1)$ on the required width is worst-case optimal remains open. We conjecture that any n-node plane graph G admits an st-ordering

with which the greedy algorithm produces a visibility representation for G that is no wider than $\frac{4n}{3} + O(1)$. We also believe that the worst-case width could be further reduced to $n + O(1)$ if the plane embedding of the input graph could be altered.

Acknowledgment. We thank the anonymous reviewers for their helpful comments.

REFERENCES

[1] N. Bonichon, C. Gavoille, and N. Hanusse, *An information-theoretic upper bound of planar graphs using triangulation*, in Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science, H. Alt and M. Habib, eds., Lecture Notes in Comput. Sci. 2607, Springer-Verlag, Berlin, 2003, pp. 499–510.

[2] N. Bonichon, B. Le Saëc, and M. Mosbah, *Wagner's theorem on realizers*, in Proceedings of the 29th International Colloquium on Automata, Languages, and Programming, P. Widmayer, F. Triguero, R. Morales, M. Hennessy, S. Eidenbenz, and R. Conejo, eds., Lecture Notes in Comput. Sci. 2380, Springer-Verlag, Berlin, 2002, pp. 1043–1053.

[3] H.-L. Chen, C.-C. Liao, H.-I. Lu, and H.-C. Yen, *Some applications of orderly spanning trees in graph drawing*, in Proceedings of the 10th International Symposium on Graph Drawing, S. G. Kobourov and M. T. Goodrich, eds., Lecture Notes in Comput. Sci. 2528, Springer-Verlag, Berlin, 2002, pp. 332–343.

[4] Y.-T. Chiang, C.-C. Lin, and H.-I. Lu, *Orderly spanning trees with applications to graph encoding and graph drawing*, in Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2001, pp. 506–515.

[5] M. Chrobak and G. Kant, *Convex grid drawings of* 3*-connected planar graphs*, Internat. J. Comput. Geom. Appl., 7 (1997), pp. 211–223.

[6] R. C.-N. Chuang, A. Garg, X. He, M.-Y. Kao, and H.-I. Lu, *Compact encodings of planar graphs via canonical ordering and multiple parentheses*, in Proceedings of the 25th International Colloquium on Automata, Languages, and Programming, K. G. Larsen, S. Skyum, and G. Winskel, eds., Lecture Notes in Comput. Sci. 1443, Springer-Verlag, Berlin, 1998, pp. 118–129.

[7] H. de Fraysseix, P. Ossona de Mendez, and P. Rosenstiehl, *On triangle contact graphs*, Combin. Probab. Comput., 3 (1994), pp. 233–246.

[8] H. de Fraysseix, J. Pach, and R. Pollack, *How to draw a planar graph on a grid*, Combinatorica, 10 (1990), pp. 41–51.

[9] G. Di Battista, R. Tamassia, and I. G. Tollis, *Constrained visibility representations of graphs*, Inform. Process. Lett., 41 (1992), pp. 1–7.

[10] S. Even and R. E. Tarjan, *Computing an st-numbering*, Theoret. Comput. Sci., 2 (1976), pp. 436–441.

[11] U. Fössmeier, G. Kant, and M. Kaufmann, 2*-visibility drawings of planar graphs*, in Proceedings of the 4th International Symposium on Graph Drawing, S. North, ed., Lecture Notes in Comput. Sci. 1190, Springer-Verlag, Berlin, 1997, pp. 155–168.

[12] X. He, *On floor-plan of plane graphs*, SIAM J. Comput., 28 (1999), pp. 2150–2167.

[13] X. He, *A simple linear time algorithm for proper box rectangular drawings of plane graphs*, J. Algorithms, 40 (2001), pp. 82–101.

[14] A. Kanevsky, R. Tamassia, G. Di Battista, and J. Chen, *On-line maintenance of the four-connected components of a graph*, in Proceedings of the 32nd Annual Symposium on Foundations of Computer Science, IEEE, Piscataway, NJ, 1991, pp. 793–801.

[15] G. Kant, *A more compact visibility representation*, in Proceedings of the 19th Workshop on Graph-Theoretic Concepts in Computer Science, J. van Leeuwen, ed., Lecture Notes in Comput. Sci. 790, Springer-Verlag, Berlin, 1994, pp. 411–424.

[16] G. Kant, *Drawing planar graphs using the canonical ordering*, Algorithmica, 16 (1996), pp. 4–32.

[17] G. Kant, *A more compact visibility representation*, Internat. J. Comput. Geom. Appl., 7 (1997), pp. 197–210.

[18] G. Kant and X. He, *Two algorithms for finding rectangular duals of planar graphs*, in Proceedings of the 19th Workshop on Graph-Theoretic Concepts in Computer Science, J. van Leeuwen, ed., Lecture Notes in Comput. Sci. 790, Springer-Verlag, Berlin, 1994, pp. 396–410.

[19] G. Kant and X. He, *Regular edge labeling of* 4*-connected plane graphs and its applications in graph drawing problems*, Theoret. Comput. Sci., 172 (1997), pp. 175–193.

[20] C.-C. Liao, H.-I. Lu, and H.-C. Yen, *Floor-planning via orderly spanning trees*, in Proceedings of the 9th International Symposium on Graph Drawing, P. Mutzel, M. Jünger, and S. Leipert, eds., Lecture Notes in Comput. Sci. 2265, Springer-Verlag, Berlin, 2002, pp. 367–377.

[21] C.-C. Lin, H.-I. Lu, and I.-F. Sun, *Improved compact visibility representation of planar graph via Schnyder's realizer*, in Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science, H. Alt and M. Habib, eds., Lecture Notes in Comput. Sci. 2607, Springer-Verlag, Berlin, 2003, pp. 14–25.

[22] H.-I. Lu, *Improved compact routing tables for planar networks via orderly spanning trees*, in Proceedings of the 8th International Conference on Computing and Combinatorics, O. H. Ibarra and L. Zhang, eds., Lecture Notes in Comput. Sci. 2387, Springer-Verlag, Berlin, 2002, pp. 57–66.

[23] C.-I. Nakano, *Planar drawings of plane graphs*, IEICE Trans. Inform. Systems, E83-D (2000), pp. 384–391.

[24] J. Nummenmaa, *Constructing compact rectilinear planar layouts using canonical representation of planar graphs*, Theoret. Comput. Sci., 99 (1992), pp. 213–230.

[25] R. Otten and J. van Wijk, *Graph representations in interactive layout design*, in Proceedings of the IEEE International Symposium on Circuits and Systems, 1978, pp. 914–918.

[26] P. Rosenstiehl and R. E. Tarjan, *Rectilinear planar layouts and bipolar orientations of planar graphs*, Discrete Comput. Geom., 1 (1986), pp. 343–353.

[27] M. Schlag, F. Luccio, P. Maestrini, D. T. Lee, and C. K. Wong, *A visibility problem in VLSI layout compaction*, in Advances in Computing Research, vol. 2, F. P. Preparata, ed., JAI Press, Greenwich, CT, 1985, pp. 259–282.

[28] W. Schnyder, *Planar graphs and poset dimension*, Order, 5 (1989), pp. 323–343.

[29] W. Schnyder, *Embedding planar graphs on the grid*, in Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 1990, pp. 138–148.

[30] R. Tamassia and I. G. Tollis, *A unified approach to visibility representations of planar graphs*, Discrete Comput. Geom., 1 (1986), pp. 321–341.

[31] R. Tamassia and I. G. Tollis, *Planar grid embedding in linear time*, IEEE Trans. Circuits Systems, 36 (1989), pp. 1230–1234.

[32] K. Wagner, *Bemerkungen zum vierfarbenproblem*, Jahresber. Deutsch. Math.-Verein, 46 (1936), pp. 26–32.

[33] H. Zhang and X. He, *Compact visibility representation and straight-line grid embedding of plane graphs*, in Proceedings of the 8th International Workshop on Algorithms and Data Structures, F. Dehne, J. R. Sackand, and M. Smid, eds., Lecture Notes in Comput. Sci. 2748, Springer-Verlag, Berlin, 2003, pp. 493–504.

SIAM J. DISCRETE MATH.
Vol. 18, No. 1, pp. 30–40

ON THE HARDNESS OF 4-COLORING A 3-COLORABLE GRAPH*

VENKATESAN GURUSWAMI† AND SANJEEV KHANNA‡

Abstract. We give a new proof showing that it is NP-hard to color a 3-colorable graph using just 4 colors. This result is already known [S. Khanna, N. Linial, and S. Safra, *Combinatorica*, 20 (2000), pp. 393–415], but our proof is novel because it does not rely on the PCP theorem, while the known one does. This highlights a qualitative difference between the known hardness result for coloring 3-colorable graphs and the factor n^{ϵ} hardness for approximating the chromatic number of general graphs, as the latter result is known to imply (some form of) PCP theorem [M. Bellare, O. Goldreich, and M. Sudan, *SIAM J. Comput.*, 27 (1998), pp. 805–915].

Another aspect in which our proof is novel is in its use of the PCP theorem to show that 4-coloring of 3-colorable graphs remains NP-hard even on bounded-degree graphs (this hardness result does not seem to follow from the earlier reduction of Khanna, Linial, and Safra). We point out that such graphs can always be colored using $O(1)$ colors by a simple greedy algorithm, while the best known algorithm for coloring (general) 3-colorable graphs requires $n^{\Omega(1)}$ colors. Our proof technique also shows that there is an $\varepsilon_0 > 0$ such that it is NP-hard to legally 4-color even a $(1 - \varepsilon_0)$ fraction of the edges of a 3-colorable graph.

Key words. graph coloring, PCP theorem, NP-hardness, hardness of approximation

AMS subject classifications. 68Q17, 05C15, 68R10

DOI. 10.1137/S0895480100376794

1. Introduction.

The graph coloring problem is to assign colors to vertices of a graph G such that no two adjacent vertices receive the same color; such a coloring is referred to as a *legal* coloring of G. The minimum number of colors required to perform a legal coloring is known as the *chromatic number* of G and is denoted $\chi(G)$. Graph coloring is a fundamental and extensively studied problem. In addition to its theoretical significance as a canonical NP-hard problem [17], it also arises naturally in a variety of applications including register allocation and timetable/examination scheduling.

Since coloring a graph G with the minimum number $\chi(G)$ of colors is NP-hard [17], we shift our focus to efficiently coloring a graph with an approximately optimum number of colors. Garey and Johnson [10] proved that it is NP-hard to approximate the chromatic number within a factor of $(2-\epsilon)$ for any $\epsilon > 0$. The best known algorithm for general graphs appears in [14] and colors a graph using a number of colors that is within a factor of $O(n(\log\log n)^2/\log^3 n)$ of the optimum (here and elsewhere, n refers to the number of vertices in the graph). There is strong evidence that one cannot do substantially better than this for general graphs, as the recent connection between probabilistically checkable proofs (PCPs) and hardness of approximations [7, 2, 1]

*Received by the editors August 8, 2000; accepted for publication (in revised form) October 28, 2003; published electronically July 2, 2004. A preliminary version of this paper appeared in the *Proceedings of the* 15*th Annual IEEE Conference on Computational Complexity*, Florence, Italy, 2000, pp. 188–197.

http://www.siam.org/journals/sidma/18-1/37679.html

†Department of Computer Science, University of Washington, Seattle, WA 98195 (venkat@cs.washington.edu). This work was done while the author was at the Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, and while he was visiting Bell Laboratories, Murray Hill, NJ.

‡Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104 (sanjeev@cis.upenn.edu). This work was done while the author was at Bell Laboratories, Murray Hill, NJ.

has led to strong hardness results for graph coloring also. The first such result was established by Lund and Yannakakis [20], who proved that chromatic number is hard to approximate within n^ϵ for some constant $\epsilon > 0$. Feige and Kilian [8], using the powerful PCP constructions due to Håstad [15], prove that, unless NP $\subseteq$ ZPP, one cannot approximate the chromatic number within a factor of $n^{1-\epsilon}$ for any constant $\epsilon > 0$.

However, *none* of these inapproximability results apply to the case when the input graph is k-colorable for some small constant k. Indeed, better performance guarantees are known in this case. For instance, a polynomial time algorithm that colors 3-colorable graphs using $\tilde{O}(n^{3/14})$ colors is known [23, 5, 16, 6]. It is known that for every constant h, there exists a large enough constant k such that coloring k-colorable graphs using kh colors is NP-hard [20, 19]; it is, however, not known if the order of quantifiers above can be reversed. Khanna, Linial, and Safra [19] proved that it is NP-hard to color a 3-colorable graph using only 4 colors, and to this date no improvement to this hardness result has been obtained.

Our results. Our main result in this paper is a new proof of the above result of [19]. Our result is stated formally below.

THEOREM 1.1 (main theorem). *It is NP-hard to color a* 3*-colorable graph with only* 4 *colors.*

The proof of Khanna, Linial, and Safra [19] uses the result that MAX CLIQUE is NP-hard to approximate within a factor of two, a consequence of the PCP theorem [2, 1]. An important distinguishing aspect of our proof is that it *does not require the PCP theorem* and only relies on the NP-hardness of the MAX CLIQUE problem. The hardness for 3-colorable graphs is the most intricate of the results in [19] and has not been improved upon or simplified ever since. Our work represents the most progress made on this important problem after the result of [19], and we hope our work will spur further improvements. Not relying on PCP machinery implies that this hardness result could have been obtained almost three decades ago, long before the arrival of the PCP theorem. In contrast, the hardness result (for approximating within n^ϵ, for example) for general graph coloring implies some form of PCP [3]; our result therefore also highlights a qualitative difference between the hardness of general graph coloring and the hardness of coloring 3-colorable graphs.[1]

As in essentially all previous reductions showing hardness of graph coloring, our reduction too starts from the hardness of INDEPENDENT SET (MAX CLIQUE): it transforms an instance G of INDEPENDENT SET to an instance H of graph coloring such that a large independent set in G translates into a small collection of (in our case, three) independent sets in H, which together cover all vertices in H. But in addition, our proof is based only on local gadgets and easily leads to the hardness of 4-coloring even bounded-degree instances of 3-colorable graphs, albeit only by resorting to the PCP theorem.

THEOREM 1.2. *There is a constant* Δ *such that given a* 3*-colorable graph with maximum degree at most* Δ*, it is NP-hard to color it using just* 4 *colors.*

Note that, since such graphs can be colored using $O(1)$ colors (in fact, $(\Delta + 1)$ colors) by a simple greedy algorithm, while the best algorithm for general 3-colorable

[1] From a strictly logical point of view, the PCP theorem is true, so every result implies it, including our hardness result for 4-coloring 3-colorable graphs. When we say a hardness of approximation result implies a nontrivial PCP, we mean that one can get such a PCP result, via a *simple* reduction from the inapproximability result, without going through the steps of the current complicated proof of the PCP theorem. We hope this does not cause any confusion for the reader.

graphs uses $n^{\Omega(1)}$ colors, this hardness result is stronger than that of Theorem 1.1. Another strengthening of Theorem 1.1, which the degree-bounded result enables us to deduce, is the following.

THEOREM 1.3. *There is a constant $\varepsilon_0 > 0$ such that it is NP-hard, given a graph G, to distinguish between the case when G is 3-colorable and when any 4-coloring of G miscolors at least an ε_0 fraction of its edges.*

Both of these results do not seem to follow from the proof technique of [19] and therefore appear to be new. Note that the latter claim also generalizes the result of Petrank [22] which shows that there is an $\varepsilon > 0$ such that it is NP-hard to legally color a $(1 - \varepsilon)$ fraction of the edges of a 3-colorable graph using only 3 colors.

Inapproximability results and PCPs. In light of our main result, it is natural to ask how far nonPCP techniques can go in proving hardness results for coloring 3-colorable graphs. It turns out that an inapproximability factor of $\Omega(\log n)$ does imply a nontrivial PCP verifier for languages in NP. This follows from a result of Blum [4] (see also [3]), which shows that if coloring a 3-colorable graph using $c \log n$ colors is hard for every constant c, then for every $\epsilon > 0$, it is hard to approximate MAX CLIQUE within a factor of $n^{1-\epsilon}$; using the "reversal" of the FGLSS connection presented in [3], this implies the PCP theorem (in fact a very strong version of it; see [3] for details).[2] However, it seems possible that any $o(\log n)$ hardness bound can be proved for coloring 3-colorable graphs without resorting to PCP techniques.[3]

Expanding the scope of our investigation, it is natural to ask which inapproximability results really require PCPs. It is known, for example, that PCPs are inherent in obtaining strong hardness results for approximating MAX SAT, MAX CLIQUE, chromatic number, and vertex cover. Concretely, inapproximability results for these problems imply, via a simple reduction, a nontrivial PCP verifier for NP languages. Recent work in [12] and [18] proves strong (in fact near-tight) inapproximability results for disjoint paths and longest path problems without requiring PCPs; prior results for these problems always began with the PCP theorem and yet turned out to be weaker. Together with our result, these raise similar questions about the hardness results for certain other fundamental problems like set cover, nearest codeword problem, shortest vector problem, etc. In each of these cases it is interesting to see if a reverse connection to PCPs exists or if PCPs are only an artifact of the current proof techniques.

Notation. We use the standard notation to denote graph-theoretic parameters. For a graph G, we denote by $\chi(G)$, $\alpha(G)$, $\omega(G)$, and $\theta(G)$ the chromatic number of G, the size of a largest independent set in G, the size of a largest clique in G, and the clique cover number of G (the minimum number of cliques to cover all the vertices of G), respectively. Clearly $\alpha(G) = \omega(\bar{G})$ and $\chi(G) = \theta(\bar{G})$, where $\bar{G}$ is the complement of the graph G.

Organization. We present the proof of our main theorem (Theorem 1.1) in section 2. Section 3 describes the hardness result for bounded-degree 3-colorable graphs and sketches the proof of Theorem 1.3.

[2] Since the reduction from 3-coloring to finding large cliques is only a Turing reduction, strictly speaking, we can only conclude that every language in NP Turing reduces to a language in a certain PCP class.

[3] Actually, such a hardness result *does* imply the existence of very good *covering PCPs*, a notion recently introduced in [13] for the purpose of studying minimization problems like coloring. Constructing a "good" covering PCP without resorting to the PCP theorem appears very difficult, so such a PCP-free hardness result for coloring 3-colorable graphs might be hard to come by.

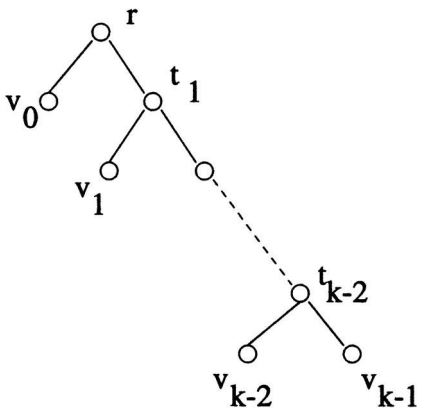


FIG. 1. *High-level structure of each T_i.*

2. Proof of the main theorem. We describe a reduction from the INDEPENDENT SET problem. Specifically, we start with instances of the following form: We are given a graph G along with a partition of the vertices of G into r cliques R_i, $1 \leq i \leq r$, each with exactly k vertices. Clearly, $\alpha(G) \leq r$. It is NP-hard to determine if $\alpha(G) = r$ on instances with this structure even when the partition into the R_i's is given as part of the input. This hardness even holds with $k = 3$—the standard reduction for NP-hardness of INDEPENDENT SET in fact produces such instances [11]. Thus the proof of Theorem 1.1 only requires us to consider the case $k = 3$. However, we will present here a construction for any arbitrary k. This is because the starting point for Theorems 1.2 and 1.3 are INDEPENDENT SET instances that are generated by PCP constructions, and k is a suitably large constant in this case.

Starting with such an instance G, we construct (in polynomial time) a graph H which will have the property that $\chi(H) = 3$ if $\alpha(G) = r$ and $\chi(H) \geq 5$ otherwise. This will clearly prove Theorem 1.1.

2.1. Overview of the reduction. Let G be a graph with vertices partitioned into r cliques R_i, $1 \leq i \leq r$, with exactly k vertices in each clique; i.e., let $R_i = \{v_{i,0}, \ldots, v_{i,k-1}\}$ for $1 \leq i \leq r$. The graph H is comprised of r "tree-like" structures, say $T_1, \ldots, T_r$, one for each clique R_i of G, together with a specific interconnection pattern between the leaves of the different tree structures based on the adjacency of vertices in G. The following are two key properties satisfied by the construction of the T_i's:

- Any 4-coloring of a T_i can be interpreted as "selecting" a unique vertex $v_{i,p}$ in the clique R_i of graph G (section 2.2).
- The edges between the T_i's are such that no 4-coloring is feasible if 2 vertices that are adjacent in G are selected from 2 different trees (section 2.3).

In other words, any 4-coloring of H can be interpreted as selecting a vertex in each of the r cliques R_i of G such that the selected vertices induce an independent set of size r in G, ensuring that if $\alpha(G) < r$, then in fact $\chi(H) > 4$. The other part, namely, that H is 3-colorable if $\alpha(G) = r$, will also be easily seen to hold for our reduction.

2.2. The structure of each T_i. Each T_i will have the structure of a binary tree with k leaves, $\{v_{i,j} : 0 \leq j < k\}$, one for each of the k vertices of G in the clique R_i (see Figure 1). It also has $(k-1)$ additional internal nodes $\{t_{i,j} : 0 \leq j < k-1\}$

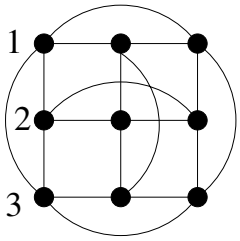


FIG. 2. *The basic template.*

with $t_{i,0}$ being the "root" r_i; by $t_{i,k-1}$ we mean the leaf node $v_{i,k-1}$. (The subscript i is omitted in Figure 1 for the sake of readability. The exact "shape" of the tree T_i is not important; any binary tree with k leaves and with all internal nodes having exactly two children will suffice for our purposes.) Each individual node of T_i is made up of the template shown in Figure 2. This basic template, denoted H_{basic}, may be viewed as a 3×3 grid such that the vertices in each row and in each column of the grid induce a 3-clique. The vertices in the first column of any such template are referred to as *ground vertices* and are in fact *shared* across all such templates in all the tree-structures. Since the ground vertices form a clique, any legal coloring will assign 3 *distinct* colors to them; we refer to these colors as 1, 2, and 3.

The connection pattern between the template at an internal node $t_{i,p}$ and its children, templates $v_{i,p}$ and $t_{i,p+1}$, is best understood by the schematic depicted in Figure 3. (Nodes P, L, and R will play the roles of $t_{i,p}$, $v_{i,p}$, and $t_{i,p+1}$, respectively.) In addition to the templates at these nodes, there are two 3-cliques that are connected to templates at $t_{i,p}, v_{i,p}$, and $t_{i,p+1}$ via appropriate edges. All nodes in the schematic are labeled as 3-tuples of the form $\langle xyz \rangle$, where $x, y, z \in \{1, 2, 3\}$. The edges (not shown) between the various vertices are given by the following simple rule: Two vertices are adjacent if and only if their labels *differ in all three coordinates*.

2.2.1. Node selection. A node of the tree is called *selected* if at least one of the three rows in its template has colors which, reading from left to right, form an even permutation of $\{1, 2, 3\}$ (i.e., the first row has colors 1, 2, 3; the second has 2, 3, 1; or the third has 3, 1, 2). Similarly, we say that a node is *not selected* if at least one of the three rows in its template has colors which, reading from left to right, form an odd permutation. It is easy to see that, in any legal 4-coloring, a node can never be simultaneously selected and not selected. Moreover, in any 4-coloring a node is always either selected or not selected.

2.2.2. Enforcing selection of a leaf node. Our goal now is to enforce that, for any legal 4-coloring of the tree-structure T_i, at least one leaf node is selected. Broadly speaking, our approach here will be to "hardwire" the selection of the root node and then introduce gadgets to ensure that, whenever a node is selected, one of its two children is selected as well. In other words, our construction propagates selection from the root to some leaf node. While one can imagine, at least for the case $k = 3$, that one can construct a "direct" 1-*out-of*-3 gadget, which will ensure that 1 of 3 nodes is always selected, this "top-down" approach works for any value of k and is also more modular and easier to present.

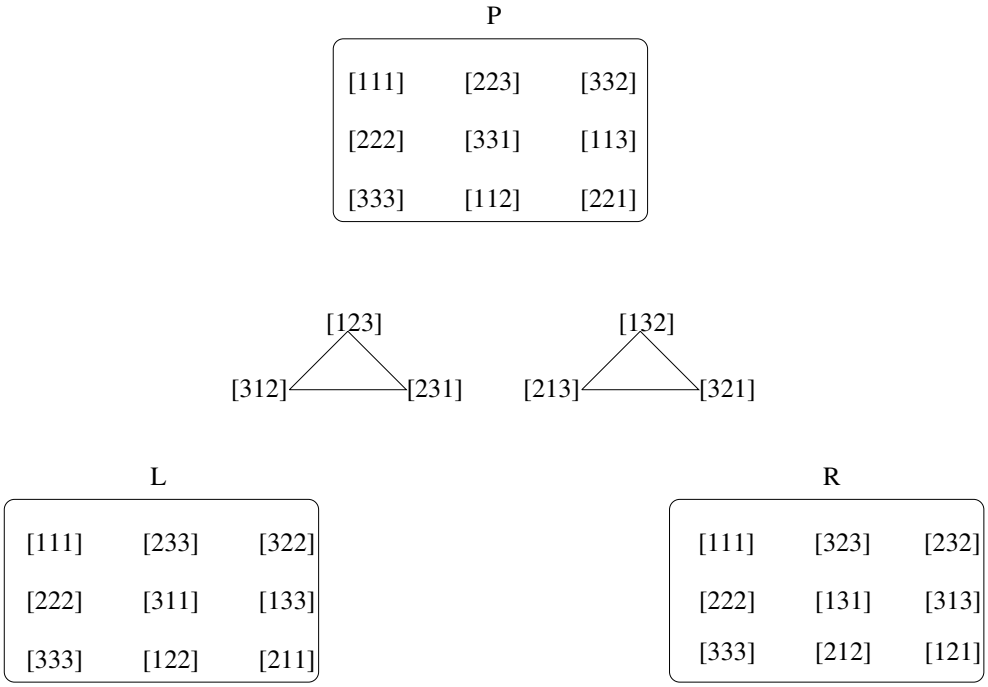


FIG. 3. *The connection pattern between the templates at a node and its children.*

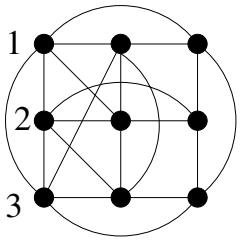


FIG. 4. *Enforcing selection at a root.*

Root selection. In each tree T_i, $1 \leq i \leq r$, we enforce selection of the root using the gadget shown in Figure 4. It is obtained by adding, for each $j \in \{1, 2, 3\}$, edges from the ground vertex colored j to the first vertex in row $(j \bmod 3 + 1)$ of the copy of H_{basic} at the root node r_i of T_i. This ensures that, in any 4-coloring of H, there will be 1 row of (each) root which will be selected (and hence the root itself will be selected). Indeed, there must exist 1 row whose vertices are not colored using 4, say, for concreteness, the third row. But since we added an edge between the ground vertex colored 2 and the first vertex in the third row, this vertex cannot be colored 2, and it follows that the third row of the root must be colored $(3, 1, 2)$, as desired.

Propagating the selection. Next, we show how the selection of a node in the tree can be propagated to at least one of the node's children. This ensures that, in each tree, at least one leaf node must be selected. Consider again the schematic in Figure 3 and assign the following interpretation to the node labels:

- Colors in the first coordinate of each node correspond to the situation in which $t_{i,j}$ is selected and these colors correspond to selection at $v_{i,j}$.
- Colors in the second coordinate of each node correspond to the situation in which $t_{i,j}$ is selected and these colors correspond to selection at $t_{i,j+1}$.
- Colors in the third coordinate of each node correspond to the situation in which $t_{i,j}$ is not selected and these colors correspond to both $v_{i,j}$ and $t_{i,j+1}$ not being selected.

It is tedious but straightforward to verify that, for any $l \in \{1,2,3\}$, if we assign colors 1, 2, and 3 to the nodes as specified by their lth coordinate, then a feasible coloring is formed. Moreover, for any choice of a leaf node to be selected in T_i, coloring the nodes along the unique root-leaf path as selected (i.e., coloring the 3 rows of the corresponding templates as $\{1,2,3\}$, $\{2,3,1\}$, and $\{3,1,2\}$), and the remaining nodes in T_i as not selected (i.e., coloring the 3 rows of the corresponding templates as $\{1,3,2\}$, $\{2,1,3\}$, and $\{3,2,1\}$), yields a legal 3-coloring of T_i. The following is thus evident for our construction.

LEMMA 2.1. *For each i, $1 \le i \le r$, and for all j, $0 \le j < k$, there is a 3-coloring of the vertices in the tree-structure T_i such that the leaf corresponding to $v_{i,j}$ is the only selected leaf in T_i.*

We can now establish the following key lemma.

LEMMA 2.2. *In any 4-coloring of a tree T_i, whenever an internal node is selected, one of its 2 children must be selected.*

Proof. Consider again the schematic of Figure 3, with P being the parent whose selection we argue implies the selection of one of its children L and R. We consider the following two cases.

Case 1. Both vertices in 1 of the pairs $\{\langle 112\rangle, \langle 113\rangle\}$, $\{\langle 221\rangle, \langle 223\rangle\}$, and $\{\langle 331\rangle, \langle 332\rangle\}$ receive color 4 in the 4-coloring of H.

Suppose it is the pair $\{\langle 331\rangle, \langle 332\rangle\}$ that receives color 4. Since P is selected, the third row of P must be colored $(3,1,2)$ in this case. We now claim that 1 of L and R will in fact be selected with their third row being colored $(3,1,2)$. Indeed, none of the vertices $\langle 122\rangle$, $\langle 211\rangle$, $\langle 212\rangle$, and $\langle 121\rangle$ (which are the third row nonground vertices of L and R) receive the color 4 as they are all adjacent to $\langle 331\rangle$ or $\langle 332\rangle$. Thus if neither L nor R is selected, $\langle 122\rangle, \langle 212\rangle$ get colored 2 and $\langle 211\rangle, \langle 121\rangle$ get colored 1. Now it is easy to see that each of the vertices $\langle 123\rangle$, $\langle 231\rangle$, and $\langle 312\rangle$ has color 1 as well as color 2 neighbors. Specifically, $\langle 123\rangle$ is adjacent to $\langle 211\rangle$ and $\langle 212\rangle$, $\langle 231\rangle$ is adjacent to $\langle 112\rangle$ and $\langle 122\rangle$, and $\langle 312\rangle$ is adjacent to $\langle 121\rangle$ and $\langle 221\rangle$ (recall that $\langle 112\rangle$ is colored 1 and $\langle 221\rangle$ is colored 2 since the third row of P is colored $(3,1,2)$). Thus, all 3 vertices must be colored either 3 or 4. But this is impossible because these 3 vertices form a clique. Therefore, L or R must be selected.

Similar arguments will hold if both vertices $\langle 112\rangle, \langle 113\rangle$ receive color 4 or if both vertices $\langle 221\rangle, \langle 223\rangle$ receive color 4. So it remains to consider the following case.

Case 2. At most 1 of the vertices in each of the pairs $\{\langle 112\rangle, \langle 113\rangle\}$, $\{\langle 221\rangle, \langle 223\rangle\}$, and $\{\langle 331\rangle, \langle 332\rangle\}$ receives color 4 in the 4-coloring of H.

In this case we first claim the following.

CLAIM 1. *At least one of the vertices $\langle 112\rangle, \langle 113\rangle$ gets colored 1, one of $\langle 221\rangle, \langle 223\rangle$ gets colored 2, and one of $\langle 331\rangle, \langle 332\rangle$ gets colored 3.*

To see this, note that P is selected, so we may assume, without loss of generality, that the third row of P is colored $(3, 1, 2)$. Thus, the above claim is trivially verified for colors 1 and 2 (since $\langle 112\rangle$ is colored 1 and $\langle 221\rangle$ is colored 2). Now if neither $\langle 331\rangle$ nor $\langle 332\rangle$ is colored 3, then in fact both must be colored 4 (since, for instance, $\langle 332\rangle$ cannot be colored either 1 or 2 because it is adjacent to both $\langle 111\rangle$ and $\langle 221\rangle$). But this contradicts the hypothesis of this case, and therefore our claim holds. □

We are now ready to finish the proof for Case 2. Suppose P is selected, but neither L nor R is selected. We will call a row of a node *pure* if none of its vertices are colored 4. Clearly, at least one of the rows of both L and R is pure. Since the entire gadget is totally symmetric, assume for definiteness that the third row of L is pure so that it is colored $(3, 2, 1)$ (recall that L is *not* selected, so it cannot be colored $(3, 1, 2)$). Now if the third row of R is pure, then it will also be colored $(3, 2, 1)$, and we will get a contradiction exactly as we obtained in the analysis of Case 1. So one of the first or second rows of R is pure; say, again without loss of generality, that the first row of R is pure so that it is colored $(1, 3, 2)$. The upshot of all this is that the vertices $\langle 122\rangle, \langle 211\rangle, \langle 323\rangle, \langle 232\rangle$ receive colors $2, 1, 3, 2$, respectively.

Now consider the vertex $\langle 231\rangle$. It is adjacent (among other vertices) to $\langle 122\rangle$ (which is colored 2), to $\langle 323\rangle$ (which is colored 3), and to both $\langle 112\rangle$ and $\langle 113\rangle$, one of which is colored 1 by Claim 1. It follows therefore that $\langle 231\rangle$ is colored 4. A similar argument shows that $\langle 123\rangle$ must also be colored 4—indeed $\langle 123\rangle$ is adjacent to $\langle 211\rangle$ (colored 1), to $\langle 232\rangle$ (colored 2), and to both $\langle 331\rangle$ and $\langle 332\rangle$, one of which is colored 3 by Claim 1. But now both $\langle 231\rangle$ and $\langle 123\rangle$ are colored 4 and are adjacent, which is a contradiction. This completes the analysis for Case 2, and the proof is now complete. □

2.3. The structure across the trees. We now specify how the nodes across different T_i's are connected. For every pair of leaf nodes $v_{i,p} \in T_i$ and $v_{j,q} \in T_j$ such that $v_{i,p}$ and $v_{j,q}$ are adjacent in G, we insert a gadget (actually a combination of more than one gadget) that prevents both of these leaf nodes from being selected simultaneously in any legal 4-coloring of H. Observe that this would immediately imply that if H is 4-colorable, then there must be an independent set of size at least r in G. This follows from Lemma 2.2 which shows that, in any 4-coloring of H, every tree has at least one selected leaf, and no two vertices of G corresponding to selected leaves can be adjacent in G.

The leaf-level gadget consists of two parts, as shown in Figures 5 and 6. Given two nodes, each a copy of the basic template H_{basic}, we use two kinds of gadgets. The first kind, shown in Figure 5, prevents both nodes from being selected because of the *same* row (for example, because the third row of both nodes is colored $(3, 1, 2)$)—we use three such gadgets, one for each row. It is easy to check that the gadget in Figure 5 is 3-colorable as long as at least one of the two third rows is colored $(3, 2, 1)$, but the gadget is not even 4-colorable if both the third rows are colored $(3, 1, 2)$.

The second kind of leaf-level gadget, shown in Figure 6, ensures that the two nodes are not both selected because of *different* rows, and this gadget is even simpler than the first. Once again it is completely straightforward to check that the gadget works as desired; for instance, for the gadget shown, there exists a valid 3-coloring as long as either the third row of the left-hand-side node is $(3, 2, 1)$ or the first row of the right-hand-side node is $(1, 3, 2)$ (i.e., at least one is not selected), but there is no valid 4-coloring if these rows are colored $(3, 1, 2)$ and $(1, 2, 3)$ (i.e., if both are selected).

The preceding discussion has thus established the following.

LEMMA 2.3. *If the graph H constructed as above is 4-colorable, then $\alpha(G) = r$.*

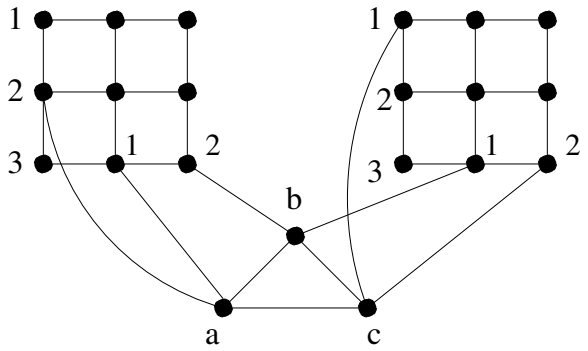


FIG. 5. *The leaf-level gadget: "Same row kind."*

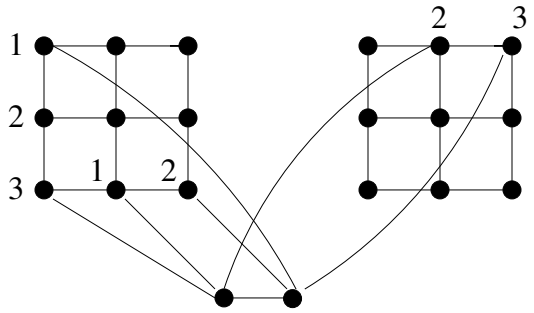


FIG. 6. *The leaf-level gadget: "Different rows kind."*

LEMMA 2.4. *If $\alpha(G) = r$, then H is 3-colorable.*

Proof. Let $K = \{v_{i,p_i} : 1 \leq i \leq r\}$ be an independent set of size r in G, where $0 \leq p_i < k$ for each i. By Lemma 2.1, we can legally color all the vertices of the tree-structures T_i using only three colors such that, for each tree T_i, the leaf corresponding to v_{i,p_i} is the only one that is selected. It remains only to color the vertices used in the leaf-level gadgets. By the argument above we can color the vertices of any leaf-level gadget using just three colors, provided at least one of the two leaf nodes it "connects" is not selected. But this condition is met for every leaf-level gadget in our case, since K is an independent set, and therefore there is no leaf-level gadget between any two of our selected leaf nodes. The entire graph H is thus 3-colorable. □

Theorem 1.1 now follows from Lemmas 2.4 and 2.3 since the construction of H can be clearly accomplished in polynomial time.

Tightness of our analysis. We point out here that the graph H constructed in the reduction above is *always* 5-colorable. Our analysis of the reduction is therefore tight in this regard. Indeed, by letting exactly one arbitrarily chosen leaf node of each tree-structure be selected, we can legally color all vertices in the tree-structures using three colors, say 1, 2, and 3. We claim it is now possible to legally color all the vertices in the leaf-level gadgets using only two more colors. Indeed, there are only two new nodes in the leaf-level gadgets of the "different rows kind" (Figure 6), and thus they can be colored 4, 5 arbitrarily. For the leaf-level gadgets of the "same row kind" (Figure 5), we need only worry about the situation where the two leaf nodes to which the gadget connects are *both* selected. This follows from our "completeness"

analysis (Lemma 2.4), where we showed that the vertices in the leaf-level gadgets can be properly colored using just colors 1, 2, and 3 when at most one of the two leaf nodes are selected. The case when both leaf nodes are selected is exactly the situation depicted in Figure 5, and it is easily seen that in this case the three new vertices in the leaf-level gadget concerned can be properly colored using the colors 3, 4, and 5.

3. Hardness for degree-bounded 3-colorable graphs. We now show that the result of Theorem 1.1 holds even if the input graph G has degree bounded by some constant Δ, thus establishing Theorem 1.2. Unlike Theorem 1.1, however, we do not see how to prove the result below without using the PCP theorem. Specifically, we use Proposition 3.1 below, which follows from the PCP theorem and MAX SNP-hardness of MAX 3-SAT instances, where each variable appears in at most a constant number of, say five, clauses [21].

PROPOSITION 3.1. *For every constant $t > 1$ there exist constants q, Δ such that, given a graph G whose vertices can be partitioned into r cliques each containing exactly q vertices and in which each vertex has degree at most Δ, it is NP-hard to distinguish between the cases $\alpha(G) = r$ and $\alpha(G) < r/t$.*

Proof of Theorem 1.2. We employ (essentially) the same reduction as in the proof of Theorem 1.1, except that we now start from a hard instance of INDEPENDENT SET, as in Proposition 3.1, with a "gap" (in independent set size) of $t = 24$. The graph H thus constructed will satisfy $\chi(H) = 3$ if $\alpha(G) = r$, while $\chi(H) \geq 5$ if $\alpha(G) < r$. By the nature of the reduction presented in section 2, and the fact that the maximum degree of G is at most Δ, it is easy to see that all vertices in H have very small degree except the three *ground vertices*, which are shared across all the r tree-like structures in H (that correspond to the r cliques in G). We get around this by simply using a distinct set of three ground vertices in each of the r tree-structures to give a new degree-bounded graph H'. By a pigeonhole argument, since there are only 24 different colorings of a (labeled) 3-clique using 4 colors, there are at least $r/24$ of the tree-structures whose ground vertices in rows $1, 2, 3$ are colored using the same three colors c_1, c_2, c_3; we just label these colors as $1, 2, 3$, respectively. Now, applying the argument used in the proof of Lemma 2.3 to the subgraph of G induced by the vertices in the $r/24$ cliques corresponding to these tree-structures, we conclude that, if H' is 4-colorable, then $\alpha(G) \geq r/24$. Of course in the case when $\alpha(G) = r$, the same coloring used to establish Lemma 2.4 with all copies of the ground vertices being colored as $1, 2, 3$ properly implies that H' is 3-colorable. Combining this reduction with Proposition 3.1, therefore, gives us our claimed result. □

It turns out that the above argument also suffices to establish Theorem 1.3.

Proof of Theorem 1.3. Use the same reduction to get a graph H as in the above proof, except now start from a hard instance of INDEPENDENT SET with a "gap" of $t = 48$. If n, m are, respectively, the number of vertices and number of edges in H, then we have $n = O(r)$, and since H is degree-bounded, $m = O(n)$. Thus $m = O(r) \leq c_0 r$ for some absolute constant c_0. Now define $\varepsilon_0 = 1/4c_0$. If a 4-coloring of H miscolors at most $\varepsilon_0 m$ edges, then since $\varepsilon_0 m \leq r/4$, there are at least $r/2$ tree-like structures such that they, and the leaf-level gadgets associated with them, are all legally colored using only 4 colors. Arguing as in the proof of Theorem 1.2, we can now conclude $\alpha(G) \geq r/48$. Thus when $\alpha(G) < r/48$, every 4-coloring of H legally colors at most $(1 - \varepsilon_0)$ fraction of the edges. □

Acknowledgment. We would like to thank Madhu Sudan for several useful discussions.

REFERENCES

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, *Proof verification and hardness of approximation problems*, J. ACM, 45 (1998), pp. 501–555.

[2] S. Arora and S. Safra, *Probabilistic checking of proofs: A new characterization of NP*, J. ACM, 45 (1998), pp. 70–122.

[3] M. Bellare, O. Goldreich, and M. Sudan, *Free bits, PCPs and nonapproximability—towards tight results*, SIAM J. Comput., 27 (1998), pp. 804–915.

[4] A. Blum, *Algorithms for Approximate Graph Coloring*, Ph.D. thesis, Laboratory for Computer Science, MIT, Cambridge, MA, 1991.

[5] A. Blum, *New approximation algorithms for graph coloring*, J. ACM, 41 (1994), pp. 470–516.

[6] A. Blum and D. R. Karger, *An $\tilde{O}(n^{3/14})$-coloring algorithm for 3-colorable graphs*, Inform. Process. Lett., 61 (1997), pp. 49–53.

[7] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy, *Interactive proofs and the hardness of approximating cliques*, J. ACM, 43 (1996), pp. 268–292.

[8] U. Feige and J. Kilian, *Zero-knowledge and the chromatic number*, J. Comput. System Sci., 57 (1998), pp. 187–199.

[9] M. Fürer, *Improved hardness results for approximating the chromatic number*, in Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science, IEEE, Piscataway, NJ, 1995, pp. 414–421.

[10] M. R. Garey and D. S. Johnson, *The complexity of near-optimal graph coloring*, J. ACM, 23 (1976), pp. 43–49.

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

[12] V. Guruswami, S. Khanna, R. Rajaraman, B. Shepherd, and M. Yannakakis, *Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems*, J. Comput. System Sci., 67 (2003), pp. 473–496.

[13] V. Guruswami, J. Håstad, and M. Sudan, *Hardness of approximate hypergraph coloring*, SIAM J. Comput., 31 (2002), pp. 1663–1686.

[14] M. M. Halldórsson, *A still better performance guarantee for approximate graph coloring*, Inform. Process. Lett., 45 (1993), pp. 19–23.

[15] J. Håstad, *Clique is hard to approximate within $n^{1-\epsilon}$*, Acta Math., 182 (1999), pp. 105–142.

[16] D. R. Karger, R. Motwani, and M. Sudan, *Approximate graph coloring using semidefinite programming*, J. ACM, 45 (1998), pp. 246–265.

[17] R. M. Karp, *Reducibility among combinatorial problems*, in Complexity of Computer Computations, Plenum Press, New York, 1972, pp. 85–103.

[18] A. Björklund, T. Husfeldt, and S. Khanna, *Approximating Longest Directed Path*, Report TR03-032, Electronic Colloquium on Computational Complexity (ECCC), http://eccc.uni-trier.de/eccc-reports/2003/TR03-032/index.html (2003).

[19] S. Khanna, N. Linial, and S. Safra, *On the hardness of approximating the chromatic number*, Combinatoria, 20 (2000), pp. 393–415.

[20] C. Lund and M. Yannakakis, *On the hardness of approximating minimization problems*, J. ACM, 41 (1994), pp. 960–981.

[21] C. H. Papadimitriou and M. Yannakakis, *Optimization, approximation and complexity classes*, J. Comput. System Sci., 43 (1991), pp. 425–440.

[22] E. Petrank, *The hardness of approximation: Gap location*, Comput. Complex., 4 (1994), pp. 133–157.

[23] A. Wigderson, *Improving the performance guarantee for approximate graph coloring*, J. ACM, 30 (1983), pp. 729–735.

SIAM J. DISCRETE MATH.
Vol. 18, No. 1, pp. 41–70

AN EAR DECOMPOSITION APPROACH TO APPROXIMATING THE SMALLEST 3-EDGE CONNECTED SPANNING SUBGRAPH OF A MULTIGRAPH*

HAROLD N. GABOW†

Abstract. This paper gives a 3/2 approximation algorithm for the smallest 3-edge connected spanning subgraph of an undirected multigraph. The previous best algorithm of Khuller and Raghavachari [*J. Algorithms*, 21 (1996), pp. 434–450] has approximation ratio 5/3. The algorithm of Cheriyan and Thurimella [*SIAM J. Comput.*, 30 (2000), pp. 528–560] achieves ratio 3/2 for simple graphs. Our approach, based on the close relationship between an ear decomposition of a 2-edge connected graph and 3-edge connected components, enables us to achieve running time $O(m\alpha(m,n))$.

Key words. approximation algorithms, network design, multigraphs, graph connectivity, edge connectivity, ear decomposition, depth-first search

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

DOI. 10.1137/S0895480102405476

1. Introduction.

Finding the smallest k-edge connected spanning subgraph is a natural problem in network design. Since the problem is NP-complete even for $k = 2$, a large number of approximation algorithms have been developed. This paper provides an algorithm for $k = 3$ that has improved accuracy and runs in almost-linear time.

We begin by surveying the most relevant past work. Throughout this paper all graphs are undirected and parallel edges are allowed. n and m always denote the number of vertices and edges of the given graph, respectively.

Khuller and Raghavachari [8] give a 1.85 approximation algorithm for the smallest k-edge connected spanning subgraph for any k. A simpler version of that algorithm [7] achieves ratio $2 - 1/k$ and runs in linear time. For $k = 3$ this gives ratio 5/3, the best previous accuracy bound for our problem. Fernandes [4] shows the 5/3 bound is tight. She also improves the general bound to 1.75 (1.7 for large enough k) when the graph is simple.

Cheriyan and Thurimella [3] give more accurate algorithms for simple graphs. The performance bound is $1 + 2/(k + 1)$. For $k = 3$ this is 3/2. The time for the algorithm for $k = 3$ is $O(\sqrt{n}m + n^2)$. As pointed out in [3], the analysis relies on properties of simple graphs that need not hold for multigraphs. Indeed, [5] exhibits a family of multigraphs for every $k \geq 2$ where the approximation ratio of the algorithm is 2.

For $k = 2$ Vempala and Vetta approximate the smallest k-edge connected spanning subgraph to the ratio 4/3 [10]. As in [3], their approach is based on matching.

We use a simpler depth-first search approach. The approximation ratio is $\leq 3/2$ and the running time is $O(m\alpha(m,n))$ where α is the inverse Ackermann function. The starting point of our approach is the observation that, in an ear decomposition of

*Received by the editors January 15, 2002; accepted for publication (in revised form) August 21, 2003; published electronically July 2, 2004. A preliminary, abbreviated version of this paper appeared in *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, Philadelphia, ACM, New York, 2002, pp. 84–93.
http://www.siam.org/journals/sidma/18-1/40547.html
†Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80309-0430 (hal@cs.colorado.edu).

a 2-edge connected graph, the ends of each ear are 3-edge connected. Cheriyan, Sebő, and Szigeti [2] use ear decomposition to approximate the smallest 2-edge connected subgraph.

Our lower bound is based on a generalization of the component lower bound introduced in [6]. We use a new operation,"breaking off," to strengthen that bound. We provide a simple proof that our algorithm has approximation ratio at most 14/9, and a more involved proof of the 3/2 accuracy bound. The latter requires combining three lower bounds (each one an instance of the component lower bound) and keeping track of the slack in those bounds. We give an example in which the algorithm has approximation ratio 17/12, i.e., 1/12 below our upper bound.

After the initial writing of this paper, we presented in [5] an improved analysis of the above-mentioned algorithm of Khuller and Raghavachari for the smallest k-edge connected spanning subgraph. The performance ratio of that algorithm is shown to be < 1.61 for any $k > 1$. To achieve this for odd values of k requires using the ear decomposition algorithm presented here as the base case. Furthermore, the analysis of [5] requires a stronger version of the performance ratio of the ear decomposition algorithm. The proof of the stronger version uses some parts of the argument for the 3/2 bound of this paper. For that reason the proof of the stronger version has been added as an appendix to this paper.

Section 2 gives basic facts that are used in the ear decomposition algorithm. Section 3 presents the algorithm. Section 4 gives a simple analysis showing that the approximation ratio is $\leq 14/9$. Section 5 refines the analysis to show the desired 3/2 bound. Section 6 shows the time bound. Appendix A gives the stronger version of the approximation ratio that is needed by [5] for general k. This section closes with our terminology and a background review. We use much of the notation of [2].

We often denote a singleton set $\{x\}$ by x. When we say a set is partitioned into subsets, each subset is required to be nonempty. For a family $\mathcal{S}$ of pairwise disjoint sets of vertices, the graph $G/\mathcal{S}$ is formed by contracting each set of $\mathcal{S}$ to a single vertex. We retain parallel edges but not loops.

We denote edges by juxtaposing the two vertices, e.g., vw. If vw is a tree edge or back edge of a depth-first search, the order of the vertices is significant: For a tree edge, v is the parent of w; for a back edge, v is a descendant of w. If the edge is not known to be a tree or back edge, the order is irrelevant.

In a graph $G = (V, E)$ with degree function d, if X and Y are disjoint sets of vertices, then $d(X, Y)$ is the total number of edges joining X and Y. $d(X)$ stands for $d(X, V - X)$. If H is a subgraph, then d_H denotes its degree function. (Sections 3 and 6 use the function d to denote depth in a tree, but this is clearly indicated.)

Two vertices are *k-edge connected* if they are joined by k edge-disjoint paths. Equivalently, the two vertices remain connected after deleting any $< k$ edges. This binary relation is an equivalence relation. A graph is k-edge connected if every two distinct vertices are k-edge connected. *k-ECSS* stands for k-edge connected spanning subgraph. For any k,

$$\varepsilon_k = \text{ the minimum number of edges in a } k\text{-ECSS}.$$

Throughout this paper we abbreviate ε_3 to ε.

We assume paths are simple, but they can be open or closed. A closed path has a distinguished vertex that plays the role of both endpoints. For a path P, $I(P)$ denotes the internal vertices of P, i.e., all the vertices except the endpoints. The symbol P denoting a path may reference the vertex set or the edge set of the path, as determined by context.

An *ear decomposition* of a graph is a partition of the edges into paths P_i, $i = 0, \ldots, q$, such that P_0 is a single vertex r and, for $i > 0$, each P_i has its ends and no other vertices in common with previous paths, i.e., $V(P_i) \cap (\bigcup_{j=0}^{i-1} V(P_j)) = V(P_i) - I(P_i)$ for $i = 1, \ldots, q$. A graph is 2-edge connected if and only if it has an ear decomposition [11]. For any vertex v the first ear containing v is denoted P_v. Any $v \neq r$ has $v \in I(P_v)$ (and for $v = r$, $I(P_r) = \emptyset$). An ear is *short* if its length is one, i.e., it has no internal vertices, otherwise the ear is *long*.

Given a spanning tree, a nontree edge *covers* every edge in its fundamental circuit. In a rooted tree we distinguish between *ancestor* and *proper ancestor*, the former relation being reflexive and the latter irreflexive. These distinctions hold similarly for *descendant* and *proper descendant*.

Khuller and Vishkin [9] define a *(dfs) tree carving* for any depth-first spanning tree T as follows. Do a bottom-up traversal of T, constructing a set of back edges B according to the following rule: When backing up from a vertex v to its parent p, if the tree edge pv is not covered by an edge of B, then add to B the back edge c that goes from a descendant of v to a vertex closest to the root. Edge c exists and covers pv, assuming G is 2-edge connected.

Each tree edge pv that forces a back edge to be added to B is a *carving edge*. Deleting the carving edges from T gives a forest called the dfs tree carving. Define

$$\gamma = \text{the number of carving edges.}$$

Khuller and Vishkin show that for any k,

$$\varepsilon_k \geq k\gamma.$$

This follows from the fact that no back edge covers two carving edges.

2. Basic facts. We increase the edge-connectivity by one using the following proposition, a slight strengthening of Lemma 3.1 of [8]. Let G be k-edge connected. Let K be a $(k-1)$-ECSS of G. Let $\mathcal{S}$ be a partition of V, each set of which is k-edge connected in K (i.e., any two vertices of the same set have $\geq k$ edge-disjoint paths between them). Let F be a maximal spanning forest of the graph $(G - K)/\mathcal{S}$.

PROPOSITION 2.1. *$K + F$ is k-edge connected.*

The next lemma gives our basic relation between an ear decomposition and 3-edge connectedness. (See Figure 1.) Consider an ear decomposition P_i, $i = 0, \ldots, q$, of a 2-edge connected graph. For any vertex v let the "closure" $Cl(v)$ be the smallest subset of $\{P_i\}$ that defines an ear decomposition of a (not necessarily spanning) subgraph containing v. (Closure can be defined inductively: For $P_0 = \{r\}$, $Cl(r) = P_0$. For $v \neq r$, if P_v goes from a to z, then $Cl(v) = \{P_v\} \cup Cl(a) \cup Cl(z)$.)

LEMMA 2.2. *Consider an ear decomposition of a 2-edge connected graph.*

(i) *The two endpoints of an ear are 3-edge connected.*

(ii) *If an ear has endpoints a and z, then each endpoint of an ear in $Cl(a) \oplus Cl(z)$ is 3-edge connected with a and z.*

Proof. Let P be an ear joining a and z.

(i) Let H be the subgraph $Cl(a) \cup Cl(z)$. a and z are 2-edge connected in H. Hence H contains two edge-disjoint az-paths. No edge of P belongs to H. This makes three edge-disjoint az-paths.

(ii) By symmetry it suffices to show that an arbitrary endpoint v of an ear in $Cl(a) - Cl(z)$ is 3-edge connected with z. Let H be the 2-edge connected graph $Cl(z) \cup Cl(v)$. The ears in $Cl(a) - Cl(v)$ contain a path Q from v to a. Combining

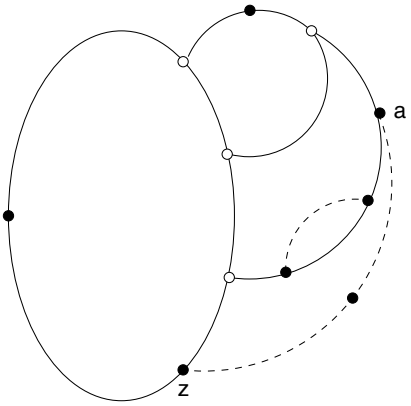


FIG. 1. *Illustration of Lemma* 2.2: $Cl(a)$ *includes the* 3 *solid ears but not the* 2 *dashed ears. The* 4 *hollow vertices are the endpoints of ears in* $Cl(a) \oplus Cl(z)$ *and so are* 3*-edge connected with* a *and* z.

Q with the given ear P gives a path from v to z that is edge-disjoint from H. As in part (i), v and z are 3-edge connected. □

Our algorithm forms an ear decomposition based on depth-first search. Let G be 2-edge connected and let T be a dfs tree of G. We shall construct ears consisting of a tree path followed by a back edge. We use the triple of vertices a, y, z to identify the ear consisting of the tree path from a to y followed by the back edge yz. Here z is an ancestor of a, which is an ancestor of y. (If $a = y$, the ear is short.)

To construct the ear decomposition for T let the first ear be r, the root of T. Suppose we have constructed a number of ears that collectively contain vertices $X \subset V$. Choose a tree edge ab with $a \in X$, $b \notin X$. Choose a back edge yz that covers ab. (yz exists since G is 2-edge connected.) The next ear is defined by the triple a, y, z. Adding this ear enlarges X by the vertices in the tree path from a to y. Repeat this step until $X = V$. Finally, make any back edge that is not yet in an ear into a short ear.

For the rest of this paper all ear decompositions are constructed in this manner. We use Figure 2 to illustrate various concepts involving ear decompositions.

We focus on the *first vertex* of an ear, i.e., vertex a in a, y, z. For each vertex x let $f(x)$ be the first vertex of ear P_x. f is represented by F, a tree on vertex set V: The root of F is r. The parent of a vertex $x \neq r$ is $f(x)$. In Figure 2 the first three proper ancestors of j in F are i, d, and a.

In this paper we use both the dfs tree T and the first vertex tree F. By default all tree terminology refers to T. For instance, we use "ancestor" (referring to T), "ancestor in T" (when there is danger of confusion), and "ancestor in F." r always denotes the root of T (r is also the root of F). It is easy to see that the proper descendants of x in F are the vertices that, in T, descend from a child x' of x with $x' \notin P_x$.

COROLLARY 2.3. *For an ear* a, y, z*, vertex* z *is* 3*-edge connected with every vertex that, in* F*, is an ancestor of* a *but not* $f(z)$.

Remark. It is possible that a is the only ancestor satisfying the conditions of the corollary.

Proof. If p is the parent of vertex x in T, then obviously $Cl(p) \subseteq Cl(x)$ (equality holds if $p \neq f(x)$). Iterating shows that any ancestor b of x has $Cl(b) \subseteq Cl(x)$. Thus

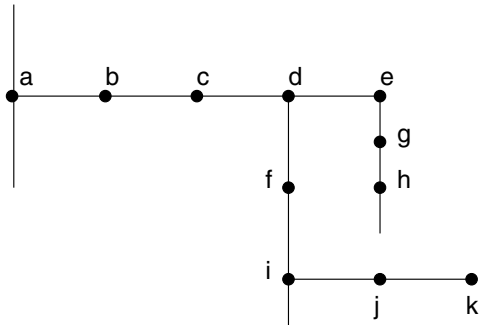


FIG. 2. *Schematic figure illustrating four long ears. Each horizontal or vertical line represents the tree path of one ear. Back edges of ears are not drawn. For example, one ear consists of the path from a to e followed by a back edge from e to an ancestor of a.*

in the statement of the corollary, $Cl(z) \subseteq Cl(a)$. In fact we get $Cl(a)$ from $Cl(z)$ by adding ears whose first vertices are the vertices that, in F, are proper ancestors of a but not proper ancestors of z. (Example: In Figure 2 we get $Cl(k)$ from $Cl(c)$ by adding two ears, with first vertex d and i, respectively.) Lemma 2.2 now implies the corollary. □

Garg, Santosh, and Singla [6] introduced the "component lower bound" for 2-ECSS. It was refined in [2]. We use the following generalization to k-ECSS. For any graph H let $c(H)$ be the number of connected components of H.

LEMMA 2.4 (component lower bound). *For any integer k let $G = (V, E)$ be k-edge connected. For any partition $\mathcal{S}$ of V, $\varepsilon_k \geq (k/2) \sum_{S \in \mathcal{S}} c(G - S)$.*

Remark. The degree lower bound $\varepsilon_k \geq kn/2$ amounts to the special case of the lemma where each $S \in \mathcal{S}$ consists of one vertex. The tree carving lower bound $\varepsilon_k \geq k\gamma$ is the special case where each S is a set of the tree carving.

Proof. Let H be a k-ECSS with ε_k edges. For any set S, each connected component of $G - S$ is joined to S by $\geq k$ edges of H. Hence

$$2\varepsilon_k = \sum_{v \in V} d_H(v) = \sum_{S \in \mathcal{S}} \sum_{v \in S} d_H(v) \geq \sum_{S \in \mathcal{S}} kc(G - S). \qquad \square$$

The rest of this section presents a method for strengthening the component lower bound. It is only used in section 5, so readers interested in just the 14/9 upper bound can skip this material.

We first give the idea and then a formalization. Suppose a set $S \subseteq V$ can be partitioned into sets S_0, S_1 so that ≤ 1 connected component of $G - S$ is adjacent to both S_0 and S_1. Then

$$c(G - S_0) + c(G - S_1) \geq c(G - S) + 1.$$

This follows since, for $i = 0$ or 1, a component of $G - S$ that is adjacent only to S_i contributes to $c(G - S_i)$. So ≤ 1 component of the right-hand side is not counted by the left-hand side. In addition, each $G - S_i$ has a component containing a vertex of S_{1-i}, which is also counted by the left-hand side.

We now define a configuration in which this principle can be applied repeatedly.

DEFINITION 2.5. *Consider a partition of a set $S \subseteq V$ into sets S_i, $i = 0, \ldots, h$, where each index i, $0 < i \leq h$ has a "parent" index $p(i)$, $0 \leq p(i) < i$. An edge joining S_i and $S_{p(i)}$ is a* bridging edge. *A component of $G - S$ that is adjacent to both S_i and*

$S_{p(i)}$ but no other set S_j is a bridging component. *The partition is an* enhancement *(of S) if the following two conditions hold:*

(i) *Every edge joining two distinct sets S_j is a bridging edge.*

(ii) *Every component of $G - S$ adjacent to two or more distinct sets S_j is a bridging component. Furthermore, at most one bridging component joins any two distinct S_j's.*

Note that the definition allows any number of components of $G-S$ to be adjacent to a single set S_i. We also remark that the following lemma remains valid for a slightly broader definition of enhancement, but Definition 2.5 suffices for our purposes.

LEMMA 2.6. *If $S \subseteq V$ has an enhancement S_i, $i = 0, \ldots, h$, then*

$$\sum_{i=0}^{h} c(G - S_i) \geq c(G - S) + h.$$

Proof. The proof is by induction on h. The above argument shows

$$c(G - (S - S_h)) + c(G - S_h) \geq c(G - S) + 1.$$

So if $h = 1$, we are done. For $h > 1$ we claim that the partition of $S - S_h$ into S_i, $i = 0, \ldots, h-1$, is an enhancement. This claim implies $\sum_{i=0}^{h-1} c(G - S_i) \geq c(G - (S - S_h)) + h - 1$ by induction. Together with the preceding inequality, this completes the inductive step.

To prove the claim, use the same parent function. Obviously condition (i) holds. The components of $G - (S - S_h)$ can be derived from the components of $G - S$ as follows: Let $\mathcal{C}$ be the family of components of $G - S$ that are adjacent to S_h and no other set S_i. Let B be the bridging component for S_h and $S_{p(h)}$, if it exists. The components of $G - (S - S_h)$ are those of $G - S$ with $\mathcal{C}$ and B replaced by one component, $S_h \cup B \cup \bigcup \mathcal{C}$. Since this new component is adjacent only to $S_{p(h)}$ (or to no set S_i, $i < h$), condition (ii) continues to hold. □

We form enhancements by starting with set S and repeatedly replacing it by $S - S_i$, where S_i is the next set of the enhancement. We call this operation *breaking off* the set S_i. So the enhancement of Definition 2.5 corresponds to breaking off sets $S_h, S_{h-1}, \ldots, S_1$ in that order. Details of the breaking off operation employed in this paper are given in section 5.2.

An *enhancement* of a partition $\mathcal{S}$ of V is formed by enhancing each set of $\mathcal{S}$. Suppose we do a total of β break off operations in various sets of $\mathcal{S}$. Lemmas 2.4 and 2.6 imply

$$\varepsilon_k \geq (k/2) \left(\sum_{S \in \mathcal{S}} c(G - S) + \beta \right). \tag{1}$$

The analysis of section 5 uses the degree lower bound, the carving lower bound, and (1). Furthermore, it analyzes the slack in these three lower bounds. We now present the version of (1) with slack terms.

Fix a k-ECSS H with ε_k edges. We will apply (1) to the graph H, *not* G. Start with a partition $\mathcal{S}$ of V. Form an enhancement, in H, by doing a total of β break off operations. An edge of H is nonbridging if it joins two vertices in the same set S_i of the enhanced partition; for any $S \in \mathcal{S}$, a component of $H - S$ is nonbridging if it is adjacent (in H) only to vertices in one set S_i of the enhanced partition. The reader should not forget that an enhancement may be valid in H but not in G. For instance,

a component may be nonbridging in H but may have neighbors in many different sets S_i in G.

Say that a component has s *surplus edges* if $k+s$ edges of H leave it. Define

θ_a = the total number of surplus edges for all nonbridging components of H;
θ_b = the number of nonbridging edges of H.

Note that θ_a is computed by summing, for every $S \in \mathcal{S}$, the number of surplus edges for all connected components of $H - S$ that are adjacent to only one set S_i of the enhancement. Introducing the terms θ_a and θ_b in the proof of Lemma 2.4 and using Lemma 2.6 gives

$$\varepsilon_k \geq (k/2)\left(\sum_{S\in\mathcal{S}} c(H-S) + \beta\right) + \theta_a/2 + \theta_b. \tag{2}$$

3. Approximation algorithm. Assume the given graph G is biconnected. If not, each block of G is 3-edge connected, and it suffices to run our approximation algorithm on each block. The algorithm consists of three phases that collectively construct a 3-ECSS A. The most involved part is Phase II. We begin by stating all three phases, and then we describe Phase II in detail.

Phase I does a depth-first search to construct a dfs tree carving. Let T denote the depth-first search tree and let r be its root. Phase I sets A to T. Let $C \subseteq T$ be the set of carving edges and write $\gamma = |C|$.

Phase II adds γ back edges to A, making A 2-edge connected. These back edges are chosen to also make a large number of pairs of vertices 3-edge connected. This is done by building A in the form of an ear decomposition. To create even more 3-edge connected pairs, additional back edges are added to A as short ears.

Phase III makes A 3-edge connected. This is done by adding a maximal forest of edges that span the 3-edge connected components of A, as described in Proposition 2.1.

The details of Phases I and III are straightforward. The rest of this section is devoted to Phase II, which is given by the pseudocode of Figure 3. The routines of Figure 3 use an auxiliary procedure `Multi-Merge` and an associated data structure to keep track of 3-edge connected pairs. We now describe both of these.

The data structure is a partition of V into sets of vertices that are known to be 3-edge connected in A. For $x \in V$, $t(x)$ denotes the set containing x. Thus x is 3-edge connected to every vertex of $t(x)$ in the subgraph A. The sets $t(x)$ are called *t-sets* and the corresponding partition of V is called the *t-partition*. The algorithm maintains the t-partition using the disjoint-set data structure, with operations `Union`(x, y) and `Find`(x) [1].

Phase II builds the first vertex tree F (defined in section 2) as it builds the ear decomposition of A. Say that an ear from a to z (with z an ancestor of a) *traverses* the set $t(z)$ and all the sets $t(x)$ where, in tree F, x is an ancestor of a but not an ancestor of $f(z)$. For instance, in Figure 2 an ear i, k, b traverses $t(i), t(d)$, and $t(b)$. In general, all the traversed sets can all be merged together according to Corollary 2.3. The purpose of `Multi-Merge`(a, z) is to execute this merge. Specifically, `Multi-Merge`(a, z) performs `Union`(z, x) for every distinct set $t(x) \neq t(z)$ traversed by the ear from a to z. Observe that an ear traversing s distinct t-sets causes exactly $s - 1$ union operations.

We turn to Figure 3. Let s be the child of r in the dfs tree T. (The root has a unique child since G is biconnected.) Here and throughout this section, $d(v)$ denotes the depth of vertex v in T.

```
Phase II
1.  Long_Ear(s);
2.  Short_Ear(s, 3);
3.  Short_Ear(s, 2);

Long_Ear(b)
1.  let a be the parent of b in T;
2.  choose an edge c ∈ C that descends from edge ab and is covered by a back
    edge yz with y descending from c and z an ancestor of a;
3.  choose the above yz so
    (i) z ∉ t(a) if possible /* merging ear */
    (ii) subject to (i), the depth d(y) is maximal;
/* the new ear P_b consists of the tree path from a to y followed by edge yz */
4.  add the back edge yz to A;
5.  Multi-Merge(a, z);
6.  for each tree edge xx' with x ∈ I(P_b), x' ∉ P_b do Long_Ear(x');

Short_Ear(b, i)
1.  for each tree edge xx' with x ∈ I(P_b), x' ∉ P_b do Short_Ear(x', i);
2.  for each x ∈ I(P_b) do
3.      if some back edge xz traverses > i distinct t-sets then { /* short ear */
4.          let xz be such an edge with minimum depth d(z);
5.          add xz to A;
6.          Multi-Merge(x, z) }
```

FIG. 3. *Algorithms for Phase* II.

Phase II has three main steps (see the top of Figure 3). It starts with every vertex being a singleton t-set. `Long_Ear` enlarges A from the dfs tree T to a 2-edge connected graph. This is done in a top-down traversal of T, as described in section 2, determining the back edges of A that form long ears. Next, the first execution of `Short_Ear` enlarges A with back edges that form short ears, each one causing ≥ 3 unions by `Multi-Merge`. This is done in a bottom-up traversal of T. Then the second execution of `Short_Ear` adds short ears that cause 2 unions by `Multi-Merge`.

The recursive procedure `Long_Ear`(b) starts by constructing a new ear whose first internal vertex is b. To do this, in lines 2–3 we choose the back edge yz of the new ear a, y, z. The back edge yz covers a carving edge c. It is easy to see that if $ab \notin C$, then the carving edge c descends from b; on the other hand, if $ab \in C$, then $c = ab$. The existence of carving edge c and back edge yz is guaranteed by the definition of tree carving.

A long ear is classified as *merging* if $z \notin t(a)$ in line 3(i); otherwise it is *nonmerging*. It is clear that the call to `Multi-Merge` (line 5) performs one or more unions if the ear is merging. The remark after Lemma 4.1 shows that no union is performed for a nonmerging ear.

Line 3(ii) ensures that the depth $d(y)$ is maximal; i.e., if a, y, z is nonmerging, then no ear a, y', z' with y' properly descending from y is possible, and if a, y, z is merging, then no merging ear a, y', z' with y' properly descending from y is possible.

Line 4 updates the tree F when it adds yz to A. Specifically, each vertex of $I(P_b)$ is made a child of a. Finally, line 6 of `Long_Ear` grows the rest of the ear decomposition in recursive calls.

`Short_Ear`(b, i) starts by recursively processing the ears descending from P_b. Then it adds short ears that have their deeper vertex in $I(P_b)$ and cause i or more unions. (In line 3, it makes sense to speak of the t-sets traversed by back edge xz since xz is an ear.) Note that in line 4, xz is not already in A since a back edge of A traverses at most two distinct t-sets. Also note that xz need not be the back edge with minimum $d(z)$. For instance, in Figure 2 if $t(a) = \{a, b, d\}$, then the back edge jc traverses more t-sets than ja or jb.

We note that the call to `Short_Ear`$(s, 3)$ is irrelevant to section 4: All results in that section hold if the call is omitted. We also note that section 5.1 adds some natural rules for choices made in the algorithm of Figure 3.

Figure 4 illustrates the algorithm on a family of graphs in which the approximation ratio approaches 17/12. (The illustration obeys the rules given in section 5.1 too.) First we describe the graph. The number of vertices is divisible by 4, so we write $n = 2h$ with h even. The vertices are identified by the integers $0, \ldots, n-1$. All arithmetic on vertex numbers is done modulo n. It is convenient to describe the edges as a union of five sets. An important property is that no two even-numbered vertices are adjacent. Figure 4(a) illustrates the first edge set,

$$H = \{(2i, 2i-1), (2i, 2i+1), (2i, 2i+3) : 0 \le i < h\}.$$

It is easy to see that the edges of H induce a 3-edge connected graph. Since each vertex has degree 3, H is a smallest 3-ECSS. The second edge set constitutes the dfs tree shown in Figure 4(b)–(c),

$$T = \{(2i-1, 2i+1) : 1 \le i < h\} \cup \{(2h-1, 2i) : 0 \le i < h\}.$$

The third edge set, shown in Figure 4(b), contains the back edges that complete merging ears,

$$M = \{(2i, 2h-2i-3) : h/2 - 1 \le i \le h-2\}.$$

The fourth edge set consists of the back edges that form short ears, shown in Figure 4(c). Writing $a = 3h/2$, the set is

$$S = \{(2i, 3h-2i-3) : a/2 - 1 \le i \le h-2\}.$$

The last edge set forms the spanning forest added to A in Phase III,

$$F = \{(1, 2i) : 0 \le i \le a/2 - 2 \text{ or } i = h-1\} \cup \{(1, 2i+1) : a/2 \le i \le h-2\}.$$

Since no two even-numbered vertices are adjacent, T is a valid dfs tree of the entire graph. Phase I constructs T as the dfs tree. The carving edges are the h edges incident to the leaves of T. In Phase II, `Long_Ear` works as follows (Figure 4(b)): The first ear is $1, 3, 5, \ldots, 2h-1, 2h-2, 1$. (The back edge comes from H.) The remaining long ears each have first vertex $2h-1$. First, $h/2$ merging ears are formed using the back edges of M. These ears build up a t-set of $h/2+1$ vertices, drawn black in Figure 4(b), $t(1) = \{2i-1,\ 2h-1 : 1 \le i \le h/2\}$. All other t-sets are singletons. Next, $h/2-1$ nonmerging ears are formed using back edges contained in H, $(h-4, h-3)$, $\ldots$, $(2, 3)$, $(0, 1)$. (The choice of the second vertex of these edges

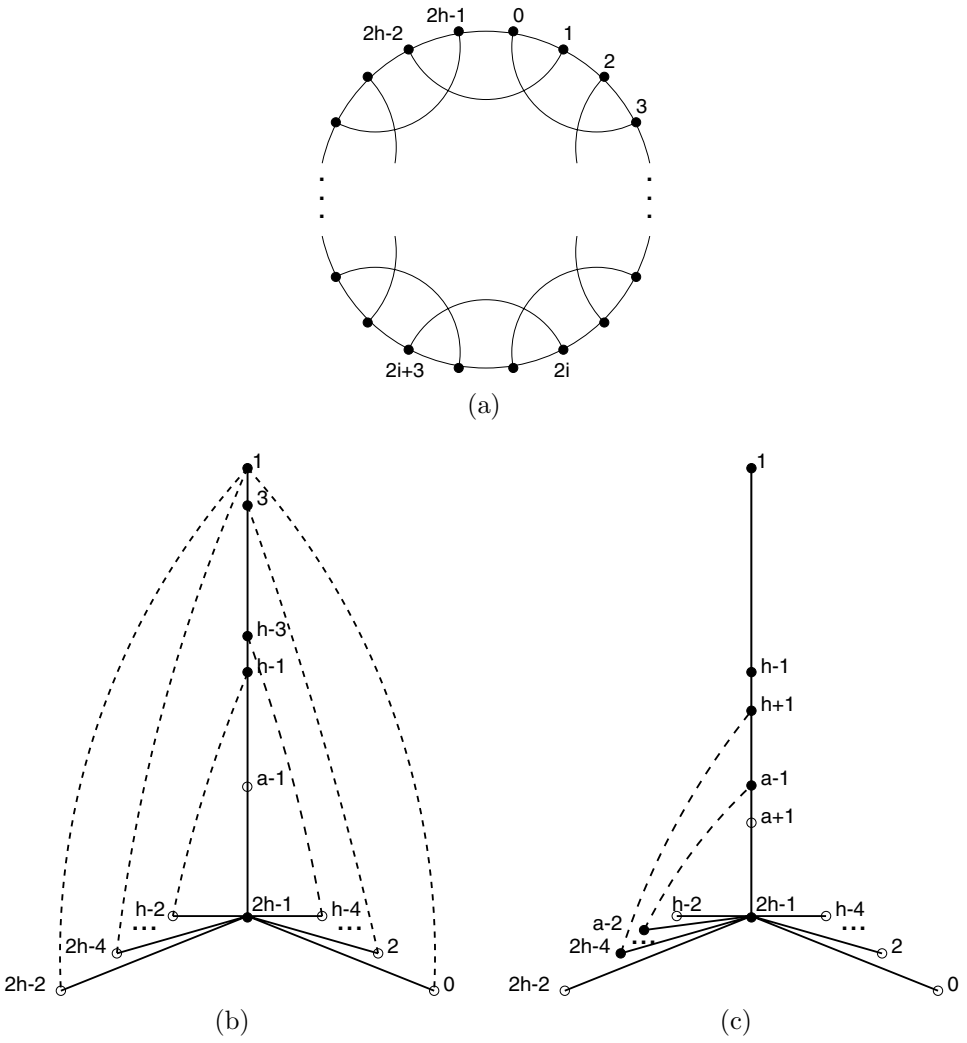


FIG. 4. *Execution of Phase* II *on an example graph.* (a) *Minimum* 3*-ECSS.* (b) *Subgraph* A *after* `Long_Ear`*. Dfs tree edges are solid. Back edges of long ears are dashed.* (c) *Dfs tree with the edges added in* `Short_Ear`*. The parameters of section* 4 *for this example are* $\gamma = h,\ \mu = \overline{\mu} = h/2,\ \sigma = h/4,\ \nu = h,\ \chi = 0,\ \kappa = 3h/4$.

is not crucial.) Observe that no merging ear is possible for any of the corresponding carving edges.

Figure 4(c) illustrates the rest of Phase II, i.e., the two executions of `Short_Ear`. `Short_Ear`$(s, 3)$ does nothing. `Short_Ear`$(s, 2)$ adds the edges of S as $h/4$ short ears. These add $h/2$ vertices to $t(1)$, drawn black. No short ears are added from vertices $a-4, \ldots, 2, 0$, since each back edge from these vertices traverses only two t-sets.

Phase III adds the forest F to A, joining each hollow vertex of Figure 4(c) to vertex 1. Phases I, II, and III add $2h-1$, $h + h/4 = 5h/4$, and $h-1$ edges to A, respectively. The approximation ratio is $(17h/4 - 2)/3h$, which approaches $17/12$ as $h \to \infty$.

Returning to our general discussion, it is clear that Phase II works correctly,

and in fact the whole algorithm correctly produces a 3-ECSS. We note one further property of Phase II.

LEMMA 3.1. *If a nonmerging ear has exactly one internal vertex, that vertex is a leaf.*

Proof. Consider a nonmerging ear P whose first edge is ab. Assume ab is a carving edge; otherwise, obviously P has > 1 internal vertex. P is created in `Long_Ear`(b). Assume b is not a leaf of T. Hence b has a child v. Since b is not an articulation point, some back edge e joins a descendant of v to an ancestor of a. Since P is nonmerging, e can be chosen in line 3(ii), which is a contradiction. Hence P has > 1 internal vertex. □

4. Basic analysis. This section proves some basic properties of the algorithm, leading to the conclusion that the approximation ratio is $\leq 14/9$. We begin by bounding the size of the algorithm's subgraph A. Define the following quantities that satisfy (3):

$$\begin{aligned} \mu &= \text{the number of merging ears,} \\ \overline{\mu} &= \text{the number of nonmerging ears.} \\ \gamma &= \mu + \overline{\mu}. \end{aligned} \tag{3}$$

By definition, a merging ear does at least one nontrivial union operation in `Multi-Merge`. A short ear does at least two nontrivial union operations. All other unions executed by `Multi-Merge` are called *surplus unions*. So an ear causes s surplus unions if it is a merging ear causing $1 + s$ unions, or a short ear (in `Short_Ear`$(s, 3)$) causing $2 + s$ unions. Define the following quantities that satisfy (4):

$$\begin{aligned} \sigma &= \text{the number of short ears,} \\ \nu &= \text{the total number of unions,} \\ \chi &= \text{the number of surplus unions.} \\ \nu &= \mu + 2\sigma + \chi. \end{aligned} \tag{4}$$

These quantities are illustrated in Figure 4.

Phase I adds $n - 1$ tree edges. Phase II adds γ back edges for long ears and σ edges that are short ears. Phase III adds $\leq (n - 1) - \nu$ edges to make A 3-edge connected. Thus the number of edges added by the algorithm is

$$|A| \leq (n-1) + \gamma + \sigma + (n-1) - \nu = 2(n-1) + \overline{\mu} - \sigma - \chi.$$

We turn to lower-bounding ε. We first prove a structural property of the t-partition: At any point in time, any set $t(x)$ contains $f(x)$ if it contains an ancestor of $f(x)$. In fact, we prove the following more general property.

LEMMA 4.1. *At any point in time, any set $t(x)$ contains any ancestor of x in F that has an ancestor in T belonging to $t(x)$.*

Example. In Figure 2, if $t(j)$ contains a proper ancestor of a, then it contains i, d, and a. The lemma is not true if we change F to T; e.g., we may have $b \in t(j)$, but $c, f \notin t(j)$.

Proof. We claim the following.

CLAIM. *At any point in time for any vertex x, $t(x)$ contains $f(x)$ if it contains a vertex not descending (in T) from $I(P_x)$.*

This implies the lemma as follows: Suppose $a \in t(x)$ is an ancestor in T of $f(x)$. The claim implies $f(x) \in t(x)$. Iterating this argument gives the lemma.

We will use this simple consequence of the claim: Call a vertex x *ancestral* if every vertex of $t(x)$ descends (in T) from $I(P_x)$. The claim implies that if $f(x) \notin t(x)$, then x is ancestral.

We now prove the claim by contradiction. Consider the first time the claim fails, say, as a result of the operation `Multi-Merge`(y, z). The new t-set τ formed by this operation must violate the claim. τ is the union of all t-sets traversed by the ear from y to z. Specifically, if W is the set of vertices that, in F, are ancestors of y but not $f(z)$, $\tau = t(z) \cup \bigcup_{x \in W} t(x)$. (Throughout this argument the notation $t(u)$ refers to the t-set of u immediately before `Multi-Merge`(y, z).) Let w be the vertex of W that is shallowest in F. Since some back edge goes from a descendant of y to z, w is a descendant of z; also $f(w) = f(z)$.

We now show that τ consists of $t(z)$, $t(w)$, and some descendants of w in T. A vertex $u \in \tau - t(w) - t(z)$ comes from a set $t(x)$, where $x \in W - t(w)$ and, without loss of generality, $f(x) \notin t(x)$. The latter implies that x is ancestral. This implies u descends from $f(x)$ in T. Since $f(x)$ descends from w in F it descends from w in T. Hence u descends from w in T, as desired.

To show the claim actually is not violated, consider a vertex $v \in \tau$ with $f(v) \notin \tau$. Clearly $f(v) \notin t(v)$, so v is ancestral. We first show that $v \in t(w) \cup t(z)$. Suppose not. Let $x \in W \cap t(v)$ with $f(x) \notin t(v)$. (x exists since $t(v) \neq t(w)$.) $f(x)$ gets added to τ since $f(x) \in W$. Furthermore, $f(x) = f(v)$ since both v and x are ancestral. But this contradicts $f(v) \notin \tau$.

We have shown either $v \in t(z)$ or $v \in t(w)$. Also note that v is ancestral, so every vertex in $t(v)$ descends from $I(P_v)$. We consider four cases depending on how $f(z)$ relates to $t(w)$ and $t(z)$.

Suppose $f(z) \in t(w) - t(z)$. (This is possible even though w descends from z: recall the above example.) Thus every vertex of $t(z)$ descends from $I(P_z)$, $v \in t(w) - t(z)$, and every vertex of $t(w) \cup P_z$ descends from $I(P_v)$. The latter implies that every vertex of τ descends from $I(P_v)$. Thus the claim holds.

The three other possibilities are $f(z) \notin t(w) \cup t(z)$, $f(z) \in t(z) - t(w)$, and $f(z) \in t(w) \cap t(z)$ (i.e., $t(w) = t(z)$). The argument for each is similar to the one just given. □

The lemma justifies the term "nonmerging ear": It is easy to see that `Multi-Merge` does not perform any unions for a nonmerging ear.

For the rest of this paper, all sets $t(a)$ refer to their value at the end of Phase II unless explicitly stated otherwise. The *(component) cluster* of an ear P with first vertex a consists of all descendants in F of vertices in $I(P) - t(a)$. For example, in Figure 2 if $P_b \cap t(a) = \{a, c, e\}$, then the cluster of P_b is $\{b, d, f, i, j, k\}$. A cluster can be empty; i.e., we can have $I(P) \subseteq t(a)$. For instance, any short ear has an empty cluster. We will be interested only in nonempty clusters, which occur only for long ears. The next lemma gives basic properties of clusters; the term "cluster" is motivated by property (ii).

LEMMA 4.2. *Let K be the cluster of an ear with first vertex a.*

(i) $K \cap t(a) = \emptyset$.

(ii) *K is a union of connected components of $G - t(a)$.*

Proof. Let K be the cluster of ear P.

(i) Suppose $y \in K \cap t(a)$. In F, y has an ancestor $x \in I(P) - t(a)$. Now $t(y)$ contains a, an ancestor of x, but not x itself. This contradicts Lemma 4.1.

(ii) By part (i), it suffices to show that every edge leaving K goes to $t(a)$. A tree edge leaving K must be an edge of P. Since $P \subseteq K \cup t(a)$, the edge goes to $t(a)$. Now

suppose a back edge yb (with b an ancestor of y) leaves K but does not go to $t(a)$. There are two possibilities.

Case 1. $y \in K$ and $b \notin K \cup t(a)$.

$b \notin P$ since $P \subseteq K \cup t(a)$. Hence b is a proper ancestor of a. Edge yb traverses $t(y)$, $t(a)$, and $t(b)$. These sets are distinct at the end of Phase II. ($b \notin t(a)$ by Case 1, $y \notin t(a)$ by part (i), and $b \notin t(y)$ by Lemma 4.1 with $a \notin t(y)$.) Hence the sets are distinct when line 2 of `Short_Ear` is executed with $x = y$ and $i = 2$. Thus lines 4–5 add a back edge yz with z an ancestor of b. The subsequent execution of `Multi-Merge`(y, z) merges $t(y)$ and $t(a)$, which is a contradiction.

Case 2. $y \notin K \cup t(a)$ and $b \in K$.

The definition of K implies that $b \in P$. Let x be the deepest ancestor of y in P. Since $y \notin K$, $x \in t(a)$. Thus b is a proper ancestor of x. Edge yb traverses $t(y)$, $t(x)$, and $t(b)$. These sets are distinct at the end of Phase II. ($y, b \notin t(x) = t(a)$ by Case 2 and $b \notin t(y)$ by Lemma 4.1 with $x \notin t(y)$.) Now the argument follows Case 1: Lines 4–5 of `Short_Ear` add a back edge yz with z an ancestor of b, and `Multi-Merge`(y, z) merges $t(y)$ and $t(x) = t(a)$, which is a contradiction. □

Define

$$\kappa = \text{the total number of nonempty clusters.}$$

(See Figure 4.) Call an ear *depleted* if all its vertices belong to the same t-set. Equivalently, ear P defined by a, y, z is depleted if $I(P) \subseteq t(a)$ (since we always have $z \in t(a)$). Thus κ equals the number of nondepleted ears.

LEMMA 4.3. $\varepsilon \geq (3/2)(n - 1 + \kappa - \nu)$.

Proof. Use the t-partition in the component lower bound. Consider a set $t(a)$. Let κ_a nondepleted ears have their first vertex in $t(a)$. Each of these ears gives a nonempty cluster. All of these clusters for $t(a)$ are pairwise disjoint (Lemma 4.2(i)). Hence $c(G - t(a)) \geq \kappa_a$. This gives a total of $\geq \kappa$ components in the component lower bound.

If $r \notin t(a)$, then $t(a)$ has at least one more component. To prove this, it suffices to show that r does not belong to any cluster of $t(a)$. This follows since r, as the root of F, does not descend (in F) from any internal vertex of any ear.

The number of distinct sets $t(a)$ is n decreased by the number of union operations, $n - \nu$. So the previous observation gives $n - \nu - 1$ more components in the component lower bound. We conclude that the total number of components in the component lower bound is $\geq \kappa + n - \nu - 1$. This gives the lemma. □

The following inequality has some slack in it, but see the remark after Lemma 4.5.

LEMMA 4.4. $\kappa \geq \overline{\mu} - \nu/2$.

Proof. Consider a nonmerging depleted ear P with first vertex a. Since each internal vertex gets merged into $t(a)$, we can associate $|I(P)|$ unions with P. We will prove $|I(P)| \geq 2$. Since a nonmerging ear is either depleted or nondepleted, we get $2\overline{\mu} \leq \nu + 2\kappa$ as desired.

We need only show that a nonmerging ear P with one internal vertex is not depleted. Let P be a, y, z. Lemma 3.1 shows that y is a leaf of T. So it suffices to prove the following claim. The claim drops the assumption that a, y, z has one internal vertex, since we need this more general fact in section 5.

CLAIM. *A nonmerging ear a, y, z with y a leaf has $t(y)$ a singleton.*

To prove this, we need only show that Phase II does not add a back edge yw as a short ear. When ear a, y, z is created, any back edge yw has w either descending from a or belonging to $t(a)$. So Lemma 4.1 (with $x = a$) shows that yw only traverses

two distinct sets, $t(y)$ and $t(w)$. Thus line 5 of `Short_Ear` does not add a back edge from y, even when $i = 2$. □

We can now bound the approximation ratio.

LEMMA 4.5. *The algorithm has approximation ratio* $\leq 14/9$.

Proof. Define the quantity δ to satisfy

$$\overline{\mu} - \sigma - \chi = (2/3)\gamma + \delta. \tag{5}$$

Thus $|A| \leq 2(n-1) + (2/3)\gamma + \delta$. We will show that

$$(4/3)\varepsilon \geq 2(n-1) + \delta.$$

The tree carving lower bound $\varepsilon \geq 3\gamma$ implies $(2/9)\varepsilon \geq (2/3)\gamma$. Thus $|A| \leq (4/3 + 2/9)\varepsilon = (14/9)\varepsilon$ as desired.

If $\delta \leq 0$, then the degree lower bound $\varepsilon \geq (3/2)n$ implies $(4/3)\varepsilon \geq 2n \geq 2(n-1) + \delta$ as desired. Hence we assume $\delta > 0$. Lemmas 4.3 and 4.4 combined give $\varepsilon \geq (3/2)(n - 1 + \overline{\mu} - 3\nu/2)$.

Some algebra shows $\overline{\mu} - 3\nu/2 \geq 3\delta$ as follows: Substitute (3) into (5) to get $\overline{\mu}/3 - \sigma - \chi = 2\mu/3 + \delta$ or, equivalently,

$$\overline{\mu} = 2\mu + 3(\sigma + \chi + \delta).$$

Combining with (4),

$$\overline{\mu} - 3\nu/2 = \mu/2 + 3\chi/2 + 3\delta \geq 3\delta.$$

We have shown $\varepsilon \geq (3/2)(n + 3\delta - 1)$. Hence $(4/3)\varepsilon \geq 2(n + 3\delta - 1) \geq 2(n-1) + \delta$. □

As already mentioned, Lemma 4.4 has some slack. However, the interested reader can check that even if we replace the term $\overline{\mu}$ with γ in that lemma, an argument similar to the above does not yield a lower approximation ratio.

5. Sharper analysis. This section proves that a natural implementation of the algorithm has approximation ratio $\leq 3/2$. Section 5.1 states three rules we require in the implementation; it also introduces some basic concepts for the analysis. Section 5.2 discusses how we use the breaking off operation. Section 5.3 proves the 3/2 bound, assuming a key inequality. Finally, section 5.4 proves the key inequality.

5.1. Algorithm rules and basic notions. We begin with some additional terminology. We often designate an ear a, y, z as a, z. The last internal vertex of a long ear (y) is its *tip*. Each vertex v is uniquely classified as a tip or nontip since P_v is unique. We often apply terminology for an ear to its tip; e.g., a merging tip is a tip whose ear is merging. The main issue of section 5 is bounding the number of nonmerging depleted ears (recall the proof of Lemma 4.4). Towards this end, let $\mathcal{Y}$ denote the set of all nonmerging depleted tips. A *child ear* of vertex x is a long ear whose first edge goes to a child of x.

To prove the upper bound, we incorporate several rules specifying choices made by the algorithm. For each rule, we specify the line number to which it applies.

Rule 1 (line 3(ii) of `Long_Ear`). Once y is chosen, z is chosen so a merging ear from y merges as many t-sets as possible.

Rule 2 (line 6 of `Long_Ear`). Vertex x progresses through the internal vertices of P_b in order; i.e., x moves from b to y.

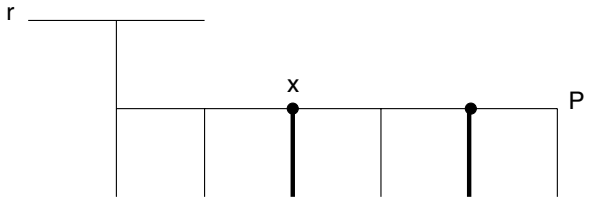


FIG. 5. *$t(x)$ contains the two solid vertices but not the first vertex of P. The root cluster of $t(x)$ consists of all vertices except the ones on the two heavy paths.*

Rule 3 (line 6 of `Long_Ear`). The first child ear of vertex x is merging if possible.

Rule 3 refers to the first child ear created at vertex x. We also call this the *first merging ear at x* (Lemma 5.10).

The following simple consequence of line 3(ii) is useful.

PROPOSITION 5.1. *Consider a back edge xz, where x properly descends from a tip y.*

(i) *If y is nonmerging, then z is a proper descendant of $f(y)$.*

(ii) *If y is merging, then z descends from $f(y)$ or belongs to $t(f(y))$.* □

The analysis of section 4 uses the notion of a cluster of an ear, and the proof of Lemma 4.3 shows there are additional components containing the root. We will call these "root clusters," defined formally as follows. Consider a vertex $x \neq r$ that is the shallowest vertex in $t(x)$. Every vertex in $t(x)$ has an ancestor in $t(x) \cap I(P_x)$. (This follows from the claim of Lemma 4.1.) The *root cluster* of $t(x)$ consists of all vertices that do not descend in F from $I(P_x) \cap t(x)$. (Contrast this with the definition of ear cluster in section 4; see Figure 5.) Equivalently, the root cluster consists of all vertices that do not descend in F from a vertex of $t(x)$.

In addition we use the term *clusters of $t(x)$* to refer to all clusters associated with the set $t(x)$; specifically the term refers to the root cluster of $t(x)$ plus the cluster of each ear a, y, z that has $a \in t(x)$.

We will bound the number of nonmerging depleted ears by associating various sets with them. The following notions are used to accomplish this.

DEFINITION 5.2. *For a vertex $y \in \mathcal{Y} \cup r$, T_y denotes the set of all vertices that descend from y but no deeper vertex of $\mathcal{Y} \cup r$. The* representative vertex *of a set is*

- *for an edge, its deeper vertex;*
- *for an ear, its first vertex;*
- *for the cluster C of an ear, the shallowest vertex of C;*
- *for the root cluster of $t(a)$, the shallowest vertex of $t(a)$;*
- *for a component C of a t-set, the representative of the cluster containing C.*

A set with representative vertex $v \in T_y$ is launched by *any ancestor of v in T_y. The set is* properly launched *if $v \in T_y - y$.*

Note that the representative of a short ear e is the same when we consider e to be an ear or an edge. In the last bulleted item, a component C of t-set $t(a)$ refers to a connected component of $G - t(a)$ (recall the proof of Lemma 4.3).

Any set S with a representative vertex is launched by exactly one vertex of $\mathcal{Y} \cup r$. So S is launched by at most one tip $y \in \mathcal{Y}$. This is our mechanism for establishing "ownership" of sets. Note that, in such an ownership relation, we cannot rely on any particular relationship between $t(y)$ and any t-set associated with S. For instance, if S is a cluster of $t(a)$ that is launched by tip $y \in \mathcal{Y}$, we might hope that $t(y) = t(a)$, but this is not true in general.

Throughout this entire section we fix a smallest 3-ECSS H of G for the analysis.

5.2. Breaking off operation. Our analysis starts with the t-partition of V and forms an enhancement, called the *t*-partition*, by doing a number of break off operations. This section describes these operations. It also proves a lemma, allowing us to enhance the t-partition to the t*-partition yet carry out the subsequent analysis by referring only to t-sets.

Section 2 defined the operation of breaking off in general. Now we specify this operation for our analysis. Initially the t*-partition is identical to the t-partition. Let b be a vertex with parent p. To *break off vertex* b means to replace $t^*(p)$ with $t^*(p)-B$ and B, for B the set of descendants of b belonging to $t^*(p)$. Vertex b is *breakable* if

(i) $t(p)$ contains at least one descendant of b;

(ii) at most one component of $H - t(p)$ is adjacent (in H) to both descendants and nondescendants of b;

(iii) b or p is a tip;

(iv) $b \notin \mathcal{Y}$.

Conditions (iii)–(iv) can be weakened, but they suffice for our purposes.

The t*-partition is formed by breaking off zero or more breakable vertices of each t-set. Let us describe how a t-set, say $t(a)$, gets partitioned into t*-sets. Choose a as the shallowest vertex in $t(a)$. Suppose we break off h vertices b_i whose parent p_i belongs to $t(a)$, $i = 1, \ldots, h$. The set $t(a)$ gets partitioned into $h+1$ t*-sets, specifically the vertices of $t(a)$ that descend from b_i but no deeper b_j, for $i = 0, \ldots, h$. Here we take a to be b_0. It is easy to see that condition (i) implies each of these t*-sets is nonempty. In Definition 2.5 the parent of the t*-set of b_i is $t^*(p_i)$ $(i \geq 1)$.

LEMMA 5.3. *Suppose the t*-partition is defined by breaking off* ≤ 1 *breakable vertex in each set* T_y.

(i) *The t*-partition is an enhancement of the t-partition (in graph* H*).*

(ii) *If no vertex of* T_y *is broken off, then an edge properly launched by* y *with both ends in the same t-set is a nonbridging edge (of the t*-partition).*

(iii) *Suppose* y *launches a bridging component* C *of* $H - t(a)$ *for some vertex* a*. Then a vertex* $b \in T_y$ *whose parent belongs to* $t(a)$ *was broken off and* C *is adjacent to both descendants and nondescendants of* b*.*

Proof. For any $y \in \mathcal{Y} \cup r$, let $\overline{T}_y$ be the set T_y enlarged at its leaves; i.e., add to T_y each $w \in \mathcal{Y}$ having $f(w) \in T_y$. Let $X_y = I(P_y) \cup \overline{T}_y$. (For $y = r$ recall $I(P_r) = \emptyset$.)

CLAIM 1. *For any vertex* a*,* $t(a) \cap X_y$ *is contained either in one t*-set or in some t*-set and its parent set.*

Proof. The only possible breakable vertices of X_y are the breakable vertices b of T_y and the first vertex b' of $I(P_y)$ (recall (iii)–(iv) of the definition). If we break off b', $t(a) \cap X_y$ is still contained in one t*-set. If we break off a vertex $b \in T_y$ with parent p, the t*-set changes only if $p \in t(a)$. In that case, $t(a) \cap X_y$ is contained in a t*-set descending from b and its parent set $t^*(p)$. □

CLAIM 2. *An edge* vz *with deeper vertex* $v \in t(z)$ *either has both ends in the same t*-set or joins a t*-set and its parent set.*

Proof. Choose y so $v \in \overline{T}_y - y$. Proposition 5.1(i) shows $z \in X_y$. Together with Claim 1 this implies Claim 2. □

Claim 2 gives condition (i) of Definition 2.5 for the t*-partition. It also gives (ii) of the current lemma.

CLAIM 3. *Consider a cluster* C *of* $t(a)$ *with representative vertex* c*. Suppose* $c \in T_y - y$ *for* $y \in \mathcal{Y} \cup r$*. Then any edge with exactly one end in* C *has the other end in* X_y*.*

Proof. We remind the reader that C is either a root cluster or an ear cluster. Our assertions must be checked in both cases! For root clusters it is convenient to refer to Figure 5.

We begin by noting that vertex y of the claim always exists; this is equivalent to $c \notin \mathcal{Y} \cup r$. P_c is not depleted (for both cluster types). This implies $c \notin \mathcal{Y}$. $c \neq r$ follows from the fact that r is not the representative of any cluster (of either type).

Now let vz be an edge, with deeper vertex v, having exactly one end in C. (That end can be v or z.) Suppose v is a proper ancestor of c. If C is an ear cluster, this implies $v, z \notin C$. If C is a root cluster, it implies $v, z \in C$. In both cases, vz violates its definition. So assume that v descends from c.

Since $c \neq y$, we have $v \neq y$. If $v \in \overline{T}_y - y$, then $z \in X_y$ (Proposition 5.1(i)) and the claim holds (regardless of which vertex is in C). So assume that v is a proper descendant of some $w \in \overline{T}_y \cap \mathcal{Y} - y$.

Let b be the first vertex of $I(P_w)$. We show that (a) b is an ancestor of z, (b) $b \notin P_c$, and (c) b descends from c. (a) follows from the definition of b and Proposition 5.1(i). (b) follows since P_c is not depleted but P_w is. For (c), recall our assumption that v descends from c. The tree path from v to c enters T_y along P_w and then goes to P_c. (c) follows.

(b) and (c) imply either that all descendants of b belong to C or that none do (for either ear type). With (a) this makes vz violate its definition. □

Now consider a component C of $H - t(a)$. Since C is contained in a cluster, Claims 3 and 1 show that all the neighbors of C belong either to the same t*-set or to some t*-set and its parent. Together with condition (ii) of the definition of breakable, this gives condition (ii) of Definition 2.5. It also gives (iii) of the current lemma. □

5.3. The approach. In the following definitions we fix a t*-partition formed according to Lemma 5.3. If a cluster of $t(x)$ contains $1 + s$ connected components of $H - t(x)$, it has s *surplus components*. If $3 + s$ edges of H cover an edge $e \in C$, then e is *redundantly covered* s times. Define

κ = the number of nondepleted ears, plus the number of surplus components in clusters;
β = the number of break offs that form the t*-partition;
θ_a = the total number of surplus edges for all nonbridging components of H;
θ_b = the number of nonbridging edges of H;
θ_c = the number of edges of H not covering an edge of C or redundantly covering an edge of C;
θ_d = the number of vertices that have degree > 3 in H.

(Subscript b stands for "both ends," c stands for "carving," and d stands for "degree.") Our new definition of κ generalizes the definition in section 4. β, θ_a, and θ_b are defined as at the end of section 2.

The next section bounds the number of nonmerging ears, showing

$$\overline{\mu} \leq \kappa + \chi + \beta + \theta_d + (\theta_a + 2\theta_b + \theta_c)/3. \tag{6}$$

We now demonstrate that this implies the approximation ratio is $\leq 3/2$.

We use the following three lower bounds:

$$\varepsilon \geq 3\gamma + \theta_c; \tag{7}$$

$$\varepsilon \geq (3/2)n + \theta_d/2; \tag{8}$$

$$\varepsilon \geq (3/2)(n - 1 + \kappa + \beta - \nu) + \theta_b + \theta_a/2. \tag{9}$$

It is clear that (7) and (8) are true. To prove (9), recall that Lemma 4.3 is proved using the component lower bound. Instead use (2), which is our extension of the component lower bound. Inequality (9) follows easily.

Define the quantity δ to satisfy

$$\overline{\mu} - \sigma - \chi = \gamma/2 + \delta. \tag{10}$$

(δ may be positive, negative, or 0.) Thus

$$|A| \le 2(n-1) + \gamma/2 + \delta.$$

Combining 1/6 times (7) with 4/3 times (8) gives $(3/2)\varepsilon \ge 2n + \gamma/2 + \theta_c/6 + (2/3)\theta_d$. Hence we can assume

$$\delta > \theta_c/6 + (2/3)\theta_d$$

since otherwise we are done. To handle this case we will show that (7) and (9) imply

$$(3/2)\varepsilon \ge 2(n-1) + \gamma/2 + 4\delta - 2\theta_d - \theta_c/2. \tag{11}$$

Inequality (11), together with our assumption $3\delta > \theta_c/2 + 2\theta_d$, implies the desired result.

We begin by reexpressing $\overline{\mu}$ as follows: Substitute (3) into (10) to get $\overline{\mu}/2 - \sigma - \chi = \mu/2 + \delta$ or, equivalently, $\overline{\mu} = \mu + 2(\sigma + \chi + \delta)$. Combining with (4), we have

$$\overline{\mu} - \nu - \chi = 2\delta.$$

Combining this with (6) gives

$$\kappa + \beta - \nu \ge \overline{\mu} - \chi - \nu - \theta_d - (\theta_a + 2\theta_b + \theta_c)/3 = 2\delta - \theta_d - (\theta_a + 2\theta_b + \theta_c)/3.$$

Thus (9) gives

$$\varepsilon \ge (3/2)(n - 1 + 2\delta - \theta_d - (\theta_a + 2\theta_b + \theta_c)/3) + \theta_b + \theta_a/2 = (3/2)(n - 1 + 2\delta - \theta_d - \theta_c/3).$$

Combining 4/3 times this inequality with 1/6 times (7) gives the desired inequality (11).

5.4. Bounding $|\mathcal{Y}|$. We begin by noting that this section is concerned with both the given graph G and the fixed 3-ECSS H. Proposition 5.4 and Lemmas 5.5–5.9 are properties of G alone. The last three results, Lemmas 5.10–5.12, depend on H.

PROPOSITION 5.4. *Let a 3-edge connected graph $G = (V, E)$ have degree function d. Suppose sets X, X', and x form a partition of V, and $d(x) = 3$. Then $d(X, X') \ge 2$.*

Proof. Without loss of generality, assume $d(x, X) \le 1$. Hence $3 \le d(X) = d(X, X') + d(X, x) \le d(X, X') + 1$ as desired. □

A long ear is *tight* if its last tree edge is a carving edge. Otherwise the ear is *loose*. Every pendant edge of T is obviously tight. The following lemma generalizes this fact.

LEMMA 5.5. *The tip of a loose ear has a merging child ear.*

Proof. Let the loose ear contain the carving edge vw, and let its last edge be xy. Thus y is the ear's tip; possibly $w = x$. Consider the depth-first search of Phase I that finds carving edges. Edge xy does not force a back edge to be added. So the search added a back edge uz from a descendant u of y to an ancestor z of x. Furthermore,

the tree path from y to u contains a carving edge. When `Long_Ear` constructs P_y, y, u, z is a possible merging ear at y. So Rule 3 shows y has a merging child. □

We note a related property for use in Lemma 5.9. A tight ear has only nonmerging children. This follows since a back edge covers at most one carving edge.

LEMMA 5.6. *Let y be a nonmerging tip. An ear of A with its first vertex descending from y has its last vertex properly descending from $f(y)$.*

Proof. Let the ear be a, u, z and, for the sake of contradiction, assume z is an ancestor of $f(y)$. Since u is a descendant of y, Proposition 5.1(i) implies $u = y$. Since u descends from a, we get $a = y$; i.e., the ear is short. But the algorithm never adds a short ear originating at a nonmerging tip (this is proved in the claim of Lemma 4.4). □

The next lemma consists of two similar parts. After stating the lemma we give an example showing that a plausible generalization is false.

LEMMA 5.7. *Let an ear P have first vertex a and tip u.*

(i) *The first ear that adds a vertex of $I(P)$ to $t(a)$ is launched by a.*

(ii) *The first ear that adds a proper ancestor of u to $t(u)$ is launched by u.*

Example 1. For an arbitrary vertex x, the first ear that adds a proper ancestor of x to $t(x)$ need not be launched by x. In Figure 2 take x to be d. Let $e \in \mathcal{Y}$. After the short ear h, c is added the short ear g, d adds c to $t(d)$. But g, d is launched by e and is not launched by d. A similar example uses merging ears: The first merging ear goes from e to c and the second from e to d.

Proof. (i) Consider the first execution `Multi-Merge`(b, z) that adds a vertex of $I(P)$ to $t(a)$. b descends from $I(P)$ and z is an ancestor of a. We claim that ear b, z was launched by the first vertex of $I(P)$. (This is slightly stronger than the lemma.) We prove this by showing that the tree path from a to b does not contain a vertex $w \in \mathcal{Y} - a$.

Suppose w exists. $f(w)$ is a descendant of a (possibly equal to it). So Lemma 5.6 shows z is a proper descendant of a, which is a contradiction.

(ii) Consider the first execution `Multi-Merge`(b, z) that adds a proper ancestor of u to $t(u)$. b descends from u and z is a proper ancestor of u. Suppose the tree path from u to b contains a vertex $w \in \mathcal{Y} - u$. Since u is a tip, $f(w)$ is a descendant of u (possibly equal to it). So Lemma 5.6 shows that z is a proper descendant of u, which is a contradiction. □

Remark. The stronger version of the lemma that we proved leads to a stronger version of Lemma 5.8, but we do not require it.

A *surplus ear* is a merging or short ear that causes a surplus union (i.e., a union counted in χ).

LEMMA 5.8. *Suppose $y \in \mathcal{Y}$ does not launch a surplus ear and some back edge xz has $x \in T_y - y$.*

(i) *If z is an ancestor of $f^3(x)$, then $f^3(x)$ launches a merging ear.*

(ii) *If z is a proper ancestor of $f^2(x)$ and $f^2(x)$ is a tip, then $f^2(x)$ launches a merging ear.*

Remark. A plausible common generalization of (i)–(ii) fails: Assuming the lemma's hypothesis, z can be a proper ancestor of $f^2(x)$ without $f^2(x)$ launching a merging ear. For instance, let xz be the back edge jc in Figure 2, and assume the merging ear scenario of Example 1.

Proof. Proposition 5.1(i) shows that z is a proper descendant of $f(y)$. Hence any vertex $f^i(x)$ that descends from z, properly or not, belongs to T_y, e.g., $i \leq 3$ in (i) and $i \leq 2$ in (ii).

(i) Suppose $f^3(x)$ does not launch a merging ear. We prove the lemma by showing y launches a surplus ear. Lemma 5.7(i) shows that $x, f(x), f^2(x)$, and $f^3(x)$ are in distinct t-sets at the end of `Long_Ear`. Hence Lemma 5.7(i) implies $f^3(x)$ launches a surplus ear during `Short_Ear`$(s, 3)$—either before x is scanned or when a back edge at x (e.g., xz) is added as a short ear.

(ii) The argument is similar to (i). Suppose $f^2(x)$ does not launch a merging ear. We show y launches a surplus ear. Lemma 5.7(i)–(ii) shows that $x, f(x), f^2(x)$, and z are in distinct t-sets at the end of `Long_Ear`. Hence Lemma 5.7(i)–(ii) implies that $f^2(x)$ launches a surplus ear during `Short_Ear`$(s, 3)$—either before x is scanned or when a back edge at x (e.g., xz) is added as a short ear. □

An ear P with first vertex a is *penetrated* if $I(P) \cap t(a) \neq \emptyset$.

LEMMA 5.9. *A nonleaf tight tip either launches a surplus ear or has a penetrated child ear.*

Proof. Let u be a nonleaf tight tip. In this proof, say that $t(u)$ *is enlarged by ear* b, z if the execution `Multi-Merge`(b, z) changes $t(u)$ from a singleton set to a nonsingleton. Clearly, in this case z is an ancestor of u, which is an ancestor of b. If $b \neq u$, then some vertex x with $f(x) = u$ gets added to $t(u)$. This makes P_x a penetrated child ear of u.

The rest of the argument is in three cases. First suppose $t(u)$ is enlarged by a merging ear b, z. No child ear of u is merging (as remarked after Lemma 5.5). So b properly descends from u. The opening remark gives the lemma.

Suppose $t(u)$ is enlarged during `Short_Ear`$(s, 3)$ by a short ear b, z. Either $b \neq u$ or u launches a surplus ear. In both cases the lemma holds.

Finally, suppose $t(u)$ is a singleton at the end of `Short_Ear`$(s, 3)$. Let P be the first child ear of u. Some back edge xz joins a descendant x of $I(P)$ to a proper ancestor z of u, since G has no articulation point. Since `Short_Ear`$(s, 2)$ works bottom-up, some short ear launched by a descendant of $I(P)$ enlarges $t(u)$. (This either occurs before x is scanned or when a back edge at x, e.g., xz, is added as a short ear.) Again the lemma holds. □

LEMMA 5.10. *Suppose $y \in \mathcal{Y}$ does not launch a surplus component or a surplus ear. If the first merging ear at a given vertex of T_y is depleted and its tip u is a leaf, then u is breakable.*

Proof. Let $a = f(u)$. Thus $a \in T_y \cap t(u)$. We claim u is adjacent to at most one cluster C of $t(u)$; furthermore

$$C \cap P_a \neq \emptyset$$

if C exists. This claim implies the lemma. To prove this, we need only verify condition (ii) of the definition of breakable. (For condition (i) note that the parent of u belongs to $t(u)$.) Assume C exists; otherwise, condition (ii) is vacuous. The above set inequality makes P_a nondepleted (since it contains vertices of two t-sets). Hence $f(a) \in T_y$, and C (root cluster or ear cluster) is launched by y. Now the lemma's hypothesis shows that C consists of exactly one component of $H - t(u)$. Hence (ii) holds and u is breakable.

To prove the claim, first suppose $f(a) \notin t(u)$. So any ancestor of u belongs to $t(u)$ or to the root cluster C of $t(u)$. Since $f(a) \in C$, we have $C \cap P_a \neq \emptyset$, and the claim holds.

Next suppose $f(a) \in t(u)$. Consider an edge uz with $z \notin t(u)$, and suppose z does not descend from $f(a)$. When ear P_u is created, $t(a)$ is a singleton by Rules 2–3. Hence $a, f(a)$, and z are in distinct t-sets at that time. (Note that at the end of Phase II, $t(u) = t(a) = t(f(a))$, so $z \notin t(f(a))$.) Rule 1 shows P_u is a surplus ear.

But the lemma assumes this is not the case (recall $a \in T_y$). Hence z descends from $f(a)$. We conclude that cluster C of P_a is the only cluster of $t(u)$ that is adjacent to u. Obviously $C \cap P_a \neq \emptyset$ if $C \neq \emptyset$. □

For any $y \in \mathcal{Y}$ define

$\kappa(y)$ = the number of merging nondepleted ears launched by y plus the number of surplus components launched by y;
$\chi(y)$ = the number of surplus ears launched by y;
$\beta(y)$ = the number of breakable vertices in $T_y - y$;
$\theta_a(y)$ = the number of surplus edges of H leaving components launched by y;
$\theta_b(y)$ = the number of edges of H properly launched by y having both ends in the same t-set;
$\theta_c(y)$ = the number of edges of H launched by y not covering a carving edge or redundantly covering a carving edge launched by y;
$\theta_d(y)$ = the number of vertices of T_y having degree > 3 in H.

$\chi(y)$ is the only one of the above quantities that is a function of G, not H. None of the quantities involve the t*-partition. The definitions of $\kappa(y), \theta_a(y)$, and $\theta_b(y)$ differ slightly from their counterparts in section 5.3. The differences are reconciled in the last argument of this section.

LEMMA 5.11. *For $y \in \mathcal{Y}$ let $u \in T_y$ be the tip of a depleted ear. Suppose u does not launch a merging ear and u has a penetrated child ear. Then either*

(i) *u has a breakable child belonging to T_y, or*

(ii) *$\theta_a(y) + 2\theta_b(y) + \theta_c(y) \geq 3$, or*

(iii) *$\kappa(y) + \chi(y) + \theta_d(y) \geq 1$.*

Proof. The argument is illustrated in Figure 6. Lemma 5.5 shows u is tight. Let U be the set of proper ancestors of u. Consider the components of $H - t(u)$. For an arbitrary ear X, C_X denotes the cluster of X (which may be empty). For $S \subseteq V$, $\overline{S}$ denotes $V - S$. We assume the lemma is false and derive a contradiction.

CLAIM 1. *If S is the set of all descendants of a child of u, then $d_H(S, U) \geq 2$. Furthermore each edge from S to U is launched by y.*

Proof. For the first part we have $d_H(u) = 3$, since otherwise $\theta_d(y) \geq 1$ and (iii) holds. Now Proposition 5.4 shows $d_H(S, U) = d_H(S, \overline{S} - u) \geq 2$.

For the second part, an edge e from S to U is launched by y since u is tight and e cannot cover two carving edges. □

Remark. If u has ≥ 3 children, the claim shows $\theta_c(y) \geq 3$ and we are done. However, we do not use this principle.

The rest of the argument focuses on P, a penetrated child ear of u. Let p be the tip of P. Let D be the set of all descendants of the child of u that belongs to P.

CLAIM 2. *Let xz be an edge of G with $x \in D$, $z \in \overline{D}$.*

(a) *$z \in t(u)$.*

(b) *Suppose x is a proper descendant in F of a vertex $w \in P$. Then $w \neq p$, the tree path from w to x contains no carving edge, and $x \in T_y$.*

(c) *x belongs to P or a child ear of $I(P)$. In the latter case $z = u$.*

Proof. (a) Since u is depleted, this part follows from Proposition 5.1(i)–(ii).

(b) Since P is nonmerging, $w \neq p$ by Proposition 5.1(i). Hence $w \in T_y$. The lemma's hypothesis shows w is not the first vertex of a merging ear. So Rule 3 implies there is no carving edge on the tree path from w to x. This makes $x \in T_y$.

(c) First suppose x is a proper descendant of a child ear of $I(P)$. (b) implies $x \in T_y$. Now Lemma 5.8(i) and the hypothesis of our lemma show y launches a surplus ear. Thus $\chi(y) \geq 1$, and (iii) holds.

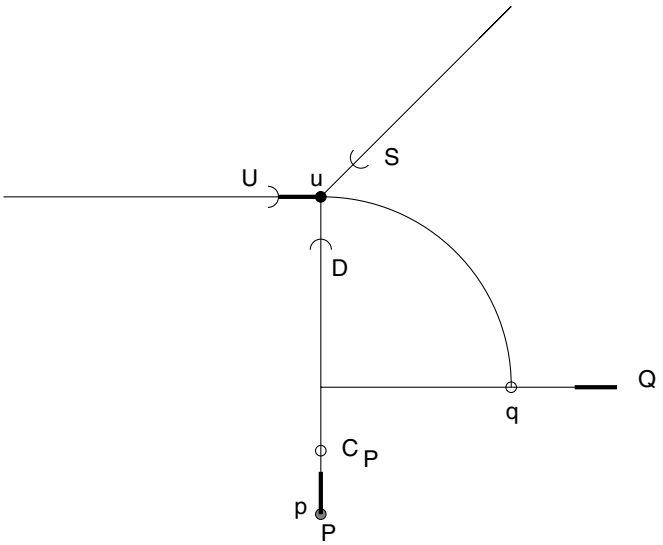


FIG. 6. *Proof of Lemma* 5.11. *Hollow vertices are not in* $t(u)$. *Heavy lines denote carving edges.*

Next suppose x belongs to a child ear of $I(P)$ and $z \in U$. As before, $x \in T_y$. Since u is a tip, Lemma 5.8(ii) and the hypothesis of our lemma show y launches a surplus ear. This gives (iii). □

CLAIM 3. *P is nondepleted and tight. Hence cluster C_P is nonempty.*

Proof. If P is depleted, then by Claims 2(a) and 2(c) the two edges of Claim 1 have both ends in $t(u)$. The deeper end of each edge belongs to $T_y - y$ since u is tight and no edge covers two carving edges. This makes $\theta_b(y) \geq 2$, so (ii) holds. We conclude P is nondepleted.

This implies $p \in T_y$. Hence p does not have a merging child. Lemma 5.5 shows P is tight. □

CLAIM 4. *H has an edge with deeper end in $D \cap T_y$ that does not cover a carving edge. So $\theta_c(y) \geq 1$.*

Proof. The child of u on P is not breakable (otherwise (i) holds). Since P is penetrated, this means ≥ 2 components of $H - t(u)$ are adjacent (in H) to both D and $\overline{D}$. Since C_P is launched by y, it does not contain a surplus component (otherwise $\kappa(y) \geq 1$ and (iii) holds). So another cluster of $t(u)$, root or ear, is adjacent to both D and $\overline{D}$. Claim 2(a) shows the cluster is C_Q for an ear Q descending from $I(P)$. Claim 2(c) shows Q is a child ear of $I(P)$ and a vertex $q \in C_Q \cap Q$ is adjacent to u in H. Since P is tight, Claim 2(b) shows qu does not cover a carving edge. It also shows $q \in T_y$. Hence $\theta_c(y) \geq 1$. Claim 4 follows. □

CLAIM 5. *Every edge from D to $\overline{D}$ in H goes from a cluster of $t(u)$.*

Proof. Let xz be an edge of H with $x \in D$ and $z \in \overline{D}$. Assume $x \in t(u)$; otherwise we are done. Claim 2(a) shows $z \in t(u)$. Now it suffices to show $x \in T_y$. For that makes $\theta_b(y) \geq 1$ which, with $\theta_c(y) \geq 1$ (Claim 4), gives (ii).

If $x \in P$, then Claim 3 shows $x \in T_y$. If $x \notin P$, then Claims 2(b) and 2(c) show $x \in T_y$. □

CLAIM 6. *$d_H(C_P, U) \geq 2$. $d_H(C_P, D - C_P) \geq 2$. Hence $\theta_a(y) \geq 1$.*

Proof. Claim 1 shows that at least two edges e of H join D and U. Claim 5 shows each e has exactly one end in a cluster C of $t(u)$ with $C \subseteq D$. Now Claim 2(c) shows

$C = C_P$. So the two edges e give the first assertion of Claim 6.

For the second assertion, first note that $D - C_P \neq \emptyset$ since P is penetrated. We have already assumed $d_H(u) = 3$. By the preceeding paragraph, an edge of H leaving $D-C_P$ does not go to U. Since V is partitioned into sets $D-C_P$, $X = V-(D-C_P)-u$, and u, Proposition 5.4 shows $d_H(D - C_P, C_P) = d_H(D - C_P, X) \geq 2$.

We turn to the third assertion of Claim 6. C_P is launched by y, and we have already noted that it does not contain a surplus component (Claim 4). Hence $\theta_a(y) \geq 1$. □

CLAIM 7. *u has only one child. $d_H(U, u) \leq 1$.*

Proof. Suppose either part of the claim is false. We first show at least four edges of H launched by y cover the carving edge of P_u. If u has two children, this follows from Claim 1. If $d_H(U, u) \geq 2$, it follows from these two edges plus Claim 1 applied to D.

Now Claim 4 shows $\theta_c(y) \geq 2$. With $\theta_a(y) \geq 1$ (Claim 6) we get (ii). □

CLAIM 8. *$d_H(D, u) \leq 1$.*

Proof. Claim 5 shows $d_H(D, u) = \sum\{d_H(C, u) : \text{cluster } C \subseteq D\}$. First consider $C = C_P$. If $d_H(C_P, u) \geq 1$, then five edges leave C_P (Claim 6) so $\theta_a(y) \geq 2$. With $\theta_c(y) \geq 1$ (Claim 4), we get (ii). We conclude $d_H(C_P, u) = 0$.

Next consider $C \neq C_P$. Claim 2(b) shows that an edge from C to u does not cover a carving edge and, further, C is launched by y. If H contains two such edges, then $\theta_c(y) \geq 2$, and $\theta_a(y) \geq 1$ (Claim 6) gives (ii). We conclude there is only one such edge, as desired. □

Claims 7 and 8 show $d_H(u) \leq 2$, the desired contradiction. □

LEMMA 5.12. *Any tip $y \in \mathcal{Y}$ has*

$$(\kappa + \chi + \beta + \theta_d)(y) + (\theta_a + 2\theta_b + \theta_c)(y)/3 \geq 1.$$

Proof. Assume $\kappa(y) = 0$; otherwise we are done. Hence any merging ear launched by y is depleted. Choose vertex $u \in T_y$ to be the tip of a depleted ear as follows. If y does not launch a merging ear, then $u = y$. In the opposite case, choose u as the tip of a merging ear a, u, z, where $a \in T_y$ has the greatest depth possible. If two merging ears have the same first vertex a, choose u on the first merging ear at a.

Vertex u belongs to T_y. Hence it does not launch a merging ear. Lemma 5.5 shows u is tight. Now consider two cases.

First suppose u is a leaf. P_u is merging since a nonmerging depleted tip is not a leaf (by the claim of Lemma 4.4). Assume $\chi(y) = 0$; otherwise we are done. Since $\kappa(y) = 0$, Lemma 5.10 shows that u is breakable. Hence $\beta(y) \geq 1$.

Now suppose u is not a leaf. Since we are assuming $\chi(y) = 0$, Lemma 5.9 shows that u has a penetrated child ear. Now the desired conclusion follows from Lemma 5.11. □

We can now achieve the goal of proving inequality (6). Recall the definitions of all the right-hand quantities from section 5.3. Apply Lemma 5.12 to each $y \in \mathcal{Y}$. Define the t*-partition by breaking off a breakable vertex in $T_y - y$ whenever $\beta(y) \geq 1$. Lemma 5.3(i) shows that this gives a valid enhancement. This defines the quantity β of (6).

Clearly

$$\sum\{\chi(y) : y \in \mathcal{Y}\} \leq \chi.$$

(Inequality may hold since we ignore surplus ears launched by r and, further, we do not count the total number of surplus unions.) Analogous inequalities hold for θ_c and

θ_d. Lemma 5.3(ii) shows

$$\sum\{\theta_b(y) : y \in \mathcal{Y},\ \beta(y) = 0\} \le \theta_b.$$

Lemma 5.3(iii) shows the analogous inequality for θ_a. Let $\overline{\mu}_n$ be the number of nonmerging nondepleted ears. Then

$$\sum\{\kappa(y) : y \in \mathcal{Y}\} \le \kappa - \overline{\mu}_n.$$

Since $|\mathcal{Y}| = \overline{\mu} - \overline{\mu}_n$, Lemma 5.12 implies

$$\overline{\mu} - \overline{\mu}_n \le \kappa - \overline{\mu}_n + \chi + \beta + \theta_d + (\theta_a + 2\theta_b + \theta_c)/3.$$

This amounts to (6).

6. Efficient implementation. This section presents an implementation of the algorithm that runs in time $O(m\alpha(m,n))$. It is straightforward to find the biconnected components and implement Phases I and III in linear time. So we limit our attention to Phase II. The implementation must incorporate Rules 1 and 3 of section 5 (Rule 2 is trivial). This section first describes how the t-partition is maintained and manipulated. Then it describes how long ears are constructed. The remaining details of Figure 3 are obvious.

6.1. The t-partition. We maintain the t-partition using the following properties of F. For any vertex x, let $\ell(x)$ be the ancestor of x in F that belongs to $t(x)$ and has minimum depth. Lemma 4.1 shows that $t(x)$ contains every vertex on the path in F from x to $\ell(x)$. Write $fl(x)$ for $f(l(x))$. Note that every $y \in t(x)$ has $f\ell(y) = f\ell(x)$. (Recall Figure 5.) In the following lemma assume $f(r) = r$.

LEMMA 6.1. *For any $i \ge 0$, a back edge xz traverses $> i+1$ distinct t-sets if and only if z is an ancestor of $[f\ell]^i(x)$ in T and $z \notin t([f\ell]^i(x))$.*

Proof. If $r \notin t(x)$, then the deepest ancestor of x in F not belonging to $t(x)$ is $fl(x)$. Hence for $i \ge 1$, if $r \notin t([fl]^{i-1}x)$, then the path in F from x to $[fl]^i(x)$ contains vertices in exactly $i+1$ distinct t-sets. If z is an ancestor (in T) of $[fl]^i(x)$ but not $[fl]^{i+1}(x)$ and $z \notin t([fl]^i(x))$, then an edge xz traverses exactly $i+2$ t-sets. □

The t-partition is maintained by a disjoint-set data structure. In addition, each set $t(x)$ is labelled by the node $f\ell(x)$. As already noted, this label is well defined. The labels allow `Multi-Merge` to be implemented using a number of finds proportional to the number of unions. Line 3 of `Short_Ear` is implemented using Lemma 6.1. It performs $O(1)$ finds per back edge, since i equals 2 or 3. Line 3(i) of `Long_Ear` uses one find per back edge.

To implement Rule 1 in line 3(ii) of `Long_Ear`, tentatively choose the back edge yz from y that has $z \notin t(a)$ and minimum $d(z)$. (Throughout this section $d(v)$ denotes the depth of vertex v in tree T, as in section 3.) Find the maximum index i with vertex $v = [fl]^i(a)$ descending from z. A merging ear with tip y merges at most $i+2$ t-sets. Any back edge yz' with z' an ancestor of v and $z' \notin t(v)$ gives such an ear. If no such z' exists, the back edge yz gives an ear merging $i+1$ t-sets, the maximum possible in this case. In either case, `Multi-Merge` performs at least i unions for this ear.

We conclude that the disjoint-set data structure performs a total of $O(m)$ find operations in a universe of n elements. Thus the total time for manipulating the t-partition is $O(m\alpha(m,n))$ [1].

6.2. Constructing long ears. This section describes how lines 2–3 of `Long_Ear` find edge $c \in C$ and back edge yz to construct the new ear. We accomplish this in linear time, implying the desired time bound for Phase II.

At any point during the execution of `Long_Ear`, let R be the subtree of edges of T that already belong to long ears. The idea of the implementation is to maintain an "active" subtree S, $R \subseteq S \subseteq T$. When xx' is chosen in `Long_Ear` (line 6), the subtree of S rooted at x' will lead to the edges of C that can be covered by new ears having xx' as their first edge. We now provide the details.

Phase I labels each back edge e with the carving edge $c[e]$ covered by e, if it exists; if not, then $c[e] = \Lambda$. It is easy to do this labelling in linear time after identifying the connected components of $T - C$.

In Phase II, say that vertex x gets *visited* the first time it is reached in line 6 of `Long_Ear`. Also, as we have implicitly done, say that a back edge is *directed to* its shallower vertex. `Long_Ear` maintains a list $L[c]$ for each $c \in C$, defined by

$$L[c] = \{e : e \text{ is a back edge covering } c \text{ and directed to an already visited vertex}\}.$$

Subtree S is maintained according to the following invariant.

S-Invariant. The pendant edges of $S - R$ are precisely the edges of $c \in C - R$ with $L[c] \neq \emptyset$.

`Long_Ear` constructs L lists and grows S as follows. When line 6 visits x, it first does some processing, described below, that implements Rule 3. Then it scans each back edge e directed to x. If $c[e] \neq \Lambda$, do the following: Add e to $L[c[e]]$. If $c[e]$ is not in S, join it into S by the tree path to its first ancestor already in S.

This procedure maintains the definition of L. It also preserves the S-Invariant, since the definition of tree carving shows the edges added to S do not descend from a pendant edge of $S - R$.

In order to implement Rule 3, we use another variable, edge c^*. If c^* is defined, then it is used as edge c of line 2 to form the merging ear required by Rule 3. We define c^* when visiting x. The complete procedure for visiting x is as follows.

When x is first reached in line 6, if x already has descendants in S not in P_x, follow a path in S from x to a pendant edge. Take that pendant edge to be c^*. Then scan the back edges directed to x, as described above. Finally, if c^* is defined, choose x' (for the first child ear at x) as the child of x that is an ancestor of c^*. Otherwise choose x' arbitrarily.

Note that when this procedure begins, all proper ancestors of x have been visited (by Rule 2) but no descendant of x has been visited. Hence for every edge $c \in C - R$ descending from x, $L[c]$ contains precisely the edges that can be used to form a merging child ear at x (by the definition of L). If such a c exists, the procedure's edge c^* qualifies, since the S-Invariant guarantees that $c^* \in C$ and $L[c^*] \neq \emptyset$.

Now we describe the implementation of lines 2–3 of `Long_Ear`(b). The purpose of line 2 is to define c. If c^* is defined, then $c = c^*$. This gives a merging ear, satisfying Rule 3.

If c^* is not defined, then follow a path in S from b to a pendant edge of S. Take that edge as c. The S-Invariant guarantees that $c \in C$ and $L[c] \neq \emptyset$. Furthermore, any edge of $L[c]$ can be used to form an ear P_b. This follows from the definition of $L[c]$, since a and all its ancestors have been visited, but no descendant of b has been visited.

Next we describe line 3. If $L[c]$ contains an edge yz with $z \notin t(a)$, choose one with maximum $d(y)$ to define a merging ear. Otherwise choose any edge $yz \in L[c]$ with

maximum $d(y)$ to define a nonmerging ear. In both cases, 3(ii) is satisfied. Note that for merging ears the procedure described in section 6.1 determines the final merging ear. Finally, line 4 adds the tree path from b to y to both R and S.

Appendix A. Analysis for *k*-edge connectivity. This appendix extends the analysis of section 5 to k-edge connectivity. Specifically, let B denote the set of nontree edges in the approximation algorithm's solution graph. (B consists of all edges added after Phase I.) We show that for any integer $k \geq 3$,

$$|B| \leq (5/2k)\ \epsilon_k. \tag{12}$$

(It is easy to see that this implies $|A| \leq (9/2k)\epsilon_k$. So for $k = 3$ we again have a 3/2 performance ratio.) This result is used in [5] to get an approximation algorithm for k-ECSS. We still assume Rules 1–3 of section 5.1 are used in the algorithm, but there are no other additional rules.

The core of the derivation is a new version of Lemma 5.11. (Taking $k = 3$, the new version can replace the one in section 5, but the new argument is slightly longer.) There are a number of additional changes, but all changes are in sections 5.3 and 5.4. The new versions of these sections are Appendices A.1 and A.2, respectively.

A.1. The approach for general *k*. We make some small changes in the definitions of our fundamental quantities as follows: If $k + s$ edges of H cover an edge $e \in C$, then e is *redundantly covered* s times. A vertex with degree $k + s$ in H has *surplus degree* s. Quantities $\kappa, \beta, \theta_a, \theta_b$, and θ_c are defined exactly as before. (For θ_c use the new definition of redundant covering.) The new definition of θ_d is

$$\theta_d = \text{the total surplus degree of all vertices in } H.$$

We modify the key inequality (6) in two ways, changing a factor 3 to k and changing the term involving θ_d as follows:

$$\overline{\mu} \leq \kappa + \chi + \beta + (\theta_a + 2\theta_b + \theta_c + \theta_d)/k. \tag{13}$$

The proof that this inequality implies (12) is entirely analogous to the proof given in section 5.3. For completeness, the rest of this section gives all the details.

Our three lower bounds are

$$\varepsilon_k \geq k\gamma + \theta_c; \tag{14}$$

$$\varepsilon_k \geq (k/2)n + \theta_d/2; \tag{15}$$

$$\varepsilon_k \geq (k/2)(n - 1 + \kappa + \beta - \nu) + \theta_a/2 + \theta_b. \tag{16}$$

It is obvious that (15) holds for the new definition of θ_d. The two other inequalities are the same as in section 5.3 with the factor 3 changed to k.

For convenience we restate here the previous equation defining δ as follows:

$$\overline{\mu} - \sigma - \chi = \gamma/2 + \delta. \tag{17}$$

The definition of B gives

$$|B| \leq \gamma + \sigma + (n - 1 - \nu) = n - 1 + \overline{\mu} - \sigma - \chi = n - 1 + \gamma/2 + \delta.$$

Combining $1/(2k)$ times inequality (14) with $2/k$ times inequality (15) gives $(5/2k)\,\varepsilon_k \geq n + \gamma/2 + \theta_c/(2k) + \theta_d/k$. Hence we can assume

$$\delta > \theta_c/(2k) + \theta_d/k$$

since otherwise we are done. To handle this case, we will show that (14) and (16) imply

$$(5/2k)\,\varepsilon_k \geq n - 1 + \gamma/2 + 2\delta - \theta_c/(2k) - \theta_d/k. \tag{18}$$

Inequality (18), together with our assumption on δ, implies the desired result.

Reexpress $\overline{\mu}$ exactly as in section 5.3 as follows: Substituting (3) into (17) and using (4) gives

$$\overline{\mu} - \nu - \chi = 2\delta.$$

Combining this with (13) gives

$$\kappa + \beta - \nu \geq \overline{\mu} - \chi - \nu - (\theta_a + 2\theta_b + \theta_c + \theta_d)/k = 2\delta - (\theta_a + 2\theta_b + \theta_c + \theta_d)/k.$$

Thus (16) gives

$$\varepsilon_k \geq (k/2)(n-1+2\delta-(\theta_a+2\theta_b+\theta_c+\theta_d)/k)+\theta_a/2+\theta_b = (k/2)(n-1+2\delta-(\theta_c+\theta_d)/k).$$

Combining $2/k$ times this inequality with $1/(2k)$ times (14) gives the desired inequality (18).

A.2. Bounding $|\mathcal{Y}|$ for general k. As in section 5.4, we still deal with the given graph G and the fixed 3-ECSS H. We do not use Proposition 5.4. Lemmas 5.5–5.9 are properties of G alone and do not involve the connectivity of G, so they are still valid. Lemma 5.10 involves H but not its connectivity, so it is still valid. Our analogue of Lemma 5.11 uses the same quantities $\kappa(y), \chi(y), \beta(y), \theta_a(y)$, and $\theta_b(y)$. The two remaining quantities are defined slightly differently as follows:

$\theta_c(y)$ = the number of edges of H launched by y not covering a carving edge or redundantly covering the carving edge of an ear launched by y;

$\theta_d(y)$ = the total surplus degree of vertices of T_y in H.

The new definition of $\theta_d(y)$ makes it consistent with θ_d. The definition of $\theta_c(y)$ differs from section 5.4 in a substantive way: It associates a carving edge vw that has $v \in T_y$ and $w \in \mathcal{Y}$ with y rather than with w as in section 5.4.

LEMMA A.1. *For $y \in \mathcal{Y}$ let $u \in T_y$ be the tip of a depleted ear. Suppose u does not launch a merging ear and u has a penetrated child ear. Then either*

(i) *u has a breakable child belonging to T_y, or*

(i) *$\theta_a(y) + 2\theta_b(y) + \theta_c(y) + \theta_d(y) \geq k$, or*

(iii) *$\kappa(y) + \chi(y) \geq 1$.*

Proof. As in Lemma 5.11, we argue by contradiction. As in that lemma, u is tight. We use the same notation U, C_X, $\overline{S}$, P, p, D with one change noted at the beginning of Case 2 below. We use Claims 2 and 4 but no others. Their proofs are unchanged. The proof of Claim 4 shows that there is an ear Q with first vertex in P, nonempty cluster C_Q, and vertex $q \in C_Q \cap Q \cap T_y$ adjacent to u in H. Furthermore, edge qu does not cover a carving edge. Throughout this proof we write d for the degree function in H, d_H.

Case 1. P is depleted.

We show that (ii) holds in this case, more specifically, $2\theta_b(y) + \theta_d(y) \geq k$. Claim 2(c) and the fact that H is k-edge connected implies these two inequalities:

$$\begin{array}{lcll} d(D) & = & d(P,U) + d(D,u) & \geq \quad k, \\ d(D+u) & = & d(P,U) + d(u,\overline{D}) & \geq \quad k. \end{array}$$

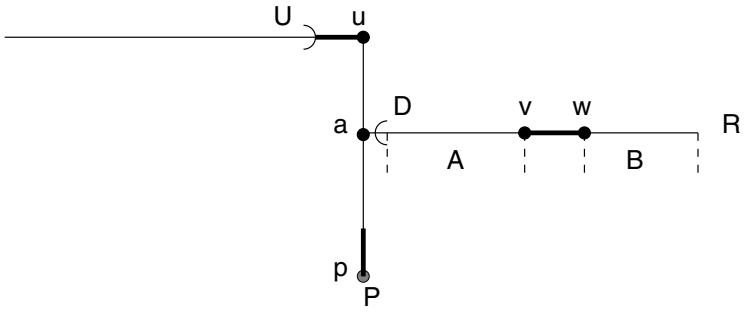


FIG. 7. *Proper child ear R.*

Adding them gives $2d(P,U)+d(u) \geq 2k$. With $\theta_d(u) = d(u)-k$ this gives $2d(P,U)+\theta_d(y) \geq k$. Now it suffices to show that every edge $xz \in E(H)$ from P to U is counted in $\theta_b(y)$. This requires that (a) x and z belong to the same t-set, and (b) xz is properly launched by y, i.e., $x \in T_y - y$. For (a) observe that Claim 2(a) shows $z \in t(u)$, and P depleted shows $x \in t(u)$. For (b) observe that $x \neq p$ since otherwise edge xz covers two carving edges (recall Figure 6).

Case 2. P is nondepleted.

As in Claim 3, the assumption implies P is tight. Consider (see Figure 7) a child ear R of P whose first vertex is a and whose carving edge is vw (with v the parent of w as usual). Say R is a *proper* child ear of P if $a \neq p, u, v$. For instance, the ear Q from Claim 4 is proper, by Claim 2(b).

Now consider a proper child ear R of P. Redefine D to be the set of all descendants of $I(R)$. (We will manipulate this new D in a manner analogous to the original D in Case 1.) Let $\theta_c(D)$ denote the contribution to $\theta_c(y)$ of all edges that have at least one end in D.

OBSERVATION 1. $\theta_c(D) \geq k/2$.

Proof. Partition D into two sets: A consists of all vertices that descend in F from a vertex of $I(R)$ that precedes (or equals) v; B consists of all vertices that descend in F from a vertex of $I(R)$ that follows (or equals) w. Both sets are nonempty ($v \in A, w \in B$). Claim 2(c) and H k-edge connected imply

$$\begin{aligned} d(D) &= d(A,P)+d(B,P) \geq k, \\ d(A) &= d(A,P)+d(A,B) \geq k. \end{aligned}$$

Adding the inequalities gives $2d(A,P)+d(B) \geq 2k$. Equivalently, $d(A,P)+(d(B)-k)/2 \geq k/2$. So it suffices to show $\theta_c(D) \geq d(A,P)+(d(B)-k)$.

$d(B)$ equals the number of edges covering the carving edge vw. Hence $d(B)-k$ is the number of times vw is redundantly covered. This is included in $\theta_c(D)$ because R is launched by y. (Note that this is not true if we use the original definition of $\theta_c(y)$.)

It remains to show that every edge $xz \in E(H)$ from A to P is counted in $\theta_c(y)$. For this it suffices to prove (a) xz does not cover a carving edge, and (b) xz is launched by y, i.e., $x \in T_y$.

We prove (a) by contradiction. Suppose xz covers $e \in C$. Let s be the deepest vertex of R that is an ancestor of x. We will show that e descends from s in F. $x \in A$ shows that the tree path from s to $f(s)$ does not contain the carving edge of R. R proper and P tight implies that the tree path from $f(s)$ to z does not contain the carving edge of P. Hence the carving edge e covered by xz must descend from s in F.

Now Rule 3 shows that R has a child ear, launched by u, that is merging. This follows since, after `Long_Ear` has constructed ear R, the possible ear s, x, z is merging (as $s \neq z$). But the existence of such a merging ear contradicts the hypothesis of Lemma A.1.

The contradiction proves (a). It is easy to see that (a) implies (b). □

Consider again the child ear Q of Claim 4. Edge qu is not a short ear (since $q \notin t(u)$). Hence $f(q) \in t(u)$ when `Short_Ear`$(\cdot, 2)$ scans q (i.e., when line 2 of `Short_Ear` has $x = q$ and $i = 2$). Consider the first ear that adds a vertex of $I(P)$ to $t(u)$. Since `Short_Ear` works bottom-up, Claim 2(c) implies this ear is a short ear xu for some vertex x internal to a child ear R of P. Claim 2(b) shows that x is an ancestor of the carving edge of R, so R is a proper child.

Case 2.1. $R \neq Q$.

Observation 1 applies to both R and Q, since both are proper. Write $D(R)$ for the set of all descendants of $I(R)$ and, similarly, write $D(Q)$. No edge contributes to both $\theta_c(D(R))$ and $\theta_c(D(Q))$. (More generally, let Q' range over all child ears of P except R. An edge leaving $D(R)$ does not leave $D(Q')$ or cover any edge of Q'. Similarly, an edge covering an edge of R does not leave $D(Q')$ or cover any edge of Q'.) Now Observation 1 shows $\theta_c(y) \geq k/2 + k/2 = k$. Hence Lemma A.1(ii) holds.

Case 2.2. $R = Q$.

For consistency we refer to the ear as R. Let vw be the carving edge of R. Recall vertices $x \in I(R) \cap t(u)$ and $q \in I(Q) - t(u)$, both adjacent to u. This implies $x \neq q$, and both vertices are proper ancestors of w (Claim 2(b)).

Let D denote the set of all descendants of $I(R)$. Partition D into three sets: A (C) consists of all vertices that descend in F from the first (last) vertex of $I(R)$, respectively; B consists of all vertices that descend in F from a vertex of $I(R)$ other than the first or last. (For the remainder of the proof we are discarding the use of "C" as the set of carving edges.) All three sets are nonempty: Since $R = Q$ is tight, $w \in C$. Since x and q are proper ancestors of w in $I(R)$, $A, B \neq \emptyset$.

Claim 2(c) and H k-edge connected imply

$$\begin{array}{rcl} d(A) & = & d(A,P) + d(A,B) + d(A,C) \quad \geq \quad k, \\ d(B) & = & d(B,P) + d(B,A) + d(B,C) \quad \geq \quad k, \\ d(D) & = & d(A,P) + d(B,P) + d(C,P) \quad \geq \quad k. \end{array}$$

Adding the inequalities gives $2d(A \cup B, P) + 2d(A, B) + d(C) \geq 3k$. Equivalently, $d(A \cup B, P) + d(A, B) + (d(C) - k)/2 \geq k$. We will show $\theta_c(y) \geq d(A \cup B, P) + d(A, B) + (d(C) - k)$, giving Lemma A.1(ii) as desired.

As in the proof of Observation 1, $d(C) - k$ is the number of times the carving edge vw is redundantly covered, and it is included in $\theta_c(y)$ since R is launched by y. Similarly, the proof of Observation 1 shows that every edge $xz \in E(H)$ from $A \cup B$ to P is counted in $\theta_c(y)$. Finally, we must show the same for edges from A to B. This follows by the same argument as in Observation 1. □

The analogue of Lemma 5.12 is that any tip $y \in \mathcal{Y}$ has

$$(\kappa + \chi + \beta)(y) + (\theta_a + 2\theta_b + \theta_c + \theta_d)(y)/k \geq 1.$$

This is proved by exactly the same argument as before, using Lemma A.1 in place of Lemma 5.11. The desired inequality (13) is then proved by the argument for (6) in section 5.4.

REFERENCES

[1] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, McGraw-Hill, New York, 1990.

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

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

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

[5] H. N. Gabow, *Better performance bounds for finding the smallest k-edge connected spanning subgraph of a multigraph*, in Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, ACM, New York, 2003, pp. 460–469.

[6] N. Garg, V. S. Santosh, and A. Singla, *Improved approximation algorithms for biconnected subgraphs via better lower bounding techniques*, in Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, ACM, New York, 1993, pp. 103–111.

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

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

[9] S. Khuller and U. Vishkin, *Biconnectivity approximations and graph carvings*, J. ACM, 41 (1994), pp. 214–235.

[10] S. Vempala and A. Vetta, *Factor* 4/3 *approximations for minimum* 2*-connected subgraphs*, in Approximation Algorithms for Combinatorial Optimization, K. Jansen and S. Khuller, eds., Lecture Notes in Comput. Sci. 1931, Springer-Verlag, Berlin, 2000, pp. 262–273.

[11] D. B. West, *Introduction to Graph Theory*, 2nd ed., Prentice-Hall, Upper Saddle River, NJ, 2001.

SIAM J. DISCRETE MATH.
Vol. 18, No. 1, pp. 71–82

COUNTING STRINGS WITH GIVEN ELEMENTARY SYMMETRIC FUNCTION EVALUATIONS II: CIRCULAR STRINGS*

C. R. MIERS† AND F. RUSKEY‡

Abstract. Let α be a string over an alphabet that is a finite ring, R. The kth elementary symmetric function evaluated at α is denoted $T_k(\alpha)$. In a companion paper we studied the properties of $\mathbf{S}_R(n;\tau_1,\tau_2,\ldots,\tau_k)$, the set of length n strings for which $T_i(\alpha)=\tau_i$. Here we consider the set, $\mathbf{L}_R(n;\tau_1,\tau_2,\ldots,\tau_k)$, of equivalence classes under rotation of aperiodic strings in $\mathbf{S}_R(n;\tau_1,\tau_2,\ldots,\tau_k)$, sometimes called Lyndon words. General formulae are established and then refined for the cases where R is the ring of integers $\mathbb{Z}_q$ or the finite field $\mathbb{F}_q$.

Key words. elementary symmetric function, combinatorial enumeration, integers mod q, Lyndon word, Möbius inversion, multinomial coefficient, finite field

AMS subject classifications. 05A15, 05E05, 05A19

DOI. 10.1137/S0895480103428053

1. Introduction.

The main purpose of this paper is to count certain equivalence classes of strings over $\mathbb{Z}_q$, the ring of integers mod q, and over the finite field $\mathbb{F}_q$. The equivalence classes contain all strings that are rotationally equivalent (sometimes called *conjugate* [7]) and that achieve specified values when regarded as the parameters of elementary symmetric functions. Aside from the intrinsic interest of the enumerative formulae and the techniques used to derive them, this paper can be viewed as part of a program to enumerate certain classes of polynomials with coefficients in a finite ring and whose coefficients are prescribed. In [2], degree n monic irreducible polynomials over $\mathbb{F}_2$ with prescribed coefficients for x^{n-1} and x^{n-2} were enumerated. If such a polynomial is factored in a splitting field, these coefficients can be interpreted as the first and second elementary symmetric functions evaluated at the string of coefficients in the factorization. The techniques in [2] (and in [11]) rely on the relationship between Lyndon words and irreducible polynomials. The relationship between strings, polynomials, and elementary symmetric functions generalizes. If a string α has its alphabet in a finite commutative ring R, we can evaluate the kth elementary symmetric function T_k at α. This evaluation depends on the *profile* $\mathbf{k}=\langle k_1,k_2,\ldots,k_{|R|-1}\rangle$, where k_i is the frequency with which ring element x_i occurs in α. The relationship between strings, polynomials, and elementary symmetric functions is contained in the map $\alpha\mapsto A_{\mathbf{k}}(z)=\prod_{j=1}^{|R|-1}(1+x_jz)^{k_j}$, since $T_m(\alpha)=[z^m]A_{\mathbf{k}}(z)$. In [8] we exploit this relationship to compute $S_{\mathbb{Z}_p}(n;\tau_1,\tau_2,\ldots,\tau_t)$, the number of strings over $\mathbb{Z}_p$ of length n for which $T_m(\alpha)=\tau_m$. Related results can be found in [5], [9], [11].

2. Notation and preliminaries.

In what follows we will assume R is a finite commutative ring with identity, denoted $\mathbf{1}$. In this case R has a characteristic c which is the least positive integer such that the c-fold sum $\mathbf{1}+\mathbf{1}+\cdots+\mathbf{1}=0$. If $d\in\mathbb{Z}^+$, then $d\in R$, where d is the d-fold sum $\mathbf{1}+\mathbf{1}+\cdots+\mathbf{1}\bmod(c)$.

*Received by the editors May 15, 2003; accepted for publication (in revised form) February 17, 2004; published electronically July 20, 2004.
http://www.siam.org/journals/sidma/18-1/42805.html
†Department of Mathematics, University of Victoria, Victoria, BC V8W-3P6, Canada (crmiers@math.uvic.ca). This author's research was supported in part by a UVic faculty research grant.
‡Department of Computer Science, University of Victoria, Victoria, BC V8W-3P6, Canada. This author's research was supported in part by NSERC.

We consider strings $\alpha = a_1 a_2 \cdots a_n$, where each $a_i \in R$, and we define the *k-trace* of α, denoted $T_k(\alpha)$, as the sum

$$T_k(\alpha) = \sum_{1 \le i_1 < i_2 < \cdots < i_k \le n} a_{i_1} a_{i_2} \cdots a_{i_k}.$$

These are the elementary symmetric functions of $a_1, a_2, \ldots, a_n$. Occasionally we will call T_1 the *trace*, T_2 the *subtrace*, and T_3 the *subsubtrace*. The trace terminology is used, in analogy with the theory of finite fields, since $(-1)^k T_k(\alpha)$ is the coefficient of z^{n-k} in the polynomial $(z-a_1)(z-a_2)\cdots(z-a_n)$ (see [10]).

By $S_R(n; \tau_1, \tau_2, \ldots, \tau_k)$ we denote the number of strings α over R of length n for which $T_i(\alpha) = \tau_i$ for $i = 1, 2, \ldots, k$. Obviously if $k = 0$, then $S_R(n) = q^n$, where q is the number of elements in R. It is also true that $S_R(n; t) = q^{n-1}$ for any $t \in R$, since $T_1(\alpha x)$ takes on distinct values for each $x \in R$; we use here only the fact that R is an additive group.

The notation $[\![P]\!]$ for a proposition P has the value 1 if P is true and the value 0 if P is false. This is "Iverson's convention," as used in [6].

The numbers $S_R(n; \tau_1, \tau_2, \ldots, \tau_k)$ satisfy the following recurrence relation. If $n = 1$, then $S_R(n; \tau_1, \tau_2, \ldots, \tau_k) = [\![\tau_2 = \cdots = \tau_k = 0]\!]$, and for $n > 0$,

$$S_R(n; \tau_1, \tau_2, \ldots, \tau_k) = \sum_{x \in R} S_R(n-1; \rho_1, \rho_2, \ldots, \rho_k), \tag{2.1}$$

where $\rho_0 = 1$, and $\rho_j = \tau_j - \rho_{j-1} x$ for $j = 1, 2, \ldots, k$. This recurrence relation holds even if R is not commutative. It allows us to evaluate $S_R(n; \tau_1, \tau_2, \ldots, \tau_k)$ in $O(nr^k)$ ring and integer operations (by, in effect, creating a size nr^k table of S_R evaluated on all strings of length at most n on all possible values of the first k k-traces). The properties of $S_{\mathbb{Z}_p}$ for p prime are studied in [8].

A *rotation* of a string α is any string β that can be written as $\beta = \gamma\delta$, where $\alpha = \delta\gamma$. A string α is *aperiodic* if there are no nonempty strings γ and δ such that $\alpha = \gamma\delta = \delta\gamma$. Let $A_R(n; \tau_1, \tau_2, \ldots, \tau_k)$ denote the number of aperiodic strings α over R of length n for which $T_i(\alpha) = \tau_i$ for $i = 1, 2, \ldots, k$. Since every rotation of an aperiodic string is distinct, $A_R(n; \tau_1, \tau_2, \ldots, \tau_k)$ is divisible by n. The number $L_R(n; \tau_1, \tau_2, \ldots, \tau_k) = (1/n) A_R(n; \tau_1, \tau_2, \ldots, \tau_k)$ is the number of equivalence classes of aperiodic strings under rotation. The lexicographically least representatives of these equivalence classes are often called *Lyndon words* [7].

LEMMA 2.1. *For all $k \ge 1$ and $d \ge 1$,*

$$T_k(\alpha^d) = \sum_{\nu_1 + 2\nu_2 + \cdots + k\nu_k = k} \binom{d}{\nu_1, \ldots, \nu_k, d-(\nu_1+\cdots+\nu_k)} T_1(\alpha)^{\nu_1} T_2(\alpha)^{\nu_2} \cdots T_k(\alpha)^{\nu_k}.$$

Proof. From the string $\alpha^d = \alpha_1 \alpha_2 \cdots \alpha_d$, where $\alpha_i = \alpha$ for all i, we need to select k positions in all possible ways. We classify those ways according to the distribution $(\nu_1, \nu_2, \ldots, \nu_k)$, where ν_j is the number of α_i's containing j of the selected positions. Such an α_i will contribute a multiplicative factor of $T_j(\alpha)$ to the sum, with $T_1(\alpha)^{\nu_1} T_2(\alpha)^{\nu_2} \cdots T_k(\alpha)^{\nu_k}$ being the total contribution for a given distribution and selection of the α_i's. There are $\binom{d}{\nu_1, \ldots, \nu_k, d-(\nu_1+\cdots+\nu_k)}$ ways to associate a distribution with particular α_i's. Finally, a distribution is valid if and only if $\nu_1 + 2\nu_2 + \cdots + k\nu_k = k$. □

Note that the multinomial coefficient can be written as

$$\binom{d}{\nu_1, \ldots, \nu_k, d-V_k} = \binom{d}{\nu_1} \binom{d-V_1}{\nu_2} \cdots \binom{d-V_{k-1}}{\nu_k}, \tag{2.2}$$

where $V_j = \nu_1 + \nu_2 + \cdots + \nu_j$.

If $\mathbf{t} = (t_1, t_2, \ldots, t_k) \in R^k = R \times R \times \cdots \times R$ and d is a natural number, define the map $\theta_d : R^k \to R^k$ as $\theta_d(\mathbf{t}) = \mathbf{u}$, where $\mathbf{u} = (u_1, u_2, \ldots, u_k)$ has the value, mod c,

$$u_j = \sum_{\nu_1+2\nu_2+\cdots+j\nu_j=j} \binom{d}{\nu_1, \ldots, \nu_j, d-(\nu_1+\cdots+\nu_j)} t_1^{\nu_1} t_2^{\nu_2} \cdots t_j^{\nu_j} \tag{2.3}$$

$$= \sum_{\nu_1+2\nu_2+\cdots+j\nu_j=j} d^{(\nu_1+\nu_2+\cdots+\nu_j)} \frac{t_1^{\nu_1}}{\nu_1!} \frac{t_2^{\nu_2}}{\nu_2!} \cdots \frac{t_j^{\nu_j}}{\nu_j!}. \tag{2.4}$$

We use in (2.4) the notation $d^{(m)} = d(d-1)\cdots(d-m+1)$ for the falling factorial. In light of Lemma 2.1, since every periodic string is the repeated concatenation of an aperiodic string,

$$S_R(n; \mathbf{u}) = \sum_{d|n} \sum_{\mathbf{t} \in R^k} [\![\theta_d(\mathbf{t}) = \mathbf{u}]\!] A_R\left(\frac{n}{d}; \mathbf{t}\right). \tag{2.5}$$

In principle (2.5) may be inverted recursively as long as all the solutions $\mathbf{t}$ to the equations $\mathbf{u} = \theta_d(\mathbf{t})$ can be determined. That is, when $d = 1$, the only solution is $t_j = u_j$ for $j = 1, 2, \ldots, k$, giving the term $A_R(n; t_1, t_2, \ldots, t_k)$. All other terms have the first parameter less than n. However, our aim is to invert (2.5) explicitly whenever possible.

In what follows it often happens that the equation $\mathbf{u} = \theta_d(\mathbf{t})$ has at most one solution for particular values of n and $\mathbf{u}$; i.e., if it has a solution, then $\mathbf{t} = \theta_d^{-1}(\mathbf{u})$. Then (2.5) becomes

$$S_R(n; \mathbf{u}) = \sum_{d|n} [\![\theta_d^{-1}(\mathbf{u}) \text{ exists}]\!] A_R\left(\frac{n}{d}; \theta_d^{-1}(\mathbf{u})\right). \tag{2.6}$$

Let us explicitly write out (2.4) for $k = 1, 2, 3, 4$ as a preparation for some examples to follow and to better understand the nature of the equation.

$$u_1 = dt_1, \tag{2.7}$$

$$u_2 = dt_2 + \binom{d}{2} t_1^2, \tag{2.8}$$

$$u_3 = dt_3 + d(d-1)t_1t_2 + \binom{d}{3} t_1^3, \tag{2.9}$$

$$u_4 = dt_4 + d(d-1)t_1t_3 + \binom{d}{2} t_2^2 + (d-2)\binom{d}{2} t_1^2 t_2 + \binom{d}{4} t_1^4. \tag{2.10}$$

Next we state a fundamental multiplicative property of the mapping θ.

LEMMA 2.2. *For all natural numbers a and b,*

$$\theta_a(\theta_b(\mathbf{t})) = \theta_{ab}(\mathbf{t}).$$

Proof. Let $h_d(z) = \sum_{n\geq 1} u_n z^n = \sum_{n\geq 1} (n! u_n) z^n / n!$. Then $h_d(z) = f_d(g(z))$, where

$$f_d(z) = \sum_{n\geq 1} d^{(n)} \frac{z^n}{n!} = \sum_{n\geq 1} \binom{d}{n} z^n \quad \text{and} \quad g(z) = \sum_{n\geq 1} n! t_n \frac{z^n}{n!} = \sum_{n\geq 1} t_n z^n,$$

by the Faà di Bruno formula (see Comtet [3, pp. 137–138]). Our lemma then reduces to the statement that $f_a(f_b(g(z))) = f_{ab}(g(z))$, which we can prove by showing that $f_a(f_b(z)) = f_{ab}(z)$. But this is a trivial substitution since $f_a(z) = (1+z)^a - 1$. □

Note that the lemma holds where a and b are formal variables; but we will use it only when they are members of $\mathbb{Z}_q$.

For fixed k and q, we will be interested in the period of the sequence $\binom{n}{k} \bmod q$ for $n = 0, 1, 2, \ldots$. The value of this period has been determined by Zabek [12], and we state this result below.

THEOREM 2.3 (Zabek). *Let the prime factorization of q be*

$$q = p_1^{n_1} p_2^{n_2} \cdots p_e^{n_e},$$

where the p_i's are distinct primes and the n_i's are positive integers. The period of the sequence $(\binom{0}{j}, \binom{1}{j}, \binom{2}{j}, \ldots) \bmod q$ is denoted q'_j and is equal to

$$q'_j = \prod_{i=1}^{e} p_i^{n_i + d_i}, \quad \textit{where } d_i = \lfloor \log_{p_i} j \rfloor.$$

COROLLARY 2.4. *If p is prime, then the period of the sequence $(\binom{0}{j}, \binom{1}{j}, \binom{2}{j}, \ldots) \bmod p$ is $p^{1+\lfloor \log_p j \rfloor}$.*

We note that Zabek's theorem (together with (2.2)) implies that $\theta_a(\mathbf{t}) : R^k \to R^k$ is periodic in the sense that

$$\theta_{a+q'_k}(\mathbf{t}) = \theta_a(\mathbf{t}). \tag{2.11}$$

Hence we will consider the integer subscripts of θ_a as integers $\bmod q'_k$; i.e., $a \in \mathbb{Z}_{q'_k}$. By $\mathbb{Z}_q^*$ we denote the group of units (invertible elements) of $\mathbb{Z}_q$.

COROLLARY 2.5. *If $a \in \mathbb{Z}_{q'_k}^*$, then θ_a is invertible and $\theta_a^{-1} = \theta_{a^{-1}}$.*

Proof. This follows from the fact that $\theta_{\mathbf{1}}$ is the identity mapping and from Lemma 2.2. □

3. A generalized Möbius inversion. In this section we prove a generalized Möbius inversion that is very useful in obtaining expressions for $A_R(n; \tau_1, \tau_2, \ldots, \tau_k)$ and $L_R(n; \tau_1, \tau_2, \ldots, \tau_k)$, when $R = \mathbb{Z}_q$ or $R = \mathbb{F}_q$. In this section q can be any positive integer. In the expressions below the reader should be careful about the context in which d is used. We use, here and throughout the remainder of the paper, the notation $d \equiv x(q)$ to mean $d \equiv x \bmod q$.

LEMMA 3.1. *If $n \bmod q \in \mathbb{Z}_q^*$, then*

$$\sum_{x \in \mathbb{Z}_q^*} \sum_{\substack{d \mid n \\ d \equiv x(q)}} \mu\left(\frac{n}{d}\right) = [\![n = 1]\!].$$

Proof. The defining recurrence relation for the Möbius function is $\sum_{d|n} \mu(d) = [\![n = 1]\!]$ (see, e.g., [6]). The lemma follows from this and the observation that if $n \bmod q \in \mathbb{Z}_q^*$ and $d|n$, then $d \bmod q \in \mathbb{Z}_q^*$. □

The following theorem was proven for $q = 2$ in [4] and for $q = 4$ in [2].

THEOREM 3.2. *Let f_x and g_x be sets of functions indexed by $x \in \mathbb{Z}_q^*$. The following two statements are equivalent. For all $x \in \mathbb{Z}_q^*$,*

$$f_x(n) = \sum_{a \in \mathbb{Z}_q^*} \sum_{\substack{d \mid n \\ d \equiv a(q)}} g_{ax}\left(\frac{n}{d}\right). \tag{3.1}$$

For all $x \in \mathbb{Z}_q^*$,

$$g_x(n) = \sum_{a\in\mathbb{Z}_q^*} \sum_{\substack{d|n \\ d\equiv a(q)}} \mu(d) f_{ax}\left(\frac{n}{d}\right). \tag{3.2}$$

Proof. Let X be the right-hand side of (3.1), and assume that (3.2) is true. Then

$$\begin{aligned} X &= \sum_{a\in\mathbb{Z}_q^*} \sum_{\substack{d|n \\ d\equiv a(q)}} g_{ax}\left(\frac{n}{d}\right) \\ &= \sum_{a\in\mathbb{Z}_q^*} \sum_{\substack{d|n \\ d\equiv a(q)}} \sum_{b\in\mathbb{Z}_q^*} \sum_{\substack{d'|(n/d) \\ d'\equiv b(q)}} \mu(d') f_{abx}\left(\frac{n/d}{d'}\right). \end{aligned}$$

We now make the substitutions $dd' = m$ and $ab = c$ and interchange the order of summation to obtain

$$\begin{aligned} X &= \sum_{c\in\mathbb{Z}_q^*} \sum_{b\in\mathbb{Z}_q^*} \sum_{\substack{m|n \\ m\equiv c(q)}} \sum_{\substack{d|m \\ d\equiv cb^{-1}(q)}} \mu\left(\frac{m}{d}\right) f_{cx}\left(\frac{n}{m}\right) \\ &= \sum_{c\in\mathbb{Z}_q^*} \sum_{\substack{m|n \\ m\equiv c(q)}} f_{cx}\left(\frac{n}{m}\right) \sum_{b\in\mathbb{Z}_q^*} \sum_{\substack{d|m \\ d\equiv cb^{-1}(q)}} \mu\left(\frac{m}{d}\right) \\ &= \sum_{c\in\mathbb{Z}_q^*} \sum_{\substack{m|n \\ m\equiv c(q)}} f_{cx}\left(\frac{n}{m}\right) [\![m = 1]\!] \\ &= f_x(n). \end{aligned}$$

The second equality above uses Lemma 3.1, noting that the condition $m \equiv c(q)$ on the second summation implies that $m \bmod q \in \mathbb{Z}_q^*$.

Verification in the other direction is similar and is omitted. □

4. General results. In this section we present some results that apply over various finite commutative rings. We assume throughout that R has r elements and prime characteristic p.

The following formula for $A_R(n)$ is well known and depends only on the number of elements in the ring and not on its algebraic structure.

$$A_R(n) = \sum_{d|n} \mu\left(\frac{n}{d}\right) r^d = \sum_{d|n} \mu(d) r^{n/d}. \tag{4.1}$$

The following two lemmas will be useful in simplifying certain later sums.

LEMMA 4.1. *Let a be a natural number, let b and j be positive integers, and let f be a function from the positive integers to a commutative ring with identity. Then*

$$\sum_{\substack{d|n \\ d\equiv ja(jb)}} f(d) = [\![j|n]\!] \sum_{\substack{d|\frac{n}{j} \\ d\equiv a(b)}} f(jd). \tag{4.2}$$

Proof. The condition $d \equiv ja(jb)$ implies that $j|d$. Let $d = jd'$. Observe that

$$[\![d|n]\!][\![d \equiv ja(jb)]\!] = [\![jd'|n]\!][\![jd' \equiv ja(jb)]\!] = [\![j|n]\!]\left[\!\!\left[d'|\frac{n}{j}\right]\!\!\right][\![d' \equiv a(b)]\!].$$

Summation index d is used on the left-hand side of (4.2), and d' is used on the right-hand side. □

In the arguments to follow, we will use Lemma 4.1 with two sets of values for j, a, b. First, in the next lemma we use $j = q$, $a = 0$, and $b = 1$. In this case the congruence in the right-hand sum becomes $d \equiv 0(1)$ which is vacuously satisfied and can be omitted. Later, we will use $j = 2$, $a = 1$, and $b = 2$.

LEMMA 4.2. *Let n and q be positive integers. Then*

$$\sum_{\substack{d|n \\ d\equiv 0(q)}} A_R\left(\frac{n}{d}\right) = [\![q|n]\!] \sum_{d|\frac{n}{q}} A_R\left(\frac{n/q}{d}\right) = [\![q|n]\!] r^{n/q}.$$

Proof. The first equality follows from Lemma 4.1 and the second from the fact that every string is the repeated concatenation of some aperiodic string. □

LEMMA 4.3. *Let R have prime characteristic p. If $k < p$ (and $\mathbf{0}$ is the k-tuple $(0, 0, \ldots, 0)$), then*

$$S_R(n;\mathbf{0}) = \sum_{\substack{d|n \\ d\equiv 0(p)}} A_R\left(\frac{n}{d}\right) + \sum_{\substack{d|n \\ d\not\equiv 0(p)}} A_R\left(\frac{n}{d};\mathbf{0}\right) = [\![p|n]\!] r^{n/p} + \sum_{\substack{d|n \\ d\not\equiv 0(p)}} A_R\left(\frac{n}{d};\mathbf{0}\right).$$

Proof. The second equality follows from Lemma 4.2. To prove the first equality, take (2.5) and break the sum into two parts depending on whether or not $d \equiv 0(p)$. Recall (2.4).

Consider first the case where $d \equiv 0(p)$. From $p|d$ and $j \geq 1$ it follows that $p|d^{(\nu_1+\nu_2+\cdots+\nu_j)}$. Since $\nu_i \leq k < p$, we have $p \nmid \nu_1!\nu_2!\cdots\nu_j!$. Thus p divides $d^{(\nu_1+\nu_2+\cdots+\nu_j)}/(\nu_1!\nu_2!\cdots\nu_j!)$, from which it follows that $u_j = 0$ irrespective of the values of $\mathbf{t}$. Thus

$$\sum_{\mathbf{t}\in R^k} [\![\theta_d(\mathbf{t}) = \mathbf{0}]\!] A_R\left(\frac{n}{d};\mathbf{t}\right) = \sum_{\mathbf{t}\in R^k} A_R\left(\frac{n}{d};\mathbf{t}\right) = A_R\left(\frac{n}{d}\right).$$

Now consider the case where $d \not\equiv 0(p)$. In the notation of Theorem 2.3, $q'_k = p$ since $k < p$. Since $d \in \mathbb{Z}_p^*$, by Corollary 2.5 the function θ_d is invertible, and $\theta_d^{-1} = \theta_{d^{-1}}$. Thus $\mathbf{t} = \theta_{d^{-1}}(\mathbf{0}) = \mathbf{0}$, and hence

$$\sum_{\mathbf{t}\in R^k} [\![\theta_d(\mathbf{t}) = \mathbf{0}]\!] A_R\left(\frac{n}{d};\mathbf{t}\right) = \sum_{\mathbf{t}\in R^k} [\![\mathbf{t} = \mathbf{0}]\!] A_R\left(\frac{n}{d};\mathbf{t}\right) = A_R\left(\frac{n}{d};\mathbf{0}\right). \qquad \square$$

COROLLARY 4.4. *If R is a ring of prime characteristic p with $k < p$, then*

$$L_R(n;\mathbf{0}) = \frac{1}{n} \sum_{\substack{d|n \\ d\not\equiv 0(p)}} \mu(d) \left(S_R\left(\frac{n}{d};\mathbf{0}\right) - [\![pd|n]\!] r^{n/(pd)}\right),$$

where R contains r elements and $\mathbf{0}$ is the k-tuple $(0, 0, \ldots, 0)$.

Proof. Note that the sum in Lemma 4.3 is over $\{1, 2, \ldots, p-1\} = \mathbb{Z}_p^*$ for prime p. Apply Theorem 3.2 with $f_x(n) = S_R(n;\mathbf{0})$ and $g_x(n) = A_R(n;\mathbf{0})$ for all x. □

5. Strings over the ring $\mathbb{Z}_q$. In [9] we showed that

$$L_{\mathbb{Z}_q}(n;t) = \frac{1}{qn} \sum_{\substack{d|n \\ \gcd(d,q)|t}} \mu(d) \gcd(d,q) q^{n/d}. \tag{5.1}$$

From this the next lemma follows.

LEMMA 5.1. *If* $\gcd(q,t) = \gcd(q,t')$, *then* $L_{\mathbb{Z}_q}(n;t) = L_{\mathbb{Z}_q}(n;t')$.

Note that $x \in \mathbb{Z}_q^*$ if and only if $x \in \mathbb{Z}_{q'_k}^*$ since $\gcd(x,q) = 1$ if and only if $\gcd(x,q'_k) = 1$.

LEMMA 5.2. *For all* $n \geq 1$ *and primes* p

$$\sum_{\substack{d|n \\ d\not\equiv 0(p)}} A_{\mathbb{Z}_p}\left(\frac{n}{d};1\right) = p^{n-1}. \tag{5.2}$$

Proof. This follows from the equation

$$p^{n-1} = S(n;1) = \sum_{d|n}\sum_{de=1} A_R\left(\frac{n}{d};e\right) = \sum_{e\in\mathbb{Z}_p^*}\sum_{d|n} A_R\left(\frac{n}{d};e^{-1}\right). \qquad \square$$

Let us say that a parameter pair $(n;\mathbf{t})$ is *unit invertible* if the equation $\mathbf{u} = \theta_d(\mathbf{t})$ has a unique solution for all $d \in \mathbb{Z}_{q'_k}^*$ and has no solution if $d \notin \mathbb{Z}_{q'_k}^*$. For example, $(n;\mathbf{t})$ is unit invertible if $t_1 \in \mathbb{Z}_q^*$ or if $n \in \mathbb{Z}_q^*$.

THEOREM 5.3. *If* $(n;\mathbf{t})$ *is unit invertible, then*

$$L_{\mathbb{Z}_q}(n;\mathbf{t}) = \frac{1}{n}\sum_{r\in\mathbb{Z}_{q'_k}^*}\sum_{\substack{d|n \\ d\equiv r^{-1}(q'_k)}} \mu(d) S_{\mathbb{Z}_q}\left(\frac{n}{d};\theta_{r^{-1}}(\mathbf{t})\right). \tag{5.3}$$

Proof. Under the stated hypotheses we can write (2.6) as

$$S(n;\mathbf{t}) = \sum_{a\in\mathbb{Z}_{q'_k}^*}\sum_{\substack{d|n \\ d\equiv a(q'_k)}} A\left(\frac{n}{d};\theta_a^{-1}(\mathbf{t})\right). \tag{5.4}$$

By Corollary 2.5, we have $\theta_a^{-1}(\mathbf{u}) = \theta_{a^{-1}}(\mathbf{u})$. Substitute $\mathbf{t} = \theta_x(\mathbf{u})$ in (5.4), and use the multiplicative property $\theta_{a^{-1}}(\theta_x(\mathbf{u})) = \theta_{a^{-1}x}(\mathbf{u})$ to obtain

$$S(n;\theta_x(\mathbf{u})) = \sum_{a\in\mathbb{Z}_{q'_k}^*}\sum_{\substack{d|n \\ d\equiv a(q'_k)}} A\left(\frac{n}{d};\theta_{a^{-1}x}(\mathbf{u})\right). \tag{5.5}$$

Written in this form we can apply Theorem 3.2 with $f_x(n) = S(n;\theta_x(\mathbf{u}))$ and $g_x(n) = A(n;\theta_{x^{-1}}(\mathbf{u}))$ to obtain (5.3). $\square$

Example. If $q = k = 3$, then $q'_k = 9$, and $\theta_{d^{-1}}(1,0,0)$ takes on the values

$$\{(1,0,0),(2,1,0),(1,0,1),(2,1,1),(1,0,2),(2,1,2)\}$$

for $d = 1,2,4,5,7,8$. The number, $L_{\mathbb{Z}_3}(n;1,0,0)$, of length n Lyndon words over $\mathbb{Z}_3$ with $(t_1,t_2,t_3) = (1,0,0)$ is therefore equal to

$$\frac{1}{n}\sum_{j\in\mathbb{Z}_3}\left(\sum_{\substack{d|n \\ d\equiv(3j+1)^{-1}(9)}} \mu(d) S_{\mathbb{Z}_3}\left(\frac{n}{d};1,0,j\right) + \sum_{\substack{d|n \\ d\equiv(3j+2)^{-1}(9)}} \mu(d) S_{\mathbb{Z}_3}\left(\frac{n}{d};2,1,j\right)\right),$$

giving rise to the sequence of numbers 1, 1, 1, 1, 1, 1, 6, 36, 141, 422, 1062, 2371, 4995, 11082, 29230, 90735 for $n = 1,2,\ldots,16$.

According to the results of [8], over $\mathbb{Z}_3$ the traces (t_1, t_2, t_3) determine the traces t_4 and t_5 so that $L_{\mathbb{Z}_3}(n; 1, 0, 0) = L_{\mathbb{Z}_3}(n; 1, 0, 0, 0, 0)$. Furthermore, the $S_{\mathbb{Z}_3}$ numbers can be expressed as sums of multinomial coefficients; e.g., for $S_{\mathbb{Z}_3}(n; 1, 0, 0)$ we have

$$S_{\mathbb{Z}_3}(n; 1, 0, 0) = \sum_{\substack{k_0+k_1+k_2=n \\ k_2 \equiv 0(3) \\ k_1-k_2 \equiv 1(9)}} \binom{n}{k_0, k_1, k_3}.$$

6. Strings over the ring $\mathbb{F}_q$.

6.1. The field $\mathbb{F}_q$ for q odd. In this section we consider the computation of the number of strings in the various classes over $\mathbb{F}_q$, where $q = p^m$, with p an odd prime.

In [9] we reproved a result of Carlitz [1] that, if $t \neq 0$, then

$$L_{\mathbb{F}_q}(n; t) = \frac{1}{qn} \sum_{\substack{d|n \\ p \nmid d}} \mu(d) q^{n/d}. \tag{6.1}$$

Here we generalize this to the first $p-1$ traces.

THEOREM 6.1. *If $q = p^m$, where p is an odd prime and $k < p$, then*

$$L_{\mathbb{F}_q}(n; \mathbf{t}) = \begin{cases} \dfrac{1}{n} \displaystyle\sum_{\substack{d|n \\ p \nmid d}} \mu(d) \left(S_{\mathbb{F}_q}\left(\frac{n}{d}; \mathbf{0}\right) - [\![pd|n]\!] q^{n/(pd)} \right) & \text{if } \mathbf{t} = \mathbf{0}, \\ \dfrac{1}{n} \displaystyle\sum_{\substack{d|n \\ p \nmid d}} \mu(d) S_{\mathbb{F}_q}\left(\frac{n}{d}; \theta_{d^{-1}}(\mathbf{t})\right) & \text{otherwise.} \end{cases}$$

Proof. The $\mathbf{t} = \mathbf{0}$ case follows from Corollary 4.4. In the other case there is some index $j \leq k$ such that $t_1 = \cdots = t_{j-1} = 0$ and $t_j \neq 0$. Consider the equation $\mathbf{t} = \theta_d(\mathbf{u})$ in (2.5). If $d \equiv 0(p)$, then we must have $\mathbf{t} = \mathbf{0}$. Thus $d \not\equiv 0(p)$. Hence $u_1 = u_2 = \cdots = u_{j-1} = 0$ and $t_j = du_j$ so that $u_j = d^{-1}t_j$. Repeated substitution will give unique values for $u_j, u_{j+1}, \ldots, u_k$. We can therefore use (2.6) and write

$$\begin{aligned} S_{\mathbb{F}_q}(n; \mathbf{t}) &= \sum_{\substack{d|n \\ d \not\equiv 0(p)}} A_{\mathbb{F}_q}\left(\frac{n}{d}; \theta_{d^{-1}}(\mathbf{t})\right) \\ &= \sum_{x \in \mathbb{Z}_p^*} \sum_{\substack{d|n \\ d \equiv x^{-1}(p)}} A_{\mathbb{F}_q}\left(\frac{n}{d}; \theta_{x^{-1}}(\mathbf{t})\right), \end{aligned}$$

which can then be inverted by Theorem 3.2 to obtain the stated result. ∎

The case where $p = 2$ will be handled in the next section.

6.2. The field $\mathbb{F}_{2^m}$. In this section $p = 2$. Since $p = 2$, if $k < p$, then $k = 0$ or $k = 1$. However, the value of $L_{\mathbb{Z}_{2^m}}(n; \mathbf{t})$ is known for $k = 0, 1$ ((4.1) and (6.2)), so unlike in the previous subsection here we have $k \geq p$. In this section we will consider in detail the $k = 3$ case, which is the largest value for which $p'_k = 2^2 = 4$. In other words, we derive a formula for $L_{\mathbb{F}_{2^m}}(n; t_1, t_2, t_3)$. We also state without proof the result for $L_{\mathbb{F}_{2^m}}(n; t_1, t_2)$.

Here the values of $\binom{d}{2}$ mod 2 follow the pattern 0,0,1,1 mod 4 and the values of $\binom{d}{3}$ mod 2 follow the pattern 0,0,0,1, so we consider the value of d mod 4 in (2.7),

(2.8), and (2.9) taken mod 2 (but with the roles of u and t reversed). If $d \equiv 0(4)$, then $t_1 = t_2 = t_3 = 0$, but u_1, u_2, and u_3 are unrestricted. If $d \equiv 1(4)$, then $u_1 = t_1$, $u_2 = t_2$, and $u_3 = t_3$. If $d \equiv 2(4)$, $t_1 = 0$, $u_1^2 = t_2$, and $t_3 = 0$. Fortunately, in a field of characteristic 2, square roots always exist and are unique, so we can set $u_1 = \sqrt{t_2}$. Finally, if $d \equiv 3(4)$, then $u_1 = t_1$, $u_2 = t_2 + t_1^2$, and $t_3 = u_3 + t_1^3$. Thus,

$$\begin{aligned} S_{\mathbb{F}_{2^m}}(n; t_1, t_2, t_3) = {} & [\![t_1 = 0]\!][\![t_2 = 0]\!][\![t_3 = 0]\!] \sum_{\substack{d|n \\ d\equiv 0(4)}} A_{\mathbb{F}_{2^m}}\left(\frac{n}{d}\right) \\ & + \sum_{\substack{d|n \\ d\equiv 1(4)}} A_{\mathbb{F}_{2^m}}\left(\frac{n}{d}; t_1, t_2, t_3\right) \\ & + [\![t_1 = 0]\!][\![t_2 = 0]\!] \sum_{\substack{d|n \\ d\equiv 2(4)}} A_{\mathbb{F}_{2^m}}\left(\frac{n}{d}; \sqrt{t_2}\right) \\ & + \sum_{\substack{d|n \\ d\equiv 3(4)}} A_{\mathbb{F}_{2^m}}\left(\frac{n}{d}; t_1, t_2 + t_1^2, t_3 + t_1^3\right). \end{aligned}$$

We now consider the different values of the trace, subtrace, and subsubtrace. If $t_1 = t_2 = t_3 = 0$, then

(6.2)
$$S_{\mathbb{F}_{2^m}}(n; 0, 0, 0) = \sum_{\substack{d|n \\ d\equiv 0(4)}} A_{\mathbb{F}_{2^m}}\left(\frac{n}{d}\right) + \sum_{\substack{d|n \\ d\equiv 2(4)}} A_{\mathbb{F}_{2^m}}\left(\frac{n}{d}, 0\right) + \sum_{\substack{d|n \\ d \text{ odd}}} A_{\mathbb{F}_{2^m}}\left(\frac{n}{d}, 0, 0, 0\right).$$

If $t_2 \neq 0$ but $t_1 = t_3 = 0$, then

$$S_{\mathbb{F}_{2^m}}(n; 0, t_2, 0) = \sum_{\substack{d|n \\ d\equiv 2(4)}} A_{\mathbb{F}_{2^m}}\left(\frac{n}{d}; \sqrt{t_2}\right) + \sum_{\substack{d|n \\ d \text{ odd}}} A_{\mathbb{F}_{2^m}}\left(\frac{n}{d}; 0, t_2, 0\right). \tag{6.3}$$

If $t_1 = 0$ but $t_3 \neq 0$, then

$$S_{\mathbb{F}_{2^m}}(n; 0, t_2, t_3) = \sum_{\substack{d|n \\ d \text{ odd}}} A_{\mathbb{F}_{2^m}}\left(\frac{n}{d}; 0, t_2, t_3\right). \tag{6.4}$$

The equations where $t_1 \neq 0$ come in parameter pairs, (t_1, t_2, t_3) and $(t_1, t_2 + t_1^2, t_3 + t_1^3)$. The quantity $S_{\mathbb{F}_{2^m}}(n; t_1, t_2, t_3)$ is equal to

$$\sum_{\substack{d|n \\ d\equiv 1(4)}} A_{\mathbb{F}_{2^m}}\left(\frac{n}{d}; t_1, t_2, t_3\right) + \sum_{\substack{d|n \\ d\equiv 3(4)}} A_{\mathbb{F}_{2^m}}\left(\frac{n}{d}; t_1, t_2 + t_1^2, t_3 + t_1^3\right). \tag{6.5}$$

We can use Theorem 3.2 to invert the pairs (6.5) to obtain

$$\begin{aligned} L_{\mathbb{F}_{2^m}}(n; t_1, t_2, t_3) = {} & \frac{1}{n} \sum_{\substack{d|n \\ d\equiv 1(4)}} \mu(d) S_{\mathbb{F}_{2^m}}\left(\frac{n}{d}; t_1, t_2, t_3\right) \\ & + \frac{1}{n} \sum_{\substack{d|n \\ d\equiv 3(4)}} \mu(d) S_{\mathbb{F}_{2^m}}\left(\frac{n}{d}; t_1, t_2 + t_1^2, t_3 + t_1^3\right). \end{aligned}$$

To invert (6.3) we will need the following lemma and corollary. The lemma holds over general finite fields.

LEMMA 6.2. *Let $q = p^m$ with p prime. For all $n \geq 1$,*

$$\sum_{\substack{d|n \\ d\not\equiv 0(p)}} A_{\mathbb{F}_q}\left(\frac{n}{d}; 1\right) = q^{n-1}, \quad \textit{and} \tag{6.6}$$

$$\sum_{\substack{d|n \\ d\not\equiv 0(p)}} A_{\mathbb{F}_q}\left(\frac{n}{d}; 0\right) = q^{n-1} - [\![p|n]\!] q^{n/p}. \tag{6.7}$$

Proof. To prove (6.6) consider the equation below, where $y \in \mathbb{Z}_p^*$.

$$q^{n-1} = S_{\mathbb{F}_q}(n; y) = \sum_{d|n} \sum_{dx \equiv y} A_{\mathbb{F}_q}\left(\frac{n}{d}; x\right) = \sum_{\substack{d|n \\ d\in\mathbb{Z}_p^*}} A_{\mathbb{F}_q}\left(\frac{n}{d}; d^{-1}y\right) = \sum_{\substack{d|n \\ d\in\mathbb{Z}_p^*}} A_{\mathbb{F}_q}\left(\frac{n}{d}; 1\right).$$

The second equality is a restatement of (2.5). The equation $dx = y$ has a solution only if $d \in \mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$, namely, $x = d^{-1}y \bmod p$, giving the third equality. The equalities $S_{\mathbb{F}_q}(n; y) = \sum_{\substack{d|n \\ d\in\mathbb{Z}_p^*}} A_{\mathbb{F}_q}(\frac{n}{d}; d^{-1}y)$ can be inverted by Theorem 3.2 to obtain

$$A_{\mathbb{F}_q}(n; y) = \sum_{\substack{d|n \\ d\in\mathbb{Z}_p^*}} \mu(d) S_{\mathbb{F}_q}\left(\frac{n}{d}; d^{-1}y\right), \tag{6.8}$$

thereby implying that $A_{\mathbb{F}_q}(n; y) = A_{\mathbb{F}_q}(n; 1)$ for all $y \in \mathbb{Z}_p^*$ and justifying the last equality. □

The following corollary generalizes Lemma 5 from [2].

COROLLARY 6.3. *Let m be a positive integer. Then*

$$\sum_{\substack{d|n \\ d\equiv 2(4)}} A_{\mathbb{F}_{2^m}}\left(\frac{n}{d}, 1\right) = [\![n \textit{ even}]\!](2^m)^{n/2-1}, \tag{6.9}$$

$$\sum_{\substack{d|n \\ d\equiv 0(4)}} A_{\mathbb{F}_{2^m}}\left(\frac{n}{d}\right) + \sum_{\substack{d|n \\ d\equiv 2(4)}} A_{\mathbb{F}_{2^m}}\left(\frac{n}{d}, 0\right) = [\![n \textit{ even}]\!](2^m)^{n/2-1}. \tag{6.10}$$

Proof. To prove (6.9) we first use Lemma 4.1 with $j = b = 2$ and $a = 1$. This produces a sum of the form of (6.6), except with $n/2$ substituted for n, and 2^m substituted for q.

To prove (6.10), note that the first term of the left-hand side is $[\![4|n]\!](2^m)^{n/4}$ by Lemma 4.2 with $q = 4$. By Lemma 4.2 the second term of the left-hand side is equal to $[\![n \text{ even}]\!] \sum_{d|(n/2), d\not\equiv 0(2)} A(n/(2d), 0)$. By Lemma 6.2 this is in turn equal to $[\![n \text{ even}]\!]((2^m)^{n/2-1} - [\![2|(n/2)]\!](2^m)^{n/4})$. Adding the two terms together, we get $[\![n \text{ even}]\!](2^m)^{n/2-1}$. □

Note from (6.1) that $A_{\mathbb{F}_{2^m}}(n; 1) = A_{\mathbb{F}_{2^m}}(n; \sqrt{t_2})$ for any $t_2 \neq 0$. In view of Corollary 6.3, for any t_2, we can write

$$S_{\mathbb{F}_{2^m}}(n; 0, t_2, 0) = [\![n \text{ even}]\!](2^m)^{n/2-1} + \sum_{\substack{d|n \\ d \text{ odd}}} A_{\mathbb{F}_{2^m}}\left(\frac{n}{d}; 0, t_2, 0\right).$$

This equation can be inverted using the Möbius inversion of Theorem 3.2 to obtain

$$L_{\mathbb{F}_{2^m}}(n;0,t_2,0)=\frac{1}{n}\sum_{\substack{d|n\\ d\ \text{odd}}}\mu(d)\left(S_{\mathbb{F}_{2^m}}\left(\frac{n}{d};0,t_2,0\right)-[\![n/d\ \text{even}]\!](2^m)^{n/(2d)-1}\right).$$

The various cases are summarized in the following theorem.

THEOREM 6.4. *If $q=2^m$, then the value of $L_{\mathbb{F}_q}(n;t_1,t_2,t_3)$ is*

$$\begin{cases}\dfrac{1}{n}\displaystyle\sum_{\substack{d|n\\ d\ \text{odd}}}\mu(d)\left(S_{\mathbb{F}_q}\left(\frac{n}{d};0,t_2,0\right)-[\![r=0]\!]\left[\!\left[2|\frac{n}{d}\right]\!\right]q^{n/(2d)-1}\right) & \text{if } t_1=0,\\ \dfrac{1}{n}\displaystyle\sum_{\substack{d|n\\ d\ \text{odd}}}\mu(d)S_{\mathbb{F}_q}\left(\frac{n}{d};t_1,t_2+\frac{d-1}{2}t_1^2,t_3+\frac{d-1}{2}t_1^3\right) & \text{if } t_1\neq 0.\end{cases}$$

By similar arguments, or by summing over t_3 in the preceding theorem, we obtain the theorem below.

THEOREM 6.5. *If $q=2^m$, then*

$$L_{\mathbb{F}_q}(n;t_1,s_1)=\begin{cases}\dfrac{1}{n}\displaystyle\sum_{\substack{d|n\\ d\ \text{odd}}}\mu(d)\left(S_{\mathbb{F}_q}\left(\frac{n}{d};0,t_2\right)-\left[\!\left[\frac{n}{d}\ \textit{even}\right]\!\right]q^{n/(2d)-1}\right) & \text{if } t_1=0,\\ \dfrac{1}{n}\displaystyle\sum_{\substack{d|n\\ d\ \text{odd}}}\mu(d)S_{\mathbb{F}_q}\left(\frac{n}{d};t_1,t_2+\frac{d-1}{2}t_1^2\right) & \text{if } t_1\neq 0.\end{cases}$$

7. Final remarks. Tables of some of the numbers discussed in this paper for $k=1,2$ may be accessed from the page www.theory.cs.uvic.ca/∼cos/inf/trs/. There are many relevant sequence numbers in Neil J. Sloane's online encylopedia of integer sequences. For example, over $\mathbb{Z}_3$ it contains $L_{\mathbb{Z}_3}(n;0,0)=$ A053548, $L_{\mathbb{Z}_3}(n;0,1)=$ A053560, $L_{\mathbb{Z}_3}(n;0,2)=$ A053561, $L_{\mathbb{Z}_3}(n;1,0)=L_{\mathbb{Z}_3}(n;2,0)=$ A053562, $L_{\mathbb{Z}_3}(n;1,1)=L_{\mathbb{Z}_3}(n;2,1)=$ A053563, $L_{\mathbb{Z}_3}(n;1,2)=L_{\mathbb{Z}_3}(n;2,2)=$ A053564.

Acknowledgments. We wish to thank the referee and the editor for helpful comments.

REFERENCES

[1] L. CARLITZ, *A theorem of Dickson on irreducible polynomials*, Proc. Amer. Math. Soc., 3 (1952), pp. 693–700.

[2] K. CATTELL, F. RUSKEY, C. R. MIERS, J. SAWADA, AND M. SERRA, *The number of irreducible polynomials over GF(2) with given trace and subtrace*, J. Combin. Math. Combin. Comput., 47 (2003), pp. 31–64.

[3] L. COMTET, *Advanced Combinatorics*, D. Reidel, Dordrecht, Holland, 1974.

[4] D. JUNGNICKEL, *Finite Fields: Structure and Arithmetics*, Bibliographisches Institut, Mannheim, Germany, 1993.

[5] E. N. KUZ'MIN, *On a class of irreducible polynomials over a finite field*, Dokl. Akad. Nauk SSSR, 313 (1990), pp. 552–555 (in Russian); translation in Soviet Math. Dokl., 42 (1991), pp. 45–48.

[6] D. E. KNUTH, R. L. GRAHAM, AND O. PATASHNIK, *Concrete Mathematics*, Addison–Wesley, Reading, MA, 1989.

[7] M. LOTHAIRE, *Combinatorics on Words*, Addison–Wesley, Reading, MA, 1983.

[8] C. R. MIERS AND F. RUSKEY, *Counting strings with given elementary symmetric function evaluations* I: *Strings over $\mathbb{Z}_p$ with p prime*, SIAM J. Discrete Math., 17 (2004), pp. 675–685.

[9] F. Ruskey, C. R. Miers, and J. Sawada, *The number of irreducible polynomials and Lyndon words with given trace*, SIAM J. Discrete Math., 14 (2001), pp. 240–245.
[10] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Their Applications*, Cambridge University Press, Cambridge, UK, 1994.
[11] J. L. Yucas and G. L. Mullen, *Irreducible polynomials over GF*(2) *with prescribed coefficients*, Discrete Math., 274 (2004), pp. 265–279.
[12] S. Zabek, *Sur la periodicite modulo m des suits de nombres* $\binom{n}{k}$, Ann. Univ. Mariae Curie-Skłodowska Sect. A, 10 (1956), pp. 37–47.

SIAM J. DISCRETE MATH.
Vol. 18, No. 1, pp. 83–102

RECOGNIZING POWERS OF PROPER INTERVAL, SPLIT, AND CHORDAL GRAPHS*

LAP CHI LAU† AND DEREK G. CORNEIL†

Abstract. In this paper, we study the complexity of recognizing powers of chordal graphs and its subclasses. We present the first polynomial time algorithm to recognize squares of proper interval graphs and give an outline of an algorithm to recognize kth powers of proper interval graphs for every natural number k. These are the first results of this type for a family of graphs that contains arbitrarily large cliques. On the other hand, we show the NP-completeness of recognizing squares of chordal graphs, recognizing squares of split graphs, and recognizing chordal graphs that are squares of some graph.

Key words. proper interval graphs, split graphs, chordal graphs, graph roots, graph powers, graph algorithms

AMS subject classifications. 05C12, 05C75, 68R10

DOI. 10.1137/S0895480103425930

1. Introduction.

Root and *root finding* are concepts familiar in most branches of mathematics. In graph theory, H is a *root* of $G = (V, E)$ if there exists a positive integer k such that x and y are adjacent in G if and only if their distance in H is at most k. If H is a kth root of G, then we write $G = H^k$ and call G the kth power of H. Note that the terms "power" and "root" are used because of the close relationship with matrix multiplication. Graph roots are associated with problems in distributed computing [26] and computational biology [31, 24], where graph roots are useful in the reconstruction of phylogeny.

For any class of graphs, recognition is a fundamental structural and algorithmic problem; in this paper, we study recognition problems on graph powers. Generally, it is a difficult task to determine whether a given graph G has a kth root or not. In 1967, Mukhopadhyay [30] characterized general graphs that possess a square root, and in the following year Geller [13] solved the problem for general digraphs. In 1974, Escalante, Montejano, and Rojano [9] characterized graphs and digraphs with a kth root. However, all of these characterizations on powers of general graphs are not polynomial in the sense that they do not yield a polynomial time algorithm. The complexity of graph power recognition was unresolved until 1994, when Motwani and Sudan [29] proved the NP-completeness of recognizing squares of graphs. About the same time, Lin and Skiena [25] gave a linear time algorithm for recognizing squares of trees. Somewhat surprisingly, until very recently, trees were the only nontrivial family of graphs where the square recognition problem is known to have a polynomial time algorithm. (For classes of graphs with diameter at most 2, such as the class of cographs, the square recognition problem is trivial since the square is always a clique.) In [22], Lau presented a polynomial time algorithm to recognize squares of bipartite graphs. (It is worth noting that Motwani and Sudan [29] believed that this problem would be NP-complete.) In this paper, we give a polynomial time algorithm

*Received by the editors April 15, 2003; accepted for publication (in revised form) December 30, 2003; published electronically July 20, 2004. This research was supported by the Natural Science and Engineering Research Council of Canada.
http://www.siam.org/journals/sidma/18-1/42593.html
†Department of Computer Science, University of Toronto, 10 King's College Road, Toronto, ON, M5S 3G4 Canada (chi@cs.toronto.edu, dgc@cs.toronto.edu).

to recognize squares of proper interval graphs. This is the first class of graphs that does not contain trees and furthermore allows graphs with arbitrarily large cliques. (The algorithms for trees and bipartite graphs are very dependent on the existence of no cliques of size > 2.) An obvious extension of the square recognition problem is that of kth power recognition. For $k > 2$ it was solved for trees in polynomial time [18]; for bipartite graphs it is NP-complete [22]. In this paper we show that for proper interval graphs it is in P, for every fixed k. In [16], Harary and McKee introduced the closed-neighborhood intersection multigraph as a useful multigraph version of the square of a graph. They characterized those multigraphs that are squares of chordal graphs and they gave an algorithm to go from the squared chordal graph back to its unique square root. In this paper, we show the NP-completeness of recognizing squares of chordal graphs. Also, we prove the NP-completeness of recognizing squares of split graphs and recognizing chordal graphs that are squares of some graph. Flotow [11] studied a related problem to graph powers recognition; he gave sufficient conditions for a graph whose power is a chordal graph (or an interval graph).

1.1. Overview of this paper. In this paper, we study the complexity of recognizing powers of chordal graphs and its subclasses. In section 2, we present the first polynomial time algorithm to recognize squares of proper interval graphs, and we sketch an outline of a polynomial time algorithm to recognize kth powers of proper interval graphs for every natural number k. Our approach is based on a nontrivial use of dynamic programming. In section 3, we prove the NP-completeness of recognizing squares of chordal graphs, recognizing squares of split graphs, and recognizing chordal graphs that are squares of some graph. Note that split graphs and bipartite graphs have a similar partitioning structure but squares of bipartite graphs can be recognized in polynomial time [22]. Before presenting our results we survey some important results concerning graph powers and give our terminology.

1.2. Related work. The literature is rich with results on graph roots and powers. Given a graph G with property P, does G^k have property P? Substantial work has been done on closure properties of powers of special classes of graphs, such as chordal graphs [1, 8], interval graphs [33], cocomparability graphs [6], strongly chordal graphs [27, 34, 2], circular arc graphs [34], and AT-free graphs [33, 3]. Given a graph G, what can be said about the properties of G^k? Since the number of edges increases with the index of the power of a graph, it is natural to expect that sufficiently large powers do possess some Hamiltonian-type properties. For instance, Fleischner [10] proved that the square of every 2-connected graph is Hamiltonian, and Sekanina [36] proved that the cube of every nontrivial connected graph is Hamiltonian connected. Besides the mathematical property questions on graph powers, the following is an obvious question to ask from an algorithmic point of view. Given a graph G, can we solve some optimization problems on G^k efficiently? Many optimization problems remain difficult in the case of powers of graphs [25]. On the other hand, a Hamiltonian cycle in the square of a 2-connected graph can be found in polynomial time [21], and the chromatic number of the square of a planar graph can be approximated within a constant factor in polynomial time [28]. Also, Ramachandran [32] proved, without using computers, that if G is a planar graph with a square root or a cube root, then G is 4 colorable.

1.3. Basic terminology. Our basic notation and terminology reference is [38]. We denote a *graph* G with *vertex set* $V(G)$ and *edge set* $E(G)$ by $G = (V, E)$, where n and m denote $|V|$ and $|E|$, respectively. A *loop* is an edge whose endpoints are equal.

Multiple edges are edges having the same pair of endpoints. A *simple graph* is a graph having no loops or multiple edges. When u and v are endpoints of an edge, they are *adjacent* and are *neighbors*. We write $u \leftrightarrow v$ or $uv \in E(G)$ for "u is adjacent to v". All the graphs we consider are *simple*, *undirected*, and *loopless*, unless otherwise specified. The *complement* $\overline{G}$ of a simple graph G is the simple graph with vertex set $V(G)$ defined by $uv \in E(\overline{G})$ if and only if $uv \notin E(G)$. A graph $G' = (V', E')$ is a *subgraph* of $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. G' is an *induced subgraph* of G, written $G[V']$, if it is a subgraph of G and it contains all the edges uv such that $u, v \in V'$ and $uv \in E(G)$.

The *degree* of a vertex v in a graph G, written $\deg(v)$, is the number of edges incident with v. A *pendant vertex* is a vertex with degree one. An *isolated vertex* is a vertex with degree zero. The *open neighborhood* of v, written $N_G(v)$ or $N(v)$, is the set of vertices adjacent to v. The *closed neighborhood* of v, written $N_G[v]$, is $N_G(v) \cup \{v\}$. When U is a set of vertices, $N_G[U] = \bigcup_{v \in U} N_G[v]$. If G has a u, v-path, then the *distance* from u to v, written $d_G(u, v)$, is the least length of a u, v-path. The *kth neighborhood* of v, written $N_G^k(v)$, is the set of vertices with distance k to v. A graph G is *connected* if each pair of vertices in G is connected by a path; otherwise, G is *disconnected*. A *clique* in a graph G is a set of pairwise adjacent vertices. When the set has size r, the clique is denoted by K_r. An *independent set* (or *stable set*) in a graph is a set of pairwise nonadjacent vertices.

A graph G is *chordal* if G does not have an induced cycle of length at least 4. A vertex v is *simplicial* if $G[N(v)]$ is a clique. Let $\sigma = [v_1, v_2, \ldots, v_n]$ be an ordering of the vertices in a graph G. We say that σ is a *simplicial elimination ordering* if each v_i is a simplicial vertex of $G[\{v_i, \ldots, v_n\}]$. It is well known that a graph is chordal if and only if it has a simplicial elimination ordering. A graph is *weakly chordal* if G and $\overline{G}$ contain no induced cycle of length at least 5. Let $\mathcal{I} = \{I_1, I_2, \ldots, I_n\}$ be a finite collection of intervals of the real line and let $G_{\mathcal{I}}$ be its intersection graph. G is an *interval graph* if G is the intersection graph $G_{\mathcal{I}}$ of an interval model $\mathcal{I}$. G is a *proper interval graph* if G is an interval graph with an interval model where no two intervals $I_x, I_y \in \mathcal{I}$ properly contain each other. Furthermore, G is a proper interval graph if and only if it is a *unit interval graph*, namely, an interval graph where all intervals are of the same length. Both interval graphs and proper interval graphs [5, 7] can be recognized in linear time. Furthermore, a fully dynamic algorithm for recognizing and representing proper interval graphs is available [17]. A graph is a *split graph* if there is a partition of its vertex set into a clique and a stable set. Clearly split graphs are chordal.

2. Squares of proper interval graphs. In this section, we will present the first polynomial time algorithm to solve Square of Proper Interval Graph.

Problem Square of Proper Interval Graph
Instance A graph $G = (V, E)$.
Question Does there exist a *proper interval graph* H such that $H^2 = G$?

The algorithm is based on a dynamic programming approach and it is conceptually simple. The outline of this section is as follows. First we develop some special structural properties of the square of a proper interval graph. This gives us insight into a polynomial time recognition algorithm. To facilitate our discussion, we then introduce the notation for the description of our algorithm. Then, we prove some lemmas and a decomposition theorem which is the core of our algorithm. We then present the algorithm formally, prove its correctness, and analyze the complexity. Finally, we give an outline to extend the algorithm to recognize kth powers of proper interval graphs for every natural number k.

PROBLEM	kTH POWER OF PROPER INTERVAL GRAPH
INSTANCE	A graph $G = (V, E)$.
QUESTION	Does there exist a *proper interval graph* H such that $H^k = G$?

2.1. Preliminaries. Let G be a proper interval graph and $V(G) = \{g_1, \ldots, g_n\}$. Let $I_G(g_i)$ be the corresponding interval of g_i. We denote the *left endpoint* of $I_G(g_i)$ by $\text{left}_G(g_i)$ and the *right endpoint* of $I_G(g_i)$ by $\text{right}_G(g_i)$. In the remainder of this section, we will assume that $\text{left}_G(g_1) < \text{left}_G(g_2) < \cdots < \text{left}_G(g_n)$. Since G is a proper interval graph, $\text{left}_G(g_i) < \text{left}_G(g_j)$ implies $\text{right}_G(g_i) < \text{right}_G(g_j)$. We call such an ordering a *total vertex ordering* in a proper interval graph.

A set S of vertices (intervals) is *consecutive* if $S = \{g_i, g_{i+1}, g_{i+2}, \ldots, g_j\}$ for some $1 \le i \le j \le n$, and we define $\text{left}_G(S) = \text{left}_G(g_i)$ and $\text{right}_G(S) = \text{right}_G(g_j)$. Given two nonempty sets of consecutive vertices S_1 and S_2, we say $S_1 < S_2$ if and only if S_1 and S_2 are disjoint and $\text{left}_G(S_1) < \text{left}_G(S_2)$. For technical reasons, we say $S_1 < S_2$ when $S_1 = \emptyset$ or $S_2 = \emptyset$.

A graph class $\mathcal{C}$ is *closed under powers* if for every $G \in \mathcal{C}$ and every $k, G^k \in \mathcal{C}$. $\mathcal{C}$ is *strongly closed under powers* if $G^k \in \mathcal{C}$ for some k implies $G^{k+1} \in \mathcal{C}$. The following theorem characterizes the closure property of proper interval graphs under powers.

THEOREM 2.1 (see [33]). *The class of proper interval graphs is strongly closed under powers.*

In particular, if H is a proper interval graph, then H^2 is a proper interval graph. In light of this theorem, to determine if G is the square of a proper interval graph, we can, without loss of generality, assume that G is a proper interval graph.

Given a graph G, there are $\mathcal{O}(n + m)$ algorithms (e.g., [5]) that determine if G is a proper interval graph and construct an interval representation if it is. Therefore, in the following sections, to determine if G is the square of a proper interval graph, we assume G is a proper interval graph and the corresponding interval representation is given. If G is the square of a proper interval graph, we use H to denote a proper interval graph square root of G. We will let $V(G) = \{g_1, g_2, \ldots, g_n\}$ and $V(H) = \{h_1, h_2, \ldots, h_n\}$. We define $G[i, j] = G[g_i, g_{i+1}, g_{i+2}, \ldots, g_j]$ and similarly for H. Without loss of generality, we assume G and H are connected in the rest of this section.

2.1.1. Computing the square of a proper interval graph. Before we discuss how to find a proper interval square root H of a given proper interval graph G, we first consider how to compute H^2 from a proper interval graph H. This will give us insight into how to do the reverse operation. In general graphs, we can compute the square of a graph by doing matrix multiplication. But in proper interval graphs, there is a more effective and yet very intuitive way to compute the square of H by looking at the interval representation of H. Figure 2.1 presents a proper interval graph H and Figure 2.2 shows H^2. In the figures, numbers on the left represent the vertex names while numbers on the right represent the leftmost neighbor names (to be defined later). This example will be used throughout this section.

Given a total vertex ordering of a proper interval graph, the following properties are obvious.

PROPOSITION 2.2. *Given a proper interval graph H with a total vertex ordering, $N_H[h_i]$ is consecutive for any $1 \le i \le n$.*

With this property, it is easy to compute the square of a proper interval graph. By Proposition 2.2, we know that $N_H[h_j]$ is consecutive. Let $N_H[h_j] = \{h_i, \ldots, h_k\}$. We say h_i is the *leftmost neighbor* of h_j denoted by $l_H(h_j)$ and h_k is the *rightmost*

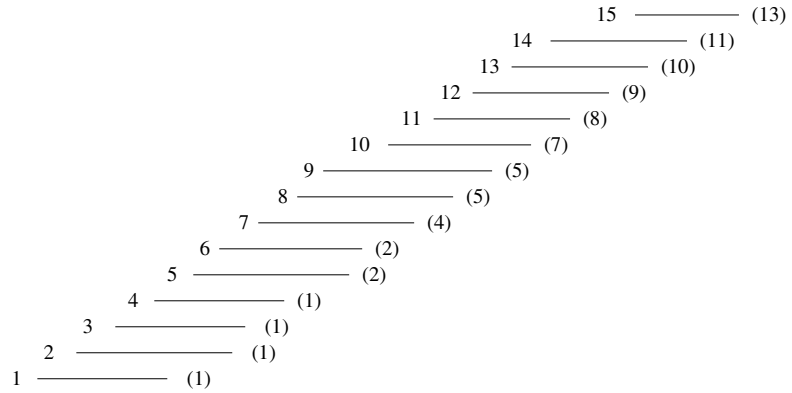


FIG. 2.1. *A proper interval graph H together with the names of the parents.*

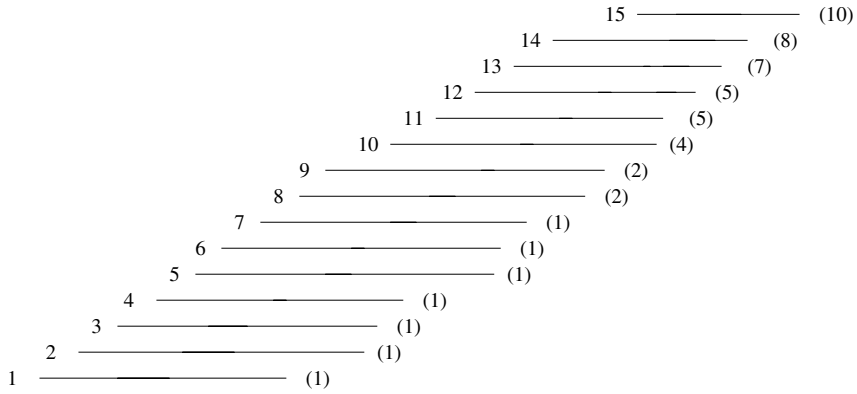


FIG. 2.2. $G = H^2$.

neighbor of h_j denoted by $r_H(h_j)$. For technical reasons, we set $l_H(h_1) = h_1$ and $r_H(h_n) = h_n$.

PROPOSITION 2.3. *Let $N_H[h_j] = \{h_i, \ldots, h_k\}$ with $1 \leq i \leq j \leq k \leq n$. Then $N_{H^2}[h_j] = N_H[h_i] \cup N_H[h_k]$ for any $1 \leq j \leq n$. In other words, $N_{H^2}[h_j] = N_H[l_H(h_j)] \cup N_H[r_H(h_j)]$.*

COROLLARY 2.4. *Given a proper interval graph H, $N^2_H[h_j] = \{l_H(l_H(h_j)), \ldots, r_H(r_H(h_j))\}$ for any $1 \leq j \leq n$.*

Notice that if $i < j$, then $\text{left}(l(l(h_i))) \leq \text{left}(l(l(h_j)))$ and $\text{right}(r(r(h_i))) \leq \text{right}(r(r(h_j)))$. So a total vertex ordering in H is a total vertex ordering in H^2. In other words, if G is the square of a proper interval graph, then it has a proper interval graph square root H such that G and H have the same vertex ordering. We will prove this fact formally later, and it is very useful when we construct a proper interval graph square root since it significantly reduces the search space.

2.1.2. Notation for the algorithm. We introduce some notation to facilitate our discussion. We say v_i is the *parent* of v_j and v_j is a *child* of v_i if v_i is the leftmost neighbor of v_j. In Figures 2.1 and 2.2, the number in parentheses beside an interval indicates the parent of the corresponding vertex. Notice that if v_j is a child of v_i, then v_j is not adjacent to v_{i-1}. For every v_i, there is a unique parent, denoted by $p(v_i)$. On the other hand, v_i may have many children and we denote the set of children of

v_i by $C(v_i)$. For example, in Figure 2.1, $C_H(h_2) = \{h_5, h_6\}$. For technical reasons, $C(v_1) = N(v_1)$ (i.e., $v_1 \notin C(v_1)$) but $p(v_1) = v_1$. Note that v_i may have no children. For example, in Figure 2.1, $C_H(h_3) = \emptyset$. If $v_i = l(l(v_j))$, then we say v_i is the *grandparent* of v_j, denoted by $gp(v_j)$, and v_j is a *grandchild* of v_i.

Let X be a set of consecutive vertices. We define $C(X) = \bigcup_{v \in X} C(v)$. Notice that it is possible that $X \cap C(X) \neq \emptyset$; however, we are interested in only the case when $X \cap C(X) = \emptyset$ and thus $X < C(X)$. $C(X)$ is the union of the set of children of vertices in X. We let $C^0(X) = X$, $C^1(X) = C(X)$, and more generally $C^i(X) = C(C^{i-1}(X))$. Notice that if $C^i(X) = \emptyset$, then $C^{i+1}(X) = \emptyset$. We define $e(X) = k$, where k is the maximum value such that $C^k(X) \neq \emptyset$, and we define $C^*(X) = \bigcup_{0 \leq i \leq e(X)} C^i(X)$. So $C^*(X)$ is the set of descendants of vertices in X together with X. We say that $C^*(X)$ is the *chain* of X and $e(X)$ is the *length* of the chain. If $v' \in C^1(v)$, then v' is a child of v, and if $v' \in C^2(v)$, then v' is a grandchild of v. Furthermore, if $v' \in C^*_G(v)$, then v' is a descendant of v. Finally, if $X < Y < C_G(X)$, then we denote $C^*_G(X) \cup C^*_G(Y)$ by $C^*_G(X, Y)$.

The remainder of this section goes as follows. In subsection 2.2, we will prove that if G is a proper interval graph square, then there exists a proper interval graph H with the same vertex ordering as in G such that $H^2 = G$. Then we will show some properties of chains in subsection 2.3 and prove the decomposition theorem in subsection 2.4. Finally, we present the algorithm and analyze its complexity in subsection 2.5 and extend it to recognize kth powers of proper interval graphs in subsection 2.6.

2.2. H^2 and H share the same vertex ordering.

LEMMA 2.5. *If G is a proper interval graph square, then there exists a proper interval graph H with the same vertex ordering as in G such that $H^2 = G$.*

Proof. We prove by induction on i that there exists H_i such that $\{g_1, \ldots, g_i\}$ in G are mapped to $\{h_1, \ldots, h_i\}$ in H_i and ${H_i}^2 = G$. Thus $H' = H_n$ is a proper interval graph with the same vertex ordering as in G.

First we prove the base case when $i = 1$. Suppose H is a proper interval graph square root of G. If g_1 in G is mapped to h_1 in H, then $H_1 = H$ and we are done. So suppose g_1 in G is mapped to h_j in H such that $1 < j$; we consider two cases.

Case 1. In H, there are $a < j < b$ such that $d_H(h_a, h_j) = 2$ and $d_H(h_j, h_b) = 2$. In H^2, h_j is adjacent to h_a and h_b. Since $I_G(g_1)$ is the leftmost interval in G, g_1 is a simplicial vertex in G and its neighborhood induces a clique in G. Since $G = H^2$, h_j is a simplicial vertex in H^2 and thus h_a and h_b are adjacent in H^2. Since $h_j h_a, h_j h_b \notin E(H)$, $\text{right}_H(h_a) < \text{left}_H(h_j)$ and $\text{right}_H(h_j) < \text{left}_H(h_b)$ in H. Therefore, h_a and h_b are not adjacent in H. Since h_a and h_b are adjacent in H^2, there exists h_c such that $h_a h_c \in E(H)$ and $h_b h_c \in E(H)$. However, this implies that $\text{left}_H(h_c) < \text{right}_H(h_a) < \text{left}_H(h_j) < \text{right}_H(h_j) < \text{left}_H(h_b) < \text{right}_H(h_c)$ and thus $I_H(h_c)$ is an interval that properly contains $I_H(h_j)$. This contradicts the assumption that H is a proper interval graph. Thus Case 1 is impossible.

Case 2. In H, $I_H(h_j)$ either intersects all the intervals on its left or intersects all the intervals on its right. Without loss of generality, we assume the former case such that h_i is a neighbor of h_j for all $i < j$. As $I_G(g_1)$ is the leftmost interval in G, g_1 is a simplicial vertex and $N_G[g_1] \subseteq N_G[g_k]$ if g_k is a neighbor of g_1 in G. Since $G = H^2$ and g_1 is mapped to h_j, $N_{H^2}[h_j] \subseteq N_{H^2}[h_i]$ for all $i < j$. On the other hand, by Corollary 2.4, $\text{right}_H(h_j) > \text{right}_H(h_i)$ implies $N_{H^2}[h_i] \subseteq N_{H^2}[h_j]$ for all $i < j$. Therefore $N_{H^2}[h_i] = N_{H^2}[h_j]$ for all $i < j$. In particular, $N_{H^2}[h_1] = N_{H^2}[h_j]$. Therefore, we can construct H_1 from H by switching the preimages of h_1 and h_j. Clearly, ${H_1}^2 = H^2 = G$. Thus the base case holds.

Now we assume $g_1, \ldots, g_k$ in G are mapped to $h_1, \ldots, h_k$ in H_k, respectively, and ${H_k}^2 = G$. We will construct H_{k+1} from H_k such that $g_1, \ldots, g_{k+1}$ in G are mapped to $h_1, \ldots, h_{k+1}$ in H_{k+1}, respectively, and $H_{k+1}^2 = G$. If g_{k+1} in G is mapped to h_{k+1} in H_k, then $H_{k+1} = H_k$ and we are done. So suppose g_{k+1} in G is mapped to h_j in H_k with $j > k+1$. Since $I_G(g_{k+1})$ is the leftmost interval in $G[k+1, n]$, by Corollary 2.4, we have $N_{G[1,k]}[g_i] \subseteq N_{G[1,k]}[g_{k+1}]$ for $i > k+1$. Since $G = {H_k}^2$ and g_{k+1} is mapped to h_j, $N_{{H_k}^2[1,k]}[h_i] \subseteq N_{{H_k}^2[1,k]}[h_j]$ for $k+1 \leq i < j$. On the other hand, since $\text{left}_{H_k}(h_i) < \text{left}_{H_k}(h_j)$ for any $k+1 \leq i < j$, by Corollary 2.4, $N_{{H_k}^2[1,k]}[h_j] \subseteq N_{{H_k}^2[1,k]}[h_i]$ for $k+1 \leq i < j$. So, $N_{{H_k}^2[1,k]}[h_i] = N_{{H_k}^2[1,k]}[h_j]$ for $k+1 \leq i < j$. By essentially the same argument, we also have $N_{{H_k}^2[k+1,n]}[h_j] = N_{{H_k}^2[k+1,n]}[h_i]$ for $k+1 \leq i < j$.

Therefore $N_{{H_k}^2}[h_i] = N_{{H_k}^2}[h_j]$ for $k+1 \leq i < j$. In particular, $N_{{H_k}^2}[h_{k+1}] = N_{{H_k}^2}[h_j]$. Thus we may construct H_{k+1} from H_k by switching the preimages of h_j and h_{k+1}. Clearly, $H_{k+1}^2 = {H_k}^2 = G$ and $g_1, \ldots, g_{k+1}$ in G are mapped to $h_1, \ldots, h_{k+1}$ in H_{k+1}. By induction, we can construct H_n. Thus $H' = H_n$, and this completes the proof. ∎

By Lemma 2.5, we can assume that if G is a proper interval graph square, then there is a proper interval graph H such that $H^2 = G$ and H and G share the same vertex ordering (i.e., g_i in G is mapped to h_i in H for all i). Later on, g_i and h_i actually mean the same vertex but g_i refers to the one in G and h_i refers to the one in H.

2.3. Parent and children relationship. Now we characterize H based on the parent-children relationship in G which is analogous to Corollary 2.4. But in the following lemma, we characterize H by just using the leftmost neighbors instead of using both leftmost and rightmost neighbors. This allows us to construct H by considering one direction only (from left to right, i.e., from 1 to n).

LEMMA 2.6. *Let G be a proper interval graph square and let H be a proper interval graph with the same vertex ordering as in G. $H^2 = G$ if and only if $gp_H(h_j) = p_G(g_j)$ for $1 \leq j \leq n$.*

Proof. Since G and H share the same vertex ordering, if $H^2 = G$, by Corollary 2.4, $gp_H(h_j) = p_G(g_j)$ for $1 \leq j \leq n$.

Now we prove the reverse direction. If $gp_H(h_j) = p_G(g_j)$ for $1 \leq j \leq n$, we prove that $H^2 = G$ by induction. First we prove the base case that $N_{H^2}[h_n] = N_G[g_n]$. By Corollary 2.4, $N_{H^2}[h_n] = \{l_H(l_H(h_n)), \ldots, h_n\}$. Since $l_H(l_H(h_n)) = gp_H(h_n) = p_G(g_n)$ and G and H share the same vertex ordering, $N_{H^2}[h_n] = N_G[g_n]$ and thus the base case holds.

Now suppose that $N_{H^2}[h_j] = N_G[g_j]$ for $k+1 \leq j \leq n$. Consider $N_{H^2}[h_k]$; by the induction hypothesis, $N_{H^2}[h_k] \cap \{h_{k+1}, \ldots, h_n\} = N_G[g_k] \cap \{g_{k+1}, \ldots, g_n\}$. Since $l_H(l_H(h_k)) = gp_H(h_k) = p_G(g_k)$ and G and H share the same vertex ordering, $N_{H^2}[h_k] \cap \{h_1, \ldots, h_k\} = N_G[g_k] \cap \{g_1, \ldots, g_k\}$. Therefore, we conclude that $N_{H^2}[h_k] = N_G[g_k]$, and this completes the induction step. ∎

The following easily proved proposition describes a basic property of a chain based on the definition.

PROPOSITION 2.7. *If $X < Y < C(X)$, then $C^i(X) < C^i(Y) < C^{i+1}(X)$ for $i \geq 0$.*

The following lemma formalizes the idea that a chain in H is composed of two chains in G.

LEMMA 2.8. *Let $H^2 = G$, where H and G share the same vertex ordering. If $X < Y < C_G(X)$ and $C_H(X) = Y$, then $C_H(C_G^i(X)) = C_G^i(Y)$ and $C_H(C_G^i(Y)) = C_G^{i+1}(X)$ for $i \geq 0$. Furthermore, $e_G(X) = e_G(Y)$ or $e_G(X) = e_G(Y) + 1$.*

Proof. Since $X < Y < C_G(X)$ and $C_H(X) = Y$, by Lemma 2.6, $C_H(Y) = C_G(X)$.

And since $H^2 = G$, if $C_G(X) \neq \emptyset$, then $Y \neq \emptyset$. By Proposition 2.7, $C_G(X) < C_G(Y)$. Repeating the same argument, since $Y < C_G(X) < C_G(Y)$ and $C_H(Y) = C_G(X)$, we have $C_H(C_G(X)) = C_G(Y)$. And since $H^2 = G$, if $C_G(Y) \neq \emptyset$, then $C_G(X) \neq \emptyset$. By induction, we have $C_H(C_G^i(Y)) = C_G^{i+1}(X)$ and $C_H(C_G^i(X)) = C_G^i(Y)$. Also, by Lemma 2.6, if $C_G^{i+1}(X) \neq \emptyset$, then $C_G^i(Y) \neq \emptyset$; if $C_G^{i+1}(Y) \neq \emptyset$, then $C_G^{i+1}(X) \neq \emptyset$. Thus $e_G(X) = e_G(Y)$ or $e_G(X) = e_G(Y) + 1$, and this completes the proof. □

Thus, a necessary condition for $C_G^*(X, Y)$ to form $C_H^*(X)$ is that the length of $C_G^*(X)$ and the length of $C_G^*(Y)$ are about the same.

The following two propositions are useful for decomposition. Their proofs follow directly from the definitions.

PROPOSITION 2.9. *Suppose $X < C(X)$. If $X_1 \cup X_2 = X$ and $X_1 < X_2$, then $C^i(X_1) \cup C^i(X_2) = C^i(X)$ and $C_G^i(X_1) < C_G^i(X_2)$ for $i \geq 0$.*

PROPOSITION 2.10. *Suppose $X < Y < C(X)$. Let $X_1 \cup X_2 = X$ and $X_1 < X_2$ and similarly $Y_1 \cup Y_2 = Y$ and $Y_1 < Y_2$. Then $C^i(X) = C^i(X_1) \cup C^i(X_2)$, $C^i(X_1) < C^i(X_2)$ and $C^i(Y) = C^i(Y_1) \cup C^i(Y_2)$, $C^i(Y_1) < C^i(Y_2)$ for $i \geq 0$. Also, $C^i(X_1) < C^i(X_2) < C^i(Y_1) < C^i(Y_2) < C^{i+1}(X_1)$ for $i \geq 0$.*

Proof. The first two statements follow from Proposition 2.9. The last statement follows from Proposition 2.7. □

2.4. A decomposition theorem. The following is an important definition for our algorithm. Intuitively, it checks if $C_G^*(X, Y)$ can form $C_H^*(X)$; in other words, it checks if $G[C_G^*(X, Y)]$ has a square root with special properties.

DEFINITION 2.11. *Suppose $X < Y < C_G(X)$; $G[C_G^*(X, Y)]$ is* matched *if and only if there exists H' with the same vertex ordering as $G[C_G^*(X, Y)]$ such that*

(P1) *$H'^2 = G[C_G^*(X, Y)]$;*

(P2) *$H'[X]$ is a clique;*

(P3) *$p_{H'}(h_i) \in X$ if and only if $h_i \in X \cup Y$.*

We say H' is a matched root *of $G[C_G^*(X, Y)]$.*

We will decompose G into small graphs that correspond to chains in H; then we will construct matched roots (i.e., chains in H) independently and combine them to form a root of G. Note that (P2) and (P3) are necessary conditions for H' to be $H[C_H^*(X)]$.

The following lemma is a rephrasing of Lemma 2.6 in the case of matched roots. It characterizes matched roots based on the parent-children relationship in G.

LEMMA 2.12. *Suppose $X < Y < C_G(X)$; let H have the same vertex set and the same vertex ordering as $G[C_G^*(X, Y)]$ and assume H satisfies* (P2) *and* (P3)*. Then $H^2 = G[C_G^*(X, Y)]$ if and only if $gp_H(h_i) = p_G(g_i)$ for all $g_i \in C_G^*(X, Y) - X - Y$.*

Proof. By Lemma 2.6, $H^2 = G[C_G^*(X, Y)]$ if and only if $gp_H(h_i) = p_{C_G^*(X,Y)}(g_i)$ for all $g_i \in C_G^*(X, Y)$. Notice that $p_{C_G^*(X,Y)}(g_i) = p_G(g_i)$ for all $g_i \in C_G^*(X, Y) - X$. Also, (P2) and (P3) holding for H guarantee $gp_H(h_i) = p_G(g_i)$ for all $g_i \in X \cup Y$. Therefore, $H^2 = G[C_G^*(X, Y)]$ if and only if $gp_H(h_i) = p_G(g_i)$ for all $g_i \in C_G^*(X, Y) - X - Y$. □

The following is a rephrasing of Lemma 2.8 in the case of a matched root. Note that the only difference between Lemmas 2.13 and 2.8 is that $C_H(C_G^0(X)) = C_G^0(Y)$ holds in Lemma 2.8.

LEMMA 2.13. *Suppose $G[C_G^*(X, Y)]$ is matched and let H be a matched root of $G[C_G^*(X, Y)]$. Then $C_H(C_G^i(X)) = C_G^i(Y)$ for $i \geq 1$ and $C_H(C_G^i(Y)) = C_G^{i+1}(X)$ for $i \geq 0$. Furthermore, $e_G(X) = e_G(Y)$ or $e_G(X) = e_G(Y) + 1$.*

Proof. By (P3) of H, $p_H(h_i) \in X$ if and only if $h_i \in X \cup Y$. Since $H^2 = G[C_G^*(X, Y)]$, $C_H(Y) = C_G(X)$. By Proposition 2.7, $C_G(X) < C_G(Y)$. Since $C_H(Y) <$

$C_G(X) < C_G(Y)$ and $C_H(Y) = C_G(X)$, by Lemma 2.8, the results follow. □

As mentioned previously, if $H^2 = G$, then we can combine matched roots to construct a root of G. The following lemma shows the reverse direction, namely, we can find matched roots in H. This justifies our approach of constructing matched roots.

LEMMA 2.14. *Let H have the same vertex ordering as in G and $H^2 = G$. If $X < Y < C_G(X)$, $H[X]$ is a clique, and $C_H(X) = Y$, then $G[C^*_G(X,Y)]$ is matched.*

Proof. Let H'=$H[C^*_H(X)]$; we will show that H' is a matched root of $G[C^*_G(X,Y)]$. First, since $H^2 = G$, $X < Y < C_G(X)$, and $C_H(X) = Y$, by Lemma 2.8, $C^*_H(X) = C^*_G(X,Y)$. We now show (P1)–(P3) are satisfied for H'. Since $H[X]$ is a clique, (P2) is satisfied. Since $C_H(X) = Y$, (P3) is satisfied. For (P1), by our construction, $p_{H'}(h_i) = p_H(h_i)$ for all $h_i \in C^*_H(X) - X$. Thus in H', $gp_{H'}(h_i) = gp_H(h_i)$ for all $h_i \in C^*_H(X) - X - C_H(X)$. Since $H^2 = G$, by Lemma 2.6, $gp_H(h_i) = p_G(g_i)$ for all i. Therefore $gp_{H'}(h_i) = p_G(g_i)$ for all $g_i \in C^*_G(X,Y) - X - Y$. By Lemma 2.12, $H'^2 = G[C^*_G(X,Y)]$ and thus (P1) is satisfied; by Definition 2.11, $G[C^*_G(X,Y)]$ is matched. □

Now we have enough machinery to prove the decomposition theorem. First, the following theorem reduces the original problem to finding a matched root. Intuitively, it corresponds to the partition of the neighborhood of g_1 in G (i.e., guessing the neighborhood of h_1 in H).

THEOREM 2.15. *G is a proper interval graph square if and only if $G[C^*_G(X,Y)]$ is matched for some X, where $X \cup Y = C_G(g_1)$ and $X < Y$.*

Proof. If G is a proper interval graph square, by Lemma 2.5, then there exists H with the same vertex ordering as in G and $H^2 = G$. Let $X = C_H(h_1)$ and $Y = C_G(g_1) - X$; clearly $X \cup Y = C_G(g_1)$ and $X < Y$. We will show that $G[C^*_G(X,Y)]$ is matched. In H, since $X = C_H(h_1)$, $H[X]$ is a clique. Since $Y = C_G(g_1) - X$ and $H^2 = G$, every vertex in Y has a neighbor in X and thus $C_H(X) = Y$. Thus $H[X]$ is a clique, $C_H(X) = Y$, and $X < Y < C_G(X)$; by Lemma 2.14, $G[C^*_G(X,Y)]$ is matched.

Now suppose $G[C^*_G(X,Y)]$ is matched for some X, where $X \cup Y = C_G(g_1)$ and $X < Y$. Let H' be a matched root of $G[C^*_G(X,Y)]$. We show how to construct H from H' such that $H^2 = G$. Since $X \cup Y = C_G(g_1)$, by Proposition 2.9, $C^*_G(X,Y) = C^*_G(C_G(g_1))$. Since $C_G(g_1) = N_G(g_1)$, $C^*_G(C_G(g_1)) = G - g_1 = \{g_2, \ldots, g_n\}$. Since H' is a matched root of $G[C^*_G(X,Y)]$, $V(H') = \{h_2, \ldots, h_n\}$. Since $H'^2 = G - g_1$, $H'[X]$ is a clique and $C_{H'}(X) = Y$, by constructing $H = H' + h_1$ with $C_H(h_1) = X$, it is easy to verify that $H^2 = G$. So G is a proper interval graph square. □

The following theorem is the core of the algorithm. It reduces the problem of finding a matched root in a graph to finding matched roots in two smaller graphs. It is the basis of the "decomposition" step in the algorithm.

THEOREM 2.16. *Assume X and Y with $X < Y < C_G(X)$ in G with $|X| \geq 2$. Let g_a be the first vertex in X. Then $G[C^*_G(X,Y)]$ is matched if and only if $G[C^*_G(g_a, A)]$ and $G[C^*_G(X - g_a, Y - A)]$ are both matched for some A, where $A < Y - A$. (Note that A could be an empty set.)*

Proof. Suppose $G[C^*_G(X,Y)]$ is matched and let H' be a matched root of $G[C^*_G(X,Y)]$. We will show $G[C^*_G(g_a, A)]$ and $G[C^*_G(X - g_a, Y - A)]$ are both matched for some A, where $A < Y - A$. In particular, we set $A = C_{H'}(h_a) - X$. Since H' has the same vertex ordering as $G[C^*_G(X,Y)]$, $A < Y - A$. If $C_G(X) = \emptyset$, by Lemma 2.13, $C^*_G(X,Y) = X \cup Y$ and clearly $G[C^*_G(g_a, A)]$ and $G[C^*_G(X - g_a, Y - A)]$ are both matched. Thus we assume $C_G(X) \neq \emptyset$; by Lemma 2.13, $C_{H'}(Y) = C_G(X)$. Since $H'^2 = G[C^*_G(X,Y)]$, by our choice of A, $C_{H'}(A) = C_G(g_a)$ and $C_{H'}(Y - A) = C_G(X - g_a)$. By Lemma 2.14, both $G[C^*_G(A, C_G(g_a))]$ and $G[C^*_G(Y - A, C_G(X - g_a))]$ are matched and we let H_1 and H_2 be the corresponding matched roots.

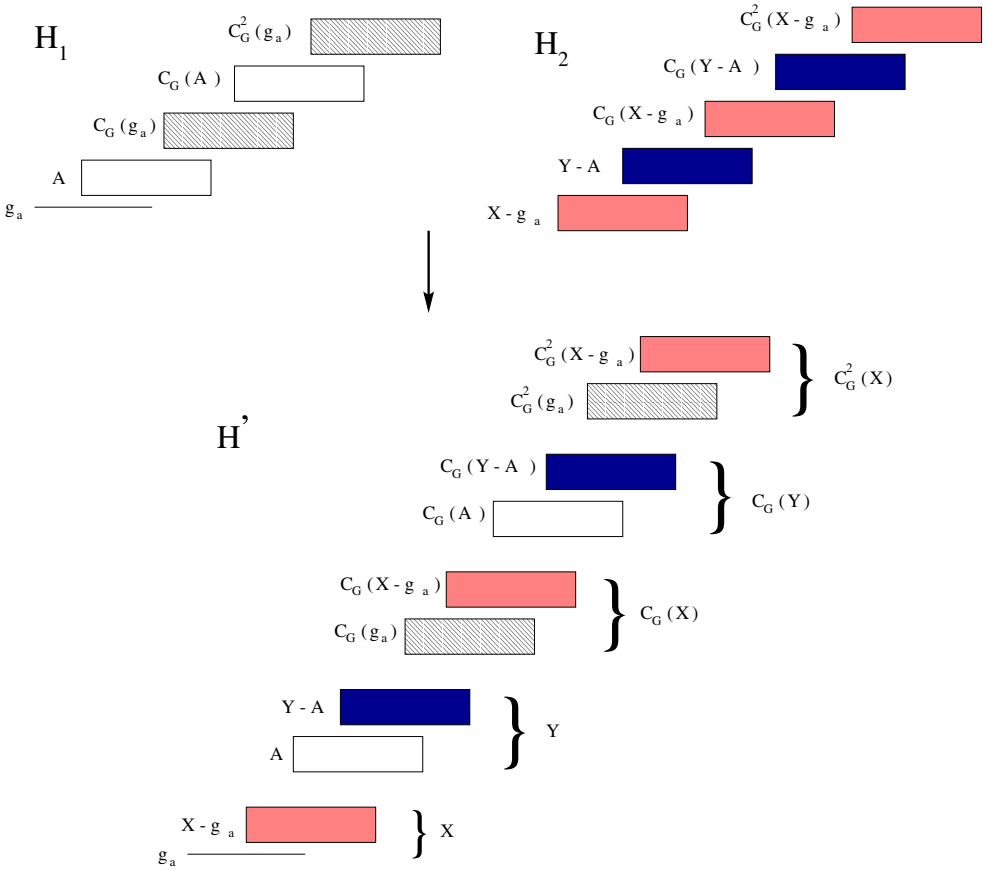


FIG. 2.3. *Combine matched root H_1 of $C^*_G(g_a, A)$ and matched root H_2 of $C^*_G(X - g_a, Y - A)$ into matched root H' of $C^*_G(X, Y)$.*

We will show that $H'_1 = H'[H_1 + h_a]$ and $H'_2 = H'[H_2 + (X - h_a)]$ are matched roots of $G[C^*_G(g_a, A)]$ and $G[C^*_G(X - g_a, Y - A)]$, respectively. By (P2) for H', $H'[X]$ is a clique, so $H'[h_a]$ and $H'[X - h_a]$ are cliques and thus (P2) is satisfied in H'_1 and H'_2. By (P3) for H', $p_{H'}(h_i) \in X$ if and only if $h_i \in X \cup Y$. Since we set $A = C_{H'}(h_a) - X$, it is clear that (P3) is satisfied in H'_1 and H'_2. By (P1) for H_1 and H_2, ${H_1}^2 = G[C^*_G(A, C_G(g_a))]$ and ${H_2}^2 = G[C^*_G(Y - A, C_G(X - g_a))]$. Since H' is a matched root of $G[C^*_G(X, Y)]$, it is easy to check that ${H'_1}^2 = G[C^*_G(g_a, A)]$ and ${H'_2}^2 = G[C^*_G(X - g_a, Y - A)]$. Thus, by Definition 2.11, $G[C^*_G(g_a, A)]$ and $G[C^*_G(X - g_a, Y - A)]$ are both matched.

Suppose $G[C^*_G(g_a, A)]$ and $G[C^*_G(X - g_a, Y - A)]$ are both matched for some A, where $A < Y - A$, and let H_1 and H_2 be the corresponding matched roots. We will show how to construct a matched root H' of $G[C^*_G(X, Y)]$. By Proposition 2.10, $C^*_G(g_a, A) \cup C^*_G(X - g_a, Y - A) = C^*_G(X, Y)$. Also by Proposition 2.10, we know the total vertex order of $H_1 \cup H_2$. We can construct H' by combining H_1 and H_2 with the order indicated by Proposition 2.10 and setting $p_{H'}(h_i) = h_a$ for $h_i \in X - h_a$. Notice that in this ordering, if $i \leq j$, then $\text{left}(p_{H'}(h_i)) \leq \text{left}(p_{H'}(h_j))$. Thus H' is constructible with the parent-children relationship unchanged as in H_1 and H_2, except for $h_i \in X - h_a$. The concept is shown in Figure 2.3, where we combine two matched roots to a matched root; recall that a matched root corresponds to two chains

in G. It is important to note that, in Figure 2.3, the parent-children relationship in H' unchanged as in H_1 and H_2 (by looking at the intersection of intervals), except for $h_i \in X - h_a$. It is also worth noting that, unlike Figure 2.3, the length of $C^*_G(g_a, A)$ and the length of $C^*_G(X - g_a, Y - A)$ could be very different.

By Lemma 2.12, $gp_{H_1}(h_i) = p_G(g_i)$ for all $g_i \in C^*_G(g_a, A) - g_a - A$ and $gp_{H_2}(h_j) = p_G(g_j)$ for all $g_j \in C^*_G(X - g_a, Y - A) - (X - g_a) - (Y - a)$. As the parent relationship in H' is unchanged as in H_1 and H_2, except for vertices in $X - g_a$, so $gp_{H'}(h_i) = p_G(g_i)$ for all $g_i \in C^*_G(X, Y) - X - Y$. Since we set $p_{H'}(h_i) = h_a$ for $h_i \in X - g_a$, $H'[X]$ is a clique and thus H' satisfies (P2). By (P3) for H_1 and H_2, $p_{H_1}(h_i) \in \{h_a\}$ if and only if $h_i \in \{h_a\} \cup A$ and $p_{H_2}(h_j) \in X - h_a$ if and only if $h_j \in (X - h_a) \cup (Y - A)$. Without changing the parent relationship except for vertices in $X - g_a$, H' satisfies (P3). By Lemma 2.12, ${H'}^2 = G[C^*_G(X, Y)]$ and thus $G[C^*_G(X, Y)]$ is matched. □

The following theorem is the basis of the "propagation" step. It reduces the problem of finding a matched root in a graph to finding a matched root in a smaller graph; in particular, it reduces the length of the chain by one.

THEOREM 2.17. *Given $X = \{g_i\}$ and Y with $\{g_i\} < Y < C_G(g_i)$, $G[C^*_G(\{g_i\}, Y)]$ is matched if and only if $G[C^*_G(Y, C_G(g_i))]$ is matched.*

Proof. Suppose $G[C^*_G(\{g_i\}, Y)]$ is matched and let H' be a matched root of $G[C^*_G(\{g_i\}, Y)]$. We will show that $H' - h_i$ is a matched root of $G[C^*_G(Y, C_G(g_i))]$. By (P3) for H', h_i is the parent of the vertices in Y in H'. So $H'[Y]$ is a clique and thus (P2) is satisfied in $H' - h_i$. Also, by Lemma 2.13, $C_{H'}(Y) = C_G(g_i)$ and thus (P3) is satisfied in $H' - h_i$. And clearly, $(H' - h_i)^2 = G - g_i$. By Definition 2.11, $G[C^*_G(Y, C_G(g_i))]$ is matched.

Suppose $G[C^*_G(Y, C_G(g_i))]$ is matched and let H_1 be a corresponding matched root. By (P2), $H_1[Y]$ is a clique. By (P3), $p_{H_1}(h_j) \in Y$ if and only if $h_j \in Y \cup C_G(g_i)$. By constructing $H' = H_1 + h_i$ with $C_H(h_i) = Y$, ${H'}^2 = G[C^*_G(\{g_i\}, Y)]$ and (P2), (P3) are satisfied in H'. Thus, by Definition 2.11, $G[C^*_G(\{g_i\}, Y)]$ is matched. □

2.5. Algorithm, correctness, and complexity. The following recursive algorithm is an implementation of the decomposition procedure. P is a *prefix* of S if $P < S - P$; note that P could be an empty set.

```
MAIN PROGRAM
    for every prefix P of C_G(g_1) do
        if MATCH (P, C_G(g_1) − P) then output "YES"
    output "NO"
```

```
MATCH (X = {g_a, ...}, Y)
    Case: X = ∅
        if Y = ∅ then return TRUE
        else return FALSE
    Case: X = {g_a}
        return MATCH (Y, C_G(g_a))
    Otherwise:
        for every prefix P of Y do
            if (MATCH ({g_a}, P) and MATCH (X − g_a, Y − P))
            then return TRUE
        return FALSE
```

THEOREM 2.18. *The algorithm is correct.*

Proof. In MAIN PROGRAM, we reduce the recognition of the square of a proper interval graph to finding a matched root. Then we use the results obtained in the previous subsection to find a matched root; this is done by the recursive function MATCH.

The correctness of the MAIN PROGRAM is justified by Theorem 2.15. Now we look into the recursive function MATCH. The first case is the base case when the X-chain ends. If the Y-chain also ends, then X and Y are matched. Otherwise, X and Y are not matched. The second case is the "propagation" step. Its correctness is justified by Theorem 2.17. The final case is the "decomposition" step. Its correctness is justified by Theorem 2.16. □

THEOREM 2.19. SQUARE OF PROPER INTERVAL GRAPH *can be solved in* $\mathcal{O}(n^5)$.

Proof. The key observation is that $X=\{g_a, g_{a+1}, \ldots, g_b\}$ and $Y=\{g_c, g_{c+1}, \ldots, g_d\}$ are consecutive sets. Thus there are at most $\mathcal{O}(n^4)$ instances of MATCH since there are at most n possibilities of each a, b, c, d. Therefore, we can use dynamic programming to solve this problem. Explicitly, we can create a table of size $\mathcal{O}(n^4)$ to store the results of all possible instances. Each entry in the table can be computed in time at most $\mathcal{O}(n)$. The total complexity is at most $\mathcal{O}(n^5)$. To implement the algorithm, we can use a standard trick of "caching" the solutions when we run the recursion so that each entry of the table is computed at most once. □

Notice that the above algorithm is for the decision problem where a proper interval graph is a proper interval graph square. To actually find a proper interval graph square root, we need to add another four-dimensional array to trace the partitions. This is a standard technique of dynamic programming and is covered in many textbooks. For simplicity in the description of the algorithm, we do not include the details of finding the partitions.

Now suppose we know the partitions; we show how to construct a proper interval square root H of G. Suppose $X = \{h_a, \ldots, h_b\}$ and $Y = \{h_c, \ldots, h_d\}$ and the partition is $(h_a, \{h_c, \ldots, h_i\})$ and $(X - h_a, \{h_{i+1}, \ldots, h_d\})$. Then we set $C_H(h_a) = \{h_c, \ldots, h_i\}$. Suppose $X = \{h_a\}$ and $Y = \{h_c, \ldots, h_d\}$; then we set $C_H(h_a) = Y$. So from the partitions, we can deduce the parent for any vertex. Then from the parents, we can obtain the adjacency matrix of a proper interval graph square root.

2.6. Outline of the recognition algorithm of *k*th powers of proper interval graphs. By applying the same idea of the recognition algorithm of the square of a proper interval graph, we can develop a polynomial time algorithm for the recognition of the kth power of a proper interval graph for any fixed k. We will not go into full details. Also, we assume that G is not a complete graph. The outline of the algorithm is as follows:

```
MAIN PROGRAM
    for every partition of C_G(g_1) into k consecutive sets S_1 < ··· < S_k
        if MATCH (S_1, S_2, ..., S_k) then output "YES"
    output "NO"
```

```
MATCH (S_1 = {g_a, ...}, S_2, ..., S_k)
    Case: S_1 = ∅
        if S_2, ..., S_k are all emptyset then return TRUE
        else return FALSE
    Case: S = {g_a}
        return MATCH (S_2, ..., S_k, C_G(g_a))
    Otherwise:
        for every combination of prefixes P_2, ..., P_k of S_2, ..., S_k do
            if (MATCH ({g_a}, P_2, ..., P_k) and
                MATCH (S_1 − g_a, S_2 − P_2, ..., S_k − P_k))
            then return TRUE
        return FALSE
```

THEOREM 2.20. *k*TH POWER OF PROPER INTERVAL GRAPH *can be solved in time* $\mathcal{O}(n^{3k-1})$ *and space* $\mathcal{O}(n^{2k})$.

Proof. To apply dynamic programming, we have to create a table of size $\mathcal{O}(n^{2k})$ since there are k pairs of integers. The main program is of complexity $\mathcal{O}(n^{k-1})$. In MATCH, it takes at most $\mathcal{O}(n^{k-1})$ to compute one entry in the table. The total complexity is at most $\mathcal{O}(n^{3k-1})$. □

3. NP-completeness. In this section, we will show that recognizing squares of chordal graphs, finding square roots of chordal graphs, and recognizing squares of split graphs are all NP-complete. We will use SET SPLITTING and INTERSECTION GRAPH BASIS as formulated in [12] for our reductions.

PROBLEM	[SP4] SET SPLITTING
INSTANCE	Collection C of finite sets of elements from S.
QUESTION	Is there a partition of S into two subsets S_1 and S_2 such that no subset in C is entirely contained in either S_1 or S_2?
NOTE	It is also known as HYPERGRAPH 2-COLORABILITY.
PROBLEM	[GT59] INTERSECTION GRAPH BASIS [19]
INSTANCE	Graph $G = (V, E)$, positive integer $K \leq \|E\|$.
QUESTION	Is G the intersection graph for a family of sets whose union has cardinality K or less, i.e., is there a K-element set S and for each $v \in V$ a subset $S[v] \subseteq S$ such that $\{u, v\} \in E$ if and only if $S[u]$ and $S[v]$ are not disjoint?

We will borrow the tail structure in [29] of a vertex v to ensure v has the same neighborhood in any square root H of G. It enables one to exactly pin down the neighborhood of v in any square root H of G.

LEMMA 3.1 (see [29]). *If a, b, c, d are vertices of G such that*

- *the only neighbors of a are b and c,*
- *the only neighbors of b are a, c, and d,*
- $c \leftrightarrow d$,

then the neighbors, in $V - \{a, b, c, d\}$, of d in any square root of G are the same as the neighbors, in $V - \{a, b, c, d\}$, of c in G. (See Figure 3.1 *for an illustration.)*

3.1. Squares of chordal graphs. In this subsection, we will show that to determine if G is the square of a chordal graph is NP-complete.

PROBLEM	SQUARE OF CHORDAL GRAPH
INSTANCE	A graph $G = (V, E)$.
QUESTION	Does there exist a *chordal graph* H such that $G = H^2$?

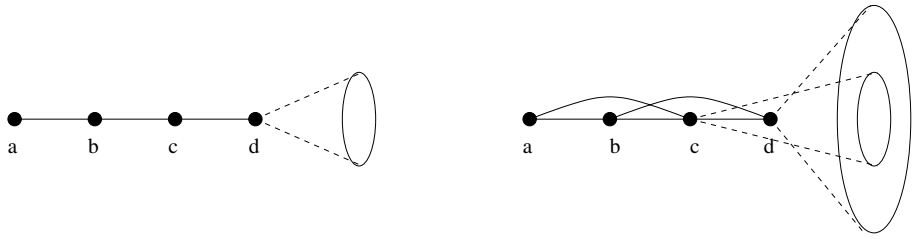


FIG. 3.1. *Tail in H and $G = H^2$.*

The rest of this section shows that SQUARE OF CHORDAL GRAPH is NP-hard by reducing SET SPLITTING to it. It is clear that SQUARE OF CHORDAL GRAPH is in NP, since guessing the square root H, verifying that H is a chordal graph, and $G = H^2$ can easily be done in polynomial time. Thus we will conclude that SQUARE OF CHORDAL GRAPH is NP-complete.

3.1.1. The reduction. Given an instance of SET SPLITTING, we construct an instance of SQUARE OF CHORDAL GRAPH. Let c_j be the set of elements in subset j and let $C = \{c_1, \ldots, c_m\}$. Let $S = \{u_1, \ldots, u_n\}$ be the ground set. The graph G is constructed as follows (note that we will be using Lemma 3.1):

Vertices of G

- *Element vertices:* U_i: $1 \leq i \leq n$ for each element u_i in S.
- *Subset vertices:* C_j for each subset $c_j \in C$ and tail vertices C_j^1, C_j^2, C_j^3 for each c_j.
- *Partition vertices:* S_1 and S_2.

Edges of G

- *Edges of tail vertices of subset vertices:* for all $c_j \in C$,
  $C_j^3 \leftrightarrow C_j^2$, $C_j^3 \leftrightarrow C_j^1$,
  $C_j^2 \leftrightarrow C_j^1$, $C_j^2 \leftrightarrow C_j$,
  $C_j^1 \leftrightarrow C_j$ and $C_j^1 \leftrightarrow U_i$ for all $u_i \in c_j$.
- *Edges of subset vertices:* for all $c_j \in C$,
  $C_j \leftrightarrow S_1$, $C_j \leftrightarrow S_2$, $C_j \leftrightarrow U_i$ for all i, and $C_j \leftrightarrow C_k$ if and only if $c_j \cap c_k \neq \emptyset$.
- *Edges of element vertices:* for all $u_i \in U$,
  $U_i \leftrightarrow U_j$ for all $j \neq i$ and $U_i \leftrightarrow S_1$ and $U_i \leftrightarrow S_2$.

Before presenting the details of the proof, we first give some intuition behind the transformation. From the tail structure, C_j^1 pins down the neighborhood of C_j in any square root H of G. So in any square root H of G, C_j is adjacent to U_i if and only if $u_i \in c_j$. Also, S_1 and S_2 are adjacent to all U_i in G to force S_1 and S_2 to have a common neighbor to all C_j in H. Moreover, S_1 and S_2 are not adjacent in G to force S_1 and S_2 to have no common neighbor in H and thus S_1, S_2 represents a partition of the ground set S.

LEMMA 3.2. *If there is a partition of S into two subsets S_1 and S_2 such that no subset in C is entirely contained in either S_1 or S_2, then there exists a chordal graph H such that $H^2 = G$.*

Proof. Edges of H.

- *Edges of subset vertices and its tail vertices:*
  $C_j^3 \leftrightarrow C_j^2$, $C_j^2 \leftrightarrow C_j^1$, $C_j^1 \leftrightarrow C_j$, and $C_j \leftrightarrow U_i$ if and only if $u_i \in c_j$.
- *Edges of partition vertices:*
  $S_k \leftrightarrow U_i$ if and only if $u_i \in S_k$.
- *Edges of subset vertices:*
  $U_i \leftrightarrow U_j$ for $i \neq j$.

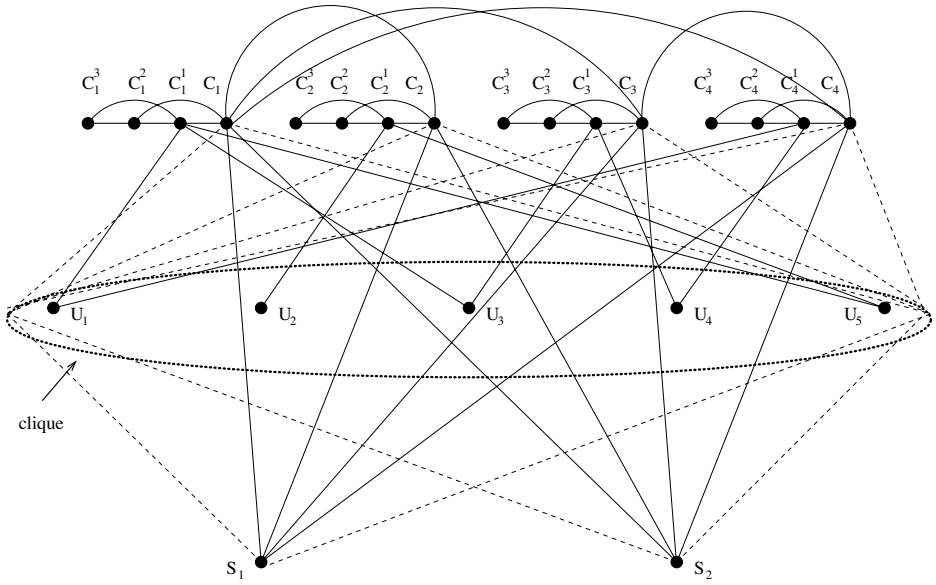


FIG. 3.2. *An example of G.*

It is a tedious but straightforward task to check that $H^2 = G$. We leave the details to the reader (see [23] for a full proof). □

For example, given $C = \{c_1, c_2, c_3, c_4\}$, $c_1 = \{u_1, u_2, u_3\}$, $c_2 = \{u_2, u_5\}$, $c_3 = \{u_3, u_4\}$, $c_4 = \{u_1, u_4\}$, and $S = \{u_1, u_2, u_3, u_4, u_5\}$, we construct G as shown in Figure 3.2. The ellipse corresponds to a clique and we omit the clique edges to keep the figure simpler. Also in the figure, C_1, C_2, C_3, C_4, S_1, and S_2 have two dotted lines to the central ellipse. This indicates that each of them is universal to the vertices in the central ellipse. In this example, $S_1 = \{u_1, u_3, u_5\}$ and $S_2 = \{u_2, u_4\}$ is a possible solution. The graph H corresponding to this solution is shown in Figure 3.3. The reader may verify that $H^2 = G$ and H is chordal. A possible perfect elimination ordering of H is $\{C_1^3, \ldots, C_4^3, C_1^2, \ldots, C_4^2, C_1^1, \ldots, C_4^1, C_1, \ldots, C_4, S_1, S_2, U_1, \ldots, U_5\}$.

We now show that if G is a square, then there is a partition of S into two subsets S_1 and S_2 such that no subset in C is entirely contained in either S_1 or S_2. First, observing that $\{C_j^3, C_j^2, C_j^1, C_j\}$ satisfies the properties of Lemma 3.1, we have the following consequence.

PROPOSITION 3.3. *If H is a square root of G, then in H, C_j is adjacent only to the following: U_i if $u_i \in c_j$, and C_j^1.*

LEMMA 3.4. *If H is a square root of G, then there is a partition of S into two subsets S_1 and S_2 such that no subset in C is entirely contained in either S_1 or S_2.*

Proof. Proposition 3.3 forces each subset vertex to be adjacent to its own elements only. Together with the fact that S_1 and S_2 are not adjacent to any tail vertices in G, S_1 and S_2 only have neighbors in the element set in H. Since $S_1 \nleftrightarrow S_2$ in G, they have no common element neighbor in H and so it is a partition on the element set. And since S_1 and S_2 are adjacent to all subset vertices in G but not in H, S_1 and S_2 have a common neighbor with C_i in the element set for all i. Therefore, $N_H(S_1)$ and $N_H(S_2)$ are the desired partitions. This completes the proof. □

Notice that in the above lemma, we didn't use the property that H is chordal. In fact, *any* square root would tell us how to do set splitting. In particular, *any chordal* root would tell us how to do set splitting. This completes the proof of NP-completeness of SQUARE OF CHORDAL GRAPH.

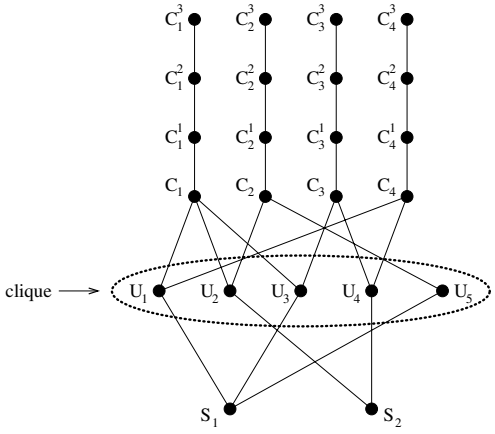


FIG. 3.3. *An example of H.*

THEOREM 3.5. SQUARE OF CHORDAL GRAPH *is NP-complete.*

Since *any* square root would tell us how to do set splitting, we have the following results.

THEOREM 3.6. *For any class X of graphs which contains the class of chordal graphs,* SQUARE OF X GRAPH *is NP-complete.*

COROLLARY 3.7. *Given G, determine if there exists a weakly chordal graph H such that $H^2 = G$ is NP-complete.*

COROLLARY 3.8. *Given G, determine if there exists a perfect graph H such that $H^2 = G$ is NP-complete.*

3.2. Square roots of chordal graphs. In this subsection, we will show that given a chordal graph G, it is NP-complete to determine if there exists H such that $H^2 = G$. Notice that the proof is almost identical to that of the previous subsection; therefore, we will omit unnecessary details.

PROBLEM. SQUARE ROOT OF CHORDAL GRAPH
INSTANCE. A *chordal graph* $G = (V, E)$.
QUESTION. Does there exist a graph H such that $H^2 = G$?

The rest of this section shows that SQUARE ROOT OF CHORDAL GRAPH is NP-hard by reducing SET SPLITTING to it. It is clear that SQUARE ROOT OF CHORDAL GRAPH is in NP, since guessing the square root H and verifying that $H^2 = G$ can be easily done in polynomial time. Thus we will conclude that SQUARE ROOT OF CHORDAL GRAPH is NP-complete.

3.2.1. The reduction. Given an instance of SET SPLITTING, we construct an instance of SQUARE ROOT OF CHORDAL GRAPH. Let c_j be the set of elements in subset j, let $C = \{c_1, \ldots, c_m\}$, and let $S = \{u_1, \ldots, u_n\}$ be the ground set. The graph G is constructed as follows (note that we will be using Lemma 3.1):

Vertices of G

- *Element vertices:* U_i: $1 \le i \le n$ for each element u_i.
- *Subset vertices:* C_j for each subset $c_j \in C$ and tail vertices C_j^1, C_j^2, C_j^3 for each c_j.
- *Partition vertices:* S_1 and S_2.

Edges of G

- *Edges of tail vertices of subset vertices:* for all $c_j \in C$,

$C_j^3 \leftrightarrow C_j^2$, $C_j^3 \leftrightarrow C_j^1$,
$C_j^2 \leftrightarrow C_j^1$, $C_j^2 \leftrightarrow C_j$,
$C_j^1 \leftrightarrow C_i$ for all i and $C_j^1 \leftrightarrow U_i$ for all $u_i \in c_j$.

- *Edges of subset vertices:* for all $c_j \in C$,
  $C_j \leftrightarrow S_1$, $C_j \leftrightarrow S_2$, $C_j \leftrightarrow U_i$ for all i, and $C_j \leftrightarrow C_k$ for all k.
- *Edges of element vertices:* for all $u_i \in U$,
  $U_i \leftrightarrow U_j$ for $i \neq j$, $U_i \leftrightarrow S_1$, and $U_i \leftrightarrow S_2$.

LEMMA 3.9. *G is a chordal graph.*

Proof. We do this by showing a simplicial elimination ordering of G. For all C_j^3, they are simplicial and we eliminate them first. Then for any C_j^2, it is adjacent only to C_j^1 and C_j, which are adjacent. Thus C_j^2 is simplicial for all j and we eliminate them. The union of the set of subset vertices and the set of element vertices induces a complete graph. For all C_j^1 and S_1 and S_2, their neighbor sets are a subset of that union. So all C_j^1 and S_1 and S_2 are simplicial and we can eliminate them. Finally, a complete graph is left, and this completes the proof that G is chordal. □

LEMMA 3.10. *If there is a partition of S into two subsets S_1 and S_2 such that no subset in C is entirely contained in either S_1 or S_2, then there exists a graph H such that $H^2 = G$.*

Proof. Edges of H.

- *Edges of subset vertices and its tail vertices:*
  $C_j^3 \leftrightarrow C_j^2$, $C_j^2 \leftrightarrow C_j^1$, $C_j^1 \leftrightarrow C_j$, $C_j \leftrightarrow U_i$ if and only if $u_i \in c_j$ and $C_j \leftrightarrow C_k$ for all k.
- *Edges of partition vertices:*
  $S_k \leftrightarrow U_i$ if and only if $u_i \in S_k$.
- *Edges of subset vertices:*
  $U_i \leftrightarrow U_j$ for $i \neq j$.

It is a routine matter to check that $H^2 = G$. (See [23] for a full proof.) □

By observing that $\{C_j^3, C_j^2, C_j^1, C_j\}$ satisfies the properties of Lemma 3.1 and using the same argument as in the previous section, we can prove the following result.

LEMMA 3.11. *If H is a square root of G, then there is a partition of S into two subsets S_1 and S_2 such that no subset in C is entirely contained in either S_1 or S_2.*

THEOREM 3.12. SQUARE ROOT OF CHORDAL GRAPH *is NP-complete.*

It should be pointed out that H is actually a Berge graph (i.e., there is no odd hole and no odd antihole in H); proofs are omitted. By the recent strong perfect graph theorem [4], it is implied that finding a perfect graph square root of a chordal graph is also NP-complete.

THEOREM 3.13. *It is NP-complete to determine if a chordal graph is the square of a perfect graph.*

3.3. Squares of split graphs. In this subsection, we will show that to determine if G is the square of a split graph is NP-complete.

Recall that an undirected graph $G = (V, E)$ is defined to be *split* if there is a partition $V = S + C$ of its vertex set into a stable set S and a complete set C. There is no restriction on edges between vertices of S and vertices of C.

PROBLEM. SQUARE OF SPLIT GRAPH
INSTANCE. Graph $G = (V, E)$.
QUESTION. Does there exist a *split graph* H such that $G = H^2$?

The motivation for studying this problem is the similarity of the structure of split graphs and the structure of bipartite graphs. While the vertex set of a bipartite graph is partitioned into two independent set, the vertex set of a split graph is partitioned

into a clique and an independent set. In [22], we prove that squares of bipartite graphs can be recognized in polynomial time. Note that since a split graph is of diameter at most 3, the tail structure cannot be applied in the reduction. In fact, we use a totally different reduction for this problem.

Given an instance of Intersection Graph Basis, we transform it into an instance of Square of Split Graph. Without loss of generality, we will assume that the graph in the instance of Intersection Graph Basis has no universal vertex and no isolated vertex. Recall that in the previous reduction, any square root will tell us the corresponding set splitting. In this reduction, we use the property that the square root is a split graph to find the corresponding intersection graph basis.

The transformation goes as follows. Given $G = (V, E)$, we construct a graph $G' = (V', E')$ by adding a set U of k universal vertices to G. Since G has no universal vertex, there are exactly k universal vertices in G'. We will show that G is an intersection graph of a family of sets whose union has cardinality at most k if and only if G' has a split root.

For example, given G as in Figure 3.4 and $k = 4$, Figure 3.4 shows the whole transformation process. In the figure, G is in the top left corner. G is the intersection graph for a family of sets whose union has cardinality 4. The 4-element set B is shown with $B[a] = \{1\}$, $B[b] = \{1, 3\}$, $B[c] = \{1, 2\}$, $B[d] = \{3, 4\}$, $B[e] = \{2, 4\}$, and $B[f] = \{4\}$. It is easy to verify that G is the intersection graph of B. The bottom left corner shows G', which comes from G by adding four universal vertices. Every vertex in G is adjacent to every vertex in U in G'. This is represented by the dotted lines between G and U in G'. Also, U itself is a clique in G'. The bottom right corner shows H', which is a split square root of G'. The reader may verify that H' is a split square root of G'. Finally, we observe that there is a one-to-one correspondence between the solution to the Intersection Graph Basis problem and the solution to the Square of Split Graph problem.

Theorem 3.14. Square of Split Graph *is NP-complete.*

Proof. We now argue that G is the intersection graph for a family of sets whose union has cardinality k or less if and only if there exists a split graph H such that $G' = H^2$.

First we prove the forward direction. If G is the intersection graph for a family of sets whose union has cardinality k or less, then we construct $H = (S+C, E)$ as follows. Each vertex c in C corresponds to an element in B. If $|B| < k$, we add some extra vertices to make $|C| = k$. Each vertex s in S corresponds to a vertex in G and s is adjacent to the vertices corresponding to $B[s]$. Now we check that $H^2 = G'$. Each vertex in the complete set is universal in H^2 and thus corresponds to a vertex in U. Two vertices u and v in the stable set have an edge in H^2 if and only if $B[u]$ and $B[v]$ are not disjoint. Thus $H^2[S]$ is precisely the G subgraph of G'. Hence $H^2 \cong G'$ as required.

Now we prove the reverse direction. If $H' = (S' + C', E')$ is a split square root of G', we construct the intersection graph basis as follows. Notice that a vertex in C' in H' is a universal vertex in H'^2. Since there are k vertices in G' that are universal vertices, there are at most k vertices in C' in H'. Furthermore, all the vertices in $G \subset V(G')$ must be in S' and so there is no edge between any two vertices in S'. Two vertices u and v in the stable set have an edge in H'^2 if and only if $N_{H'}(u)$ and $N_{H'}(v)$ are not disjoint. Now we see that G has an intersection basis B such that $|B| \leq k$ by setting $B = C'$. For $u \in G$, $B[u] = N_{H'}(u)$, thereby showing that for any split root of G', we can construct an intersection graph basis of cardinality at most k. □

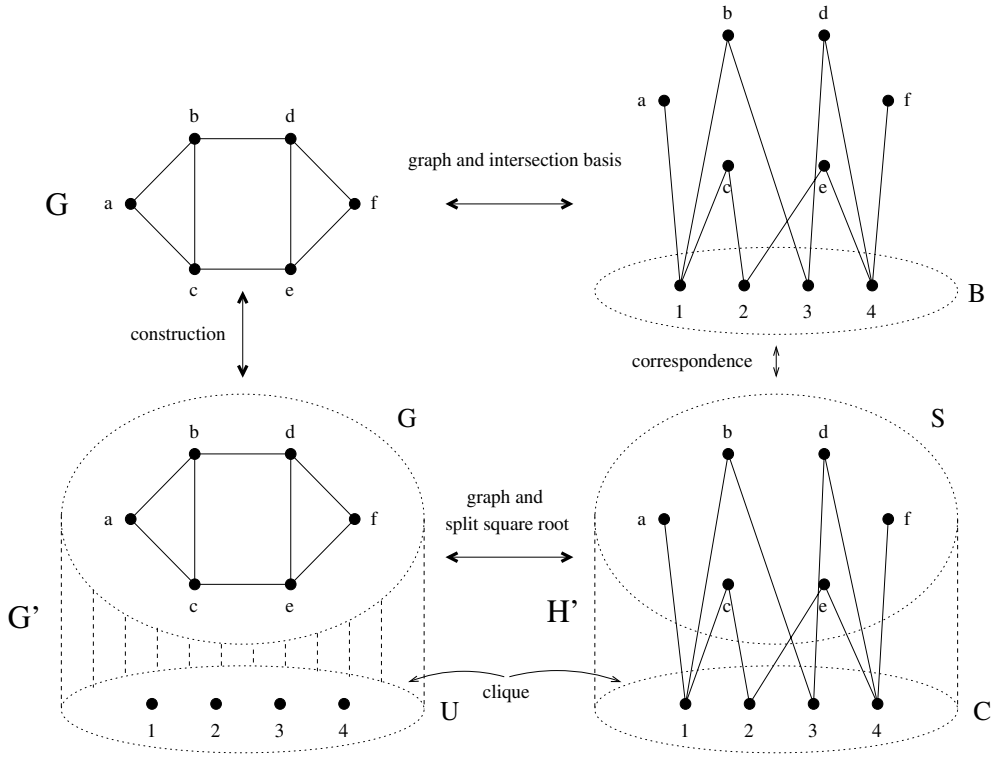


FIG. 3.4. *The whole transformation process of* SQUARE OF SPLIT GRAPH.

It is perhaps important to mention that, unlike the previous section, this result does not imply that finding square roots of a more general class (e.g., chordal graphs) is NP-complete. It is because we use the property that the square root is a split graph to force all the vertices that are not universal in the square to form an independent set in the square root.

4. Concluding remarks. For kTH POWER OF PROPER INTERVAL GRAPH, the complexity of our algorithm is exponential in k. It is open whether there is a polynomial time algorithm for kTH POWER OF PROPER INTERVAL GRAPH when k is part of the input. Also, it is open whether there is a polynomial time algorithm to recognize squares of interval graphs or, more generally, kth powers of interval graphs.

Acknowledgment. We thank the anonymous referees for their helpful comments.

REFERENCES

[1] R. BALAKRISHNAN AND P. PAULRAJA, *Powers of chordal graphs*, J. Austral. Math. Soc. Ser. A, 35 (1983), pp. 211–217.

[2] A. BRANDSTÄDT, F. DRAGAN, V. CHEPOI, AND V. VOLOSHIN, *Dually chordal graphs*, SIAM J. Discrete Math., 11 (1998), pp. 437–455.

[3] J. M. CHANG, C. W. HO, AND M. T. KO, *Powers of asteroidal triple-free graphs with applications*, Ars Combin., 67 (2003), pp. 161–173.

[4] M. CHUDNOVSKY, N. ROBERTSON, P. SEYMOUR, AND R. THOMAS, *The Strong Perfect Graph Theorem*, preprint. Available at http://www.math.gatech.edu/~thomas/spgc.html.

[5] D. G. Corneil, *A simple* 3*-sweep LBFS algorithm for the recognition of unit interval graphs*, Discrete Appl. Math., 138 (2004), pp. 371–379.
[6] P. Damaschke, *Distances in cocomparability graphs and their powers*, Discrete Appl. Math., 35 (1992), pp. 67–72.
[7] X. Deng, P. Hell, and J. Huang, *Linear-time representation algorithms for proper circular-arc graphs and proper interval graphs*, SIAM J. Comput., 25 (1996), pp. 390–403.
[8] P. Duchet, *Classical perfect graphs*, Ann. Discrete Math., 21 (1984), pp. 67–96.
[9] F. Escalante, L. Montejano, and T. Rojano, *Characterization of* n*-path graphs and of graphs having* n*th root*, J. Combin. Theory Ser. B, 16 (1974), pp. 282–289.
[10] H. Fleischner, *The square of every two-connected graph is Hamiltonian*, J. Combin. Theory Ser. B, 16 (1974), pp. 29–34.
[11] C. Flotow, *Graphs whose powers are chordal and graphs whose powers are interval graphs*, J. Graph Theory, 24 (1997), pp. 323–330.
[12] M. R. Garey and D. S. Johnson, *Computers and Intractability—A Guide to the Theory of NP-Completeness*, Freeman, Oxford, UK, 1979.
[13] D. P. Geller, *The square root of a digraph*, J. Combin. Theory Ser. B, 5 (1968), pp. 320–321.
[14] P. C. Gilmore and A. J. Hoffman, *A characterization of comparability graphs and of interval graphs*, Canad. J. Math., 16 (1964), pp. 539–548.
[15] F. Harary, R. M. Karp, and W. T. Tutte, *A criterion for planarity of the square of a graph*, J. Combin. Theory, 2 (1967), pp. 395–405.
[16] F. Harary and T. A. McKee, *The square of a chordal graph*, Discrete Math., 128 (1994), pp. 165–172.
[17] P. Hell, R. Shamir, and R. Sharan, *A fully dynamic algorithm for recognizing and representing proper interval graphs*, SIAM J. Comput., 31 (2001), pp. 289–305.
[18] P. Kearney and D. Corneil, *Tree powers*, J. Algorithms, 29 (1998), pp. 111–131.
[19] L. T. Kou, L. J. Stockmeyer, and C. K. Wong, *Covering edges by cliques with regard to keyword conflicts and intersection graphs*, Comm. ACM, 21 (1978), pp. 135–138.
[20] R. Laskar and D. Shier, *On powers and centers of chordal graphs*, Discrete Appl. Math., 6 (1983), pp. 139–147.
[21] H. T. Lau, *Finding a Hamiltonian Cycle in the Square of a Block*, Ph.D. thesis, School of Computer Science, McGill University, Montreal, QC, Canada, 1980.
[22] L. C. Lau, *Bipartite roots of graphs*, in Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2004, pp. 945–954.
[23] L. C. Lau, *Roots of Graphs*, Master's thesis, University of Toronto, Toronto, ON, Canada, 2003.
[24] G.-H. Lin, P. E. Kearney, and T. Jiang, *Phylogenetic* k*-root and Steiner* k*-root*, in Proceedings of the 11th Annual International Symposium on Algorithms and Computation, Lecture Notes in Comput. Sci. 1969, Springer-Verlag, Berlin, 2000, pp. 539–551.
[25] Y.-L. Lin and S. S. Skiena, *Algorithms for square roots of graphs*, SIAM J. Discrete Math., 8 (1995), pp. 99–118.
[26] N. Linial, *Locality in distributed graph algorithms*, SIAM J. Comput., 21 (1992), pp. 193–201.
[27] A. Lubiw, γ*-free Matrices*, Master's thesis, University of Waterloo, Waterloo, ON, Canada, 1982.
[28] M. Molloy and M. R. Salavatipour, *Frequency channel assignment on planar networks*, in Proceedings of the 10th Annual European Symposium on Algorithms, Lecture Notes in Comput. Sci. 2461, Springer-Verlag, Berlin, 2002, pp. 736–747.
[29] R. Motwani and M. Sudan, *Computing roots of graphs is hard*, Discrete Appl. Math., 54 (1994), pp. 81–88.
[30] A. Mukhopadhyay, *The square root of a graph*, J. Combin. Theory, 2 (1967), pp. 290–295.
[31] N. Nishimura, P. Ragde, and D. Thilikos, *On graph powers for leaf-labeled trees*, J. Algorithms, 42 (2002), pp. 69–108.
[32] S. Ramachandran, *Planar graphs with square or cube root are four colorable*, J. Combin. Theory Ser. B, 24 (1978), pp. 362–364.
[33] A. Raychaudhuri, *On powers of interval and unit interval graphs*, Congr. Numer., 59 (1987), pp. 235–242.
[34] A. Raychaudhuri, *On powers of strongly chordal and circular arc graphs*, Ars Combin., 34 (1992), pp. 147–160.
[35] I. C. Ross and F. Harary, *The square of a tree*, Bell System Tech. J., 39 (1960), pp. 641–647.
[36] M. Sekanina, *On an ordering of the vertices of a graph*, Časopis Pěst. Mat., 88 (1963), pp. 265–282.
[37] J. P. Spinrad, *Doubly lexical ordering of dense* 0-1 *matrices*, Inform. Process. Lett., 45 (1993), pp. 229–235.
[38] D. West, *Introduction to Graph Theory*, 2nd ed., Prentice–Hall, Englewood Cliffs, NJ, 2001.

SIAM J. DISCRETE MATH.
Vol. 18, No. 1, pp. 103–109

SMALL EXAMPLES OF NONCONSTRUCTIBLE SIMPLICIAL BALLS AND SPHERES*

FRANK H. LUTZ†

Abstract. We construct nonconstructible simplicial d-spheres with $d+10$ vertices and nonconstructible, nonrealizable simplicial d-balls with $d+9$ vertices for $d \geq 3$.

Key words. simplicial balls and spheres, constructibility, shellability, vertex-decomposability, knots in 3-balls and 3-spheres

AMS subject classifications. 57Q15, 52B05, 52B22, 57M25

DOI. 10.1137/S0895480103430521

1. Introduction.

The concepts of *vertex-decomposability*, *shellability*, and *constructibility* describe three particular ways to assemble a simplicial complex from the collection of its facets (cf. Björner [5]). The following implications are strict for (pure) simplicial complexes:

$$\text{vertex decomposable} \implies \text{shellable} \implies \text{constructible.}$$

Shellability has its origin in Schläfli's computation from 1852 [33] of the Euler characteristics of convex polytopes, where he based his calculation on the assumption that the boundary complexes of polytopes are shellable. However, this property of polytopes was justified only much later in 1970 by Bruggesser and Mani [9] and then played a crucial role in McMullen's proof of the upper bound theorem in the same year [28]. Besides in polyhedral theory, shellability has found fruitful applications in topology, combinatorics, and computational geometry; see the surveys [4], [5], [12], [35, Ch. 8], [36], and the references contained therein.

The notion of constructibility was coined by Hochster in 1972 [19] but implicitly was used long before in combinatorial topology. In particular, it follows from Newman's and Alexander's fundamental works on the foundations of combinatorial and piecewise linear (PL) topology from 1926 [29] and 1930 [1] (cf. also Björner [5]) that a constructible d-dimensional simplicial complex in which every $(d-1)$-face is contained in exactly two or at most two d-dimensional facets is a PL d-sphere or a PL d-ball, respectively. For recent surveys on constructibility see [17] and [18].

The strongest concept, vertex-decomposability, was introduced by Provan and Billera in their proof from 1980 [31] that vertex decomposable simplicial complexes satisfy the simplicial form of the famous Hirsch conjecture (cf. [13, p. 168]) of linear programming.

Although boundary spheres of simplicial polytopes are shellable, Lockeberg [24] constructed a simplicial 4-polytope with 12 vertices which is not vertex-decomposable; and there even are not vertex-decomposable simplicial 4-polytopes with 10 vertices [21] and not vertex-decomposable, nonpolytopal simplicial 3-spheres with 9 vertices [8]. For two-dimensional balls and spheres it was proved by Bing [4] that they are shellable

*Received by the editors June 26, 2003; accepted for publication (in revised form) February 10, 2004; published electronically July 20, 2004.
http://www.siam.org/journals/sidma/18-1/43052.html
†Technische Universität Berlin, Fakultät II - Mathematik und Naturwissenschaften, Institut für Mathematik, Sekr. MA 6-2, Straße des 17. Juni 136, 10623 Berlin, Germany (lutz@math.tu-berlin.de).

and by Provan and Billera [31] that they are vertex-decomposable. Klee and Kleinschmidt [21] also showed that all simplicial d-balls and all simplicial d-spheres with up to $d+3$, respectively, $d+4$ vertices, are vertex-decomposable. However, for $d \geq 3$ there are not vertex-decomposable simplicial d-balls with $d+4$ vertices and 10 facets as well as not vertex-decomposable simplicial d-spheres with $d+6$ vertices; see [8] and [27].

The first known example of a nonshellable cellular 3-ball is due to Furch and appeared in 1924 [15]. A nonshellable simplicial 3-ball with 30 vertices and 72 facets was provided by Newman in 1926 [30]. Newman's ball is *strongly nonshellable*; i.e., it has no *free* facet that can be removed from the triangulation without losing ballness. Much smaller strongly nonshellable simplicial 3-balls were obtained by Grünbaum (cf. [12]) with 14 vertices and 29 facets and by Ziegler [36] with 10 vertices and 21 facets. Rudin's 3-ball [32] with 14 vertices and 41 tetrahedra gives a strongly nonshellable rectilinear triangulation of a tetrahedron with all the vertices on the boundary; the vertices even can be moved slightly to yield a straight triangulation of a convex 3-polytope with 14 vertices [11]. Ziegler's ball is realizable as a straight yet nonconvex ball in 3-space. Coordinates for a rectilinear realization of Grünbaum's ball can be found in [17]. Vertex-minimal nonshellable 3-balls with 9 vertices are enumerated in [8]; see [26] for a geometric realization of one of these balls with 18 facets.

The existence of nonconstructible 3-balls was shown by Lickorish [22] in 1971, but it remained unclear whether there are nonshellable 3-spheres. Nonshellable cell partitions of S^3 were first constructed by Vince [34] in 1985 and then by Armentrout [3]. In 1991, Lickorish [23] described nonshellable triangulated 3-spheres that contain a knotted triangle made of the sum of (at least) three trefoil knots.

In fact, it suffices to use one single trefoil knot.

THEOREM 1 (Hachimori and Ziegler [18]). *If a triangulated* 3*-ball or* 3*-sphere contains* any *knotted triangle, then it is nonconstructible (and thus nonshellable). Moreover, a* 3*-ball with a knotted spanning arc consisting of at most* 2 *edges is nonconstructible.*

A first explicit, but large, nonconstructible triangulated 3-sphere with f-vector $f = (381, 2309, 3856, 1928)$ based on Furch's 3-ball with a knotted spanning arc consisting of one edge was constructed by Hachimori [16]. Suspensions of such spheres produce nonconstructible simplicial PL d-spheres in dimensions $d \geq 3$. Examples of small non-PL (and hence nonconstructible) d-spheres of dimensions $d \geq 5$ with $d+13$ vertices can be found in [6]; see also [7]. Their construction makes use of the double suspension theorem of Edwards [14] (respectively, of its generalization by Cannon [10]) that double suspensions of nonspherical homology d-spheres give non-PL $(d+2)$-spheres.

2. The examples. In the following, we employ the theorem of Hachimori and Ziegler to construct simplicial PL d-spheres in dimensions $d \geq 3$ with only $d+10$ vertices that are nonconstructible. From the enumeration in [8] it follows that all 3-spheres with $n \leq 10$ vertices are shellable. Hence, the nonconstructible 3-sphere $S^3_{13,56}$ with 13 vertices that we are going to obtain is, if not vertex-minimal, then close to vertex-minimality.

THEOREM 2. *There is a nonconstructible* 3*-sphere* $S^3_{13,56}$ *with* 13 *vertices and* 56 *facets. Moreover, there are two strongly nonshellable, nonconstructible* 3*-balls* $B^3_{12,37,a}$ *and* $B^3_{12,37,b}$ *with* 12 *vertices and* 37 *facets that cannot be rectilinearly embedded into* $\mathbb{R}^3$.

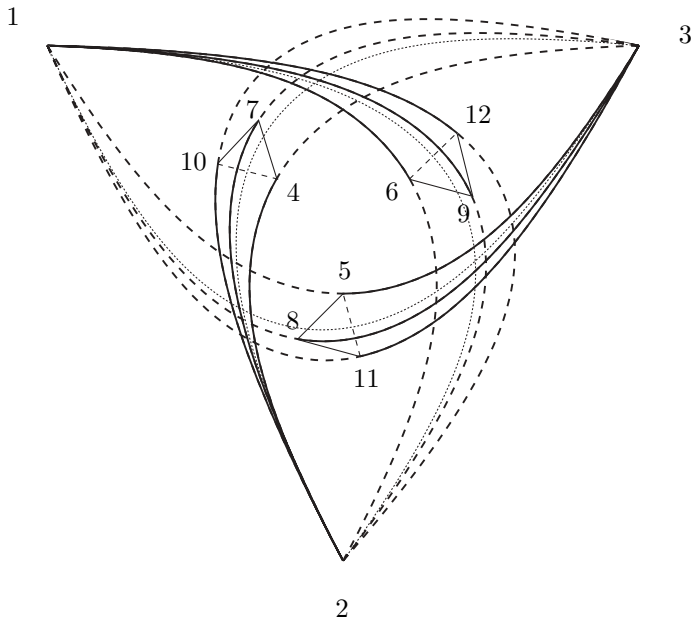


FIG. 1. *The trefoil knot with three protected edges.*

Proof. The examples are based on a trefoil knot consisting of three edges 1 2, 1 3, and 2 3 (the dotted lines in Figure 1) which we embed into $\mathbb{R}^3$. We shield off the edges by enclosing every edge with three tetrahedra, as listed in the first column of Table 1. We then close the holes of the knot by gluing in the following 16 triangles:

4 5 6	1 4 6	2 4 5	3 5 6
	1 4 7	2 5 8	3 6 9
	1 7 10	2 8 11	3 9 12
	1 5 10	2 6 11	3 4 12
	4 5 10	5 6 11	4 6 12.

TABLE 1
The ball $B^3_{16,46}$.

1 2 6 9	1 4 6 12	1 4 7 13	2 5 8 14	3 6 9 15	4 5 6 16
1 2 6 12	2 4 5 10	2 4 7 13	3 5 8 14	1 6 9 15	1 4 6 16
1 2 9 12	3 5 6 11	1 7 10 13	2 8 11 14	3 9 12 15	1 4 13 16
		2 7 10 13	3 8 11 14	1 9 12 15	2 4 13 16
1 3 5 8		1 5 10 13	2 6 11 14	3 4 12 15	2 4 5 16
1 3 5 11		2 5 10 13	3 6 11 14	1 4 12 15	2 5 14 16
1 3 8 11		1 5 8 13	2 6 9 14	3 4 7 15	3 5 14 16
		2 5 8 13	3 6 9 14	1 4 7 15	3 5 6 16
2 3 4 7					3 6 15 16
2 3 4 10					1 6 15 16
2 3 7 10					

The resulting simplicial complex C is contractible. By adding the 37 tetrahedra in the columns 2–6 of Table 1 we thicken C to a ball $B^3_{16,46}$ with 16 vertices, 46 facets, and f-vector $f = (16, 75, 106, 46)$. Since $B^3_{16,46}$ contains a trefoil knot composed of three edges, it follows from Theorem 1 of Hachimori and Ziegler that $B^3_{16,46}$ is not

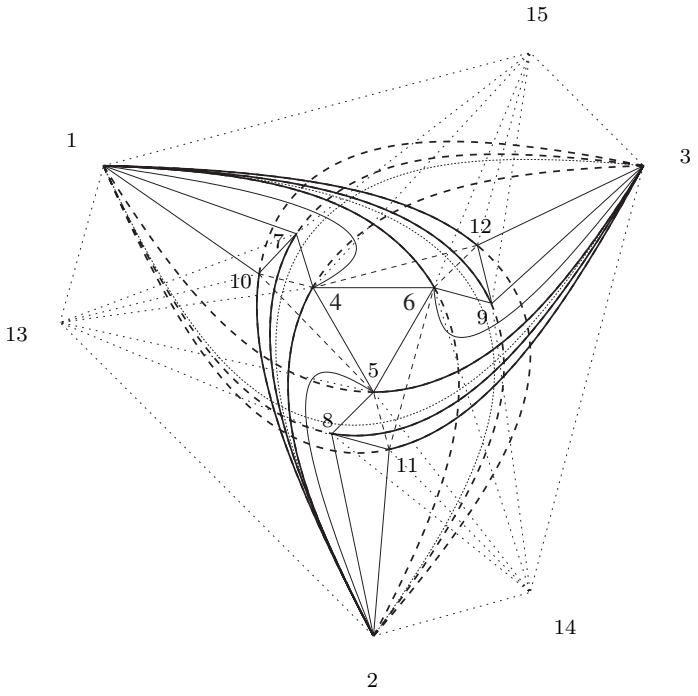


FIG. 2. *The contractible complex C with three cones.*

constructible and thus not shellable. In fact, $B^3_{16,46}$ is strongly nonshellable, as the removal of any of its facets destroys the ballness. Moreover, the presence of the 3-edge knot prevents $B^3_{16,46}$ from having a straight embedding into $\mathbb{R}^3$.

In Figure 2 we display the complex C. We also indicate the cones with respect to the vertices 13, 14, and 15 over eight of the triangles of C each, as listed in columns 3–5 of Table 1. The cone with respect to vertex 16 is then placed "above" the drawing.

The boundary of $B^3_{16,46}$ consists of 28 triangles:

1 13 16	4 5 6	4 5 10	5 6 11	4 6 12
2 13 16		1 5 10	2 6 11	3 4 12
2 14 16		1 5 11	2 6 12	3 4 10
3 14 16		1 8 11	2 9 12	3 7 10
3 15 16		2 8 11	3 9 12	1 7 10
1 15 16		2 8 13	3 9 14	1 7 15
		1 8 13	2 9 14	3 7 15.

If we add to $B^3_{16,46}$ the cone over these 28 triangles with respect to a new vertex 17, then we get a 3-sphere $S^3_{17,74}$ with $f = (17, 91, 148, 74)$. This 3-sphere still contains the complex C and with it the trefoil knot composed of the three edges 1 2, 1 3, and 2 3. Hence, $S^3_{17,74}$ is a nonconstructible, nonshellable sphere. By construction, $B^3_{16,46}$ and $S^3_{17,74}$ have a $\mathbb{Z}_3$-symmetry.

Since all 3-spheres with $n \leq 10$ vertices are shellable [8], 17 vertices is close to the minimal number of vertices that are needed for a nonshellable 3-sphere. In order to still improve on the number of vertices, we applied the bistellar flip program BISTELLAR [25] to $S^3_{17,74}$, under the additional restriction that the edges of the knot should not be touched. (The objective of BISTELLAR is to decrease the size

of a triangulation of a manifold by performing bistellar flips that locally modify the triangulation without changing the topological type; see [6] for an explicit description.) As result, we obtained a simplicial 3-sphere $S^3_{13,56}$ with $f = (13, 69, 112, 56)$ that has no nontrivial symmetry. The removal of the star of vertex 13

1 7 9 13	2 5 7 13	3 5 8 13	5 7 9 13
1 7 11 13	2 5 8 13	3 5 9 13	6 10 11 13
1 9 10 13	2 6 11 13	3 6 10 13	
1 10 11 13	2 6 12 13	3 6 12 13	
	2 7 11 13	3 8 12 13	
	2 8 12 13	3 9 10 13	

from this complex yields a 12-vertex 3-ball $B^3_{12,38}$ with 38 facets, as listed in Table 2.

TABLE 2
The ball $B^3_{12,38}$.

1 2 6 9	1 5 8 10	*2 4 5 7*	3 4 6 7	4 5 6 7
1 2 6 12	1 5 10 11	2 4 5 10	*3 4 6 10*	4 5 6 10
1 2 9 12	1 6 7 9	2 5 8 10	3 5 9 11	5 6 7 9
	1 6 7 12	2 6 9 11	3 6 7 12	5 6 9 11
1 3 5 8	1 7 8 10	2 7 8 10	3 7 10 12	5 6 10 11
1 3 5 11	1 7 8 11	2 7 8 11	3 8 9 11	
1 3 8 11	1 7 10 12	2 8 9 11	3 8 9 12	
	1 9 10 12	2 8 9 12	3 9 10 12	
2 3 4 7				
2 3 4 10				
2 3 7 10				

This ball has two free facets, 2 4 5 7 and 3 4 6 10, so is not strongly nonshellable. However, when we remove either of the two tetrahedra, we get strongly nonshellable, nonconstructible 3-balls $B^3_{12,37,a}$ and $B^3_{12,37,b}$ with 37 facets and $f = (12, 58, 84, 37)$, respectively. These two balls are not isomorphic, although they have isomorphic boundaries. (The permutation $(2,3)(5,6)(7,10)(8,12)(9,11)$ maps the boundary spheres onto each other, but, if we add to each ball the cone over its boundary with respect to a new vertex, then the resulting 3-spheres have different Altshuler–Steinberg determinants [2].) Both balls (and also the sphere $S^3_{13,56}$) still contain the original 3-edge trefoil knot for which, this time, the triangles

4 5 6	4 6 7	2 4 5	5 6 9
	1 6 7	2 5 8	3 5 9
	1 7 10	2 8 11	3 9 12
	1 5 10	2 6 11	3 6 12
	4 5 10	5 6 11	3 4 6

are glued in to close the holes of the knot; see Figure 3. □

COROLLARY 3. *For* $d \geq 3$ *there are nonconstructible* d*-spheres with* $d+10$ *vertices. Also there are nonconstructible* d*-balls,* $d \geq 3$*, with* $d + 9$ *vertices and* 37 *facets that do not have a straight embedding into* $\mathbb{R}^d$*.*

Proof. The cone over a nonconstructible, nonrealizable d-ball is a nonconstructible, nonrealizable $(d + 1)$-ball with the same number of facets. Similarly, the one-point suspension of a nonconstructible d-sphere is a nonconstructible $(d + 1)$-sphere; see [20]. □

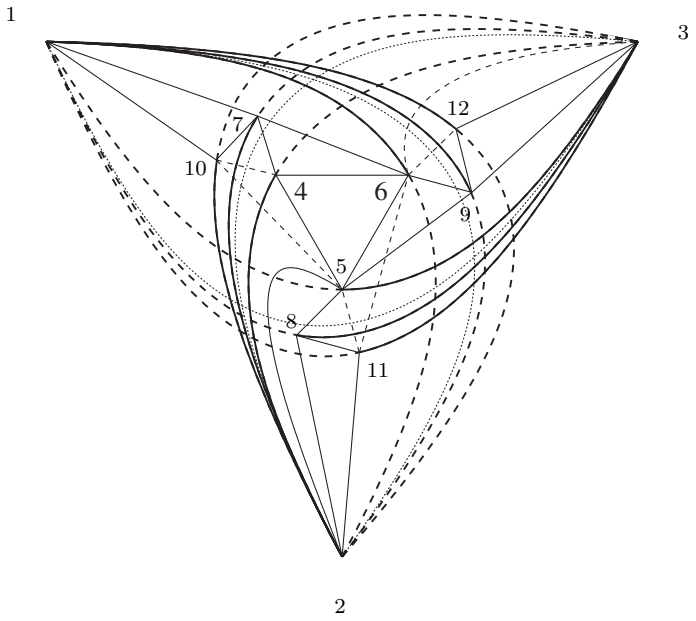


FIG. 3. *The* 3*-edge trefoil knot lying in the nonshellable sphere* $S^3_{13,56}$.

Acknowledgment. The author is grateful to Günter M. Ziegler for helpful remarks.

REFERENCES

[1] J. W. ALEXANDER, *The combinatorial theory of complexes*, Ann. of Math. (2), 31 (1930), pp. 292–320.

[2] A. ALTSHULER AND L. STEINBERG, *Neighborly* 4*-polytopes with* 9 *vertices*, J. Combin. Theory Ser. A, 15 (1973), pp. 270–287.

[3] S. ARMENTROUT, *Knots and shellable cell partitionings of* S^3, Illinois J. Math., 38 (1994), pp. 347–365.

[4] R. H. BING, *Some aspects of the topology of* 3*-manifolds related to the Poincaré conjecture*, in Lectures on Modern Mathematics, Vol. II, T. L. Saaty, ed., John Wiley & Sons, New York, 1964, pp. 93–128.

[5] A. BJÖRNER, *Topological methods*, in Handbook of Combinatorics, R. Graham, M. Grötschel, and L. Lovász, eds., Elsevier, Amsterdam, 1995, pp. 1819–1872.

[6] A. BJÖRNER AND F. H. LUTZ, *Simplicial manifolds, bistellar flips and a* 16*-vertex triangulation of the Poincaré homology* 3*-sphere*, Experiment. Math., 9 (2000), pp. 275–289.

[7] A. BJÖRNER AND F. H. LUTZ, *A* 16*-vertex triangulation of the Poincaré homology* 3*-sphere and non-PL spheres with few vertices*, Electronic Geometry Model No. 2003.04.001, 2003, http://www.eg-models.de/2003.04.001.

[8] J. BOKOWSKI, D. BREMNER, F. H. LUTZ, AND A. MARTIN, *Combinatorial* 3*-Manifolds with* 10 *Vertices*, in preparation.

[9] H. BRUGGESSER AND P. MANI, *Shellable decompositions of cells and spheres*, Math. Scand., 29 (1971), pp. 197–205.

[10] J. W. CANNON, *Shrinking cell-like decompositions of manifolds. Codimension three*, Ann. of Math. (2), 110 (1979), pp. 83–112.

[11] R. CONNELLY AND D. W. HENDERSON, *A convex* 3*-complex not simplicially isomorphic to a strictly convex complex*, Math. Proc. Cambridge Philos. Soc., 88 (1980), pp. 299–306.

[12] G. DANARAJ AND V. KLEE, *Which spheres are shellable?*, in Algorithmic Aspects of Combinatorics, Ann. Discrete Math. 2, B. Alspach, P. Hell, and D. J. Miller, eds., North–Holland, Amsterdam, 1978, pp. 33–52.

[13] G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ, 1963.
[14] R. D. Edwards, *The double suspension of a certain homology* 3*-sphere is* S^5, Notices Amer. Math. Soc., 22 (1975), p. A-334.
[15] R. Furch, *Zur Grundlegung der kombinatorischen Topologie*, Abh. Math. Sem. Univ. Hamburg, 3 (1924), pp. 69–88.
[16] M. Hachimori, *A* 3*-Sphere with a Knotted Triangle*, http://infoshako.sk.tsukuba.ac.jp/~hachi/math/library/nc_sphere_eng.html.
[17] M. Hachimori, *Nonconstructible simplicial balls and a way of testing constructibility*, Discrete Comput. Geom., 22 (1999), pp. 223–230.
[18] M. Hachimori and G. M. Ziegler, *Decompositions of simplicial balls and spheres with knots consisting of few edges*, Math. Z., 235 (2000), pp. 159–171.
[19] M. Hochster, *Rings of invariants of tori, Cohen-Macaulay rings generated by monomials, and polytopes*, Ann. of Math. (2), 96 (1972), pp. 318–337.
[20] M. Joswig and F. H. Lutz, *One-Point Suspensions and Wreath Products of Polytopes and Spheres*, http://www.arxiv.math.co/0403494, 2004.
[21] V. Klee and P. Kleinschmidt, *The d-step conjecture and its relatives*, Math. Oper. Res., 12 (1987), pp. 718–755.
[22] W. B. R. Lickorish, *An unsplittable triangulation*, Michigan Math. J., 18 (1971), pp. 203–204.
[23] W. B. R. Lickorish, *Unshellable triangulations of spheres*, European J. Combin., 12 (1991), pp. 527–530.
[24] E. R. Lockeberg, *Refinements in Boundary Complexes of Polytopes*, Dissertation, University College London, London, 1977.
[25] F. H. Lutz, `BISTELLAR`, 1997–2003, http://www.math.tu-berlin.de/diskregeom/stellar/BISTELLAR.
[26] F. H. Lutz, *A vertex-minimal non-shellable simplicial* 3*-ball with* 9 *vertices and* 18 *facets*, Electronic Geometry Model No. 2003.05.004, 2004, http://www.eg-models.de/2003.05.004.
[27] F. H. Lutz, *Vertex-minimal not vertex-decomposable balls*, Electronic Geometry Model No. 2003.06.001, 2004, http://www.eg-models.de/2003.06.001.
[28] P. McMullen, *The maximum numbers of faces of a convex polytope*, Mathematika, 17 (1970), pp. 179–184.
[29] M. H. A. Newman, *On the foundations of combinatory analysis situs* I, II, Proc. Royal Acad. Amsterdam, 29 (1926), pp. 611–626, 627–641.
[30] M. H. A. Newman, *A property of* 2*-dimensional elements*, Proc. Royal Acad. Amsterdam, 29 (1926), pp. 1401–1405.
[31] J. S. Provan and L. J. Billera, *Decompositions of simplicial complexes related to diameters of convex polyhedra*, Math. Oper. Res., 5 (1980), pp. 576–594.
[32] M. E. Rudin, *An unshellable triangulation of a tetrahedron*, Bull. Amer. Math. Soc., 64 (1958), pp. 90–91.
[33] L. Schläfli, *Theorie der vielfachen Kontinuität*, Neue Denkschriften der allgemeinen schweizerischen Gesellschaft für die Gesammten Naturwissenschaften 38, Zürcher und Furrer, Zürich, 1901 (written 1850–1852). Reprinted in Ludwig Schläfli 1814–1895, Gesammelte mathematische Abhandlungen, Band I, Birkhäuser, Basel, 1950, pp. 167–387.
[34] A. Vince, *A non-shellable* 3*-sphere*, European J. Combin., 6 (1985), pp. 91–100.
[35] G. M. Ziegler, *Lectures on Polytopes*, Grad. Texts in Math. 152, Springer-Verlag, New York, 1995; revised edition, 1998.
[36] G. M. Ziegler, *Shelling polyhedral* 3*-balls and* 4*-polytopes*, Discrete Comput. Geom., 19 (1998), pp. 159–174.

SIAM J. DISCRETE MATH.
Vol. 18, No. 1, pp. 110–113

A NOTE ON BANDITS WITH A TWIST*

AKSHAY-KUMAR KATTA† AND JAY SETHURAMAN†

Abstract. A variant of the multiarmed bandit problem was recently introduced by Dimitriu, Tetali, and Winkler. For this model (and a mild generalization) we propose faster algorithms to compute the Gittins index. The indexability of such models follows from earlier work of Nash on generalized bandits.

Key words. multiarmed bandit problem, generalized bandit problem, stochastic scheduling, priority rule, Gittins index, game

AMS subject classifications. 60J10, 66C99, 60G40, 90B35, 90C40

DOI. 10.1137/S0895480103433549

1. Introduction. The multiarmed bandit problem is a well-studied optimization problem concerned with dynamically allocating a single resource amongst several competing projects. In the basic version of this problem, there are N independent projects, each of which can be in one of many possible states. At each $t = 1, 2, \ldots$, we must operate exactly one of the projects; as a result, we earn a (possibly random) reward that may depend on the state of the operated project, which undergoes a Markovian state transition. The states of all the other projects remain frozen. Future earnings are discounted by a factor β, and our objective is to decide the order in which we must operate the various projects to maximize the expected total discounted reward earned. Gittins and Jones [7] showed that to each project i, we can attach an index that depends only on the state of project i and is independent of the states of all the other projects, and that operating a project with the largest index at any point in time is optimal. (Such problems are said to be *indexable*.) Since their original proof, many alternative and insightful proofs have appeared; see [14, 10, 12, 5, 9, 11, 1, 3]. In addition, several natural extensions and variations of the basic multiarmed bandit model have been considered; see [8, 14, 15, 13, 2]. Especially relevant to this work is the generalized bandit model of Nash [8], which considers a class of bandit problems with a more general reward structure. In such a model, the reward obtained from a transition in one project depends in a multiplicatively separable way on the states of all the other projects. Nash [8] proved that this more general class of bandit problems is indexable.

In this paper we consider a variant of the multiarmed bandit problem that was introduced in [4]. Here, as before, we are required to operate exactly one of the projects, except that we are forced to stop on reaching certain "target" states. The formulation in [4] is in terms of costs (instead of rewards) and is used to model situations in which there are multiple ways to accomplish a certain task, and the goal is to find the "best" way. Termination is assumed to be inevitable, and our objective is to operate the projects so as to minimize the expected total cost incurred until termination. By letting the multiplicative factors in the generalized bandit model be zero for the target states and one for the nontarget states, we see that the (discounted

*Received by the editors August 22, 2003; accepted for publication (in revised form) January 15, 2004; published electronically July 20, 2004. This research was supported by NSF grant DMI-0093981 and by an IBM partnership award.

http://www.siam.org/journals/sidma/18-1/43354.html

†IEOR Department, Columbia University, New York, NY 10027 (ark2001@columbia.edu, jay.sethuraman@columbia.edu).

version of the) model considered here can be viewed as a special case of the generalized bandit problem.

2. Model and related work. There are n bandit processes; the ith process is a Markov chain with a finite state space $\mathcal{S}_i$ and a *sink* $t_i \in \mathcal{S}_i$. For convenience, we assume that the state spaces of the different bandit processes are disjoint. Time is discrete and is indexed by t. If the ith bandit is at some state $x \in \mathcal{S}_i$ and is operated at time t, then the bandit moves to state $y \in \mathcal{S}_i$ with probability p_{xy}, and a (possibly random) cost C_{xy} is incurred. If $y = t_i$, we stop; otherwise we must choose a bandit to operate at time $t + 1$. Our objective is to operate the bandits over time so as to minimize the expected total cost incurred before termination, which we assume is inevitable (so the expected total cost is finite). For simplicity, we assume that the C_{xy} are deterministic, noting that much of what follows holds true for random C_{xy} by simply replacing the random variables by their expected values.

We shall call the special case in which $C_{xy} > 0$ for all nonsink states x the *positive-cost* model, distinguishing it from the *general* model in which no assumptions are made about C_{xy}. The indexability of the positive-cost model and the general-cost model can be inferred from the classical results of Gittins [6] and Nash [8], respectively, by letting $\beta \to 1$. In [4], the authors prove the indexability of the positive-cost model by adapting Weber's elegant intuitive proof to this setting; in addition, they provide two algorithms to compute the Gittins index, both with complexity $O(n^5)$, where n is the number of nonsink states. Our main observation is that standard techniques result in an $O(n^3)$ algorithm to compute the Gittins index for the general model (and hence for the positive-cost model as well); this matches the complexity of the most efficient algorithm to compute the Gittins index in the usual multiarmed bandit problem [11, 12].

3. Computing the Gittins index. Since the model considered here is indexable, we focus on a single bandit and show how the Gittins index can be computed for each of its states. Without loss of generality, we assume that the bandit has a single sink, which can be accessed from every other state. Let F_x denote the probability of going from state x to the sink in one step. Also, let $C_x \equiv \sum_y C_{xy} p_{xy}$ be the expected cost of operating the bandit when it is in state x. For convenience, we also assume that $F_x > 0$ for every nonsink state x. Later we show how this assumption can be relaxed.

An alternative characterization of the Gittins index is the key to computing it efficiently, so we discuss this briefly. Consider a "game" in which, at each step, one is faced with two choices: continuing to operate the bandit, which costs (on average) C_x if the bandit is in state x, or quitting by paying a fee of M dollars. It is well known that the Gittins index, ν_x, of a state x is the unique value of M at which one is indifferent between operating the bandit in state x and quitting.

Suppose state x has the smallest Gittins index, and suppose the bandit is currently in state x. Let the fee in the game described earlier be ν_x. By definition, it is optimal to operate the bandit once and quit by paying ν_x if the resulting state is not a sink; thus $\nu_x = C_x/F_x$. Unfortunately, we do not know the state with the smallest Gittins index, so we test all possibilities. From the alternative characterization of the Gittins index mentioned earlier, it is clear that x is a state with the smallest Gittins index if and only if

$$x = \arg\min_{y \in \mathcal{S}} \frac{C_y}{F_y}.$$

Having identified a state with the smallest Gittins index, we can now "reduce" the bandit by eliminating x in the following manner (see [11]). Consider any nonsink state $y \neq x$ with $p_{yx} > 0$. In computing the Gittins index of y, we may assume that whenever we make a transition to x, we continue to operate the bandit until we leave x to reach some state z (which may possibly be y itself or even the sink); this sequence of plays may be regarded as a single play with a "cost"

$$\hat{c}_{yz} \;=\; C_{yx} + C_{xx}\left\{\frac{1}{1-p_{xx}} - 1\right\} + C_{xz}$$

and a transition probability

$$\hat{p}_{yz} \;=\; p_{yx}\, p_{xz}/(1-p_{xx}).$$

We note that $\hat{c}_{yz}$ is the expected cost incurred during this composite play, which can be broken down into three components: the first transition from y to x, costing C_{yx}; the successive self-transitions at x, whose expected number is $1/(1-p_{xx}) - 1$, each costing C_{xx}; and the last transition from x to z, costing C_{xz}. The conditional probability of an (x, z) transition, given that a transition from x to another state occurs is $p_{xz}/(1-p_{xx})$, which justifies the expression for $\hat{p}_{yz}$. If the (y, z) arc does not already exist, we introduce one, and we let $C_{yz} = \hat{c}_{yz}$, $p_{yz} \;=\; \hat{p}_{yz}$; if the (y, z) arc already exists, the cost for a y to z transition is updated as

$$C_{yz} \leftarrow \frac{p_{yz}C_{yz} + \hat{p}_{yz}\hat{c}_{yz}}{p_{yz} + \hat{p}_{yz}},$$

and the transition probability from y to z now becomes

$$p_{yz} \leftarrow p_{yz} + \hat{p}_{yz}.$$

For a bandit with n states, a state with the smallest Gittins index can be determined in $O(n)$ time; the reduction algorithm needs to examine $O(n^2)$ pairs, each of which requires $O(1)$ time. Thus the complexity per iteration is $O(n^2)$ when there are n states. The (reduced) bandit now has one less state; we proceed as before by identifying a state with the minimum Gittins index, eliminating this state to further reduce the bandit, etc. After $(n-1)$ applications of the reduction algorithm we will have determined the Gittins index for all the nonsink states; thus the overall complexity of computing the Gittins index for an n-state bandit is easily seen to be $O(n^3)$.

We now show how the assumption $F_x > 0$ for all nonsink states x can be relaxed. Let x be a nonsink state with $F_x = 0$. If $C_x \leq 0$, then we will always operate the bandit in state x, so $\nu_x = -\infty$. (Such states must be reduced first.) If $C_x > 0$, it is clear that x cannot be a state with the minimum index; in fact, it is easy to see that some state adjacent to x must have a lower index (see [4]). In this case, the index of state x will be determined by the algorithm at a later point.

Finally, we note that the algorithm proposed here can be extended to more general versions of the problem, such as the semi-Markov version (time is not slotted) and the discounted version. We leave the obvious modifications to the reader.

Acknowledgments. A version of the problem described here was the subject of the first author's final project in a graduate course on dynamic programming taught by the second author. We thank John Tsitsiklis for sharing his thoughts on this problem and Kevin Glazebrook for telling us about the relevance of Nash's work on generalized bandits. In addition, we thank them both for their comments on an earlier version of this paper.

REFERENCES

[1] D. Bertsimas and J. Nino-Mora, *Conservation laws, extended polymatroids and multi-armed bandit problems: A polyhedral approach to indexable systems*, Math. Oper. Res., 21 (1996), pp. 257–306.

[2] J. H. Crosbie and K. Glazebrook, *Index policies and a novel performance space structure for a class of generalized branching bandit problems*, Math. Oper. Res., 25 (2000), pp. 281–297.

[3] M. Dacre, K. Glazebrook, and J. Nino-Mora, *The achievable region approach to the optimal control of stochastic systems*, J. Roy. Statist. Soc. Ser. B, 61 (1999), pp. 747–791.

[4] I. Dumitriu, P. Tetali, and P. Winkler, *On playing golf with two balls*, SIAM J. Discrete Math., 16 (2003), pp. 604–615.

[5] J. C. Gittins, *Multi-Armed Bandit Allocation Indices*, Wiley, New York, 1989.

[6] J. C. Gittins, *Bandit processes and dynamic allocation indices*, J. Roy. Statist. Soc. Ser. B, 41 (1979), pp. 148–177.

[7] J. C. Gittins and D. M. Jones, *A dynamic allocation index for the sequential design of experiments*, in Progress in Statistics, Colloq. Math. Soc. Janos Bolyai 9, J. Gani, K. Sarkadi, and I. Vincze, eds., North–Holland, Amsterdam, 1974, pp. 241–266.

[8] P. Nash, *A generalised bandit problem*, J. Roy. Statist. Soc. Ser. B, 42 (1980), pp. 165–169.

[9] R. Weber, *On the Gittins index for multiarmed bandits*, Ann. Appl. Probab., 2 (1992), pp. 1024–1033.

[10] J. N. Tsitsiklis, *A lemma on the multi-armed bandit problem*, IEEE Trans. Automat. Control, 31 (1986), pp. 576–577.

[11] J. N. Tsitsiklis, *A short proof of the Gittins index theorem*, Ann. Appl. Probab., 4 (1994), pp. 194–199.

[12] P. Varaiya, J. Walrand, and C. Buyukkoc, *Extensions of the multi-armed bandit problem: The discounted case*, IEEE Trans. Automat. Control, 30 (1985), pp. 426–439.

[13] G. Weiss, *Branching bandit processes*, Probab. Engrg. Inform. Sci., 2 (1988), pp. 269–278.

[14] P. Whittle, *Multi-armed bandits and the Gittins index*, J. Roy. Statist. Soc. Ser. B, 42 (1980), pp. 143–149.

[15] P. Whittle, *Arm acquiring bandits*, Ann. Probab., 9 (1981), pp. 284–292.

SIAM J. DISCRETE MATH.
Vol. 18, No. 1, pp. 114–129

THE NONAPPROXIMABILITY OF NON-BOOLEAN PREDICATES*

LARS ENGEBRETSEN†

Abstract. Constraint satisfaction programs where each constraint depends on a constant number of variables have the following property: The randomized algorithm that guesses an assignment uniformly at random satisfies an expected constant fraction of the constraints. Combining constructions from interactive proof systems with harmonic analysis over finite Abelian groups, Håstad [*J. ACM*, 48 (2001), pp. 798–859] showed that for several constraint satisfaction programs this naive algorithm is essentially the best possible unless $\mathbf{P} = \mathbf{NP}$. While most of the predicates analyzed by Håstad depend on a small number of variables, Samorodnitsky and Trevisan [*Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, Portland, OR, 2000, pp. 191–199] recently extended Håstad's result to predicates depending on an arbitrarily large, but still constant, number of Boolean variables.

We combine ideas from these two constructions and prove that there exists a large class of predicates on finite non-Boolean domains such that for predicates in the class, the naive randomized algorithm that guesses a solution uniformly is essentially the best possible unless $\mathbf{P} = \mathbf{NP}$. As a corollary, we show that it is **NP**-hard to approximate the Maximum k-CSP problem over domains with size d within $d^{k-2k^{1/2}} - \epsilon$, for every constant $\epsilon > 0$, unless $\mathbf{P} = \mathbf{NP}$. This lower bound extends the previously known bound for the case $d = 2$ and matches well with the best known upper bound, d^{k-1}, of Serna, Trevisan, and Xhafa [*Proceedings of the 15th Annual Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Comput. Sci. 1373, M. Morvan, C. Meinel, and D. Krob, eds., Springer-Verlag, Berlin, 1998, pp. 488–498].

Key words. combinatorial optimization, approximation, approximation hardness, maximum CSP

AMS subject classifications. 68Q17, 68Q25, 20K01

DOI. 10.1137/S0895480100380458

1. Introduction.

In a breakthrough paper, Håstad [10] studied the problem of giving approximate solutions to maximization versions of several constraint satisfaction problems (CSPs). An instance of such a problem is given as a collection of constraints, i.e., functions from some domain to $\{0, 1\}$, and the objective is to satisfy as many constraints as possible. An approximate solution of a CSP is simply an assignment that satisfies roughly as many constraints as possible. In this setting, we are interested in proving either that there exists a polynomial time algorithm producing approximate solutions with weight some constant fraction from the optimum weight or that no such algorithms exist.

Typically, each individual constraint depends on a fixed number k of the variables, and the size of the instance is given as the total number of variables that appear in the constraints. In this case, which is usually called the Max k-CSP problem, there exists a very naive algorithm that approximates the optimum within a constant factor—the algorithm that just guesses a solution at random. In his paper, Håstad [10] proved the very surprising fact that this algorithm is essentially the best possible efficient algorithm for several CSPs, unless $\mathbf{P} = \mathbf{NP}$. His proofs unify constructions from interactive proof systems with harmonic analysis over finite groups and give a general framework for proving strong impossibility results regarding the approximation of

*Received by the editors November 3, 2000; accepted for publication (in revised form) January 27, 2004; published electronically July 20, 2004. This research was partly performed while the author was visiting MIT with support from the Marcus Wallenberg Foundation and the Royal Swedish Academy of Sciences.

http://www.siam.org/journals/sidma/18-1/38045.html

†KTH, Numerical Analysis and Computer Science, SE-100 44 Stockholm, Sweden (enge@kth.se).

CSPs. Håstad [10] suggests that predicates with the property that the naive randomized algorithm is the best possible polynomial time approximation algorithm should be called *nonapproximable beyond the random assignment threshold*.

DEFINITION 1.1. *A maximization problem is* nonapproximable beyond the random assignment threshold *if, for every constant $\epsilon > 0$, it is* **NP***-hard to approximate the optimum within a factor $w - \epsilon$, where $1/w$ is the expected fraction of constraints satisfied by a solution guessed uniformly at random.*

Håstad's paper [10] deals mainly with CSPs whose constraints involve a small number of variables, typically three or four. In most of the cases, the variables are Boolean, but Håstad also treats the case of linear equations over finite Abelian groups. In the Boolean case, Håstad's techniques have been extended by Trevisan [21], Sudan and Trevisan [18], and Samorodnitsky and Trevisan [16] to some predicates involving a large, but still constant, number of Boolean variables. In this paper, we prove that those extensions can be adapted also to the non-Boolean case—a fact that is not immediately obvious from the proof for the Boolean case. This establishes nonapproximability beyond the random assignment threshold for a large class of CSPs where the domain of the variables is non-Boolean.

DEFINITION 1.2. *Given a finite Abelian group G, integers $\ell > 0$ and $m > 0$, and a set $E \subseteq [\ell] \times [m]$, let $\mathcal{F}(G, \ell, m, E)$ be the family of constraints of the form*

$$\bigwedge_{i,j:(i,j)\in E} \bigl(x_i y_j y_{i,j} = a_{i,j}\bigr),$$

where $a_{i,j}$ $((i,j) \in E)$ are constants from G and x_i $(1 \le i \le \ell)$, y_j $(1 \le j \le m)$, and $y_{i,j}$ $((i,j) \in E)$ are variables assuming values in G.

Each constraint in $\mathcal{F}(G, \ell, m, E)$ involves at most $\ell + m + |E|$ variables and tests if $|E|$ linear equations are satisfied. The constraint is satisfied if and only if all of the linear equations are satisfied. Our main result is that certain CSPs on constraints from $\mathcal{F}(G, \ell, m, E)$ are nonapproximable beyond the random assignment threshold.

THEOREM 1.3. *For every finite nontrivial Abelian group G, every positive integer ℓ, every positive integer m, and every set $E \subseteq [\ell] \times [m]$, the problem of simultaneously satisfying as many constraints as possible given a collection of constraints from $\mathcal{F}(G, \ell, m, E)$ on some set of variables is nonapproximable beyond the random assignment threshold.*

As special cases, the above theorem includes the results of Håstad [10] (set $\ell = m = |E| = 1$), Trevisan [21] (set $G = \boldsymbol{Z}_2$, $\ell = 2$, $m = 1$, and $|E| = 2$), Sudan and Trevisan [18] (set $G = \boldsymbol{Z}_2$, $\ell = 2$, and $|E| = 2m$), and Samorodnitsky and Trevisan [16] (set $G = \boldsymbol{Z}_2$). The theorem also gives approximation hardness results for the general problem of approximating the maximum of CSPs where each constraint involves at most a fixed number of variables.

DEFINITION 1.4. Max k-CSP-d *is the following maximization problem: Given a finite domain D of size d, a set X of variables assuming values in D, and a number of functions from at most k variables in X to $\boldsymbol{Z}_2$, find an assignment to the variables in X maximizing the number of functions evaluating to* 1.

COROLLARY 1.5. *For every integer $k \ge 3$, every integer $d \ge 2$, and every constant $\epsilon > 0$, it is* **NP***-hard to approximate* Max k*-CSP-*d *within $d^{k-2k^{1/2}} - \epsilon$.*

Proof. Let G be an Abelian group of order d, $\ell = m = \lfloor k^{1/2} \rfloor$ and $|E| = \min\{\ell m, k - \ell - m\}$ in Theorem 1.3. Then the theorem implies that it is **NP**-hard to approximate Max k-CSP-d within $d^{\min\{\ell m, k-\ell-m\}} - \epsilon$.

It remains to show that $\min\{\ell m, k - \ell - m\} \ge k - 2k^{1/2}$ for every $k \ge 3$. To this

end, note that it follows from the definition of the floor operation that there exists a $\xi \in [0,1)$ such that $\ell = m = k^{1/2} - \xi$. Then straightforward substitution shows that $\ell m = k - 2\xi k^{1/2} + \xi^2$ and that $k - \ell - m = k - 2k^{1/2} + 2\xi$. □

The above hardness result for Max k-CSP-d compares favorably with the currently best known polynomial time approximation algorithm [17, 20], which is guaranteed to always deliver a solution at most a factor d^{k-1} from the optimum. Choosing ℓ, m, and $|E|$ in a more careful way, it is possible to improve the inapproximability factor in Corollary 1.5 by constant powers of d.

The high-level structure of our proof is more or less identical to the one used by Samorodnitsky and Trevisan [16]. However, the fact that we are working with finite Abelian groups rather than the familiar group $\boldsymbol{Z}_2$ makes explicit several subtleties in the proof. For instance, it is easy to write an arithmetic expression that serves as in indicator for the event that some Boolean variable x evaluates to false: representing true by -1 and false by 1, this expression is simply $(1+x)/2$. Similarly, a Boolean function may in a straightforward and intuitive way equivalently be viewed as function assuming values in $\{0,1\}$ or in $\{-1,1\}$. Working with an abstract Abelian group requires that such arguments are formalized. How should the group be represented in order to facilitate arithmetic manipulations of, for instance, indicator functions? What is the "correct" way to embed group values into $\boldsymbol{R}$ or $\boldsymbol{C}$?

Our proofs use Fourier analysis of functions from finite Abelian groups to the complex numbers combined with what has now become standard constructions from the world of interactive proof systems. The "ordinary" Fourier transform of Boolean functions is well known to the computer science community. In an attempt to introduce the community to the more general Fourier transform of functions on finite Abelian groups, we have included a short description of it in the introductory part of the paper. Our main technical result, the proof of Lemma 3.4, also serves as an illustration of how this more general Fourier transform may be applied. Following the conference version of our paper [4], several papers have appeared where the Abelian Fourier transform has been applied successfully to establish strong hardness results for several combinatorial optimization problems [5, 6, 12, 13, 14]. In addition to this, the non-Abelian analogue of the Fourier transform, the so-called representation theory of finite groups, has been used by Engebretsen, Holmerin, and Russell [7] to generalize Håstad's hardness result [10] for linear equations involving three variables to all finite groups. We expect that these tools will continue to play an important role in the field of theoretical computer science.

2. Preliminaries. To show our approximation hardness result, we use the same underlying hard problem as many previous constructions. While Håstad [10] describes this problem in terms of a two-prover one-round protocol for a certain class of 3-Sat formulas, we chose one of the alternate formulations: a label cover problem with a certain "gap" property.

DEFINITION 2.1. *Given two sets R and S, an* instance of the label cover problem on R and S *is a set $\Psi \subseteq V \times \Phi$, where V and Φ are sets of variables with ranges R and S, respectively, with an onto function $\pi_{v\phi} \colon S \to R$ for every $(v,\phi) \in \Psi$. The instance is* regular *if every $v \in V$ occurs in the same number of pairs in Ψ and every $\phi \in \Phi$ occurs in the same number of pairs in Ψ.*

The following theorem is a consequence of the so-called "PCP theorem" [2] combined with a certain regularization procedure [8] and the parallel repetition theorem [15].

THEOREM 2.2 (see [2, 8, 15]). *There exists a universal constant $\mu > 0$ such that*

for every large enough constant u there exist sets R and S with $2^u \le |R| \le |S| \le 7^u$ such that it is **NP***-hard to distinguish between the following two cases given a regular instance $\Psi \subseteq V \times \Phi$ of a label cover problem on R and S:*

YES: *There exist assignments $\Pi\colon V \to R$ and $\Sigma\colon \Phi \to S$ such that for every $(v,\phi) \in \Psi$, $\Pi(v) = \pi_{v\phi}(\Sigma(\phi))$.*

NO: *There are no functions P from V to probability distributions on R and Q from Φ to probability distributions on S with the property that*

$$\mathrm{E}_{(v,\phi)\in\Psi}\left[\Pr_{\substack{r\sim P(v)\\ s\sim Q(\phi)}}\left[r = \pi_{v\phi}(s)\right]\right] \ge |R|^{-\mu}. \tag{1}$$

2.1. Our PCP. A probabilistically checkable proof (PCP) consists of a verifier and a proof. The verifier is given an input and oracle access to an alleged proof of the fact that the input belongs to some specified language. The verifier also has access to a specified amount of random bits. Based on the random bits and the input, the verifier decides which positions in the proof it should look at. Once it has examined the positions of its choice, it uses all available information to decide if the input should be accepted or rejected. Our main interest in PCPs comes from the fact that PCPs where the verifier uses a logarithmic, in the size of the input, number of random bits are intimately connected with CSPs. Indeed, enumerating the verifier's acceptance conditions for every possible outcome of the random bits gives rise to a CSP with size polynomial in the size of the input to the PCP verifier. Maximizing the number of simultaneously satisfied constraints in this CSP is the same as finding the proof maximizing the probability that the verifier accepts the input.

In our case, the PCP in some sense "simulates" the above label cover problem: the verifier is given oracle access to an alleged proof of the fact that there exist assignments Π and Σ as described in case YES above, and it is supposed to check if the proof is correct or not. The verifier in our PCP has access to several subtables, one for each $v \in V$ and one for each $\phi \in \Phi$. These tables contain purported encodings of the labels assigned to the corresponding variable. The error correcting code supposedly used to create the tables is the so-called long G-code for some finite Abelian group G, first defined by Håstad [10]. It is a generalization of the ordinary long code, first used in the context of PCPs by Bellare, Goldreich, and Sudan [3]; the ordinary long code is in fact exactly the same thing as the long $\boldsymbol{Z}_2$-code.

Definition 2.3. *For a finite Abelian group G, the* long G-code $A_{v,\sigma}$ *of an assignment σ to some variable v assuming values in some set R is a function mapping $f \in G^R$, i.e., a function from R to G, to a value in G by the map $A_{v,\sigma}\colon f \mapsto f(\sigma)$.*

As mentioned above, the proof in our PCP supposedly contains the long G-code of assignments Π and Σ satisfying property YES in Theorem 2.2. Concretely, the proof contains for each $v \in V$ a string of length $|G|^{|R|}$, interpreted as a function $A_v\colon G^R \to G$, and for each $\phi \in \Phi$ a string of length $|G|^{|S|}$ interpreted as a function $A_\phi\colon G^S \to G$. The proof is *correct* if there exist assignments Π and Σ satisfying property YES in Theorem 2.2 such that A_v is the long code of $\Pi(v)$ and A_ϕ is the long G-code of $\Sigma(\phi)$ for every v and ϕ. In section 3 we design a verifier for the proof described above and formalize the connection between our PCP and the label cover problem. Specifically, we show that if the verifier accepts with "high" probability, then there exist functions Π and Σ satisfying property YES in Theorem 2.2.

2.2. Folded proof tables. Our analysis requires that all the tables in the PCP are *folded over G* [10, sections 2.4–2.6]. That a table $A_v\colon G^R \to G$ is folded amounts

to the requirement that for $f \in G^R$ and $g \in G$, $A_v(gf) = gA_v(f)$, where the function gf is defined by the map $gf\colon r \mapsto gf(r)$. Note that a long G-code is always folded since $A_{v,\sigma}(gf) = (gf)(\sigma) = gf(\sigma) = gA_{v,\sigma}(f)$.

The requirement that a table $A_v\colon G^R \to G$ is folded can be enforced by employing the following *access conventions* in the verifier: partition G^R into equivalence classes under the relation $\sim$, where $f_1 \sim f_2$ if there exists a $g \in G$ such that $f_1 = gf_2$, this equality being an equality of functions. Denote by $\tilde{f}$ the coset representative for the equivalence class that f belongs to. Instead of reading position f in A_v, the verifier reads position $\tilde{f}$ and returns the value $gA_v(\tilde{f})$, where $g \in G$ is such that $f = g\tilde{f}$.

Intuitively, folding can be viewed as a method of restricting the possibility to construct "bad" proofs, i.e., proofs that fool the verifier to accept when it should not. Since the long G-code is folded, a correctly encoded proof is also folded; hence folding does not restrict the possibility to formulate correct proofs.

2.3. Fourier transforms. To prove a bound on the soundness of their verifiers, Håstad [10] as well as Samorodnitsky and Trevisan [16] use Fourier transforms. In this section we give a brief account of the methods involved; for more details see Håstad's paper [10] or Terras's book [19].

The aim of Fourier transforms is to express functions from some domain to the complex numbers as linear combinations of basis functions with certain nice properties. The reader may be familiar with the Boolean case, where a function is usually written in $\{-1,1\}$-notation, i.e., -1 represents true and 1 represents false. The Fourier transform of a function $f\colon \{-1,1\}^n \to \{-1,1\}$ can then be written as $f(x) = \sum_{\alpha\in\{0,1\}^n} \hat{f}_\alpha \chi_\alpha(x)$, where $\hat{f}_\alpha = 2^{-n} \sum_{x\in\{-1,1\}^n} f(x)\chi_\alpha(x) = \mathrm{E}_x[f(x)\chi_\alpha(x)]$ and $\chi_\alpha(x) = \prod_{i=1}^n x_i^{\alpha_i}$. In this paper we need a generalization of this transform. Specifically, we need to apply the Fourier transform to functions from (powers of) an arbitrary finite Abelian group to the complex numbers. It turns out that the correct way to generalize the "basis functions" χ_α is to use the set of all homomorphisms from the group in question to the set of complex numbers of unit norm. For an Abelian group G, the set of all such homomorphisms is denoted by $\hat{G}$; the members of $\hat{G}$ are called *characters of* G. It is true for every Abelian group that the number of distinct characters is equal to the order of the group. In fact something even stronger holds: for finite Abelian groups the set of characters forms a group with complex multiplication as the group operation. This group is usually called the dual group and it is isomorphic to the original group.

Example. The cyclic group $\boldsymbol{Z}_m$ represented as powers of a primitive mth root of unity with multiplication as the group operator has $\hat{\boldsymbol{Z}}_m = \{x \mapsto x^t : 0 \le t < m\}$. The members of $\hat{\boldsymbol{Z}}_m$ are functions from $\boldsymbol{Z}_m$ to $\boldsymbol{C}$; moreover, these functions form a group under multiplication. Notice that each function in $\hat{\boldsymbol{Z}}_m$ can be described by an integer mod m, namely, the power to which the argument of the function is raised. Hence we can actually equivalently view $\hat{\boldsymbol{Z}}_m$ as the group of integers mod m with addition as the group operation.

To construct characters for an arbitrary finite Abelian group, recall that every such group can be written as a direct product of cyclic groups. It turns out that the characters of $G \times H$ are precisely the functions $(g,h) \mapsto \chi(g)\psi(h)$ such that $\chi \in \hat{G}$ and $\psi \in \hat{H}$. More formally, suppose that $G \cong \boldsymbol{Z}_{m_1} \times \cdots \times \boldsymbol{Z}_{m_k}$, where $|G| = m_1 \cdots m_k$, and that $g \in G$ is represented as a k-tuple $(g_1, \ldots, g_k) \in \boldsymbol{Z}_{m_1} \times \cdots \times \boldsymbol{Z}_{m_k}$, where each $\boldsymbol{Z}_{m_i}$ is represented as powers of a primitive m_ith root of unity with multiplication as the group operator; the group operation of G corresponds to coordinatewise application of the group operations in $\boldsymbol{Z}_{m_i}$. The characters of G are then the functions $\chi_a(g) =$

$\prod_{j=1}^{k}(g_j)^{a_j}$ for all vectors $a = (a_1, \ldots, a_k)$ such that a_i is an integer mod m_i. Since each function in $\hat{G}$ corresponds to exactly one such vector a, we may view $\hat{G}$ as the group of vectors a described above with componentwise addition mod m_i as the group operation. It is notationally convenient for us to do computations in this group; we then use addition as the group operator and denote the character corresponding to the all zeroes vector, the so-called trivial character, by χ_0. The set of all characters except the trivial one is denoted by $\hat{G}^*$ in this paper.

PROPOSITION 2.4. *Let G be a finite Abelian group. Then the following identities hold for $g, g' \in G$ and $a, a' \in \hat{G}$:*

$$\chi_a(gg') = \chi_a(g)\chi_a(g'), \tag{2}$$

$$\chi_{a+a'}(g) = \chi_a(g)\chi_{a'}(g), \tag{3}$$

$$\chi_0(g) = \chi_a(1_G) = 1, \tag{4}$$

$$\overline{\chi_a(g)} = \chi_{-a}(g) = \chi_a(g^{-1}). \tag{5}$$

Moreover,

$$\frac{1}{|G|}\sum_{a\in\hat{G}}\chi_a(g) = \begin{cases}1 & \text{if } g = 1_G,\\ 0 & \text{otherwise.}\end{cases} \tag{6}$$

$$\frac{1}{|G|}\sum_{g\in G}\chi_a(g)\overline{\chi_{a'}(g)} = \begin{cases}1 & \text{if } a = a',\\ 0 & \text{otherwise.}\end{cases} \tag{7}$$

Once we have the characters, the Fourier transform of functions from G to $\boldsymbol{C}$ is defined as follows.

DEFINITION 2.5. *For every finite Abelian group G, every function $f\colon G \to \boldsymbol{C}$ can be written as $\sum_{a\in\hat{G}}\hat{f}_a\chi_a(g)$, where the* Fourier coefficients *$\{\hat{f}_a\}_{a\in\hat{G}}$ are defined as $\hat{f}_a = E_{g\in G}[f(g)\overline{\chi_a(g)}] = |G|^{-1}\sum_{g\in G} f(g)\overline{\chi_a(g)}$.*

As an illustration of these concepts, we state and prove the only theorem from classical Fourier analysis that we use in this paper, namely, Plancherel's equality.

LEMMA 2.6. *Suppose that f is a function from G to $\boldsymbol{C}$ and that its Fourier coefficients are $\{\hat{f}_a\}_{a\in\hat{G}}$. Then $\mathrm{E}_{g\in G}[|f(g)|^2] = \sum_{a\in\hat{G}}|\hat{f}_a|^2$.*

Proof. If we expand f in its Fourier series, we get

$$\mathrm{E}_{g\in G}[|f(g)|^2] = \mathrm{E}_{g\in G}[f(g)\overline{f(g)}] = \sum_{a\in\hat{G}}\sum_{a'\in\hat{G}}\hat{f}_a\overline{\hat{f}_{a'}}\,\mathrm{E}_{g\in G}[\chi_a(g)\overline{\chi_{a'}(g)}].$$

Since $\mathrm{E}_{g\in G}[\chi_a(g)\overline{\chi_{a'}(g)}]$ is one when $a = a'$ and zero otherwise according to (7), the only terms remaining in the above double sum are those where $a = a'$. □

Recall that the verifier in our PCP has access to a proof consisting of several subtables, one for each variable in the label cover instance. For each $v \in V$ there is a table A_v, interpreted as a function from G^R to G, and for each $\phi \in \Phi$ there is a table A_ϕ, interpreted as a function from G^S to G. The tables are folded, i.e., $A_v(gf) = gA_v(f)$ and $A_\phi(gh) = gA_\phi(h)$. In our analysis, we need some identities regarding the Fourier transform of a folded table and the Fourier transform of related tables. All these identities have already been obtained by Håstad [10, section 2.6] but we state and prove them here as an illustration of how characters and Fourier coefficients may be manipulated.

Since a function from R to G can be identified by a table of $|R|$ values from G, we identify G^R with the group $G^{|R|}$. For a function $f\colon R \to G$ or, equivalently, an element $f \in G^{|R|}$, we denote by $f(r)$ the coordinate in f corresponding to r. To write the Fourier transform of some function $A\colon G^R \to \boldsymbol{C}$ we need the characters of G^R. As mentioned above, they are all possible products of $|R|$ functions from $\hat{G}$. We use the notation $\hat{G}^R$ for the set of characters of G^R and often view an $\alpha \in \hat{G}^R$ as a vector of $|R|$ values from $\hat{G}$ or, equivalently, a function from R to $\hat{G}$. We can then write the character explicitly as $\chi_\alpha(f)$ as $\prod_{r\in R} \chi_{\alpha(r)}(f(r))$. Using this explicit representation of the characters, we now derive two identities needed in our main technical lemma.

LEMMA 2.7. *Given an Abelian group G and a finite set R, suppose that $A_v\colon G^R \to G$ is folded over G, define $A\colon G^R \to \boldsymbol{C}$ by the map $f \mapsto \chi_a(A_v(f))$ for some $a \in \hat{G}^*$, and let $\hat{A}_\alpha$ be the Fourier coefficients of A. Then $\hat{A}_\alpha = 0$ unless $a = \sum_{r\in R} \alpha(r)$. In particular, $\hat{A}_0 = 0$.*

Proof. Using the definition of the Fourier coefficient,

$$\hat{A}_\alpha = \mathrm{E}_f[A(f)\overline{\chi_\alpha(f)}] = |G|^{-1} \sum_{g\in G} \mathrm{E}_f[A(gf)\overline{\chi_\alpha(gf)}].$$

By the definition of A and the fact that A_v is folded, $A(gf) = \chi_a(g)A(f)$. Using the definition of $\chi_\alpha(\cdot)$ gives

$$\chi_\alpha(gf) = \prod_{r\in R} \chi_{\alpha(r)}(gf(r)) = \chi_\alpha(f) \prod_{r\in R} \chi_{\alpha(r)}(g) = \chi_\alpha(f)\chi_{\sum_{r\in R}\alpha(r)}(g),$$

where the second equality follows from (2) and the third from (3). Since $\overline{\chi_a(g)} = \chi_{-a}(g)$ by (5) we can summarize our calculations by

$$\mathrm{E}_f[A(gf)\overline{\chi_\alpha(gf)}] = \mathrm{E}_f[A(f)\overline{\chi_\alpha(f)}]\chi_a(g)\chi_{-\sum_{r\in R}\alpha(r)}(g);$$

hence

$$\hat{A}_\alpha\left(1 - |G|^{-1} \sum_{g\in G} \chi_{a-\sum_{r\in R}\alpha(r)}(g)\right) = 0.$$

This equality and (7) with $a' = 0$ together imply that either $\hat{A}_\alpha = 0$ or $a = \sum_{r\in R} \alpha(r)$. □

LEMMA 2.8. *Given an Abelian group G, two finite sets R and S, a function $f\colon R \to G$, an onto function $\pi\colon S \to R$, and a $\beta \in \hat{G}^S$, let $\pi_G(\beta) \in \hat{G}^R$ be defined by $\pi_G(\beta)(r) = \sum_{s\in\pi^{-1}(r)} \beta(s)$. Then $\chi_\beta(f \circ \pi) = \chi_{\pi_G(\beta)}(f)$.*

Proof. By definition,

$$\chi_\beta(f \circ \pi) = \prod_{s\in S} \chi_{\beta(s)}(f(\pi(s))) = \prod_{r\in R} \prod_{s\in\pi^{-1}(r)} \chi_{\beta(s)}(f(r)).$$

For each fixed r in the above product,

$$\prod_{s\in\pi^{-1}(r)} \chi_{\beta(s)}(f(r)) = \chi_{\pi_G(\beta)(r)}(f(r))$$

by the identity (3) and the definition of $\pi_G(\beta)$. Hence the lemma follows. □

Select $v \in V$ uniformly at random.
For $j = 1, \ldots, m$, select ϕ_j uniformly at random from $\{\phi : (v, \phi) \in \Psi\}$.
For $i = 1, \ldots, \ell$, select $f_i : R \to G$ uniformly at random.
For $j = 1, \ldots, m$, select $g_j : S \to G$ uniformly at random.
For all $(i, j) \in E$, choose $e_{ij} : S \to G$ such that, independently for all $s \in S$,
  with probability $1 - \delta_1$, $e_{ij}(s) = 1_G$;
  with probability δ_1, $e_{ij}(s)$ is selected uniformly at random from G.
Define h_{ij} by $h_{ij}(s) = \big(f_i(\pi_{v\phi_j}(s))g_j(s)e_{ij}(s)\big)^{-1}$ for all $s \in S$.
If for all $(i, j) \in E$, $A_v(f_i)A_{\phi_j}(g_j)A_{\phi_j}(h_{ij}) = 1_G$, then accept, else reject.

FIG. 1. *The PCP is constructed from an instance $\Psi \subseteq V \times \Phi$ of the label cover problem on R and S. The proof consists of for each $v \in V$ a table A_v defining a function from G^R to G and for each $\phi \in \Phi$ a table A_ϕ defining a function from G^S to G. The verifier is parameterized by the integers ℓ and m, a set $E \subseteq [\ell] \times [m]$, and a constant $\delta_1 > 0$. The actions of the verifier are described above, within the figure.*

3. The PCP verifier. The objective of our verifier is to check if a purported proof does in fact constitute an encoding of a labeling corresponding to the YES case in Theorem 2.2. To obtain this, we—following the construction of Samorodnitsky and Trevisan [16]—first query $2k$ positions from the proof and then, as a checking procedure, construct k^2 linear equations, each of them involving two of the first $2k$ queried positions and one extra variable. The verifier accepts if *all* these linear equations are satisfied. To give a more illustrative picture of the procedure, we let the first $2k$ queries correspond to the vertices of a complete $k \times k$ bipartite graph. The k^2 linear equations that we check then correspond to the edges of this graph.

Given a labeling corresponding to the YES case, it turns out—for our particular choice of verifier—that it is possible to construct a proof such that the verifier accepts with probability almost one. As for the nonapproximability beyond the random assignment threshold, a random assignment to the positions in the proof satisfies all k^2 linear equations simultaneously with probability $|G|^{-k^2}$—the aim of our analysis is to prove that this is essentially the best possible any polynomial time algorithm can accomplish. This follows from a connection between our PCP and the label cover problem: we assume that it is possible to satisfy a fraction $|G|^{-k^2} + \epsilon$ of the equations for some constant $\epsilon > 0$ and prove that this implies that the NO case in Theorem 2.2 is violated. The final link in the chain is the observation that since there are only polynomially many outcomes for the random choices made by the verifier, we can form a CSP with polynomial size by enumerating the checked constraints for every possible outcome. If the resulting CSP is approximable beyond the random assignment threshold, we can use it to decide the **NP**-hard label cover problem.

We remark that by checking the equations corresponding to some subset E of the edges in the complete bipartite graph we also get a predicate that is nonapproximable beyond the random assignment threshold. It is satisfied with probability $|G|^{-|E|}$ by a random assignment and our proof methodology works also for this case.

LEMMA 3.1. *Suppose that a PCP is constructed from an instance $\Psi \subseteq V \times \Phi$ of the label cover problem on R and S as described in Figure* 1. *If the case YES holds in Theorem* 2.2*, there exists a proof that convinces the verifier with probability at least $(1 - \delta_1)^{|E|}$.*

Proof. Given assignments Π and Σ guaranteed by the YES case in Theorem 2.2, construct a proof as follows. For each $v \in V$, let A_v be the long G-code of $\Pi(v)$; for

each $\phi \in \Phi$, let A_ϕ be the long G-code of $\Sigma(\phi)$. Then

$$A_v(f_i)A_{\phi_j}(g_j)A_{\phi_j}(h_{ij}) = f_i(\Pi(v))g_j(\Sigma(\phi_j))h_{ij}(\Sigma(\phi_j)).$$

Suppose that $e_{ij}(\Sigma(\phi_j)) = 1_G$ for all $(i,j) \in E$; this happens with probability at least $(1-\delta_1)^{|E|}$. Then

$$h_{ij}(\Sigma(\phi_j)) = \big(g_j(\Sigma(\phi_j))\big)^{-1}\Big(f_i\big(\pi_{v\phi_j}(\Sigma(\phi_j))\big)\Big)^{-1}$$

and hence

$$A_v(f_i)A_{\phi_j}(g_j)A_{\phi_j}(h_{ij}) = f_i(\Pi(v))\big(f_i(\pi_{v\phi_j}(\Sigma(\phi_j)))\big)^{-1}.$$

Since the assignments Π and Σ satisfy property YES in Theorem 2.2, it is guaranteed that $\pi_{v\phi_j}(\Sigma(\phi_j)) = \Pi(v)$; therefore $A_v(f_i)A_{\phi_j}(g_j)A_{\phi_j}(h_{ij}) = 1_G$. □

The remainder of the proof establishes that if the verifier accepts with probability at least $(1+\delta_2)/|G|^{|E|}$ for any constant $\delta_2 > 0$, then case NO in Theorem 2.2 cannot hold. The high-level structure of the proof is the same as in earlier constructions [10, 16]. First the acceptance predicate is arithmetized.

In the Boolean case, it is straightforward to express the indicator for the event that a Boolean variable is false with a closed form formula that is easy to analyze: the expression $(1+b)/2$ is one if b is false and zero otherwise. (Recall the we work with $\{-1,1\}$-representation of Boolean values where 1 represents false.) For an arbitrary Abelian group, the sum (6) provides us with the "right" generalization of the above formula: for an element $g \in G$, $|G|^{-1}\sum_{a\in\hat{G}}\chi_a(g)$ evaluates to one if $g = 1_G$ and zero otherwise.

LEMMA 3.2. *The test in the PCP accepts with probability* $|G|^{-|E|}\sum_{P\subseteq E}\mathrm{E}[T_P]$, *where the expectation is over the verifier's choice of* v, $\phi_1,\ldots,\phi_m$, $f_1,\ldots,\phi_\ell$, $g_1,\ldots,g_m$, *and* e_{ij} $((i,j)\in E)$, *and*

$$T_P = \prod_{(i,j)\in P}\Big(\sum_{a\in\hat{G}^*}\chi_a\big(A_v(f_i)A_{\phi_j}(g_j)A_{\phi_j}(h_{ij})\big)\Big). \tag{8}$$

Proof. The PCP tests if $|E|$ linear equations of the form $A_v(f_i)A_{\phi_j}(g_j)A_{\phi_j}(h_{ij}) = 1_G$ are satisfied. We index the equations by (i,j), and we note that the fact (4) that $\chi_0(g) = 1$ for all $g \in G$ and the summation relation (6) together imply that the expression

$$P_{ij} = \frac{1}{|G|}\Big(1 + \sum_{a\in\hat{G}^*}\chi_a\big(A_v(f_i)A_{\phi_j}(g_j)A_{\phi_j}(h_{ij})\big)\Big)$$

is one when the equation corresponding to (i,j) is satisfied and zero otherwise. Since the test accepts if all equations are satisfied,

$$\prod_{(i,j)\in E}P_{ij} = \begin{cases}1 & \text{if the test in the PCP accepts,}\\ 0 & \text{otherwise;}\end{cases}$$

hence the test accepts with probability $\mathrm{E}[\prod_{(i,j)\in E}P_{ij}]$. Expanding the product, we obtain the claim in the formulation of the lemma. □

Having arithmetized the acceptance probability, the main part of the analysis begins. The most technical part of the argument is to establish that whenever the

expression derived in Lemma 3.2 is "large" for a certain value of "large," there exist functions P and Q satisfying the inequality (1) in Theorem 2.2. This follows from manipulations of the above expression with the aid of Fourier analysis and averaging arguments. We remark that our original proof [4] was more complicated than the one given below in the proof of Lemma 3.4. The main simplification of the proof amounts to a certain way to fix many of the functions f_i, g_j, and e_{ij} at an early stage of the proof; this method was first used by Håstad and Wigderson [11]. We frequently use a straightforward consequence of Jensen's inequality in the proof and therefore first state this relation formally.

LEMMA 3.3. *Let X be a finite set. For every function $f\colon X \to \boldsymbol{C}$,*

$$\max_{x\in X} |f(x)| \geq \mathrm{E}_{x\in X}[|f(x)|] \geq |\,\mathrm{E}_{x\in X}[f(x)]|,$$
$$\max_{x\in X} |f(x)|^2 \geq \mathrm{E}_{x\in X}[|f(x)|^2] \geq |\,\mathrm{E}_{x\in X}[f(x)]|^2,$$
$$|X| \max_{x\in X} |f(x)| \geq \sum_{x\in X} |f(x)| \geq \Big|\sum_{x\in X} f(x)\Big|.$$

Proof. Jensen's inequality implies that for every set $\{z_i\}_{i=1}^n$ of complex numbers, every set $\{\lambda_i\}_{i=1}^n$ of real numbers, and every convex function $g\colon \boldsymbol{C} \to \boldsymbol{R}$, $\sum_i \lambda_i g(z_i) \geq g(\sum_i \lambda_i z_i)$ as soon as $\sum_i \lambda_i = 1$. Since the functions $z \mapsto |z|$ and $z \mapsto |z|^2$ are convex, the second equalities on each line of the lemma follow. The first inequalities follow from a simple averaging argument. □

LEMMA 3.4. *Suppose that a PCP is constructed from a regular instance $\Psi \subseteq V \times \Phi$ of the label cover problem on R and S as described in Figure 1 and that the PCP verifier accepts with probability at least $(1+\delta_2)/|G|^{|E|}$ for some constant $\delta_2 \geq 0$. Then there exist functions P from V to probability distributions on R and Q from Φ to probability distributions on S such that*

$$\mathrm{E}_{(v,\phi)\in\Psi}\left[\Pr_{\substack{r\sim P(v)\\ s\sim Q(\phi)}}\big[r = \pi_{v\phi}(s)\big]\right] \geq \frac{2\delta_1\delta_2^2}{(2^{|E|}-1)^2(|G|-1)^{2|E|}}\,. \tag{9}$$

Proof. Consider the expression for the acceptance probability derived in Lemma 3.2. Since the term corresponding to $P = \emptyset$ in the sum $|G|^{-|E|}\sum_{P\subseteq E} \mathrm{E}[T_P]$ contributes with $|G|^{-|E|}$, the assumption that the PCP verifier accepts with probability at least $(1+\delta_2)/|G|^{|E|}$ implies that

$$\delta_2 \leq \sum_{\substack{P\subseteq E\\ P\neq\emptyset}} \mathrm{E}[T_P] = \left|\sum_{\substack{P\subseteq E\\ P\neq\emptyset}} \mathrm{E}[T_P]\right| < (2^{|E|}-1)\max_{\substack{P\subseteq E\\ P\neq\emptyset}} |\,\mathrm{E}[T_P]|,$$

where the last inequality follows from Lemma 3.3. We now fix P to a value that attains the above maximum. The first phase of the proof then continues by gradually fixing more and more parameters in the expression for $|\,\mathrm{E}[T_P]|$. Switching the order of product and summation in (8) gives

$$\delta_2 \leq (2^{|E|}-1)|\,\mathrm{E}[T_P]| = \left|\sum_{\substack{a_{ij}\in\hat{G}^*\\ (i,j)\in P}} \mathrm{E}\Big[\prod_{(i,j)\in P} \chi_{a_{ij}}\big(A_v(f_i)A_{\phi_j}(g_j)A_{\phi_j}(h_{ij})\big)\Big]\right|.$$

The summation sign above denotes $|P|$ summations; each summation is over $a_{ij} \in \hat{G}^*$. By Lemma 3.3 and the fact that $|P| \leq |E|$, there exists a way to fix all a_{ij} such that

$$\delta_2 \leq (2^{|E|}-1)(|G|-1)^{|E|}\left|\mathrm{E}\Big[\prod_{(i,j)\in P}\chi_{a_{ij}}\big(A_v(f_i)A_{\phi_j}(g_j)A_{\phi_j}(h_{ij})\big)\Big]\right|,$$

and we now keep a_{ij} fixed in such a way for the rest of the analysis. We also introduce the shorthand $\delta_3 = \delta_2/(2^{|E|}-1)(|G|-1)^{|E|}$. Let (σ,τ) be an arbitrary pair in P. Since the functions are chosen independently from each other and from v and $\phi_1,\ldots,\phi_m$, the above expression can be rewritten as

$$\delta_3 \leq \left|\mathrm{E}\left[\mathrm{E}_{\substack{v,\phi_1,\ldots,\phi_m\\ f_\sigma,g_\tau,e_{\sigma\tau}}}\Big[\prod_{(i,j)\in P}\chi_{a_{ij}}\big(A_v(f_i)A_{\phi_j}(g_j)A_{\phi_j}(h_{ij})\big)\Big]\right]\right|,$$

where the outer expectation is over the functions $\{f_i : i \neq \sigma\}$, $\{g_j : j \neq \tau\}$, and $\{e_{ij} : (i,j) \neq (\sigma\tau)\}$. By Lemma 3.3, there hence exists a way to fix those functions such that

$$\delta_3 \leq \left|\mathrm{E}_{\substack{v,\phi_1,\ldots,\phi_m\\ f_\sigma,g_\tau,e_{\sigma\tau}}}\Big[\prod_{(i,j)\in P}\chi_{a_{ij}}\big(A_v(f_i)A_{\phi_j}(g_j)A_{\phi_j}(h_{ij})\big)\Big]\right|.$$

We now keep all functions but $\{f_\sigma, g_\tau, e_{\sigma\tau}\}$ fixed in such a way for the remainder of the proof. In fact, we are now in a situation similar to, but not quite identical to, the one considered by Håstad [10, Lemmas 5.2 and 5.11], who estimated an expectation of the product of three correlated functions. To see this, define the functions

$$\begin{aligned} A\colon f_\sigma &\mapsto \chi_{a_{\sigma\tau}}(A_v(f_\sigma))\prod_{\substack{(\sigma,j)\in P\\ j\neq\tau}}\chi_{a_{\sigma,j}}\big(A_v(f_\sigma)A_{\phi_j}(g_j)A_{\phi_j}(h_{\sigma,j})\big),\\ B\colon h_{\sigma\tau} &\mapsto \chi_{a_{\sigma\tau}}(A_{\phi_\tau}(h_{\sigma\tau})),\\ C\colon g_\tau &\mapsto \chi_{a_{\sigma\tau}}(A_{\phi_\tau}(g_\tau))\prod_{\substack{(i,j)\in P\\ i\neq\sigma}}\chi_{a_{ij}}\big(A_v(f_i)A_{\phi_j}(g_j)A_{\phi_j}(h_{ij})\big). \end{aligned}$$

Using these three functions and (2), the above inequality can then be rewritten as

$$\delta_3 \leq \left|\mathrm{E}_{\substack{v,\phi_1,\ldots,\phi_m\\ f_\sigma,g_\tau,e_{\sigma\tau}}}\big[A(f_\sigma)B(h_{\sigma\tau})C(g_\tau)\big]\right|.$$

Using the fact that $h_{\sigma\tau}$ is defined as $((f_\sigma \circ \pi_{v\phi_\tau})g_\tau e_{\sigma\tau})^{-1}$ and, again, the fact that the functions f_σ, g_τ, and $e_{\sigma\tau}$ are independent of v and $\phi_1,\ldots,\phi_m$, we can simplify the notation even further to

$$\delta_3 \leq \left|\mathrm{E}_{v,\phi_1,\ldots,\phi_m}\Big[\mathrm{E}_{f,g,e}\big[A(f)B(((f\circ\pi_{v\phi_\tau})ge)^{-1})C(g)\big]\Big]\right|. \tag{10}$$

The second phase of the proof concentrates on the inner expectation above. To this end, we expand each of the functions A, B, and C in their Fourier series:

$$\begin{aligned} A(f) &= \sum_{\alpha\in\hat{G}^R}\hat{A}_\alpha\chi_\alpha(f),\\ B(h) &= \sum_{\beta_1\in\hat{G}^S}\hat{B}_{\beta_1}\chi_{\beta_1}(h),\\ C(g) &= \sum_{\beta_2\in\hat{G}^S}\hat{C}_{\beta_2}\chi_{\beta_2}(g). \end{aligned}$$

Note that the function A depends implicitly on v and $\{\phi_j\}_{j\neq\tau}$ while B depends implicitly on ϕ_τ. Hence, the Fourier coefficients $\hat{A}_\alpha$ depend on v and $\{\phi_j\}_{j\neq\tau}$ and $\hat{B}_\beta$ depend on ϕ_τ. Inserting the Fourier expansions results in the identity

$$\begin{aligned}&\mathrm{E}_{f,g,e}\big[A(f)B(((f\circ\pi_{v\phi_\tau})ge)^{-1})C(g)\big]\\&\quad=\sum_{\alpha,\beta_1,\beta_2}\hat{A}_\alpha\hat{B}_{\beta_1}\hat{C}_{\beta_2}\,\mathrm{E}_{f,g,e}\big[\chi_\alpha(f)\chi_{\beta_1}(((f\circ\pi_{v\phi_\tau})ge)^{-1})\chi_{\beta_2}(g)\big].\end{aligned}$$

Using the identities (2), (3), and (5) together with Lemma 2.8, we can simplify the above sum to

$$\sum_{\alpha,\beta_1,\beta_2}\hat{A}_\alpha\hat{B}_{\beta_1}\hat{C}_{\beta_2}\,\mathrm{E}_f\big[\chi_{\alpha-\pi_G(\beta_1)}(f)\big]\,\mathrm{E}_g\big[\chi_{\beta_2-\beta_1}(g)\big]\,\mathrm{E}_e\big[\chi_{\beta_2}(e)\big].$$

From the summation identity (6) it follows that

$$\mathrm{E}_f\big[\chi_{\alpha-\pi_G(\beta_1)}(f)\big]=\begin{cases}1 & \text{if } \alpha=\pi_G(\beta_1),\\ 0 & \text{otherwise,}\end{cases}$$

and that

$$\mathrm{E}_g\big[\chi_{\beta_2-\beta_1}(g)\big]=\begin{cases}1 & \text{if } \beta_1=\beta_2,\\ 0 & \text{otherwise.}\end{cases}$$

To calculate $\mathrm{E}_e[\chi_{\beta_2}(e)]$ notice that

$$\mathrm{E}_e\big[\chi_{\beta_2}(e)\big]=\mathrm{E}_e\Big[\prod_{s\in S}\chi_{\beta_2(s)}\big(e(s)\big)\Big]=\prod_{s\in S}\mathrm{E}_{e(s)}\big[\chi_{\beta_2(s)}\big(e(s)\big)\big],$$

where the last inequality follows since e is selected in such a way that $e(s)$ is independent of all other coordinates in e. For factors such that $\beta_2(s)=0$ in the above product, (4) implies that $\mathrm{E}_{e(s)}[\chi_{\beta_2(s)}(e(s))]=1$; for the remaining factors,

$$\mathrm{E}_{e(s)}[\chi_{\beta_2(s)}(e(s))]=(1-\delta_1)\cdot 1+\delta_1\cdot\frac{1}{|G|}\sum_{g\in G}\chi_{\beta_2(s)}(g).$$

The last term above is always zero thanks to (7) with $a=\beta_2(s)\neq 0$ and $a'=0$; hence $\mathrm{E}_e[\chi_{\beta_2}(e)]=(1-\delta_1)^{|\beta_2|}$, where we introduced the notation $|\beta_2|$ to denote $|\{s:\beta_2(s)\neq 0\}|$. To summarize,

$$\mathrm{E}_{f,g,e}\big[A(f)B((fge)^{-1})C(g)\big]=\sum_{\beta\in\hat{G}^S}\hat{A}_{\pi_G(\beta)}\hat{B}_\beta\hat{C}_\beta(1-\delta_1)^{|\beta|}.$$

Inserting this expression into (10) results in the inequality

$$\delta_3\leq\Big|\mathrm{E}_{v,\phi_1,\ldots,\phi_m}\Big[\sum_{\beta\in\hat{G}^S}\hat{A}_{\pi_G(\beta)}\hat{B}_\beta\hat{C}_\beta(1-\delta_1)^{|\beta|}\Big]\Big|.\tag{11}$$

Squaring the above inequality and using Lemma 3.3 gives the bound

$$\delta_3^2\leq\mathrm{E}_{v,\phi_1,\ldots,\phi_m}\Big[\Big|\sum_{\beta\in\hat{G}^S}\hat{A}_{\pi_G(\beta)}\hat{B}_\beta\hat{C}_\beta(1-\delta_1)^{|\beta|}\Big|^2\Big].$$

Moreover, the Cauchy–Schwarz inequality shows that

$$\left|\sum_{\beta\in\hat{G}^S}\hat{A}_{\pi_G(\beta)}\hat{B}_\beta\hat{C}_\beta(1-\delta_1)^{|\beta|}\right|^2 \le \left(\sum_{\beta\in\hat{G}^S}|\hat{A}_{\pi_G(\beta)}|^2|\hat{B}_\beta|^2(1-\delta_1)^{2|\beta|}\right)\left(\sum_{\beta\in\hat{G}^S}|\hat{C}_\beta|^2\right).$$

Finally, Plancherel's equality (Lemma 2.6) shows that $\sum_\beta|\hat{C}_\beta|^2 = \mathrm{E}_g[|C(g)|^2] = 1$, where the last equality follows since $|C(g)| = 1$ for all g—recall that C is a product of characters and hence a root of unity. Putting these bounds together shows that

$$\delta_3^2 \le \mathrm{E}_{v,\phi_1,\dots,\phi_m}\left[\sum_{\beta\in\hat{G}^S}|\hat{A}_{\pi_G(\beta)}|^2|\hat{B}_\beta|^2(1-\delta_1)^{2|\beta|}\right].$$

Recall that our aim is to show that there exists a probabilistic procedure to select labels such that many of the constraints in the label cover instance are satisfied. The third, and final, phase of the proof establishes this by constructing probability distributions from the Fourier coefficients $\{\hat{A}_\alpha\}$ and $\{\hat{B}_\beta\}$.

The expectation above is over $v,\phi_1,\dots,\phi_m$ as they are selected by the PCP verifier. More precisely, first $v\in V$ is selected uniformly at random, then for all j $(1\le j\le m)$ ϕ_j is selected uniformly at random from $\{\phi:(v,\phi)\in\Psi\}$. Since the instance of label cover was assumed to be regular, the following probabilistic procedure induces the same distribution on $v,\phi_1,\dots,\phi_m$. First select (v,ϕ_τ) uniformly at random from Ψ; then for all j $(1\le j\le m\wedge j\ne\tau)$ ϕ_j is selected uniformly at random from $\{\phi:(v,\phi)\in\Psi\}$. Therefore, the above inequality can be equivalently written as

$$\mathrm{E}_{(v,\phi_\tau)\in\Psi}\left[\sum_{\beta\in\hat{G}^S}\mathrm{E}_{\phi_j(j\ne\tau)}\left[|\hat{A}_{\pi_G(\beta)}|^2\right]|\hat{B}_\beta|^2(1-\delta_1)^{2|\beta|}\right]\ge\delta_3^2, \tag{12}$$

where the inner expectation is over ϕ_j $(j\ne\tau)$ given v.

Consider the following probabilistic procedure that given $v\in V$ selects an $r\in R$ and given $\phi_\tau\in\Phi$ selects an $s\in S$. Given v, first select $\alpha\in\hat{G}^R$ according to the probability distribution given by $\mathrm{E}_{\phi_j(j\ne\tau)}[|\hat{A}_\alpha|^2]$ and then select an r such that $\alpha(r)\ne0$ uniformly at random. If no such r exists, select an $r\in R$ uniformly at random. Given $\phi_\tau\in\Phi$, first select $\beta\in\hat{G}^R$ according to the probability distribution given by $|\hat{B}_\beta|^2$ and then select an s such that $\beta(s)\ne0$ uniformly at random.

There always exists at least one s such that $\beta(s)\ne0$, for $\hat{B}_0=0$ by Lemma 2.7 since the table A_{ϕ_τ} is folded and $\{\hat{B}_\beta\}$ are the Fourier coefficients of $\chi_a\circ A_{\phi_\tau}$ for some $a\ne0$. Moreover, Plancherel's equality (Lemma 2.6) implies that $\sum_\alpha|\hat{A}_\alpha|^2 = \sum_\beta|\hat{B}_\beta|^2=1$; therefore, the procedure described above indeed defines functions P from V to probability distributions on R and Q from Φ to probability distributions on S. We now analyze the success rate of the above strategy.

Note that if $\alpha=\pi_G(\beta)\ne0$ there exists for each $r\in R$ such that $\alpha(r)\ne0$ at least one $s\in S$ such that $\beta(s)\ne0$ and $\pi_{v\phi_\tau}(s)=r$. This follows since by definition $\pi_G(\beta)(r)=\sum_{s:r=\pi_{v\phi_\tau}(s)}\beta(s)$ and has the following important consequence. Suppose that (α,β) are selected such that $\alpha=\pi_G(\beta)\ne0$ and that an r such that $\alpha(r)\ne0$ is selected. Then, if an s such that $\beta(s)\ne0$ is selected uniformly at random, $r=\pi_{v\phi}(s)$ with probability at least $|\beta|^{-1}$.

Note also that if $\hat{B}_\beta\ne0$, then $\pi_G(\beta)\ne0$. This follows from Lemma 2.7 since $\hat{B}_\beta$ is a Fourier coefficient of a nontrivial character composed with a folded table. The fact that $\hat{B}_\beta$ is nonzero implies, by Lemma 2.7, that $\sum_{s\in S}\beta(s)\ne0$. But this sum

is precisely $\sum_{r\in R}\pi_G(\beta)(r)$ by the definition of the "projected character" $\pi_G(\beta)$ in Lemma 2.8. Therefore it cannot be the case that $\pi_G(\beta)(r)=0$ for all r.

Combining these two observations, we can conclude that the probability that the above probabilistic procedure succeeds for an arbitrary pair $(v,\phi_\tau)\in\Psi$ to select values r and s such that $r=\pi_{v\phi_\tau}(s)$ is no less than

$$\sum_{\beta\in\hat{G}^S}\mathrm{E}_{\phi_j(j\neq\tau)}\big[|\hat{A}_{\pi_G(\beta)}|^2\big]|\hat{B}_\beta|^2|\beta|^{-1}\geq 2\delta_1\sum_{\beta\in\hat{G}^S}\mathrm{E}_{\phi_j(j\neq\tau)}\big[|\hat{A}_{\pi_G(\beta)}|^2\big]|\hat{B}_\beta|^2(1-\delta_1)^{2|\beta|},$$

where the inequality follows since $x^{-1}>\delta e^{-\delta x}>\delta(1-\delta)^x$ for all $x\geq 0$ and all $\delta\in(0,1)$. Finally, since

$$\mathrm{E}_{(v,\phi_\tau)\in\Psi}\left[2\delta_1\sum_{\beta\in\hat{G}^S}\mathrm{E}_{\phi_j(j\neq\tau)}\big[|\hat{A}_{\pi_G(\beta)}|^2\big]|\hat{B}_\beta|^2(1-\delta_1)^{2|\beta|}\right]\geq 2\delta_1\delta_3^2$$

by (12), the proof is completed. □

4. The reduction to non-Boolean CSPs. We now show how the above PCP can be connected with CSPs to prove that the corresponding CSPs are nonapproximable beyond the random assignment threshold.

Proof of Theorem 1.3. Select the constants $\delta_1>0$ and $\delta_2>0$ such that

$$\frac{|G|^{|E|}(1-\delta_1)^{|E|}}{1+\delta_2}\geq|G|^{|E|}-\epsilon$$

and select u in Theorem 2.2 such that $2\delta_1\delta_2^2(2^{|E|}-1)^{-2}(|G|-1)^{-2|E|}>|R|^{-\mu}$. Let $\ell=|U|$ and $m=|W|$ and consider a PCP constructed from a regular instance $\Psi\subseteq V\times\Phi$ of the label cover problem on R and S as described in Figure 1.

Construct a collections of constraints as follows. Introduce variables $x_{v,f}$ and $y_{\phi,g}$ representing $A_v(f)$ and $A_\phi(g)$, respectively. For all ways that the verifier can select v, $\phi_1,\ldots,\phi_m$, $f_1,\ldots,f_\ell$, $g_1,\ldots,g_m$, and $h_{1,1},\ldots,h_{\ell,m}$, introduce a constraint that is one exactly when $x_{v,f_i}y_{\phi_j,g_j}=y_{\phi_j,h_{ij}}$ for all $(i,j)\in E$. Set the weight of this constraint to the probability of the event that v, $\phi_1,\ldots,\phi_m$, $f_1,\ldots,f_\ell$, $g_1,\ldots,g_m$, and $h_{1,1},\ldots,h_{\ell,m}$ are chosen by the verifier in the PCP. Each constraint is a function of at most $|E|+\ell+m$ variables. The total number of constraints is definitely less than $|\Psi|^m|G|^{\ell 2^u+m2^{3u}+\ell m2^{3u}}$, which is polynomial in the size of the label cover instance if ℓ, m, $|G|$, and u are constants. The weight of the satisfied equations for a given assignment to the variables is equal to the probability that the PCP accepts the proof corresponding to this assignment. Thus, any algorithm approximating the optimum of the above instance within

$$\frac{|G|^{|E|}(1-\delta_1)^{|E|}}{1+\delta_2}\geq|G|^{|E|}-\epsilon$$

decides the **NP**-hard label cover problem. □

5. Conclusions. We have shown that it is possible to combine the harmonic analysis introduced by Håstad [10] with the recycling techniques used by Samorodnitsky and Trevisan [16] to obtain a lower bound on the approximability of Max k-CSP-d. The current state of the art regarding the (non)approximability of predicates is that there are a number of predicates—such as linear equations mod p with three unknowns

in every equation, E3-satisfiability, and the predicates of this paper—that are nonapproximable beyond the random assignment threshold [10, 16]. There also exists some predicates—such as linear equations mod p with two unknowns in every equation [1], constraints on exactly two Boolean variables [9], and a large class of so-called regular constraints on two non-Boolean variables [5]—where there are polynomial time algorithms beating the bound obtained from a random assignment.

A very interesting direction for future research is to try to determine criteria identifying predicates that are nonapproximable beyond the random assignment threshold. Some such attempts have been made for special cases. It is known that every predicate on two Boolean variables is approximated beyond the random assignment threshold [9]; for predicates of three Boolean variables, it is known that the predicates that are nonapproximable beyond the random assignment threshold are precisely those that are implied by parity [10, 22]. However, the general question remains completely open.

Acknowledgments. The author thanks Johan Håstad for many clarifying discussions on the subject of this paper and the anonymous referees for many constructive comments and suggestions that helped improve the presentation of the results in the paper.

REFERENCES

[1] G. Andersson, L. Engebretsen, and J. Håstad, *A new way of using semidefinite programming with applications to linear equations mod p*, J. Algorithms, 39 (2001), pp. 162–204.

[2] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, *Proof verification and the hardness of approximation problems*, J. ACM, 45 (1998), pp. 501–555.

[3] M. Bellare, O. Goldreich, and M. Sudan, *Free bits, PCPs, and nonapproximability—towards tight results*, SIAM J. Comput., 27 (1998), pp. 804–915.

[4] L. Engebretsen, *The non-approximability of non-Boolean predicates*, in Proceedings of the 5th International Workshop on Randomization and Approximation Techniques in Computer Science, Lecture Notes in Comput. Sci. 2129, M. Goemans, K. Jansen, J. D. P. Rolim, and L. Trevisan, eds., Springer-Verlag, Berlin, 2001, pp. 241–248.

[5] L. Engebretsen and V. Guruswami, *Is constraint satisfaction over two variables always easy?*, Random Structures Algorithms, 25 (2004), pp. 150–178.

[6] L. Engebretsen and J. Holmerin, *Three-query PCPs with perfect completeness over non-Boolean domains*, in Proc. 18th IEEE Conference on Computational Complexity, Århus, 2003, pp. 284–299.

[7] L. Engebretsen, J. Holmerin, and A. Russell, *Inapproximability results for equations over finite groups*, Theoret. Comput. Sci., 312 (2004), pp. 17–45.

[8] U. Feige, *A threshold of* $\ln n$ *for approximating set cover*, J. ACM, 45 (1998), pp. 634–652.

[9] M. X. Goemans and D. P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. ACM, 42 (1995), pp. 1115–1145.

[10] J. Håstad, *Some optimal inapproximability results*, J. ACM, 48 (2001), pp. 798–859.

[11] J. Håstad and A. Wigderson, *Simple analysis of graph tests for linearity and PCP*, Random Structures Algorithms, 22 (2003), pp. 139–160.

[12] S. Khot, *Hardness results for approximate hypergraph coloring*, in Proceedings of the 34th Annual ACM Symposium on Theory of Computing, Montréal, QC, Canada, 2002, pp. 351–359.

[13] S. Khot, *Hardness results for coloring* 3*-colorable* 3*-uniform hypergraphs*, in Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, Vancouver, BC, Canada, 2002, pp. 23–32.

[14] S. Khot, *On the power of unique* 2*-prover* 1*-round games*, in Proceedings of the 34th Annual ACM Symposium on Theory of Computing, Montréal, QC, Canada, 2002, pp. 767–775.

[15] R. Raz, *A parallel repetition theorem*, SIAM J. Comput., 27 (1998), pp. 763–803.

[16] A. Samorodnitsky and L. Trevisan, *A PCP characterization of NP with optimal amortized query complexity*, in Proceedings of the 32nd Annual ACM Symposium on Theory of Computing, Portland, OR, 2000, pp. 191–199.

[17] M. Serna, L. Trevisan, and F. Xhafa, *The (parallel) approximability of non-Boolean satisfiability problems and restricted integer programming*, in Proceedings of the 15th Annual Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Comput. Sci. 1373, M. Morvan, C. Meinel, and D. Krob, eds., Springer-Verlag, Berlin, 1998, pp. 488–498.

[18] M. Sudan and L. Trevisan, *Probabilistically checkable proofs with low amortized query complexity*, in Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science, Palo Alto, CA, 1998, pp. 18–27.

[19] A. Terras, *Fourier Analysis on Finite Groups and Applications*, London Math. Soc. Stud. Texts 43, Cambridge University Press, Cambridge, UK, 1999.

[20] L. Trevisan, *Parallel approximation algorithms by positive linear programming*, Algorithmica, 21 (1998), pp. 72–88.

[21] L. Trevisan, *Recycling queries in PCPs and in linearity tests*, in Proceedings of the 30th Annual ACM Symposium on Theory of Computing, Dallas, TX, 1998, pp. 299–308.

[22] U. Zwick, *Approximation algorithms for constraint satisfaction programs involving at most three variables per constraint*, in Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA, 1998, ACM, New York, 1998, pp. 201–210.

SIAM J. DISCRETE MATH.
Vol. 18, No. 1, pp. 130–140

ON DECIMATIONS OF ℓ-SEQUENCES*

MARK GORESKY†, ANDREW KLAPPER‡, RAM MURTY§, AND IGOR SHPARLINSKI¶

Abstract. Maximal length feedback with carry shift register sequences have several remarkable statistical properties. Among them is the property that the arithmetic correlations between any two cyclically distinct decimations are precisely zero. It is open, however, whether all such pairs of decimations are indeed cyclically distinct. In this paper we show that the set of distinct decimations is large and, in some cases, all decimations are distinct.

Key words. feedback with carry shift register, arithmetic correlation, exponential sum, binary sequence, p-adic number

AMS subject classifications. 11A07, 11B50, 11L03, 11L07, 11L26, 11T23, 94A55, 94B40

DOI. 10.1137/S0895480102403428

1. Introduction.

If $\mathbf{a} = (a_0, a_1, a_2, \dots)$ is a periodic binary sequence, let $\mathbf{a}_\tau = (a_\tau, a_{\tau+1}, a_{\tau+2}, \dots)$ denote the τ-shifted sequence. If $\mathbf{a}, \mathbf{b}$ are periodic binary sequences with the same period T we say they are *cyclically distinct* if $\mathbf{a}_\tau \neq \mathbf{b}$, for every shift τ with $0 < \tau < T$.

Associate to $\mathbf{a}$ and $\mathbf{b}_\tau$ the 2-adic integers

$$\alpha = \sum_{i=0}^{\infty} a_i 2^i \qquad \text{and} \qquad \beta_\tau = \sum_{i=0}^{\infty} b_{i+\tau} 2^i.$$

We recall that if $\bar{b}_{i+\tau} = 1 - b_{i+\tau}$ denotes the complementary bit, then $-\beta_\tau = 1 + \sum_{i=0}^{\infty} \bar{b}_{i+\tau} 2^i$. Let

$$\gamma = \alpha - \beta_\tau = \sum_{i=0}^{\infty} c_i 2^i$$

be the difference. The sequence of bits $\mathbf{c} = (c_0, c_1, \dots)$ is eventually periodic (with period T), and the *arithmetic cross-correlation* $C_{\mathbf{a},\mathbf{b}}(\tau)$ is defined to be the number of zeroes minus the number of ones in a single window of size T within the periodic part of $\mathbf{c}$. The pair of sequences $\mathbf{a}, \mathbf{b}$ is said to have *ideal arithmetic cross-correlation* if $C_{\mathbf{a},\mathbf{b}}(\tau) = 0$ for every τ. In this paper we discuss families $\mathcal{S}$ of periodic binary sequences such that every pair $\mathbf{a}, \mathbf{b} \in \mathcal{S}$ of elements has ideal arithmetic cross-correlation. Further background on 2-adic numbers can be found in a book by

*Received by the editors March 4, 2002; accepted for publication (in revised form) January 20, 2004; published electronically July 20, 2004. Parts of this research were performed while the second and fourth authors were visitors at the Institute for Mathematical Sciences, National University of Singapore.

http://www.siam.org/journals/sidma/18-1/40342.html

†The Institute for Advanced Study, School of Mathematics, Princeton, NJ 08540 (goresky@ias.edu, www.math.ias.edu/∼goresky/). This author's research was partially supported by NSF grant 0002693.

‡Department of Computer Science, 779A Anderson Hall, University of Kentucky, Lexington, KY, 40506-0046, and the Institute for Advanced Study, School of Mathematics (klapper@cs.uky.edu). This author's research was partially supported by NSF grant 9980429.

§Department of Mathematics, Queen's University, Kingston, ON K7L 3N6, Canada, and the Institute for Advanced Study, School of Mathematics (murty@mast.queensu.ca).

¶Department of Computing, Macquarie University, NSW 2109, Australia (igor@ics.mq.edu.au).

Koblitz [15] and in a paper by Klapper and Goresky [14]. Further background on arithmetic correlations can be found in another paper by Goresky and Klapper [8].

The existence of such families is surprising in light of the Welch bound [11], which states that if $\mathcal{S}$ is a collection of S cyclically distinct binary sequences of period T, then there exist $\mathbf{a}, \mathbf{b} \in \mathcal{S}$ and a shift τ such that the (usual) periodic cross-correlation

$$c_{\mathbf{a},\mathbf{b}}(\tau) = \sum_{i=0}^{T-1} (-1)^{a_i - b_{\tau+i}}$$

satisfies

$$c_{\mathbf{a},\mathbf{b}}(\tau) \geq T\sqrt{\frac{S-1}{ST-1}}.$$

Thus the Welch bound can be broached, by replacing the usual cross-correlation c with the arithmetic cross-correlation C.

The particular sequences of interest are called *long* sequences or ℓ-sequences; they are in many ways analogous to the binary m-sequences. Let q be a prime number such that 2 is a primitive root modulo q (meaning that the powers of 2 account for all the nonzero elements in $\mathbf{Z}/(q)$). Then a binary ℓ-sequence is any sequence of the form

$$a_i = (A2^{-i} \bmod q) \mod 2, \tag{1.1}$$

where $A \in \mathbf{Z}/(q)$ is nonzero. This equation means the following. Let $b = 2^{-1} \in \mathbf{Z}/(q)$ be the inverse of 2, modulo q. First compute Ab^i and reduce modulo q to obtain a number between 0 and $q-1$. Then reduce this number modulo 2. The sequence (1.1) is strictly periodic with period $q-1$, and different choices of A give rise to cyclic shifts of the same "base" sequence $a_i = (2^{-i} \bmod q) \mod 2$. (Up to a shift, this sequence may be described as the coefficient sequence of the 2-adic expansion of the fraction $-1/q$; it is also the reverse of the binary expansion of the fraction $1/q$.) These sequences have been studied since Gauss [7]. The related sequences $(g^i \bmod q) \mod \ell$ are used in the Digital Signature Standard and are important for an attack due to Nguyen and Shparlinski [18].

Such ℓ-sequences may be generated using feedback with carry shift registers as described in [13, 14], where their role in stream ciphers was investigated; see also [5] and [16]. This method of generating ℓ-sequences (and their mod p generalizations) was discovered independently by Marsaglia and Zaman [17] in special cases and by Couture and L'Ecuyer [4] in general, who proposed using them as pseudorandom number generators for Monte Carlo simulations.

These ℓ-sequences exhibit important randomness properties. In [1] it was shown that they have perfect distribution properties: for any $d < \log q$, every d-tuple of bits occurs either $\lceil (q-1)/d \rceil$ or $\lfloor (q-1)/d \rfloor$ times in a single period, where hereafter we use $\ln z$ and $\log z$ to denote the natural and binary logarithms of $z > 0$, respectively.

Let $\mathbf{x} = \mathbf{a}^d$ be the d-fold decimation of $\mathbf{a}$. That is, $x_i = a_{di}$. We say this decimation is *allowable* if d is relatively prime to $q-1$. In [8] it was shown that cyclically distinct allowable decimations of a single ℓ-sequence have ideal arithmetic cross-correlation; see the following theorem.

THEOREM 1.1. *Let q be a prime number such that 2 is a primitive root modulo q and let $\mathbf{a} = (a_0, a_1, \dots)$ be an ℓ-sequence of period $q-1$. Let $\mathbf{x} = \mathbf{a}^d$ and $\mathbf{y} = \mathbf{a}^e$*

be allowable decimations of $\mathbf{a}$ *by* d *and* e*, respectively. Suppose* $\mathbf{x}$ *and* $\mathbf{y}$ *are cyclically distinct. Then for any shift* τ *the arithmetic crosscorrelation vanishes:* $C_{\mathbf{x},\mathbf{y}}(\tau) = 0$.

This theorem provides a family S of periodic sequences with ideal arithmetic cross-correlation. Unfortunately, however, even if $d \neq e$, the sequences $\mathbf{x}$ and $\mathbf{y}$ may fail to be cyclically distinct. On the basis of extensive experimental evidence the following conjecture was made [8].

CONJECTURE 1.2. *If* $q > 13$ *is prime,* 2 *is primitive modulo* q*, and* $\mathbf{a}$ *is an* ℓ*-sequence based on* q*, then every pair of allowable decimations of* $\mathbf{a}$ *is cyclically distinct.*

It is relevant to remark that by the celebrated result of Hooley [12], under the extended Riemann hypothesis, 2 is primitive for a set of primes of positive relative density.

If Conjecture 1.2 holds for a prime q, then the resulting family $\mathcal{S}$ consists of $\varphi(q-1)$ distinct elements with ideal arithmetic correlation (where φ is the Euler function). We have verified this conjecture for all primes $q < 2,000,000$. It can be restated in very elementary terms as follows.

Let $q > 13$ be a prime number such that 2 is primitive mod q. Let E be the set of even integers $0 \leq e \leq q-1$. Fix A with $1 \leq A \leq q-1$. Suppose the mapping $x \mapsto Ax^d \mod q$ preserves (but permutes the elements within) the set E. Then $d = 1$ and $A = 1$. The equivalence between these two statements follows from the fact that $\mathbf{a}^d$ and $\mathbf{a}^e$ are cyclically distinct if and only if $\mathbf{a}$ and $\mathbf{a}^h$ are cyclically distinct, where $h = d(e^{-1} \bmod q-1)$.

2. Previous and current results. Conjecture 1.2 has turned out to be surprisingly resistant to proof. Suppose q is prime, 2 is a primitive root mod q, $\mathbf{a}$ is an ℓ-sequence with prime connection integer $q > 13$, and d is relatively prime to $q-1$. In [8] and [9] the following was shown.

THEOREM 2.1. *Suppose*

(i) *either* $d = -1$ (*or, equivalently,* $d = q-2$)*;*
(ii) *or* $q \equiv 1 \bmod 4$ *and* $d = (q+1)/2$*;*
(iii) *or*

$$1 < d \leq \frac{(q^2-1)^4}{2^{16}q^7(\ln q + 2)^4} \sim \frac{q}{(16 \ln q)^4}.$$

Then the decimation $\mathbf{a}^d$ *is cyclically distinct from* $\mathbf{a}$.

In this paper we give the complete proof of Theorem 2.1 (iii) (which was only sketched in [9]) and we improve substantially on this bound by removing the $\ln q$ factors. Let $\mathbf{a}, q, d$ be as above.

THEOREM 2.2. *If* $d > 1$ *and*

$$d \leq \frac{(q^2-1)^4}{2^{24}q^7}$$

or if $d < 0$ *and*

$$|d| \leq \frac{(q^2-1)^4}{2^{25}q^7},$$

then the decimation $\mathbf{a}^d$ *is cyclically distinct from* $\mathbf{a}$.

Finally, we show that, asymptotically for large q, the collection of counterexamples to Conjecture 1.2 is a vanishingly small fraction of the set of all allowable decimations.

THEOREM 2.3. *For any fixed $\varepsilon > 0$ there is a constant $C_0(\varepsilon) > 0$ depending only on ε, such that there are at most $C_0(\varepsilon)q^{2/3+\varepsilon}$ decimations of an ℓ-sequence $\mathbf{a}$ with connection number q that are cyclic permutations of $\mathbf{a}$.*

Finally, we show that for certain q, Conjecture 1.2 holds.

THEOREM 2.4. *If $q = 2p+1 = 8r+3$ with q, p, and r prime, and if 2 is primitive mod q, then Conjecture 1.2 holds for q sufficiently large.*

3. Preliminary estimates. Throughout this paper we fix a primitive qth root of unity, say, $\xi = e^{2\pi i/q} \in \mathbf{C}$. Define

$$S_d(a,b) = \sum_{x=0}^{q-1} \xi^{ax^d+bx}.$$

Then $S_d(0,0) = q$, $S_d(a,0) = S_d(0,b) = 0$ if a and b are nonzero, and

$$S_d(1,b) = S_d(\lambda^d, \lambda b) \tag{3.1}$$

for any $\lambda \neq 0$ (and for any b).

We need the following bound on the fourth moment of the sums $S_d(a,b)$ averaged over b; see [9].

LEMMA 3.1. *If $a \neq 0$ and $d > 1$, then*

$$\sum_{b=0}^{q-1} |S_d(a,b)|^4 \leq (d-1)q^3.$$

Proof. The proof follows the method of Davenport and Heilbronn [6]. Let $R(w,t)$ denote the number of solutions to the system of congruences

$$\begin{aligned} x + y &\equiv w \bmod q, \\ x^d + y^d &\equiv t \bmod q. \end{aligned}$$

By solving for y using the first equation, we can reduce this to a single equation of degree $d-1$. (Since $d > 1$ is odd, the terms involving x^d cancel out.) Such an equation has at most $d-1$ solutions unless $w = t = 0$, when it has q solutions. Also, $R(w,t) = 0$ if one but not both of w and t is zero. Thus we have

$$\sum_{w=1}^{q-1}\sum_{t=1}^{q-1} R(w,t) = \sum_{w=0}^{q-1}\sum_{t=0}^{q-1} R(w,t) - q = q^2 - q$$

and

$$\sum_{w=0}^{q-1}\sum_{t=0}^{q-1} R^2(w,t) \leq (d-1)\sum_{w=1}^{q-1}\sum_{t=1}^{q-1} R(w,t) + q^2 = dq^2 - (d-1)q.$$

Therefore the sum

$$T = \sum_{a=0}^{q-1}\sum_{b=0}^{q-1} |S_d(a,b)|^4$$

is given by

$$T = \sum_{a=0}^{q-1}\sum_{b=0}^{q-1}\sum_{x_1=0}^{p-1}\sum_{x_2=0}^{p-1}\sum_{x_3=0}^{p-1}\sum_{x_4=0}^{p-1} \xi^{a(x_1^d+x_2^d-x_3^d-x_4^d)+b(x_1+x_2-x_3-x_4)},$$

which is q^2 times the number of solutions (x_1, x_2, x_3, x_4) to the system

$$x_1 + x_2 \equiv x_3 + x_4 \bmod q,$$
$$x_1^d + x_2^d \equiv x_3^d + x_4^d \bmod q$$

with $0 \le x_1, x_2, x_3, x_4 \le q-1$. Counting the number of pairs (x_1, x_2) and (x_3, x_4) independently gives

$$\sum_{a=0}^{q-1}\sum_{b=0}^{q-1} |S_d(a,b)|^4 = T = q^2 \sum_{w=0}^{q-1}\sum_{t=0}^{q-1} R^2(w,t) \le dq^4 - (d-1)q^3. \tag{3.2}$$

The terms for which $a = 0$ contribute the quantity

$$\sum_{b=0}^{q-1} |S_d(0,b)|^4 = q^4.$$

Thus

$$\sum_{a=1}^{q-1}\sum_{b=0}^{q-1} |S_d(a,b)|^4 \le (d-1)(q^4 - q^3).$$

Since d is relatively prime to $q-1$, the mapping $x \mapsto x^d$ is a permutation; hence (3.1) gives

$$\begin{aligned}\sum_{b=0}^{q-1} |S_d(a,b)|^4 = \sum_{b=0}^{q-1} |S_d(1,b)|^4 &= \frac{1}{q-1}\sum_{\lambda=1}^{q-1}\sum_{b=0}^{q-1} |S_d(\lambda^d, \lambda b)|^4 \\ &\le \frac{1}{q-1}\sum_{u=0}^{q-1}\sum_{v=0}^{q-1} |S_d(u,v)|^4 \le (d-1)q^3.\end{aligned} \tag{3.3}$$

This completes the proof of Lemma 3.1. □

Let $E = \{0, 2, \dots, q-1\} \subset \mathbf{Z}/(q)$ denote the set of "even" elements. For any $b \in \mathbf{Z}$ define

$$\sigma_d(b) = \sum_{x \in E} \xi^{bAx^d} = \sum_{x=0}^{(q-1)/2} \xi^{bA2^d x^d}.$$

Then $\sigma_d(0) = |E| = (q+1)/2$.

LEMMA 3.2. *For any $b \ne 0$ we have*

$$|\sigma_d(b)| \le \frac{2^{14/4}}{\pi}(d-1)^{1/4}q^{3/4} + 4\ln q + 4 < 2^3(d-1)^{1/4}q^{3/4}.$$

Proof. Davenport and Heilbronn [6] gave estimates on certain exponential sums. If we let

$$F(n) = \sum_{x=0}^{q-1} \xi^{f(x)+nx},$$

where n is an integer, then their Lemma 4 says that for any m,

$$\sum_{x=0}^{m} \xi^{f(x)} = \frac{m}{q}F(0) + O\left(\sum_{n=1}^{q-1} \frac{1}{n}(|F(n)| + |F(-n)|)\right) + O(\ln q).$$

Let us take $f(x) = ax^d$ (where $a = bA2^d$) and $m = (q-1)/2$. Then $F(0) = 0$ and $F(n) = S_d(a, n)$. By carefully examining Davenport and Heilbronn's proof, one sees that the constant on the first big-O is $2/\pi$ and the second big-O can be replaced by $4\ln q + 4$. In other words,

$$|\sigma_d(b)| \le \frac{2}{\pi}\left(\sum_{n=1}^{q-1} \frac{1}{n}(|S_d(a, n)| + |S_d(a, -n)|)\right) + 4\ln q + 4. \tag{3.4}$$

Applying Hölder's inequality to Lemma 3.1 gives

$$\sum_{n=1}^{q-1} \frac{1}{n}|S_d(a, n)| \le 4^{3/4}(d-1)^{1/4}q^{3/4}.$$

The same bound applies to the sum using $S_d(a, -n)$ in place of $S_d(a, n)$. The lemma follows. □

4. Proof of Theorem 2.1 (iii). Although Theorem 2.2 gives a better estimate than Theorem 2.1 (iii), we briefly include our original proof of it because it illustrates a technique which may some day be refined so as to give an even better estimate. As in the previous sections we suppose that q is a prime number, that 2 is primitive modulo q, and that d is relatively prime to $q-1$. Again let $E = \{0, 2, \ldots, q-1\} \subset \mathbf{Z}/(q)$ be the set of even numbers. Define

$$f_E(x) = \begin{cases} 1 & \text{if } x \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Its Fourier transform is given by

$$\hat{f_E}(b) = \frac{1}{q}\sum_{c=0}^{q-1} f_E(c)\xi^{-bc}.$$

By the Fourier inversion formula we have

$$f_E(a) = \sum_{b=0}^{q-1} \hat{f_E}(b)\xi^{ba}.$$

Now assume that the mapping $x \mapsto Ax^d$ preserves (but permutes the elements within) the set E. Then

$$\sum_{x\in E} f_E(Ax^d) = \sum_{b=0}^{q-1} \hat{f_E}(b) \sum_{x\in E} \xi^{bAx^d} = \sum_{b=0}^{q-1} \hat{f_E}(b)\sigma_d(b).$$

The left-hand side equals $|E| = (q+1)/2$ because if $b = 0$, then $\hat{f_E}(b) = (q+1)/(2q)$ and $\sigma_d(b) = |E| = (q+1)/2$. Thus

$$\frac{q^2-1}{4q} = \left|\sum_{b=1}^{q-1} \hat{f_E}(b)\sigma_d(b)\right| \le \left(\sum_{b=1}^{q-1} |\hat{f_E}(b)|\right) \max_{b\ne 0} |\sigma_d(b)|.$$

We need the following lemma; see [9].

LEMMA 4.1. *The following inequality holds:*

$$\sum_{b=1}^{q-1} |\hat{f_E}(b)| \le 1 + \frac{1}{2}\ln\left(\frac{q-3}{2}\right) < \frac{\ln q + 2}{2}.$$

Combining this estimate with Lemma 3.2 gives

$$d > \frac{(q^2-1)^4}{2^{16}q^7(\ln q + 2)^4},$$

which completes the proof of Theorem 2.1. □

5. Proof of Theorem 2.2. In this section we use the technique for obtaining bounds from exponential sums that has been used by several authors (for example, see [3]).

As in the preceding sections, let E be the set of even integers between 0 and $q-1$ and assume that the conclusion of Theorem 2.2 is false. In other words, assume that $Ax^d \in E$ for every $x \in E$. Let W denote the set of integers between 0 and $\lfloor (q-2)/4 \rfloor$ and let $s = 2\lfloor (q-1)/4 \rfloor + 1$. It follows that the congruence

$$Ax^d \equiv 2(u-v) + s \bmod q, \qquad x \in E,\ u, v \in W,$$

has no solutions. Therefore

$$\begin{aligned} 0 &= \frac{1}{q} \sum_{u,v\in W} \sum_{x\in E} \sum_{b=0}^{q-1} \xi^{b(Ax^d - 2(u-v) - s)} \\ &= \frac{1}{q} \sum_{b=0}^{q-1} \xi^{-bs} \sigma_d(b) \sum_{u,v\in W} \xi^{2b(u-v)} = \frac{1}{q} \sum_{b=0}^{q-1} \xi^{-bs} \sigma_d(b) \left| \sum_{u\in W} \xi^{bu} \right|^2. \end{aligned}$$

The term corresponding to $b = 0$ equals $|W|^2|E|/q$. Therefore

$$\frac{|W|^2|E|}{q} = -\frac{1}{q} \sum_{b=1}^{q-1} \xi^{-bs} \sigma_d(b) \left| \sum_{u\in W} \xi^{2bu} \right|^2 \le \frac{1}{q} \sum_{b=1}^{q-1} |\sigma_d(b)| \left| \sum_{u\in W} \xi^{2bu} \right|^2. \tag{5.1}$$

Using Lemma 3.2, we derive

$$\begin{aligned} \frac{|W|^2|E|}{q} &\le 2^3(d-1)^{1/4} q^{-1/4} \sum_{b=1}^{q-1} \left| \sum_{u\in W} \xi^{bu} \right|^2 \\ &\le 2^3(d-1)^{1/4} q^{-1/4} \sum_{b=0}^{q-1} \left| \sum_{u\in W} \xi^{2bu} \right|^2 \\ &= 2^3(d-1)^{1/4} q^{-1/4} (q|W|) = 2^3(d-1)^{1/4} q^{3/4} |W|. \end{aligned} \tag{5.2}$$

Since $|W| \ge (q-1)/4$ we obtain

$$d - 1 \ge \frac{|W|^4|E|^4}{2^{12}q^7} \ge \frac{(q^2-1)^4}{2^{24}q^7}.$$

A similar argument can be made for negative d. Suppose $d = -e$ with $e > 0$. The system of congruences

$$\begin{aligned} x + y &\equiv w \bmod q, \\ x^d + y^d &\equiv t \bmod q \end{aligned}$$

is equivalent to the single congruence

$$(w - x)^e + x^e \equiv t(w - x)^e x^e,$$

which has at most $2e$ solutions. This fact can be used in the proof of Lemma 3.1, which now says, If $a \neq 0$ and if $d = -e < 0$, then

$$\sum_{b=0}^{q-1} |S_d(a, b)|^4 \leq 2eq^3.$$

Lemma 3.2 then reads as follows: for any $b \neq 0$,

$$|\sigma_d(b)| < 2^3(2e)^{1/4}q^{3/4}.$$

Now go back to the beginning of the proof of Theorem 2.2, using this estimate for $|\sigma_d(b)|$ in (5.1). The factors $(d - 1)$ become replaced by $2e$, which leads to the conclusion

$$2e > \frac{(q^2 - 1)^4}{2^{24}q^7}.$$

This completes the proof of Theorem 2.2. □

6. Proof of Theorem 2.3. It follows from the proof of Theorem 8 of Canetti et al. [2] that for any fixed $\varepsilon > 0$, the sum of the numbers of solutions to the systems of congruences

$$\begin{aligned} x_1 + x_2 &\equiv x_3 + x_4 \bmod q, \\ x_1^d + x_2^d &\equiv x_3^d + x_4^d \bmod q \end{aligned} \tag{6.1}$$

over all $d = 0, 1, \ldots, q - 2$ is bounded by a function in $O\left(q^{11/3+\varepsilon}\right)$. Let D be the set of d such that the system of congruences (6.1) has more than $q^{3-\varepsilon}$ solutions. Then the cardinality of D satisfies $|D| \in O(q^{2/3+\varepsilon})$.

We claim that if there exists $A \neq 0$ such that $x \mapsto Ax^d$ preserves the set E of even elements, then $d \in D$. Suppose the contrary: fix such an A and d, and suppose that $d \notin D$. Then the number of solutions to (6.1) is no more than $q^{3-\varepsilon}$. Thus by (3.2) we obtain the bound

$$\sum_{a=0}^{q-1}\sum_{b=0}^{q-1} |S_d(a, b)|^4 = T \in O(q^{5-\varepsilon}).$$

Hence, as in (3.3), we conclude that

$$\sum_{b=0}^{q-1} |S_d(a, b)|^4 \leq \frac{1}{q-1}\sum_{a=0}^{q-1}\sum_{b=0}^{q-1} |S_d(a, b)|^4 \in O(q^{4-\varepsilon}),$$

and thus $|S_d(a,b)| \in O(q^{1-\varepsilon/4})$ for every $b = 0, \ldots, q-1$. Hence

$$\sum_{n=1}^{q-1} \frac{1}{n} |S_d(a,n)| \in O(q^{1-\varepsilon/4} \ln q)$$

and this estimate can be used in (3.4) to give

$$|\sigma_d(b)| \in O(q^{1-\varepsilon/4} \ln q).$$

Now return to the beginning of the proof of Theorem 2.2 and use this estimate in (5.1). Then (5.2) becomes

$$\frac{|W|^2|E|}{q} \in O(q^{1-\varepsilon/4}|W| \ln q),$$

which is impossible. □

7. Proof of Theorem 2.3 and other large sets of distinct decimations. Let G denote the set of decimations of an ℓ-sequence $\mathbf{a}$ with connection integer q. The set G is a multiplicative group isomorphic to $(\mathbf{Z}/(\varphi(q)))^*$. Let H denote the set of decimations that are cyclic shifts of $\mathbf{a}$. Then H is a subgroup of G.

Let $\Delta \subseteq G$ be a set of representatives for G/H, with $1 \in \Delta$. That is, for each coset dH, there is exactly one element in $dH \cap \Delta$.

LEMMA 7.1. *The set $D = \{\mathbf{a}^d : d \in \Delta\}$ is a set of $|\Delta|$ pairwise cyclically distinct decimations with ideal arithmetic correlations.*

Proof. Suppose that $\mathbf{a}^d$ is a cyclic permutation of $\mathbf{a}^e$, with $d, e \in \Delta$. Then $\mathbf{a}^{de^{-1}}$ is a cyclic permutation of $\mathbf{a}$. Thus $de^{-1} \in H$, and by the hypotheses on Δ, $de^{-1} = 1$. That is, $d = e$. □

COROLLARY 7.2. *Let $\mathbf{a}$ be an ℓ-sequence with connection integer q. For any fixed $\varepsilon > 0$ there are constants $C_1(\varepsilon), C_2(\varepsilon) > 0$ depending only on ε, such that the following statements hold:*

(i) *The set $\{\mathbf{a}^d : d \in \Delta\}$ is a set of at least*

$$\frac{|G|}{|H|} \geq C_1(\varepsilon) q^{1/3-\varepsilon}$$

cyclically distinct sequences with ideal arithmetic correlations.

(ii) *If $\varphi(\varphi(q))$ has a prime factor $r > C_2(\varepsilon) q^{2/3+\varepsilon}$, then $\{\mathbf{a}^d : d \in \Delta\}$ is a set of at least r cyclically distinct sequences with ideal arithmetic correlations.*

Proof. The first statement follows from Theorem 2.3 and the lower bound

$$\frac{q}{\varphi(\varphi(q))} \in O((\ln \ln q)^2);$$

see Theorem 328 in [10].

We know that $|H|$ and $|\Delta|$ divide $|G| = \varphi(\varphi(q))$. Take $C_2(\varepsilon) = C_1(\varepsilon)^{-1}$. If $\varphi(\varphi(q))$ has a prime factor $r > C_2(\varepsilon) q^{2/3+\varepsilon}$, then r cannot divide $|H|$, and so $\{\mathbf{a}^d : d \in \Delta\}$ is a set of at least $r > C_2(\varepsilon) q^{2/3+\varepsilon}$ cyclically distinct sequences. This proves the second statement. □

Now consider integers q with the special form $q = 2p + 1 = 2^k r + 3$ with p and r prime. In this case $\varphi(\varphi(q)) = \varphi(2p) = p - 1 = 2^{k-1}r$. If k is small enough (for example, $k = 3$ as in the formulation of Theorem 2.4) and q is large enough, then r does not divide $|H|$. This implies that $|H|$ is a power of 2. We also have $|G| = (\mathbf{Z}/(2p))^* = (\mathbf{Z}/(p))^*$, which is a cyclic group. Thus either $|H|$ is trivial or H contains -1. However, we have already shown in Theorem 2.1 that $-1 \notin H$. It follows that all decimations are cyclically distinct. This proves Theorem 2.4. □

In fact, as we have just seen, in Theorem 2.4 one can consider more general families of primes.

There is a heuristic for the density of such primes q. Artin's conjecture, which is true if the extended Riemann hypothesis is true [12], implies that there are at least $AN/\ln N$ primes $q < N$ such that 2 is primitive modulo q. (The constant A is known as Artin's constant and is about .3739558.) Of these, we expect about $1/\ln N$ to satisfy $q = 2p + 1$ with p prime. If p is congruent to 3 modulo 4, then q is congruent to 7 modulo 8, which would imply that 2 is a quadratic residue, hence, not primitive. Thus it must be the case that p is congruent to 1 modulo 4, so $q = 8r + 3$ for some r. We expect r to be prime with probability about $1/\ln N$, so we expect more than $AN/(\ln N)^2$ primes less than N that satisfy all these requirements. Experimentation shows that this estimate is a bit conservative for $N < 1,000,000,000$.

8. Conclusions. We have significantly increased the set of decimations of an ℓ-sequence $\mathbf{a}$ that are known to be cyclically distinct from $\mathbf{a}$. For sufficiently long ℓ-sequences we have shown that there is a large family of cyclically distinct decimations. In some special cases we have in fact shown that all decimations are cyclically distinct.

REFERENCES

[1] L. Blum, M. Blum, and M. Shub, *A simple unpredictable pseudo-random number generator*, SIAM J. Comput., 15 (1986), pp. 364–383.

[2] R. Canetti, J. B. Friedlander, S. Konyagin, M. Larsen, D. Lieman, and I. E. Shparlinski, *On the statistical properties of Diffie–Hellman distributions*, Israel J. Math., 120 (2000), pp. 23–46.

[3] J. H. H. Chalk, *Polynomial congruences over incomplete residue systems modulo k*, Proc. Kon. Ned. Acad. Wetensch., A92 (1989), pp. 49–62.

[4] R. Couture and P. L'Ecuyer, *On the lattice structure of certain linear congruential sequences related to AWC/SWB generators*, Math. Comp., 62 (1994), pp. 799–808.

[5] T. W. Cusick, C. Ding, and A. Renvall, *Stream Ciphers and Number Theory*, Elsevier, Amsterdam, 1998.

[6] H. Davenport and H. Heilbronn, *On an exponential sum*, Proc. London Math. Soc., 41 (1936), pp. 449–453.

[7] C. F. Gauss, *Disquisitiones Arithmeticae*, 1801. Yale University Press, New Haven, CT, 1966 (in English).

[8] M. Goresky and A. Klapper, *Arithmetic cross-correlations of FCSR sequences*, IEEE Trans. Inform. Theory, 43 (1997), pp. 1342–1346.

[9] M. Goresky, A. Klapper, and R. Murty, *On the distinctness of decimations of ℓ-sequences*, in Sequences and Their Applications—SETA '01, T. Helleseth, P. V. Kumar, and K. Yang, eds., Discrete Math. Comput. Sci., Springer-Verlag, New York, 2002.

[10] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers*, Oxford Univ. Press, Oxford, UK, 1979.

[11] T. Helleseth and V. Kumar, *Sequences with low correlation*, in Handbook of Coding Theory, V. Pless and W. Huffman, eds., North–Holland Elsevier, Amsterdam, 1998.

[12] C. Hooley, *On Artin's conjecture*, J. Reine Angew. Math., 225 (1967), pp. 209–220.

[13] A. Klapper and M. Goresky, *2-adic shift registers*, in Fast Software Encrypt., Cambridge Security Workshop, R. Anderson, ed., Lecture Notes in Comput. Sci. 809, Springer-Verlag, New York, 1994.

[14] A. Klapper and M. Goresky, *Feedback shift registers, combiners with memory, and 2-adic*

span, J. Cryptology, 10 (1997), pp. 111–147.

[15] N. Koblitz, *p-Adic Numbers, p-Adic Analysis, and Zeta Functions*, Grad. Texts in Math. 58, Springer-Verlag, New York, 1984.

[16] J. C. Lagarias, *Pseudorandom number generators in cryptography and number theory*, Proc. Sympos. Appl. Math., 42 (1990), pp. 115–143.

[17] G. Marsaglia and A. Zaman, *A new class of random number generators*, Ann. Appl. Probab., 1 (1991), pp. 462–480.

[18] P. Q. Nguyen and I. E. Shparlinski, *The insecurity of the Digital Signature Algorithm with partially known nonces*, J. Cryptology, 15 (2002), pp. 151–176.

SIAM J. DISCRETE MATH.
Vol. 18, No. 1, pp. 141–158

EXACT WORD-RUN STATISTICS IN RANDOM ORDERINGS*

GABRIEL A. SCHACHTEL†

Abstract. An exact formula for the number of arrangements of a fixed collection of letters containing a specified set of runs is generalized from runs of letters to runs of words. The generalized formula is applicable, provided the specified run-lengths and word-types meet certain restrictions. The result is of special interest in the analysis of biomolecular sequences where such word-runs are known as microsatellites.

Key words. run test, word, sequence analysis, cluster, short tandem repeat, microsatellite

AMS subject classifications. 05A05, 68R15, 62P10, 92D20

DOI. 10.1137/S0895480101390576

1. Introduction. Different probabilistic models are utilized to measure the statistical significance of repeating patterns in letter sequences. Assuming a fixed pool $\mathcal{L}$ of letters, this paper presents a shuffling model in order to determine the exact number of arrangements of $\mathcal{L}$ containing a specified set of word-runs. A known run-formula by Morris, Schachtel, and Karlin (1993) allows one to specify any number of desired letter-runs of different types and minimum lengths. This result will be extended here from letter- to word-runs. Our generalization, however, is restricted to a certain class of so-called admissible sets of words and run-lengths.

The first comprehensive book on *runs* dates back to Bortkiewicz (1917), who studied *unconditional models*, i.e., sequences whose composition is not fixed a priori. Additional exact results on runs are provided in Mood (1940), asymptotic results in Erdös and Renyi (1970), Guibas and Odlyzko (1980), and Foulser and Karlin (1987), to name a few. *Conditional models* are less common; results can be found, for instance, in Mood (1940) and in Bradley (1968). More recently Balakrishnan and Koutras (2002) published a monograph on runs and scans with applications.

An early comprehensive publication on the combinatorics of *words* was compiled by Lothaire in 1983 and supplemented in 2002. Reinert, Schbath, and Waterman (2000) reviewed the current state of research in the area, giving an overview on the statistical and probabilistic properties concerning location and counts of words in letter sequences, introducing major aspects of the field, providing relevant techniques, warning of pitfalls associated with the analysis of words, and citing many related results and references.

With the accumulation of biomolecular sequences in the 1980s, the search for statistically significant patterns in sequences motivated many probabilistic studies. Runs of various kinds are of interest in this context, since a large fraction of genomic DNA is comprised of repetitive sequences. These so-called *tandem repeats* are of growing biological concern, because they are involved in a variety of regulatory, catalytic, immunological, and evolutionary processes. As markers in linkage analysis, DNA fingerprinting, and phylogenetic studies, they have also become valuable investigative

*Received by the editors June 11, 2001; accepted for publication (in revised form) November 30, 2003; published electronically August 26, 2004. This work was supported by the DFG Sonderforschungsbereich 299.

http://www.siam.org/journals/sidma/18-1/39057.html

†Biostatistik am FB 09, Interdisziplinäres Forschungszentrum (IFZ), Justus-Liebig-Universität, Giessen, Germany (gh69@agrar.uni-giessen.de).

tools in molecular genetics. Repeats of units of up to six nucleotides (i.e., characters over the alphabet {A, C, G, T}) are called *microsatellites* or *STRs* (short tandem repeats) and are found in many different sequence locations, e.g., in direct proximity to genes or within them; see Reddy and Housman (1997).

Much attention is currently focused on diseases involving trinucleotide repeats. Through mutation these repeats can expand dramatically and accumulate to over 40 times their nonpathological copy number; e.g., in *fragile X syndrome* the trinucleotide CGG is repeated 2000 times (healthy individuals carry only up to 52 copies). Such STR expansions are associated with about 20 inherited human diseases, such as *myotonic dystrophy* and *Huntington's disease*, both characterized by a substantially expanded CAG run (Sinden (1999)). The great biological and medical impact of STR disorders quickly prompted the development of efficient and sensitive algorithms for their detection, e.g., Karlin et al. (1988). More recently an automated program, *TRF* (tandem repeat finder) by Benson (1999) and the database *TRIPS* (tandem repeats in proteins) by Katti et al. (2000) were made available on the World Wide Web.

The vast amount of sequence data provided by modern sequencing technologies requires automated methods to identify biologically important features within those sequences. While statistical significance is neither necessary nor sufficient for biological significance, it is nevertheless a valuable indicator. The distinction between the randomness and nonrandomness of observed patterns can often serve as a benchmark to detect potentially interesting regions within biomolecular sequences and sort them out for further experimental investigation. In this paper we are concerned with sets of microsatellites encountered in a given sequence. In order to assess their statistical significance we will regard them as word-runs in random orderings of fixed length and letter composition, and derive the probability that any specified collection of word-runs will be present in such random orderings.

The remainder of the paper is organized as follows. In section 2 we briefly describe the main features of the counting approach for letter-runs and then illustrate the complications arising in the generalization from letter- to word-runs. In section 3 proper notation and terminology are introduced. In section 4 a class of admissible sets is defined, for which a uniqueness theorem is proved in section 5. Finally, section 6 provides the generalized formula, valid for any collection of word-runs obtained from an admissible set.

2. Complications associated with the generalization. Several word-properties such as overlap and self-overlap cause serious complications and prevent an immediate generalization of the Morris, Schachtel, and Karlin (1993) run-formula from letter- to word-runs. Before demonstrating these complications, let us first describe the counting methods exploited to derive the formula for letter-runs.

2.1. Counting method for letter-runs. We take as given a finite alphabet $\mathcal{A} = \{L_\alpha | \alpha = 1, 2, \ldots, \tau\}$ with $\tau \geq 2$ letter types L_α and an integer-valued, τ-dimensional vector $\vec{n} = (n_1, n_2, \ldots, n_\tau)$ with $N = \sum_{\alpha=1}^{\tau} n_\alpha$. Then $\mathcal{L}(\mathcal{A}; \vec{n})$ denotes a *letterpool* containing N letters from alphabet $\mathcal{A}$, namely n_1 letters of type L_1, n_2 of type $L_2, \ldots, n_\tau$ of type L_τ. Any distinguishable arrangement of *all* N members of the letterpool is called an *ordering* of $\mathcal{L}$. Let the sequence $Z = (z_1, z_2, \ldots, z_N)$ be one such ordering. We define a *letter-run* within Z as an uninterrupted subsequence $(z_m, z_{m+1}, \ldots, z_n)$ with $0 < m \leq n \leq N$ consisting of only one letter type bounded at either end by a letter of different type or by a boundary of the sequence (i.e., in our context, only maximal runs are considered). By this definition, the sequence BAAAABBB has (letter-) runs of B's only of lengths 1 and 3. We will sometimes refer

TABLE 2.1

Different configurations representing the same ordering. Suppose at least two runs(A) ≥ 2 *and one run*(C) ≥ 3 *are desired. Let* x = [AA] *and* y = [CCC] *be chunks representing an* A*-run of length* 2 *and a* C*-run of length* 3*, respectively. Rows* (1) *to* (4) *display different configurations of the same chunkpool, all corresponding to the ordering* (0)*, which must be acceptable since* (3) *and* (4) *are disjoint.*

(0)	**A**	**A**	**C**	**C**	**C**	**A**	**A**	**A**	**A**	**T**	**A**	**A**	**G**	*Original ordering*
(1)	[x]		[y]			A	[x]		A	T	A	A	G	*Nonendplaced configuration*
(2)	A	A	[y]			[x]			[x]	T	A	A	G	*Nonseparate configuration*
(3)	[x]		[y]			A	A	A	A	T	[x]		G	*Assignment alternatives*
(4)	[x]		[y]			A	A	[x]		T	A	A	G	

to runs of the same letter type as *separate* to emphasize that they must be separated by intervening letters of a different type.

Example 2.1. Let $\mathcal{A} = \{L_\alpha | \alpha = 1, 2, 3, 4\}$ be an alphabet of the $\tau = 4$ letter types $L_1 = A$, $L_2 = C$, $L_3 = G$, and $L_4 = \mathrm{T}$. And let $\mathcal{L}(\mathcal{A}; \vec{n})$ with $\vec{n} = (8, 3, 1, 1)$ be a letterpool containing altogether $N = 13$ letters, namely 8 A's, 3 C's, 1 G, and 1 T. Then $\mathcal{O}$ = AACCCAAAATAAG in row (0) of Table 2.1 is an ordering of $\mathcal{L}$ having two A-runs of length 2, one A-run of length 4, one C-run of length 3, one T- and one G-run each of length 1. – $\mathcal{O}'$ = ACAACAACAAATG is another ordering of the same letterpool $\mathcal{L}$.

Let $\mathcal{F}$ represent the set of conditions describing the required runs (i.e., their number, length, and letter type). The ordering Z is said to be *acceptable* (or $\mathcal{F}$-*acceptable*) if, for each run specified in $\mathcal{F}$, there exists in Z a run of this letter type and at least the specified length. Each run within Z may be used to satisfy only one run within the condition set $\mathcal{F}$; e.g., a run of length 4 does not satisfy the requirement for two runs of length 2, since they would not be separate.

To count the number of acceptable orderings, we will first count the number of distinguishable arrangements of a modified pool, called the $\mathcal{F}$-*pool* or the *chunkpool* of $\mathcal{F}$. For each run specified in $\mathcal{F}$, replace these letters with a new type of object called a *chunk*. Each chunk is an indivisible unit having the "length" and letter type attributes of the letters it replaces. The resulting pool of letters and chunks is called the chunkpool of $\mathcal{F}$. Arrangements of the chunkpool are called *configurations* to distinguish them from orderings of the full letterpool, $\mathcal{L}$. There is an obvious mapping of configurations to orderings, which replaces each chunk by its component individual letters. Although each acceptable ordering has at least one corresponding configuration (as we will show), the map is not one-to-one because some configurations map to unacceptable orderings and different configurations often map to the same ordering. For example, suppose we have a two-letter alphabet and a letter-pool, $\mathcal{L}$ of $n_1 = 7$ A's and $n_2 = 3$ B's, and we desire at least two separate runs of three or more A's. In a simplified notation, we will write this condition as $\mathcal{F} = \{\text{at least two runs(A)} \geq 3\}$. We use six of the A's to construct two chunks of three A's each, denoted by x's or by square brackets, e.g., x = [AAA]. Our $\mathcal{F}$-pool thus consists of two x chunks, one remaining A, and three B's, and each arrangement of this chunkpool is a configuration. The configuration (x, B, A, x, B, B) corresponds to the ordering (A, A, A, B, A, A, A, A, B, B), which is $\mathcal{F}$-acceptable since it has the desired runs (one of length 3 and one of length 4). The configurations (x, B, x, A, B, B) and (x, B, A, x, B, B), are different, but correspond to the same ordering. Another configuration (x, x, B,

A, B, B) corresponds to an ordering that is not $\mathcal{F}$-acceptable, since it contains only one run of three or more A's—in this case, a run of length 6.

As the above example shows, the natural correspondence between configurations and the orderings they determine is not one-to-one; nor is it surjective over all orderings, since there are many nonacceptable orderings that have no corresponding configuration. Consider the mapping that takes an ordering to its corresponding configurations. Intuitively, this can be visualized in terms of placing each chunk on a run of the correct letter type within the ordering without overlapping any two chunks. We say that each ordering is counted as many times as there are configurations corresponding to it. Since each acceptable ordering has, by definition, a collection of runs large enough to fit all of the chunks in the chunkpool, it is counted at least once. The number of configurations therefore exceeds the number of acceptable orderings, and this overcounting can be attributed to three mechanisms: (i) multiple ways of *positioning* a chunk within a longer run, (ii) multiple *occupancy* of two or more chunks within a single long run, and (iii) multiple ways for *assignment* of a number of chunks to a greater number of available runs.

To determine the number of acceptable orderings, we stepwise eliminate these three sources of overcounting as follows.

(i) *Positioning*. An acceptable ordering containing a run longer than required is counted multiple times corresponding to the number of ways the chunk can be positioned within the actual run. For example, if an x corresponds to a chunk of three A's, the ordering $\mathcal{O}_1 = (\mathrm{A},\ \mathrm{A},\ \mathrm{A},\ \mathrm{A},\ \mathrm{B})$ is counted by the configurations $\mathcal{C}_1 = (\mathrm{A},\ x,\ \mathrm{B})$ and $\mathcal{C}_2 = (x,\ \mathrm{A},\ \mathrm{B})$. To avoid this source of overcounting, we count only configurations in which no chunk is adjacent on its right to an individual letter of the same kind. We call these configurations *endplaced* (in our example, $\mathcal{C}_1$ is endplaced; $\mathcal{C}_2$ is not).

(ii) *Occupancy*. Some nonacceptable orderings have corresponding configurations in which more than one chunk is placed within a single run. For example, if two runs of three A's each are required, the ordering $\mathcal{O}_2 = (\mathrm{A},\ \mathrm{A},\ \mathrm{A},\ \mathrm{A},\ \mathrm{A},\ \mathrm{A},\ \mathrm{A},\ \mathrm{B})$ is not acceptable because it has only one separate run of length 3 or greater. However, $\mathcal{O}_2$ is counted by the configurations $\mathcal{C}_3 = (x,\ x,\ \mathrm{A},\ \mathrm{B})$, $\mathcal{C}_4 = (x,\ \mathrm{A},\ x,\ \mathrm{B})$, and $\mathcal{C}_5 = (\mathrm{A},\ x,\ x,\ \mathrm{B})$. Configurations in which no chunk is adjacent to another chunk of the same letter type are called *separate*. Configurations that are both endplaced and separate are called *disjoint*. Disjoint configurations cannot have more than one chunk to a run (we speak of runs in configurations, although technically we are referring to the runs in the corresponding ordering). If more than one chunk were placed on a run, the leftmost chunk would be adjacent on its right to an individual letter of the same type (in which case, it would not be endplaced) or to another chunk (in which case, it would not be separate). It follows that the ordering corresponding to a disjoint configuration must have a separate run for each chunk in the chunkpool and is therefore acceptable. Furthermore, an acceptable ordering, because it has a separate run for each chunk in the chunkpool, must have at least one corresponding disjoint configuration. We have thus shown that the set of disjoint configurations counts all acceptable orderings (one or more times each) and only acceptable orderings. The unacceptable ordering $\mathcal{O}_2$, for example, has no corresponding disjoint configuration ($\mathcal{C}_3$ and $\mathcal{C}_4$ are not endplaced, and $\mathcal{C}_5$ is not separate).

(iii) *Assignment*. Orderings with more than the required number of separate runs are counted by as many disjoint configurations as there are ways of assigning the chunks to the runs. If, for example, two runs of three A's are required, the

ordering (A, A, A, B, A, A, A, B, A, A, A) is counted three times corresponding to the configurations $\mathcal{C}_6 = (x,\ \mathrm{B},\ x,\ \mathrm{B},\ \mathrm{A},\ \mathrm{A},\ \mathrm{A})$, $\mathcal{C}_7 = (x,\ \mathrm{B},\ \mathrm{A},\ \mathrm{A},\ \mathrm{A},\ \mathrm{B},\ x)$, and $\mathcal{C}_8 = (\mathrm{A},\ \mathrm{A},\ \mathrm{A},\ \mathrm{B},\ x,\ \mathrm{B},\ x)$. A judiciously weighted sum of the numbers of disjoint configurations eliminates overcounting of type (iii) and gives the number of acceptable orderings. In the construction of this sum, we make use of the following standard equality:

$$\sum_{i \geq s} \Delta(i;s) \binom{k}{i} = 1 \qquad \text{for} \quad k \geq s \quad \text{and} \quad k \geq 0, \tag{2.1}$$

$$\text{where } \Delta(i;s) = \begin{cases} (-1)^{i+s} \cdot \binom{i-1}{s-1} & \text{for} \quad s > 0, \\ 1 & \text{for} \quad s \leq 0 \ \text{ and } \ i = 0, \\ 0 & \text{otherwise.} \end{cases}$$

To prove that the weighted sum of the number of disjoint configurations equals the number of acceptable orderings, Morris, Schachtel, and Karlin (1993) introduce G_m as the number of orderings of $\mathcal{L}$ with *exactly* m runs of desired length and letter type. If s is the requested number of runs with specified length and letter type, then the number $\mathrm{AO}(s)$ of acceptable orderings can be rewritten as the sum over all G_m with $m \geq s$ as follows:

$$\mathrm{AO(s)} = \mathrm{G}_s + \mathrm{G}_{s+1} + \mathrm{G}_{s+2} + \cdots. \tag{2.2}$$

Our counting method overcomes the three sources of overcounting by first discarding all nonendplaced and nonseparate (i.e., all nondisjoint) configurations and then utilizing the fact that a properly weighted sum over the number of disjoint configurations equals the sum of G_m-values, which in turn is equal to the number of acceptable orderings.

2.2. Problems arising with word-runs. The described counting strategy relies heavily on the concepts of (i) "endplacedness" and (ii) "separatedness" of configurations, as defined in section 2.1 for the case of letter-runs. And it requires (iii) that the G_m-values of (2.2) be well defined, which presumes that for each ordering the precise number and maximal length of all specified runs can be ascertained unambiguously. While these three prerequisites are easily met with respect to letter-runs, their generalization to word-runs is not at all straightforward and presents various obstacles, which are exemplified briefly in the following.

First, we will illustrate the difficulty of defining endplaced word-runs. A *letter*-run is endplaced if it is followed to the right by a letter of different type. It is nonendplaced if the letter is of the same type.

Example (letter-runs). Given the ordering $\mathcal{O}$ = AACCCCAG, we may place a desired run of three C's in two ways within $\mathcal{O}$. In AACxAG the C-run (denoted by x) is endplaced, because the next letter to the right is different from "C". In AAxCAG, on the other hand, the C-run is followed by an additional "C" and is therefore nonendplaced.

For word-runs the situation is more intricate because a *word*-run is not necessarily endplaced, even when followed to the right by a different word.

Example (word-runs). Suppose a sequence CTATATATATATG, a word w = TATA, and a desired word-run ww = TATATATA of length 2. In CwwTATG the w-run is followed by a word different from w, namely TATG (or any of its prefixes).

Nevertheless, it does not seem useful for our purposes to view "ww" in this position as endplaced, since it is not at the very end of the available stretch of TA's. Only if shifted by two positions to the right does it make sense to consider it endplaced, as in CTAwwTG.

Next, we demonstrate the difficulty of ascertaining the exact number of available word-runs in a given ordering. Such a problem does not arise for letter-runs, where each run is of unique letter type and maximal length. The latter is easily determined by the run's well-defined boundaries, i.e., on either side by the first occurrence of a letter of different type (or of no letter).

Example (letter-runs). Looking in AGGCCCCCAGGGCCCCT for the number of available runs(C) ≥ 4 and runs(G) ≥ 3 immediately reveals 2 appropriate C- and 1 appropriate G-run.

Again, for word-runs the situation is more complicated because runs may overlap and therefore exhibit blurring borders. The intervening segments between neighboring runs can be used in some circumstances to lengthen either one or the other, resulting in ambiguities about the precise number and size of available runs.

Example (word-runs). Given the ordering $\mathcal{O}$ = TAGCAGCAGCAGCCC, we seek the exact number of runs(w_1) ≥ 2 and of runs(w_2) ≥ 3 for the words w_1 = GCA and w_2 = AGC. Depending on what run we assign to the stretch behind the T, we get either a run of three w_1, since $\mathcal{O}$ = TA$w_1w_1w_1$GCCC, or a run of four w_2, since $\mathcal{O}$ = T$w_2w_2w_2w_2$CC. Hence there is ambiguity as to how many desired w_1- and w_2-runs the given ordering harbors.

These few examples demonstrate the need to extend concepts (i) to (iii) to runs of words before applying the described counting method; i.e., we have to generalize the definitions of endplaced and separate configurations and to solve the ascertainment problem in order to ensure that G_m is well defined. To that end, however, a number of modifications with respect to notation and terminology will be necessary.

3. Notation and terminology. Let $\mathcal{A}$ be a finite alphabet with τ letters, and $\mathcal{L}(\mathcal{A};\vec{n})$ be a *letterpool* containing $N = \sum_{\alpha=1}^{\tau} n_\alpha$ letters from alphabet $\mathcal{A}$; see section 2.1. A short sequence $w = (a_1, a_2, \ldots, a_r)$ with $0 \leq r \ll N$ and $a_i \in \mathcal{A}$ is called a *word*, and its length r is denoted by $|w|$. For words of length $r = 0$ the symbol ε will be reserved.

Given an ordering $\mathcal{O} = (z_1, z_2, \ldots, z_N)$ of letterpool $\mathcal{L}$ and words u, x, and v such that $u = (z_1, z_2, \ldots, z_m)$, $x = (z_{m+1}, z_{m+2}, \ldots, z_n)$, and $v = (z_{n+1}, z_{n+2}, \ldots, z_N)$, then u, x, and v are called *factors* of $\mathcal{O}$ and we may write $\mathcal{O} = uxv$. The factors u and v are called the *prefix* and *suffix*, respectively, of $\mathcal{O}$. See, for example, Crochemore and Rytter (1994).

For a word $w \neq \varepsilon$ we define a *word-run* of type w within $\mathcal{O}$ as a factor of $\mathcal{O}$ consisting of a series of abutting reiterations of w. To distinguish the number of letters in a w-run from the number of w-iterations, the latter will be referred to as the run's *iteration length*; e.g., a w-run of iteration length $l \geq 0$, denoted by $w^l = ww\ldots w$, has a total length of $l \cdot |w|$ individual letters.

Next we formally postulate the demands concerning the word-types, iteration lengths, and number of occurrences of the various desired runs. Given a set $\mathcal{W} = \{w_1, w_2, \ldots, w_p\}$ of p words, and two integer-valued ragged arrays[1] $\mathbf{S}$ and $\mathbf{L}$ of equal dimensions and with p rows, the required runs are specified by the *condition set* $\mathcal{F}_4(\mathbf{S};\mathcal{W};\mathbf{L})$ = {at least s_{ij} runs(w_i) $\geq l_{ij}$ with $l_{i1} > l_{i2} > \cdots > l_{iq_i}$; $1 \leq i \leq p$;

[1] A ragged array is a degenerate matrix where rows can have different lengths.

$w_i \in \mathcal{W}\}$, where s_{ij}, $l_{ij} > 0$ are the elements of $\mathbf{S}$ and $\mathbf{L}$, respectively. Recall, as stated in section 2.1, that we consider only maximal runs.

The "lengths-array" $\mathbf{L}$ contains the iteration lengths of the desired runs of the word-types specified in $\mathcal{W}$. The lengths l_{iq_i} of the shortest desired w_i-runs are of particular interest, and therefore are summarized in $\vec{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_p)$ with $\lambda_i = l_{iq_i}$, called the vector associated with $\mathbf{L}$.

Example 3.1 (condition set). Take the set of desired runs in Table 2.1, where $p = 2$, $\mathcal{W} = \{w_1 = \mathrm{A}, w_2 = \mathrm{C}\}$, $\mathbf{S} = \binom{2}{1}$, and $\mathbf{L} = \binom{2}{3}$. The corresponding condition set is $\mathcal{F}_4(\mathbf{S}; \mathcal{W}; \mathbf{L}) = \{$at least $s_{11} = 2$ runs$(w_1 = \mathrm{A}) \geq l_{11} = 2$; and $s_{21} = 1$ run$(w_2 = \mathrm{C}) \geq l_{21} = 3\}$.

In accordance with $\mathcal{F}_1$ to $\mathcal{F}_3$ in Morris, Schachtel, and Karlin (1993), we denote condition sets by $\mathcal{F}_4$. Notice also that the number q_i of columns in ragged arrays depends on i.

An ordering $\mathcal{O}$ of $\mathcal{L}$ is said to be $\mathcal{F}_4$-*acceptable* (or just *acceptable*) if $\mathcal{O}$ contains the required number of runs of proper type and iteration length as specified in $\mathcal{F}_4$, that is, if it contains at least $\mathbf{S}$ runs$(\mathcal{W}) \geq \mathbf{L}$.

The following order relation between ragged arrays $\mathbf{S} = \|s_{ij}\|$ and $\mathbf{H} = \|h_{ij}\|$ of equal dimension will be needed later. Let $S_j^{(i)}$ be the partial sum of the first j elements of the ith row of $\mathbf{S}$, i.e., $S_j^{(i)} = \sum_{k=1}^{j} s_{ik}$ with $S_0^{(i)} = 0$, and $H_j^{(i)}$ defined accordingly. $\mathbf{H}$ is said to be greater than or equal to $\mathbf{S}$, denoted $\mathbf{H} \geq \mathbf{S}$, if $H_j^{(i)} \geq S_j^{(i)}$ holds for all i and j.

For the intended enumeration technique we will substitute all desired runs by (elementary) *chunks*, i.e., by indivisible units of the same iteration length and word-type as the runs they represent. Chunks will be indicated by square brackets to distinguish them from sequences of individual letters (and from words, which are just shortcuts for specific letter sequences).

Let $N_\alpha(w)$ denote the number of letters of type L_α contained in word w. Then a w-chunk $x = [w^l]$ of iteration length l consumes $l \cdot N_\alpha(w)$ individual letters of type L_α, and a total of $|x| = l \cdot \sum_{\alpha=1}^{p} N_\alpha(w) = l \cdot |w|$ letters.

Example. Take the word $w =$ ACCC of length $|w| = 4$ and the w-chunk $x = [w^2] = [ww] =$ [ACCCACCC] of iteration length $l = 2$; then $N_{\mathrm{A}}(w) = 1$, $N_{\mathrm{C}}(w) = 3$, and $|x| = 8$.

Given a letterpool $\mathcal{L}(\mathcal{A}; \vec{n})$ and a condition set $\mathcal{F}_4(\mathbf{S}; \mathcal{W}; \mathbf{L})$, one generates the associated *chunkpool* $\mathcal{F}_4[\mathbf{S}; \mathcal{W}; \mathbf{L}; \mathcal{L}]$ as follows: For each word-run specified in condition set $\mathcal{F}_4(\cdot)$ the corresponding letters in $\mathcal{L}$ are replaced by one (indivisible) elementary chunk of proper length and word-type. The resulting chunkpool $\mathcal{F}_4[\,\cdot\,]$ consists of elementary chunks (for each word-type $w_i \in \mathcal{W}$ exactly s_{ij} chunks of iteration lengths l_{ij}) and remaining individual letters (for each letter type $L_\alpha \in \mathcal{A}$ exactly $t_\alpha = n_\alpha - \sum_{i=1}^{p} \sum_{j=1}^{q_i} [s_{ij} \cdot l_{ij} \cdot N_\alpha(w_i)]$).

Remark. Obviously, the number of $\mathcal{F}_4$-acceptable orderings becomes 0 whenever the letterpool is too small to satisfy the entire demand for runs, e.g., whenever $t_\alpha < 0$ for at least one α.

Example 3.2. Take letterpool $\mathcal{L}$ and condition set $\mathcal{F}_4$ in Examples 2.1 and 3.1, respectively. Generate two chunks of type $x =$ [AA]; this will reduce the number of individual letters of type A from 8 to $t_1 = 4$. Similarly, one chunk of type $y =$ [CCC] will use up all three available C's in $\mathcal{L}$, i.e., $t_2 = 0$. Hence the resulting chunkpool $\mathcal{F}_4[\mathbf{S}; \mathcal{W}; \mathbf{L}; \mathcal{L}]$ will consist of x, x, y, A, A, A, A, G, T.

Arrangements of all chunks and all individual letters of a given chunkpool are called *configurations* (in order to set them apart from orderings).

Rows (1) to (4) in Table 2.1 display four different configurations for chunkpool $\mathcal{F}_4[\mathbf{S};\mathcal{W};\mathbf{L};\mathcal{L}]$ of Example 3.2, e.g., in row (3), configuration xyAAAATxG. Chunks are regarded as indivisible units. Ordering $\mathcal{O}'$ = ACAACAACAAATG in Example 2.1 therefore has no corresponding configuration with respect to chunkpool $\mathcal{F}_4[\mathbf{S};\mathcal{W};\mathbf{L};\mathcal{L}]$. Even though $\mathcal{O}'$ possesses three C's, they are not adjacent and thus do not constitute a run(C) ≥ 3. Hence chunk y cannot be "placed" properly in ordering $\mathcal{O}'$.

The building blocks of configurations are chunks and individual letters. For our purposes it is convenient, within configurations, to distinguish between chunks and stretches of individual letters. These (maximal) stretches of letters will be called *separators*. Such a distinction will enable us to represent configurations as sequences of chunks and separators. We also allow for empty separators and can thus represent any configuration by $\mathcal{C} = \sigma_1 x_1 \sigma_2 x_2 \ldots \sigma_n x_n \sigma_{n+1}$, where σ_i and x_i denote separators and chunks, respectively. While separators are regarded as maximal stretches and therefore must not occur adjacent to each other, chunks may abut one another (with empty separators).

Example. Take ordering $\mathcal{O}$ = ACACTGTGTGTGG and chunks x = [ACAC] and y = [TGTG]. The associated configuration $\mathcal{C}_1$ = [ACAC]TGTG[TGTG]G = xTGTG yG may then be represented by $\mathcal{C}_1 = \sigma_1 x_1 \sigma_2 x_2 \sigma_3$ with $\sigma_1 = \varepsilon$, σ_2 = TGTG, σ_3 = G, $x_1 = x$, and $x_2 = y$. $-\mathcal{C}_2$ = [ACAC][TGTG][TGTG]G = xyyG is another configuration associated with the same ordering $\mathcal{O}$, and the four separators are $\sigma_1 = \sigma_2 = \sigma_3 = \varepsilon$ and σ_4 = G.

Configurations are called *separate* if none of their chunks is adjacent to another chunk of identical word-type.

As Table 2.1 illustrates, the same ordering can be represented by different configurations. Each configuration $\mathcal{C}$, on the other hand, is uniquely associated with exactly one ordering $\mathcal{O}(C)$, namely with the corresponding sequence of individual letters. This sequence is obtained by resolving every chunk or word back into the subsequence of letters it represents. Applying this natural mapping, we establish an equivalence relation "$\sim$" on the family of configurations. Two configurations $\mathcal{C}_i$ and $\mathcal{C}_j$ are said to be *equivalent*, denoted by $\mathcal{C}_i \sim \mathcal{C}_j$, if they correspond to the same ordering, i.e., to the same sequence of individual letters.

Example. Take chunks x = [ACAC] and y = [TGTG] and configurations $\mathcal{C}_1$ = xTGTGyG and $\mathcal{C}_2 = xyy$G (see the last example). The two configurations are equivalent, i.e., $\mathcal{C}_1 \sim \mathcal{C}_2$, since both correspond to the same ordering, i.e., $\mathcal{O}(\mathcal{C}_1) = \mathcal{O}(\mathcal{C}_2)$ = ACACTGTGTGTGG. But configuration $\mathcal{C}_3 = x$GGTTyG corresponds to a different ordering, namely to $\mathcal{O}(\mathcal{C}_3)$ = ACACGGTTTGTGG; therefore $\mathcal{C}_1 \not\sim \mathcal{C}_3 \not\sim \mathcal{C}_2$.

Remark. We will regard "letter sequences" as "configurations without chunks" and apply the equivalence relation to compare configurations and letter-sequences, e.g., xTT $\sim$ ACACACTT with x = [ACACAC].

4. Admissible condition sets. So far we have been able to smoothly adapt the terminology and notation from letter- to word-runs. We now turn to the more demanding task of generalizing concepts (i) to (iii) of section 2. First, we will provide a definition for endplaced chunks (of words). Then we introduce cyclic chunks and principal units, which will substantially simplify the identification of disjoint configurations. Next, we will deal with overlapping and embeddable words and chunks, two sources of ambiguities concerning the ascertainment of the exact number of available word-runs. Finally, we combine these attributes to define a class of admissible condition sets, which subsequently allows us in section 5 to eliminate all the unwanted ambiguities.

4.1. Extenders and endplaced chunks. The essential feature of endplaced configurations is that none of their chunks can be shifted any further within their harboring run, because the run does not *extend* properly to the right.

DEFINITION 1. *Let x be a chunk. A word u is called an* extender *to x if $0 < |u| \le |x|$ and if a suitable word v exists such that $xu \sim vx$ holds. The word v is called the* extension complement *of u.*

Note that the complement v of an extender u is unique and of the same length as u (i.e., $|v| = |u|$); its sequence of letters is identical to the first $|u|$ letters of chunk x (see also Lemma 1 below).

Remark. The existence of at least one extender to each chunk $[x]$ is ensured by $[x]$ "itself" (strictly speaking by the underlying sequence x); in that sense $[x]$ or x are said to be their own extenders. Generally, there is more than one extender to each chunk. Obviously, a word w always qualifies as an extender to any w-chunk $x = [w^l]$.

Example. Chunk $x = [\text{ATACATA}]$ has three extenders, $u_1 = \text{CATA}$ (with extension complement $v_1 = \text{ATAC}$), $u_2 = \text{TACATA}$ (with $v_2 = \text{ATACAT}$), and finally x "itself", i.e., $u_3 = \text{ATACATA}$.

The restrictions $|u| > 0$ and $|u| \le |x|$ in Definition 1 exclude zero-shifts and "jumps" to the next available (separate) run, respectively. We do not want, for example, to regard the 5-letter word u=ATGGG as an extender to the 3-letter chunk $x = [\text{GGG}]$, even though the equivalence $xu \sim vx$ holds for $v = \text{GGGAT}$.

Several definitions and lemmas in this paper are phrased with respect to *chunks* but easily adapted to *words* and used in this broader sense. For instance, u is an extender to the word w if it is an extender to the corresponding chunk $x = [w]$.

LEMMA 1. *Let u be an extender to chunk x with extension complement v.*

(i) *Then a word $\mu^{(x)}$ exists such that $x \sim \mu^{(x)}u = v\,\mu^{(x)}$ holds.*

(ii) *If u_1 with $|u_1| \ge |u|$ is another extender to x with extension complement v_1, then words $u^{(1)}$ and $v^{(1)}$ exist such that $u_1 = u^{(1)}u$ and $v_1 = v\,v^{(1)}$.*

Proof. For extender u the equivalence $xu \sim vx$ implies that u coincides with the last $|u|$ letters of x. Let $\mu^{(x)}$ consist of the preceding $|x| - |u|$ letters of chunk x, i.e., $x \sim \mu^{(x)}u$. Now $xu \sim vx \sim v(\mu^{(x)}u) = (v\mu^{(x)})u$ implies $x \sim v\mu^{(x)}$, which proves (i). From $x \sim \mu^{(x)}u$ and $x \sim \mu_1^{(x)}u_1$ we get $\mu^{(x)}u = \mu_1^{(x)}u_1$, and because $|u_1| \ge |u|$, both extenders must share the last $|u|$ letters. Hence a proper word $u^{(1)}$ can be found with $u_1 = u^{(1)}u$. If $|u| = |u_1|$, then $u^{(1)} = \varepsilon$. Analogously, $v\mu^{(x)} = v_1\mu_1^{(x)}$ implies $v_1 = vv^{(1)}$. □

Remark 4.1. Each chunk has exactly one shortest extender. Consider the existence of two minimal extenders e and e' to the same chunk x; then $\mu\, e \sim x \sim \mu'\, e'$ must hold by Lemma 1(i) for two possibly empty words μ and μ'. And since e and e' are of equal size, $e = e'$ follows.

The following lemma explores the relationship between extenders to words and extenders to corresponding chunks.

LEMMA 2. *Let w be a word and $x = [w^l]$ a w-chunk of iteration length l. If u_x is an extender to x with complement v_x, then an extender u_w to w with complement v_w can be found such that $u_x = u_w w^s$ and $v_x = w^s v_w$ with $0 \le s < l$.*

Proof. We approach the proof by induction on l. Let u_x denote an extender to $x = [w^l]$. For $l = 1$ we obtain $x = [w]$, and obviously $u_x = u_w w^0 = u_w$ holds. Supposing now that the statement holds for l, we show its validity for $l + 1$. For that purpose we distinguish the following two cases.

Case 1. $|u_x| \le |w|$. Recall that w is an extender to x; by Lemma 1(ii), words $u_x^{(w)}$ and $v_x^{(w)}$ exist such that $w = u_x^{(w)}u_x$ and $w = v_x v_x^{(w)}$. Moreover, a comparison of

the first $|w|$ letters on the left and on the right of $w\,w^l u_x = w^{l+1}u_x \sim x\,u_x \sim v_x x \sim v_x w^{l+1} = v_x w\,w^l = v_x u_x^{(w)} u_x w^l$ implies $w = v_x u_x^{(w)}$. Together with $w = v_x v_x^{(w)}$, this results in $u_x^{(w)} = v_x^{(w)} = \mu_x^{(w)}$. Thus $wu_x = (v_x\mu_x^{(w)})u_x = v_x(\mu_x^{(w)}u_x) = v_x w$, and hence u_x is an extender to w, i.e., $u_x = u_w w^0$ as well as $v_x = w^0 v_w$ hold.

Case 2. $|u_x| > |w|$. Application of Lemma 1(ii) provides words $u^{(x)}$ and $v^{(x)}$ such that $u_x = u^{(x)}w$ and $v_x = wv^{(x)}$. Now $ww^l u^{(x)}w = w^{l+1}u_x \sim xu_x \sim v_x x \sim wv^{(x)}w^l w$ implies $[w^l]u^{(x)} \sim v^{(x)}[w^l]$. Thus $u^{(x)}$ is an extender to chunk $[w^l]$, for which by the induction assumption, u_w and s with $0 \le s < l$ can be found such that $u^{(x)} = u_w w^s$ holds. In combination with $u_x = u^{(x)}w$ we obtain $u_x = u_w w^{s+1}$; analogously $v_x = w^{s+1}v_w$ is shown. This completes the proof. □

Everything is now at hand to define endplacedness.

DEFINITION 2. *Let $\mathcal{C} = \sigma_1 x_1 \sigma_2 x_2 \dots x_r \sigma_{r+1} \dots x_n \sigma_{n+1}$ be a configuration of chunkpool $\mathcal{F}_4[\mathbf{S}; \mathcal{W}; \mathbf{L}; \mathcal{L}]$, and σ_i and x_j separators and chunks, respectively. Then we have the following:*

(i) *chunk x_r is called* extendible *in $\mathcal{C}$ if an extender u to x_r and a word c exist such that $\sigma_{r+1} = uc$. The boundary between x_r and σ_{r+1} is called an* extension site *of x_r in $\mathcal{C}$.*

(ii) *chunk x_r is called* endplaced *in $\mathcal{C}$ if it is not extendible in $\mathcal{C}$, i.e., if σ_{r+1} is empty or if no extender u to x_r exists such that $\sigma_{r+1} = u\,c$.*

(iii) *configuration $\mathcal{C}$ of chunkpool $\mathcal{F}_4[\mathbf{S}; \mathcal{W}; \mathbf{L}; \mathcal{L}]$ is called* endplaced *if all its chunks are endplaced.*

In other words, endplaced chunks are those not followed immediately by any of their extenders. Note that two configurations can be equivalent, even if x is endplaced in one but not the other.

Example. In configuration $\mathcal{C}_1 = \mathrm{ATCG}x\mathrm{CGGTG}$ the chunk $x = [\mathrm{CGCG}]$ is not endplaced but rather extendible with $u = \mathrm{CG}$ and $c = \mathrm{GTG}$. In $\mathcal{C}_2 = \mathrm{ATCGCG}x\mathrm{GTG}$, on the other hand, x is endplaced.

4.2. Cyclic chunks. The assessment of extendibility of word-chunks can, in general, become quite complicated. For the following class of cyclic chunks, however, extendibility depends only on the presence or absence of a unique word to the right, namely the corresponding principal unit.

DEFINITION 3. *A chunk x is called* cyclic *if a word $e \neq \varepsilon$ exists such that all extenders u to x are of the form $u = e^r$ for proper integers $r > 0$. The word e is called the* principal unit *of x and is denoted by* $\mathrm{PU}(x)$.

Example. Chunk $x = [\mathrm{AGTAGT}]$ is cyclic with principal unit $e = \mathrm{AGT}$. Its only extenders $u_1 = e$ and $u_2 = e^2$ are of the required form $u_i = e^{r_i}$. Chunk $x = [\mathrm{ATA\ CATA}]$, on the other hand, is not cyclic. It has three extenders ($u_1 = \mathrm{CATA}, u_2 = \mathrm{TACATA}$, and $u_3 = \mathrm{ATACATA}$), but no principal unit e such that $u_i = e^{r_i}$ holds for all three.

Note also that chunks of letter-runs (in contrast to word-runs) are always cyclic, with letter L as the principal unit of any L-chunk.

Remark 4.2. Let x denote a cyclic chunk with principal unit $e = \mathrm{PU}(x)$, and let u be an extender to x.

(a) Then $x = e^k$ for a proper integer $k > 0$, because x is an extender to itself.

(b) The principal unit e is an extender to x, since $ex \sim e(e^k) = (e^k)e \sim xe$.

(c) Moreover, the principal unit e is the *minimal extender* to x, because $u = e^r$ with $r > 0$ implies that no extender can be shorter.

(d) All extenders to x can be written as $u = ec$ with proper c; i.e., extenders to cyclic chunks always start with the same word e. (Generally this is not true,

e.g., $u_1 =$ AGA and $u_2 =$ GAAGA are both extenders to $x =$ AGAAGA, but start differently.)

For cyclic chunks Remark 4.2(d) provides a useful and simple way to distinguish endplaced from nonendplaced chunks.

LEMMA 3. *Let x_r be a cyclic chunk with principal unit e and let $\mathcal{C} = \sigma_1 x_1 \sigma_2 x_2 \ldots x_r \sigma_{r+1} \ldots x_n \sigma_{n+1}$ be a configuration where σ_i and x_j denote separators and chunks, respectively. Then x_r is endplaced in $\mathcal{C}$ if and only if σ_{r+1} is either empty or no word c exists such that $\sigma_{r+1} = ec$.*

In other words, a cyclic chunk is endplaced in a configuration if and only if it is not followed immediately by its principal unit.

Proof. First show [x_r endplaced] $\Rightarrow$ [$\sigma_{r+1} \neq ec$]. For x_r endplaced and $e = \mathrm{PU}(x_r)$ no word c can be found (see Definition 2) such that $\sigma_{r+1} = ec$ holds. To show the converse, assume that no c with $\sigma_{r+1} = ec$ exists. Then no extender u to x_r can satisfy $\sigma_{r+1} = uc_1$ for any word c_1; otherwise $\sigma_{r+1} = uc_1 = e^r c_1 = e(e^{r-1}c_1) = e\,c$ would hold, contradicting the assumption. Hence x_r must be endplaced. □

The next lemma states that cyclic chunks are exactly those in which extenders and extension complements coincide. For a corresponding result, see Lothaire (1983).

LEMMA 4. *A chunk x is cyclic if and only if it commutes with all its extenders, i.e., if $xu \sim ux$ holds for any extender u to x.*

Proof. Let u and v denote an extender to x and its complement, respectively. Show first [$x$cyclic] $\Rightarrow$ [$u = v$]. Cyclicity of x implies $u = e^r$ and $x = e^k$ with $e = \mathrm{PU}(x)$. Therefore $vx \sim xu \sim e^k e^r = e^r e^k$, i.e., $v = e^r = u$. To prove the reverse let e denote the unique smallest extender to x (see Remark 4.1); then $ex \sim xe$ holds by assumption. Now, by Lemma 1(ii), a word u_1 exists such that $u = u_1 e$ and $u_1 xe \sim u_1 ex = ux \sim xu = xu_1 e$ holds, implying $u_1 x \sim xu_1$. Hence u_1 qualifies also as an extender to x, and (again by Lemma 1) a word u_2 exists with $u_1 = u_2 e$. Proceeding in this manner, we obtain a finite number of extenders $u_1, u_2, \ldots$ of decreasing length, the smallest of which is u_{r-1} with $0 < |u_{r-1}| \leq |e|$. Since e is the unique smallest extender, u_{r-1}=e, and $u = e^r$, which proves that x is cyclic with $e = \mathrm{PU}(x)$. □

The cyclicity of a word w and the cyclicity of an associated w-chunk x are closely related properties, as will be described next.

LEMMA 5. *Let w be a word and $x = [w^l]$ a w-chunk of iteration length l. Then x is cyclic if and only if w is cyclic. Additionally, both share the same principal unit, i.e.,* $\mathrm{PU}(x) = \mathrm{PU}(w)$.

Proof. Let u_x and u_w be extenders to x and w, respectively. Show first [w cyclic] $\Rightarrow$ [x cyclic]. With proper u_w and s, any u_x can be written as $u_x = u_w w^s$; see Lemma 2. Cyclicity of w implies $w = e_w^k$ and $u_w = e_w^r$. Now $u_x = u_w w^s = (e_w^r)(e_w^{k \cdot s})$, i.e., $u_x = e_w^t$ with $t = r + k \cdot s$. Thus x is cyclic with $e_w = \mathrm{PU}(x)$. To show the reverse, let x be cyclic. For any extender u_w the word $u_x = u_w w^{l-1}$ is an extender to x, because $xu_x \sim w^l u_w w^{l-1} = w^{l-1}(wu_w)w^{l-1} = w^{l-1}(v_w w)w^{l-1} = (w^{l-1}v_w)w^l \sim v_x x$. Since w is an extender to x, the cyclicity of x implies $w = e_x^s$ and $u_x = e_x^r$. Now $e_x^k = u_x = u_w w^{(l-1)} = u_w e_x^{s \cdot (l-1)}$, i.e., $u_w = e_x^t$ with $t = k - s \cdot (l-1)$. The uniqueness of principal units implies that $\mathrm{PU}(x) = \mathrm{PU}(w)$. □

4.3. Overlapping chunks. It was mentioned earlier that overlapping word-runs can sometimes compete for the same intervening segment, resulting in blurred borders and ambiguities in the exact number, location, and size of available runs. To avoid such ambiguities, word-type combinations with certain overlaps will be excluded from our considerations.

DEFINITION 4. *Two chunks x_1 and x_2 are called* overlapping *if words u_1 and u_2 with $|u_1| < |x_1|$ (and $|u_2| < |x_2|$) exist such that $x_1 u_2 \sim u_1 x_2$ or $x_2 u_1 \sim u_2 x_1$.*

It is immediate from the definition that two overlapping chunks share a subsequence o such that $x_1 \sim u_1 o$ and $x_2 \sim o\,u_2$ (i.e., $x_1u_2 \sim u_1 o\,u_2 \sim u_1x_2$) or $x_2 \sim u_2 o$ and $x_1 \sim o\,u_1$ (i.e., $x_2u_1 \sim u_2 o\,u_1 \sim u_2x_1$) holds. The subsequence o is called a *common overlap* of x_1 and x_2.

Example. Chunks $x_1 = [\mathrm{ATGACCTTT}]$ and $x_2 = [\mathrm{AGGTATG}]$ overlap by $o = \mathrm{ATG}$: Take $u_1 = \mathrm{ACCTTT}$, $u_2 = \mathrm{AGGT}$; then $x_2u_1 \sim u_2 o\,u_1 \sim u_2x_1$. However, a shared subsequence at the beginning or at the end of two chunks does not necessarily mean that they overlap in the sense of the definition. For instance, $x_1 = [\mathrm{TATTCC}]$ and $x_2 = [\mathrm{TATTAAAG}]$ are, by Definition 4, nonoverlapping even though they share the same start sequence $u = \mathrm{TATT}$.

The following lemma shows that the overlap property is carried over from chunks to corresponding words and vice versa.

LEMMA 6. *Let x_1 and x_2 be chunks of the words w_1 and w_2, respectively, i.e., $x_1 = [w_1^{l_1}]$ and $x_2 = [w_2^{l_2}]$. Then x_1 and x_2 overlap if and only if w_1 and w_2 overlap.*

Proof. First show [x_1 and x_2 overlap]⇒[w_1 and w_2 overlap]. W.l.o.g. let $x_1u_2 \sim u_1x_2$ and $|u_1| < |x_1|$, implying that $u_1 = w_1^r h_1$ for proper integer $r < l_1$ and a possibly empty word h_1, where $w_1 = h_1\mu$ for a word μ with $0 < |\mu| \le |w_1|$. Now let $|w_1| \le |w_2|$ hold; then $|\mu| \le |w_2|$ and a word h_2 exists such that $w_2 = \mu\,h_2$, and hence $w_1h_2 = h_1\mu\,h_2 = h_1w_2$ with $|h_1| < |w_1|$ and $|h_2| < |w_2|$; i.e., w_1 and w_2 overlap, by Definition 4. For $|w_1| \ge |w_2|$ symmetric arguments apply. To show the converse, let w_1 and w_2 overlap, i.e., $w_1h_2 = h_1w_2$ with $|h_i| < |w_i|$ for $i = 1, 2$. Then $u_1 = w_1^{l_1-1}h_1$ and $u_2 = h_2w_2^{l_2-1}$ satisfies $|u_1| < |x_1|$ and $|u_2| < |x_2|$ and $x_1u_2 \sim w_1^{l_1-1}(w_1h_2)w_2^{l_2-1} = w_1^{l_1-1}(h_1w_2)w_2^{l_2-1} \sim u_1x_2$, completing the proof. □

While Lemma 6 deals with two chunks of different word-types, the next statement considers chunks of the same but cyclic word-type.

LEMMA 7. *Let $x_1 = [w^{l_1}]$ and $x_2 = [w^{l_2}]$ be two chunks of a cyclic word w with principal unit $e = \mathrm{PU}(w)$. Then an integer $r > 0$ can be found for any common overlap o of x_1 and x_2 such that $o = e^r$ holds.*

Proof. W.l.o.g., let $d = l_1 - l_2 > 0$ and $u_1x_2 \sim u_1 o\,u_2 \sim x_1u_2$; then $u = u_2w^d$ is an extender to x_1, because of $u_1x_1 \sim u_1(w^{l_2}w^d) \sim u_1x_2w^d \sim x_1u_2w^d = x_1u$ and $0 < |u| < |x_1|$. Lemma 5 implies the cyclicity of x_1, and Lemma 4 implies that $u = u_1$. With $e = \mathrm{PU}(x_1) = \mathrm{PU}(w)$ we obtain by Definition 4 that $u_1 = e^{r_1}$ and $w = e^t$ for proper integers r_1 and t, respectively. Now, $u_2e^{dt} = u_2w^d = u = u_1 = e^{r_1} = e^{r_1-dt}e^{dt}$ implies that $u_2 = e^{r_1-dt}$. Finally, $e^{r_1}e^{l_2t} \sim u_1x_2 \sim u_1 o\,u_2 = e^{r_1} o\,e^{r_1-dt}$ leads to $o = e^{l_2t-(r_1-dt)} = e^{l_1t-r_1} = e^r$ for $r = l_1t - r_1$, which completes the proof. □

4.4. Embeddible chunks. Two chunks may not only compete for the same intermediate segment; one chunk, if a substring of the other, may compete for a segment *within* the other.

DEFINITION 5. *Let x_1 and x_2 be two chunks of word-types w_1 and w_2, respectively. Chunk x_1 is called* embeddable *in x_2, denoted by $x_1 \subseteq x_2$, if words u_L and u_R (possibly empty) exist such that $x_2 \sim u_Lx_1u_R$ holds. We say that the pair $(x_1; x_2)$ is* embeddible *if $x_1 \subseteq x_2$ or $x_2 \subseteq x_1$; accordingly, $(x_1; x_2)$ is* nonembeddable *if neither $x_1 \subseteq x_2$ nor $x_2 \subseteq x_1$.*

Contrary to cyclicity and overlapability, the property of embeddability is *not* conveyed from words to chunks or vice versa. Chunks of embeddable words may nevertheless be nonembeddable; and chunks of nonembeddable words can still be embeddable.

Example. The word $w_1 = \mathrm{AG}$ is embeddable in $w_2 = \mathrm{AGT}$, but chunk $x_1 = [w_1^2] = [\mathrm{AGAG}]$ is not embeddable in $x_2 = [w_2^3] = [\mathrm{AGTAGTAGT}]$. The words $w_1 = \mathrm{AG}$ and $w_2 = \mathrm{GA}$ are obviously nonembeddable, whereas $x_1 = [w_1^2] = [\mathrm{AGAG}]$

is nevertheless embeddable in $x_2 = [w_2^3] = [\text{GAGAGA}]$ with $u_L = \text{G}$ and $u_R = \text{A}$.

With respect to embeddability, this suggests that there is no simple relation between words and chunks. However, restricted to nonoverlapping chunks, the following statement holds.

LEMMA 8. *Let w_1 and w_2 be two words, $l_1, l_2 \geq 1$ two integers, and $x_1 = [w_1^{l_1}]$ and $x_2 = [w_2^{l_2}]$ two nonoverlapping chunks. If any two chunks $y_1 = [w_1^{s_1}]$ and $y_2 = [w_2^{s_2}]$ with $s_1 \geq l_1$ and $s_2 \geq l_2$ are embeddible, then x_1 and x_2 are embeddible too.*

Proof. W.l.o.g., assume $y_1 \subseteq y_2$, i.e., $y_2 \sim u_L y_1 u_R$ for proper u_L and u_R. Then words v and u_1 with $|v| < |w_2|$ exist such that $u_L = w_2^{r-1}v$ and $w_2 = vu_1$. Now, $w_2^{r-1}vy_1u_R = u_Ly_1u_R \sim y_2 \sim w_2^{s_2} = w_2^{r-1}vu_1w_2^{s_2-r}$ implies $y_1u_R \sim u_1w_2^{s_2-r}$. Therefore $y_1u_Rw_2^r \sim u_1w_2^{s_2-r}w_2^r \sim u_1y_2$. Setting $u_2 = u_Rw_2^r$, we obtain $y_1u_2 \sim u_1y_2$. If, however, $|u_1| < |y_1|$, then (by Definition 5) y_1 and y_2 overlap, and (by Lemma 6) therefore x_1 and x_2 overlap, but they were assumed nonoverlapping. Hence $|u_1| \geq |y_1|$ and a (possibly empty) v' exist such that $u_1 \sim y_1v'$, implying $x_2 \sim w_2w_2^{l_2-1} = vu_1w_2^{l_2-1} \sim vy_1v'w_2^{l_2-1} \sim vx_1w_1^{s_1-l_1}v'w_2^{l_2-1}$. Finally, setting $u_L^{(x)} = v$ and $u_R^{(x)} = w_1^{s_1-l_1}v'w_2^{l_2-1}$ provides $x_2 \sim u_L^{(x)}x_1u_R^{(x)}$, i.e., $x_1 \subseteq x_2$. □

The formal negation of Lemma 8 provides the following important conclusion.

Remark 4.3. If two chunks are neither overlapping nor embeddable, then any two longer chunks of the same two word-types are also neither overlapping nor embeddable.

4.5. Condition set restrictions. In section 2 we described several complications emerging with the Morris, Schachtel, and Karlin (1993) counting method, if applied directly to word-runs. Combining the properties introduced in sections 4.2–4.4, we next define a class of condition sets to which the method can be applied without further difficulties.

DEFINITION 6. *Let $\mathcal{W}$ be a set of words, $\mathbf{L}$ a lengths-array, and $\vec{\lambda}$ its associated vector. The pair $(\mathcal{W}; \mathbf{L})$ is called* admissible *if the following holds for $1 \leq i, j \leq p$:*

(i) *All words $w_i \in \mathcal{W}$ are cyclic.*

(ii) *Any two words $w_i, w_j \in \mathcal{W}$ are either identical or nonoverlapping.*

(iii) *Chunks $x_i = [w_i^{\lambda_i}]$ and $x_j = [w_j^{\lambda_j}]$ are nonembeddable, provided $w_i, w_j \in \mathcal{W}$ are not identical.*

The condition set $\mathcal{F}_4(\mathbf{S}; \mathcal{W}; \mathbf{L})$ as well as the associated $\mathcal{F}_4$-chunkpool are called admissible *if $(\mathcal{W}; \mathbf{L})$ is admissible.*

Recalling Remark 4.3, we note that Definition 6(iii) ensures for admissible chunkpools that chunks of different word-types are nonembeddable.

5. Ascertaining the number of word-runs. The derivation of the word-run formula involves the summation of $\text{G}_\mathbf{M}$-values, where $\text{G}_\mathbf{M}$ denotes the number of orderings of $\mathcal{L}$ with exactly $\mathbf{M}$ runs$(\mathcal{W}) \geq \mathbf{L}$. It is therefore crucial that these values be well defined, i.e., that the number m_{ij} of runs$(w_i) \geq l_{ij}$ can be unambiguously determined for any ordering of $\mathcal{L}$. While this is not the case for word-runs in general (see section 2), we will show that $\text{G}_\mathbf{M}$ is always well defined, as long as $(\mathcal{W}, \mathbf{L})$ is admissible.

To this end we introduce the class of $\vec{\lambda}$-maximal configurations and establish in Theorem 1 a function assigning to each ordering of $\mathcal{L}$ a unique $\vec{\lambda}$-maximal configuration. For each such configuration the numbers m_{ij} of chunks$(w_i) \geq l_{ij}$ are well defined and coincide with the corresponding m_{ij}-values of the associated ordering, thus proving $\text{G}_\mathbf{M}$ to be well defined.

5.1. $\vec{\lambda}$-maximal configurations. We start by defining the class of $\vec{\lambda}$-maximal configurations as follows.

DEFINITION 7. *Let* $(\mathcal{W};\mathbf{L})$ *be admissible and* $\vec{\lambda}$ *associated with* $\mathbf{L}$*, i.e.,* $\lambda_i = l_{iq_i}$*. Also let* $\sigma_1,\sigma_2,\ldots,\sigma_{n+1}$ *denote separators and* $x_j = [w_{i_j}^{r_j}]$ *chunks of types* $w_{i_j} \in \mathcal{W}$ *and lengths* $r_j \geq \lambda_{i_j}$*, with* $1 \leq j \leq n$ *and* $1 \leq i, i_j \leq p$*. Then any configuration* $\mathcal{C} = \sigma_1 x_1 \sigma_2 x_2 \ldots \sigma_n x_n \sigma_{n+1}$ *is called* $\vec{\lambda}$-maximal *with respect to* $(\mathcal{W};\mathbf{L})$ *if*

(i) $\sigma_{j+1} = e_i v$ *for* $w_i \in \mathcal{W}$ *and* $e_i = \mathrm{PU}(w_i)$ *implies* $w_i \neq w_{i_j}$,
(ii) $\sigma_{j+1} = \varepsilon$ *implies* $w_{i_j} \neq w_{i_{j+1}}$,
(iii) *no word* v *exists such that* $\sigma_j = v\, w_{i_j}$,
(iv) *no* $w_i^{\lambda_i}$ *is embeddable in any* σ_j.

The four requirements (i)–(iv) in Definition 7 force $\vec{\lambda}$-maximal configurations to be *disjoint* (i.e., endplaced and separate), and their chunks to be maximal in length and number. $\vec{\lambda}$-maximal configurations are endplaced (see Lemma 3) because by Definition 7(i), no separator starts with the principal unit of the preceding chunk; they are separate because by Definition 7(ii), empty separators may occur only between chunks of distinct word-types. Chunks of $\vec{\lambda}$-maximal configurations are of maximal iteration length because extensions are excluded to the right by (i) and to the left by (iii) of Definition 7; and by (iv) the number of desired chunks has to be maximal, because no separator harbors an additional run of desired length and word-type.

Remark 5.1. For any $\vec{\lambda}$-maximal configuration $\mathcal{C}$ the ascertainment of the exact number m_{ij} of chunks$(w_i) \geq l_{ij}$ is unambiguous (and straightforward). The number of $\vec{\lambda}$-maximal configurations with exactly $\mathbf{M}$ chunks$(\mathcal{W}) \geq \mathbf{L}$ is therefore well defined. Furthermore, these m_{ij} coincide with the numbers of runs$(w_i) \geq l_{ij}$ in the corresponding ordering $\mathcal{O}(\mathcal{C})$.

5.2. Uniqueness theorem. While each configuration $\mathcal{C}$ can be mapped in a natural way to exactly one ordering $\mathcal{O}(\mathcal{C})$, the reverse is not generally true. Usually the same ordering can be represented by several different configurations. However, as Theorem 1 states, each ordering corresponds to only one $\vec{\lambda}$-*maximal* configuration.

THEOREM 1. *Let* $(\mathcal{W};\mathbf{L})$ *be admissible and* $\mathcal{O}$ *an ordering of letterpool* $\mathcal{L}$*. Then exactly one* $\vec{\lambda}$*-maximal configuration* $\mathcal{C}_\mathcal{O}$ *exists with* $\mathcal{C}_\mathcal{O} \sim \mathcal{O}$*.*

Proof. We have to show the existence and uniqueness of configuration $\mathcal{C}_\mathcal{O}$.

Existence. Shown by construction. Start with word-type $w_1 \in \mathcal{W}$; scan ordering $\mathcal{O}$; beginning from the right, replace the first occurring subsequence that matches $w_1^{\lambda_1}$ by a corresponding chunk, preliminarily labeled $x_1^{(1)}$. Then lengthen $x_1^{(1)}$ as long as it is immediately preceded by a word of type w_1; e.g., if a total of three additional w_1 precede $x_1^{(1)}$, then the chunk becomes $x_1^{(1)} = [\, w_1^{\lambda_1+3}\,]$ and substitutes for the whole subsequence of (λ_1+3) iterations of type w_1 at this position of $\mathcal{O}$. The next $w_1^{\lambda_1}$-match is treated in the same manner, substituting it by a properly lengthened chunk $x_1^{(2)}$, etc. Eventually this results in configuration $\mathcal{C}_1 = \sigma_1^{(n_1+1)} x_1^{(n_1)} \sigma_1^{(n_1)} x_1^{(n_1-1)} \ldots x_1^{(1)} \sigma_1^{(1)}$, which satisfies $\mathcal{C}_1 \sim \mathcal{O}$ and meets with respect to w_1 and λ_1 all requirements of Definition 7. Next, one proceeds analogously with respect to $w_2 \in \mathcal{W}$, scanning all separators $\sigma_1^{(j)}$ of $\mathcal{C}_1$ for $w_2^{\lambda_2}$-matches and then relabeling properly to obtain configuration $\mathcal{C}_2 = \sigma_2^{(n_2+1)} x_2^{(n_2)} \sigma_2^{(n_2)} x_2^{(n_2-1)} \ldots x_2^{(1)} \sigma_2^{(1)}$, which satisfies $\mathcal{C}_2 \sim \mathcal{O}$ and meets the requirements of Definition 7 with respect to w_1, w_2, λ_1, and λ_2. Note that all $x_2^{(k)}$ are either w_1- or w_2-chunks. After p steps, one attains $\mathcal{C}_p = \mathcal{C}_\mathcal{O}$, which is $\vec{\lambda}$-maximal with respect to $(\mathcal{W};\mathbf{L})$ and fulfills $\mathcal{C}_\mathcal{O} \sim \mathcal{O}$.

Uniqueness. Let $\mathcal{C}_\mathcal{O} = c_1 c_2 c_3 \ldots c_{2n+1}$, where for *even* $k \leq 2n$ each c_k denotes a (nonempty) chunk x_r, and for *odd* $k \leq 2n+1$ a (possibly empty) separator σ_r, with $r = k/2$ or $r = (k+1)/2$, respectively. Assume now that $\mathcal{C}_\mathcal{O}$ is *not* unique, i.e.,

a second, distinct $\vec{\lambda}$-maximal configuration $\mathcal{C}'_{\mathcal{O}} = c'_1c'_2c'_3\dots c'_{2m+1}$ exists, such that $\mathcal{C}'_{\mathcal{O}} \sim \mathcal{C}_{\mathcal{O}} \sim \mathcal{O}$ holds. Let κ with $1 \le \kappa \le 2n$ be the smallest integer for which $c_\kappa \ne c'_\kappa$ holds, and w.l.o.g. let $|c_\kappa| < |c'_\kappa|$. Two cases need consideration: (1) κ *even*, i.e., c_κ, c'_κ are chunks, and (2) κ *odd*, i.e., c_κ, c'_κ are separators.

Case 1. Suppose that κ even and $c_\kappa = x_r, c'_\kappa = x'_r$ are chunks with $|x_r| < |x'_r|$. Then x_r and x'_r overlap and are (by Lemma 6 and by (i) and (ii) of Definition 6) of identical (cyclic) word-type, say $x_r = [w^l]$ and $x'_r = [w^{l+d}]$ with $d \ge 1$. This means that x_r is immediately followed by at least one w, and thus a word u exists such that $x_rw = x_r\sigma_{r+1}u$, i.e., $w = \sigma_{r+1}u$ (recall that $w = e^t$ holds by cyclicity), and $|\sigma_{r+1}| < |e| \le |w|$ by Definition 7(i). At the same time x'_r and x_{r+1} overlap at least by u, implying that x_{r+1} is also of word-type w, hence σ_{r+1} (which precedes x_{r+1}) is, by Definition 7(ii), nonempty and also followed by at least one w. Thus a word u' exists with $\sigma_{r+1}w = \sigma_{r+1}(uu') = (\sigma_{r+1}u)u' = wu'$, making σ_{r+1} an extender to w. Now the cyclicity of w implies $\sigma_{r+1} = e^s$ for proper integer $s > 0$. But since this contradicts Definition 7(i), κ cannot be even.

Case 2. Suppose that κ is odd and $c_\kappa = \sigma_r, c'_\kappa = \sigma'_r$ are separators with $|\sigma_r| < |\sigma'_r|$. Then $|\sigma'_r| < |\sigma_rx_r|$; otherwise x_r were embeddible in σ'_r, violating Definition 7(iv). This means that x_r and x'_r must overlap or be embeddible; in either circumstance they are of identical (cyclic) word-type, say w. We have to distinguish three situations:

(I) If $|\sigma'_rx'_r| = |\sigma_rx_r|$, then $\sigma'_rw^l = \sigma_rw^{l+d}$ with $d \ge 1$ (since $|x_r| > |x'_r|$). But this would imply $\sigma'_r = \sigma_rw^d$, violating Definition 7(iii).

(II) If $|\sigma'_rx'_r| > |\sigma_rx_r|$, then a word u_2 ($|u_2| < |x'_r|$) exists with $\sigma_rx_ru_2 \sim \sigma'_rx'_r$, and because of $|\sigma_r| < |\sigma'_r|$ a word $u_1(|u_1| < |x_r|)$ with $\sigma'_r = \sigma_ru_1$. Hence $\sigma_rx_ru_2 \sim \sigma'_rx'_r \sim \sigma_ru_1x'_r$, and therefore $u_1x'_r \sim u_1o\,u_2 \sim x_ru_2$ with $o = e^r_w$ by Lemma 7 and $e_w = \mathrm{PU}(w)$, leading to $e^s_w \sim x'_r \sim o\,u_2 = e^r_wu_2$ and thus $u_2 = e^{(s-r)}_w$; i.e., x_r is immediately followed by e_w. By Definition 7(i), the separator σ_{r+1} must not start with e_w; therefore $|\sigma_{r+1}| < |e_w|$. Hence σ_{r+1} is followed by a word u with $e_w = \sigma_{r+1}u$. But then x_{r+1} overlaps x'_r and is of type w; i.e., σ_{r+1} is followed by e_w, and a word v exists with $\sigma_{r+1}e_w = \sigma_{r+1}u\,v = e_wv$. Thus σ_{r+1} is either empty, violating Definition 7(ii), or an extender to e_w and thereby to w, violating the minimality of e_w.

(III) If $|\sigma'_rx'_r| < |\sigma_rx_r|$, then arguments analogous to (II) apply.

Now it follows from (I), (II), and (III) that $|\sigma'_rx'_r|$ is neither equal to, nor larger or smaller than $|\sigma_rx_r|$. Hence κ cannot be odd.

Altogether, Cases 1 and 2 imply that no even or odd κ exists, with $c_\kappa \ne c'_\kappa$. Therefore $\mathcal{C}'_{\mathcal{O}} = \mathcal{C}_{\mathcal{O}}$; i.e., $\mathcal{C}_{\mathcal{O}}$ is unique. □

Remark 5.2. For admissible $(\mathcal{W}; \mathbf{L})$, Theorem 1 ensures that for each ordering $\mathcal{O}$ of letterpool $\mathcal{L}$ a unique $\vec{\lambda}$-maximal configuration $\mathcal{C}_{\mathcal{O}}$ with $\mathcal{C}_{\mathcal{O}} \sim \mathcal{O}$ exists. Furthermore, we know from Remark 5.1 that the exact numbers m_{ij} of chunks$(w_i) \ge l_{ij}$ can be assessed unambiguously for $\mathcal{C}_{\mathcal{O}}$ and that these m_{ij} coincide with the numbers of runs$(w_i) \ge l_{ij}$ in ordering $\mathcal{O}$. Hence $\mathrm{G}_{\mathbf{M}}$ is well defined for all admissible $(\mathcal{W}; \mathbf{L})$.

6. The number of acceptable orderings. At this point all preparations have been made to apply the Morris, Schachtel, and Karlin (1993) counting methods to word-runs. The remaining steps closely resemble their approach, and the necessary lemmas will be listed, but for most proofs we will refer to the original publication.

We start this section by introducing so-called generalized chunkpools. Then, in Lemma 9, we provide the number of unrestricted configurations, generated from these generalized chunkpools. Next, in Lemma 10, the number of endplaced configurations (EC) and of disjoint configurations (DC) is established, and in Lemma 11, the number of acceptable orderings is calculated as a weighted sum of DC-values. Finally, Theorem

2 presents the resulting run-formula, valid for word-runs of admissible condition sets.

6.1. Generalized pools. In section 2 we introduced ragged arrays **S** and **H**, both of equal dimension. Array **S** is associated with the condition set $\mathcal{F}_4(\mathbf{S};\mathcal{W};\mathbf{L})$. Its elements s_{ij} are fixed, providing the numbers of requested occurrences for each run specified by $(\mathcal{W};\mathbf{L})$. Array **H**, on the other hand, may range over all $\mathbf{H} \geq \mathbf{S}$ and will serve as a summation index.

Extending the notion of chunkpools, we can form *generalized* pools by the following two operations:

(i) *Fusion of chunks*: two or more adjacent elementary chunks of equal word-type are fused to form one *megachunk*. For simplicity all chunks in generalized pools are called megachunks, even the nonfused (elementary) ones.

(ii) *Extension of megachunks*: megachunks of word-type w_i are extended by their principal unit $e_i = \mathrm{PU}(w_i)$. Note that for any $w_i \in \mathcal{W}$ admissibility ensures the existence of e_i, which contains exactly $N_\alpha(e_i)$ letters of type $L_\alpha \in \mathcal{A}$.

Generalized $\mathcal{F}_4$-pools are characterized by an array **H** together with two integer-valued vectors $\vec{d} = (d_1, d_2, \ldots, d_p)$ and $\vec{\delta} = (\delta_1, \delta_2, \ldots, \delta_p)$ such that they contain exactly d_i megachunks originating from a total of $H^{(i)} = \sum_{j=1}^{q_i} h_{ij}$ elementary w_i-chunks, exactly δ_i of which are extended. The vectors $\vec{d}$ and $\vec{\delta}$ are constrained by $\min(H^{(i)};1) \leq d_i \leq H^{(i)}$ and $0 \leq \delta_i \leq d_i$, respectively. There are multiple ways to form $\vec{d}$ megachunks by fusing elementary chunks of a given $\mathcal{F}_4$-pool and to extend $\vec{\delta}$ of them. Thus, the characterization by $\vec{d}$ and $\vec{\delta}$ is not unique; usually more than one generalized $\mathcal{F}_4$-pool is associated with the same pair of vectors $\vec{d}$ and $\vec{\delta}$. However, all generalized $\mathcal{F}_4[\mathbf{H};\mathcal{W};\mathbf{L};\mathcal{A};\vec{n};\,\vec{d};\,\vec{\delta}]$-pools exhibit the same number d_i of w_i-megachunks and the same number $t_\alpha = n_\alpha - \sum_{i=1}^{p}[\,\delta_i \cdot N_\alpha(e_i) + \sum_{j=1}^{q_i}[h_{ij} \cdot l_{ij} \cdot N_\alpha(w_i)]\,]$ of remaining letters L_α, where $\alpha = 1, 2, \ldots, \tau$.

Next we derive the total number UC of "unrestricted" configurations generated from corresponding generalized $\mathcal{F}_4$-pools.

LEMMA 9. *Provided all $t_\alpha \geq 0$, the number* UC *of unrestricted configurations from generalized $\mathcal{F}_4[\mathbf{H};\mathcal{W};\mathbf{L};\mathcal{A};\vec{n};\,\vec{d};\,\vec{\delta}]$-pools is*

$$\mathrm{UC}(\mathbf{H};\mathcal{W};\mathbf{L};\mathcal{A};\vec{n};\,\vec{d};\,\vec{\delta}) = \frac{(D+T)!}{\prod_{\alpha=1}^{\tau}(t_\alpha!)} \cdot \prod_{i=1}^{p}\left[\frac{H^{(i)}!}{(d_i!)\cdot\prod_{j=1}^{q_i}(h_{ij}!)} \cdot \binom{H^{(i)}-1}{H^{(i)}-d_i} \cdot \binom{d_i}{\delta_i}\right],$$

where $D = \sum_{i=1}^{p} d_i$ and $T = \sum_{\alpha=1}^{\tau} t_\alpha$.

Proof. For the $H^{(i)}$ elementary chunks of type $w_i \in \mathcal{W}$, exactly $H^{(i)}!/\prod_{j=1}^{q_i}(h_{ij})!$ distinguishable arrangements exist. In each arrangement there are $\binom{H^{(i)}-1}{H^{(i)}-d_i}$ ways to choose the $(H^{(i)} - d_i)$ fusion sites out of a total of $(H^{(i)} - 1)$ available, and then $\binom{d_i}{\delta_i}$ possibilities for choosing δ_i out of the d_i available w_i-megachunks for extension. For each word-type w_i, this results in $\frac{H^{(i)}!}{\prod_{j=1}^{q_i}(h_{ij}!)} \cdot \binom{H^{(i)}-1}{H^{(i)}-d_i} \cdot \binom{d_i}{\delta_i}$ ways to obtain the required d_i megachunks, to extend δ_i of them, and to order them. Finally, the $D = \sum_{i=1}^{p} d_i$ generated megachunks and the remaining $T = \sum_{\alpha=1}^{\tau} t_\alpha$ single letters need to be arranged. Considering that the d_i megachunks of each word-type w_i are ordered already, and that the $t_\alpha \geq 0$ remaining single letters of type L_α are indistinguishable, this arrangement of megachunks and letters can be conducted in $(D+T)!/[\prod_{\alpha=1}^{\tau}(t_\alpha!)\cdot\prod_{i=1}^{p}(d_i!)]$ different ways. This completes the proof. □

The following lemma, in concert with Lemma 9, provides a formula for the number of disjoint configurations in admissible chunkpools.

LEMMA 10. *Let* $(\mathcal{W};\mathbf{L})$ *be admissible and all* $t_\alpha \geq 0$.

(i) *The number* EC *of endplaced configurations generated from generalized* $\mathcal{F}_4$*-pools with exactly* $\vec{d}$ *megachunks is*

$$\mathrm{EC}(\mathbf{H};\,\mathcal{W};\,\mathbf{L};\,\mathcal{A};\vec{n};\,\vec{d}\,) = \sum_{0\leq\vec{\delta}\leq\vec{d}} (-1)^{\eta(\vec{\delta})}\,\mathrm{UC}(\mathbf{H};\,\mathcal{W};\,\mathbf{L};\,\mathcal{A};\vec{n};\,\vec{d};\,\vec{\delta}\,),$$

where $\eta(\vec{\delta}) = \sum_{i=1}^{p}\delta_i$ *is the total number of extended megachunks and* UC *is as in Lemma* 9.

(ii) *The number* DC *of disjoint configurations of chunkpool* $\mathcal{F}_4[\mathbf{H};\mathcal{W};\mathbf{L};\mathcal{A};\vec{n}\,]$ *is*

$$\mathrm{DC}(\mathbf{H};\,\mathcal{W};\,\mathbf{L};\,\mathcal{A};\vec{n};\,) = \sum_{\vec{b}\leq\vec{d}\leq\vec{H}} (-1)^{\varphi(\vec{d})}\,\mathrm{EC}(\mathbf{H};\,\mathcal{W};\,\mathbf{L};\,\mathcal{A};\vec{n};\,\vec{d}\,),$$

where $\varphi(\vec{d}) = \sum_{i=1}^{p}(H^{(i)} - d_i)$ *is the total number of connected fusion sites within all megachunks, vector* $\vec{H} = (H^{(1)}, H^{(2)}, \ldots, H^{(p)})$ *with* $H^{(i)} = \sum_{k=1}^{q_i} h_{ik}$ *is the total number of elementary* w_i*-chunks,* $b_i = \min(H^{(i)};1)$ *is the* i*th component of* $\vec{b}$*, and* EC *is as in* (i).

Proof. See Lemmas 2 and 3 in Morris, Schachtel, and Karlin (1993) for a proof of this result. □

6.2. The word-run formula. Let $\mathcal{F}_4(\mathbf{S};\mathcal{W};\mathbf{L})$ be an admissible condition set and let $\mathrm{G}_{\mathbf{M}}$ be the number of orderings of $\mathcal{L}$ with exactly $\mathbf{M}$ runs$(\mathcal{W}) \geq \mathbf{L}$. Then $\mathrm{G}_{\mathbf{M}}$ is, according to Remark 5.2, well defined. Further, let AO denote the number of $\mathcal{F}_4$-acceptable orderings. Applying now (2.2) of section 2.1, AO can be written as $\mathrm{AO} = \sum_{\mathbf{M}\geq\mathbf{S}} \mathrm{G}_{\mathbf{M}}$. Based on this sum and on the relation between $\mathrm{G}_{\mathbf{M}}$ and DC (see (ii) in Lemma 11), and by exploiting the standard equality (2.1) of section 2.1, we will express AO as a weighted sum of DC-values with $\Delta(h;s)$ as weights.

LEMMA 11. *Let* $\mathcal{L}$ *be a letterpool and* $\mathcal{F}_4[\mathbf{H};\mathcal{W};\mathbf{L};\mathcal{L}]$ *an admissible chunkpool.*

(i) *Let* $\mathcal{O}$ *be an ordering of* $\mathcal{L}$ *with exactly* $\mathbf{M}$ *runs*$(\mathcal{W}) \geq \mathbf{L}$*. The number* RC *of disjoint configurations* $\mathcal{C}_i \sim \mathcal{O}$ *with exactly* $\mathbf{H}$ *chunks*$(\mathcal{W})$ *of lengths* $\mathbf{L}$ *is*

$$\mathrm{RC}(\mathbf{M};\mathbf{H}) = \prod_{i=1}^{p}\prod_{j=1}^{q_i}\binom{M_j^{(i)} - H_{j-1}^{(i)}}{h_{ij}},$$

where $\mathbf{M} = ||m_{ik}||$ *and* $M_j^{(i)} = \sum_{k=1}^{j} m_{ik}$; $\mathbf{H}$ *and* $H_j^{(i)}$ *accordingly.*

(ii) *Let* $\mathrm{G}_{\mathbf{M}}$ *denote the number of orderings of* $\mathcal{L}$ *with exactly* $\mathbf{M}$ *runs*$(\mathcal{W}) \geq \mathbf{L}$*. The number* DC *of disjoint configurations of chunkpool* $\mathcal{F}_4[\mathbf{H};\mathcal{W};\mathbf{L};\mathcal{L}]$ *is*

$$\mathrm{DC}(\mathbf{H};\,\mathcal{W};\,\mathbf{L};\,\mathcal{L}) = \sum_{\mathbf{M}\geq\mathbf{H}}\left[\prod_{i=1}^{p}\prod_{j=1}^{q_i}\binom{M_j^{(i)} - H_{j-1}^{(i)}}{h_{ij}}\right]\cdot \mathrm{G}_{\mathbf{M}},$$

with $\mathbf{M}$ *and* $\mathbf{H}$ *as in* (i).

(iii) *Let* $\mathcal{F}_4(\mathbf{S};\mathcal{W};\mathbf{L})$ *be an admissible condition set. The number AO of* $\mathcal{F}_4$*-acceptable orderings of* $\mathcal{L}$ *is*

$$\mathrm{AO}(\mathbf{S};\,\mathcal{W};\,\mathbf{L};\,\mathcal{L}) = \sum_{\mathbf{H}\geq\mathbf{S}}\left(\prod_{i=1}^{p}\prod_{j=1}^{q_i}\Delta(h_{ij};S_j^{(i)} - H_{j-1}^{(i)})\right)\cdot \mathrm{DC}(\mathbf{H};\,\mathcal{W};\,\mathbf{L};\,\mathcal{L}).$$

Proof. See (6) and (12) and Result 3 in Morris, Schachtel, and Karlin (1993) for a proof. □

Finally, Theorem 2 combines Lemmas 9–11 to supply an explicit formula for the number of orderings of $\mathcal{L}$ with at least $\mathbf{S}$ word-runs$(\mathcal{W}) \geq \mathbf{L}$, provided $(\mathcal{W}, \mathbf{L})$ is admissible.

THEOREM 2. *Let $\mathcal{L}(\mathcal{A}; \vec{n})$ be a letterpool, $\mathcal{F}_4(\mathbf{S}; \mathcal{W}; \mathbf{L})$ an admissible condition set, and $t_\alpha \geq 0$ for $1 \leq \alpha \leq \tau$. The number* AO *of $\mathcal{F}_4$-acceptable orderings of $\mathcal{L}$ is*

$$\mathrm{AO}(\mathbf{S};\, \mathcal{W};\, \mathbf{L};\, \mathcal{L}) =$$

$$\sum_{\mathbf{H} \geq \mathbf{S}} \sum_{\substack{\vec{b} \leq \vec{d} \leq \vec{H} \\ 0 \leq \vec{\delta} \leq \vec{d}}} \frac{(-1)^{\varphi(\vec{d})+\eta(\vec{\delta})} \cdot E!}{\prod_{\alpha=1}^{\tau} (t_\alpha!)} \cdot \prod_{i=1}^{p} \left[\frac{H^{(i)}!}{d_i!} \binom{H^{(i)} - 1}{H^{(i)} - d_i} \binom{d_i}{\delta_i} \prod_{j=1}^{q_i} \frac{\Delta(h_{ij};\, S_j^{(i)} - H_{j-1}^{(i)})}{h_{ij}!} \right],$$

where $H_j^{(i)} = \sum_{k=1}^{j} h_{ik}, H^{(i)} = H_{q_i}^{(i)}$, $S_j^{(i)}$ and $S^{(i)}$ analogously. Further, $\varphi(\vec{d}) = \sum_{i=1}^{p} (H^{(i)} - d_i), \eta(\vec{\delta}) = \sum_{i=1}^{p} \delta_i, D = \sum_{i=1}^{p} d_i, T = \sum_{\alpha=1}^{\tau} t_\alpha$, and $E = D + T$, $b_i = \min(H^{(i)}; 1)$ the ith component of $\vec{b}$, and $\Delta(h; s)$ is as in (2.1).

Proof. The proof follows by Lemmas 9–11. □

Acknowledgments. Thanks to Boris Hollas, Santani Teng, and Rob O'Neill for valuable discussions and suggestions.

REFERENCES

N. BALAKRISHNAN AND M.V. KOUTRAS (2002), *Runs and Scans with Applications*, Wiley, New York.
G. BENSON (1999), *Tandem repeats finder: A program to analyze DNA sequences*, Nucleic Acids Research 27, pp. 573–580.
L. BORTKIEWICZ (1917), *Die Iterationen*, Springer-Verlag, Berlin.
J.V. BRADLEY (1968), *Distribution Free Statistical Tests*, Prentice–Hall, Englewood Cliffs, NJ.
M. CROCHEMORE AND W. RYTTER (1994), *Text Algorithms*, Oxford University Press, New York.
P. ERDÖS AND A. RENYI (1970), *On a new law of large numbers*, J. Analyse Math. 23, pp. 103–111.
D.E. FOULSER AND S. KARLIN (1987), *Maximal success durations for the distribution of the scan statistic*, Stochastic Process Appl. 24, pp. 203–224.
L.J. GUIBAS AND A.M. ODLYZKO (1980), *Long repetitive patterns in random sequences*, Z. Wahrsch. Verw. Gebiete, 53, pp. 241–262.
S. KARLIN, M. MORRIS, G. GHANDOUR, AND M.Y. LEUNG (1988), *Efficient algorithms for molecular sequence analysis*, Proc. Natl. Acad. Sci. USA, 85, pp. 841–845.
M.V. KATTI, R. SAMI-SUBBU, P.K. RANJEKAR, AND V.S. GUPTA (2000), *Amino acid repeat patterns in protein sequences: Their diversity and structural-functional implications*, Protein Sci., 9, pp. 1203–1209.
M. LOTHAIRE (1983), *Combinatorics of Words*, Addison-Wesley, London.
M. LOTHAIRE (2002), *Algebraic Combinatorics on Words*, Cambridge University Press, Cambridge, UK.
A.M. MOOD (1940), *The distribution theory of runs*, Ann. Math. Statist., 11, pp. 367–392.
M. MORRIS, G. SCHACHTEL, AND S. KARLIN (1993), *Exact formulas for multitype run statistics in a random ordering*, SIAM J. Discrete Math., 6, pp. 70–86.
P.S. REDDY AND D.E. HOUSMAN (1997), *The complex pathology of trinucleotide repeats*, Curr. Opin. Cell. Biol., 9, pp. 364–372.
G. REINERT, S. SCHBATH, AND M.S. WATERMAN (2000), *Probabilistic and statistical properties of words: An overview*, J. Comput. Biol., 7, pp. 1–46.
R.R. SINDEN (1999), *Biological implications of the DNA structures associated with disease-causing triplet repeats*, Am. J. Hum. Genet., 64, pp. 346–353.

SIAM J. DISCRETE MATH.
Vol. 18, No. 1, pp. 159–176

A CHARACTERIZATION OF ACYCLIC SWITCHING CLASSES OF GRAPHS USING FORBIDDEN SUBGRAPHS*

JURRIAAN HAGE† AND TERO HARJU‡

Abstract. We characterize the switching classes that do not contain an acyclic graph. The characterization is by means of a set of forbidden induced subgraphs. We prove that in addition to switches of the cycles C_n for $n \geq 7$, there are only finitely many such graphs in 24 switching classes, all having at most 9 vertices. We give a representative of each of the 24 switching classes.

Key words. graph, switching class, Seidel switching, acyclic graph, trees, forbidden induced subgraph, critically cyclic graph

AMS subject classifications. 05C75, 05C22, 05C38

DOI. 10.1137/S0895480100381890

1. Introduction.

For a finite undirected graph $G = (V, E)$ and a set $\sigma \subseteq V$, the *switch* of G by σ is defined as the graph $G^\sigma = (V, E')$, which is obtained from G by removing all edges between σ and its complement $\bar{\sigma}$ and adding as edges all nonedges between σ and $\bar{\sigma}$. The switching class $[G]$ determined by G consists of all switches G^σ for subsets $\sigma \subseteq V$.

A switching class is an equivalence class of graphs under switching. The initiators of the theory of switching classes of graphs were Van Lint and Seidel [10]. They used the model in their investigation of elliptic geometry. For a survey of switching classes of graphs, and especially their many connections to other parts of mathematics, we refer to Seidel [7], Seidel and Taylor [8], and Cameron [2]. Recently a book by Ehrenfeucht, Harju, and Rozenberg was published on 2-structures that has a number of chapters on switching classes of graphs and their generalizations [4]. A book completely devoted to the subject is the first author's thesis (Hage [5]). Part of the motivation for the general model treated in these two books is that they constitute a way in which to model the semantics of a certain type of network of processors.

In this paper we solve a problem raised by Acharya [1] and mentioned by Zaslavsky in his dynamic survey [11], which asks for a characterization of those graphs that have an acyclic switch. We are concerned with those graphs that do not have an acyclic switch. Obviously, any graph which contains such a graph as an induced subgraph also does not have an acyclic switch. For this reason we are interested in the graphs that are minimal in this respect: they do not have an acyclic switch, but all their induced subgraphs do have an acyclic switch. We call these graphs and the corresponding switching class *critically cyclic*. We show that apart from the simple cycles C_n for $n \geq 7$, there are only finitely many critically cyclic graphs. In fact, we shall prove that a critically cyclic graph $G \notin [C_n]$ has order at most 9. These graphs are partitioned into 24 switching classes, and altogether there are 905 critically cyclic graphs (up to isomorphism and excluding switches of the cycles C_n).

In order to save the reader from long—and occasionally tedious—technical constructions for the small graphs, we rely on a computer program (in fact, two indepen-

*Received by the editors December 4, 2000; accepted for publication (in revised form) February 18, 2004; published electronically August 19, 2004.
http://www.siam.org/journals/sidma/18-1/38189.html
†Institute of Information and Computing Science, University of Utrecht, P. O. Box 80.089, 3508 TB Utrecht, The Netherlands (jur@cs.uu.nl).
‡Department of Mathematics, University of Turku, FIN-20014 Turku, Finland (harju@utu.fi).

dent programs) for the cases of order at most 9. Therefore our purpose is to prove that if G is a critically cyclic graph of order $n \geq 10$, then $G \in [C_n]$. The proof of this result uses the characterization from [6] of the acyclic graphs G—henceforth called the *special acyclic graphs*—that have a nontrivial acyclic switch (see section 4).

The paper is structured as follows. After some preliminaries we list the necessary details of the special acyclic graphs from [6]. We proceed by proving that critically cyclic graphs can have only a limited number of isolated vertices. As a consequence, a vertex in a critically cyclic graph has only a limited number of leaves adjacent to it. We prove that each switching class consisting of critically cyclic graphs and, different from $[C_n]$ for $n \geq 8$, contains a graph G that is almost a special acyclic graph. We then prove by case analysis, relying on the types of the special acyclic graphs, that a critically cyclic graph must have order at most 9. At the end of the paper we shall spend some time discussing the computer programs that were used to search for the small critically cyclic graphs. We shall also consider the question of why not all of the critically cyclic switching classes are used in our proof.

2. Preliminaries. For a (finite) set V, let $|V|$ be the cardinality of V. We shall often identify a subset $A \subseteq V$ with its characteristic function $A\colon V \to \mathbf{Z}_2$, where $\mathbf{Z}_2 = \{0, 1\}$ is the cyclic group of order 2, by the convention that for $x \in V$, $A(x) = 1$ if and only if $x \in A$. The symmetric difference of two sets A and B is denoted by $A + B$, and for the difference between A and B we write $A - B$.

The set $E(V) = \{\{x, y\} \mid x, y \in V,\ x \neq y\}$ is the set of all unordered pairs of distinct elements of V. A *graph* is a pair $G = (V, E)$, where V is the set of vertices and $E \subseteq E(V)$ the set of edges. We write xy or yx for the undirected edge $\{x, y\} \in E$; we call x and y *adjacent*. The graphs in this paper will be finite, undirected, and simple; i.e., they contain no loops or multiple edges. The cardinalities $|V|$ and $|E|$ are called the *order* and the *size* of G. Analogously to sets, a graph G will be identified with the characteristic function $G\colon E(V) \to \mathbf{Z}_2$ of its set of edges defined by $G(xy) = 1$ for $xy \in E$ and $G(xy) = 0$ for $xy \notin E$. Later we shall use both notations, $G = (V, E)$ and $G\colon E(V) \to \mathbf{Z}_2$, for graphs.

A graph $H = (X, E')$ is a *subgraph* of $G = (V, E)$, if $X \subseteq V$ and $E' \subseteq E$. Moreover, if $H \neq G$, then H is a *proper* subgraph of G. Also, H is an *induced subgraph* or a *subgraph induced by* X if for all distinct vertices $x, y \in X$, $H(xy) = G(xy)$. As shorthand we write $G - x$ for the subgraph induced by $V - \{x\}$ and, more generally, we write $G - I$ for the subgraph induced by $V - I$. Let H be a subgraph induced by a nonempty set $X \subseteq V$. If $H(xy) = 1$ for all distinct $x, y \in X$, then H is called a *clique*. On the other hand, if $H(xy) = 0$ for all distinct $x, y \in X$, then X is said to be *independent*.

For two graphs G and H on the vertex set V, we define $G + H$ to be the graph such that $(G + H)(xy) = G(xy) + H(xy)$ for all $xy \in E(V)$, where $+$ is addition modulo 2. We extend this operation to graphs on different sets of vertices V and V', respectively, by first extending G and H to graphs on $V \cup V'$ by setting all new edges to 0.

The disjoint union of two graphs G and H, on the other hand, is denoted $G \cup H$. We use $k \cdot G$ as shorthand for the disjoint union of k copies of G.

Some graphs we shall encounter in what follows are K_n, the clique on n vertices, and $K_{m,n}$, the complete bipartite graph on two disjoint sets of m and n vertices, respectively. P_n denotes a path of n vertices, and C_n denotes a cycle on n vertices.

For a vertex $v \in V$ of a graph G, the neighborhood $N_G(v) \subseteq V$ is the set of vertices adjacent to v in G. The *degree* of v is defined by $d_G(v) = |N_G(v)|$. An

isolated vertex has degree zero, a *leaf* degree one. A vertex v is a *leaf at* z if v is a leaf adjacent to z.

A graph is *acyclic* if it has no cycles. A *tree* is a connected acyclic graph.

A *selector* for $G = (V, E)$ is a subset $\sigma \subseteq V$, or alternatively a function $\sigma\colon V \to \mathbf{Z}_2$. A *switch* of a graph G by σ is the graph G^σ such that for all $xy \in E(V)$,

$$G^\sigma(xy) = \sigma(x) + G(xy) + \sigma(y).$$

It should be clear that this definition of switching is equivalent to the one given in the introduction. One of the switches of the graph (7-3) of Figure 3.1 is the graph (7-3') of Figure 3.4. In the figures we shall usually indicate a selector by the black vertices.

For a singleton set $\sigma = \{x\}$ we shall write G^x instead of $G^{\{x\}}$.

The set $[G] = \{G^\sigma \mid \sigma \subseteq V\}$ is called the *switching class* of $G = (V, E)$. We reserve lower case σ, τ for selectors (subsets) used in switching.

We always have $G^\sigma = G^{\bar\sigma}$ for the complemented selector $\bar\sigma = V - \sigma$.

A selector σ is said to be *constant* on a subset $X \subseteq V$ if σ is a constant function on X, that is, if $X \subseteq \sigma$ or $X \cap \sigma = \emptyset$. Note that if σ is constant on X, then the subgraphs induced by X in G and in G^σ are equal.

3. Critically cyclic graphs. A graph G, as well as its switching class $[G]$, is called *critically cyclic* if all proper induced subgraphs of G have an acyclic switch, but G itself does not have any acyclic switches. These graphs are *forbidden*, if we want to avoid switching classes with acyclic graphs. It is clear that a switch of a critically cyclic graph is also critically cyclic.

We say that an acyclic graph G has a *singular acyclic switch* if G has a unique acyclic switch different from G, that is, if, whenever σ and τ are any two nonconstant selectors for which both G^σ and G^τ are acyclic, then G^σ and G^τ are equal (not only isomorphic).

Let G be a critically cyclic graph. By definition, for each $x \in V$, there is a switch G^σ such that $G^\sigma - x$ is acyclic. We have the following simple result that will be used quite often without reference in the proofs.

LEMMA 3.1. *Let G be a critically cyclic graph and let x be a vertex of G.*

(i) *The proper induced subgraphs of G all have acyclic switches.*

(ii) *There exists a switch G^σ of G such that the induced subgraph $G^\sigma - x$ is acyclic. In this case all cycles of G^σ and of $(G^x)^\sigma$ go through x.*

Note that it is not true that in every critically cyclic graph G there is a vertex x such that $G - x$ is acyclic; the graph $K_{3,3} \cup 3 \cdot K_1$ of Figure 3.3 (9-2) is a counter example.

Example 3.2. Let G be the graph (7-3') in Figure 3.4. We prove that it is a critically cyclic graph. For this we need to show that it has no acyclic switches and that removing any of the vertices allows for an acyclic switch. For the latter it is sufficient to observe that the vertices $2, \ldots, 6$ are all on the unique cycle of G, and the induced subgraphs $G^{\{2,5\}} - 7$ and $G^{\{3,6\}} - 1$ are acyclic.

To prove that G has no acyclic switch observe that G has seven edges and that an acyclic graph can have at most six. We prove that applying any selector will not decrease the number of edges, and thereby we have proved that there is no acyclic switch of G. Since $G^\sigma = G^{\bar\sigma}$ for all selectors σ, we can assume that σ has at most three vertices.

First of all, for all vertices x, $d_G(x) \le 3 = (n-1)/2$. Hence applying a singleton selector cannot decrease the number of edges.

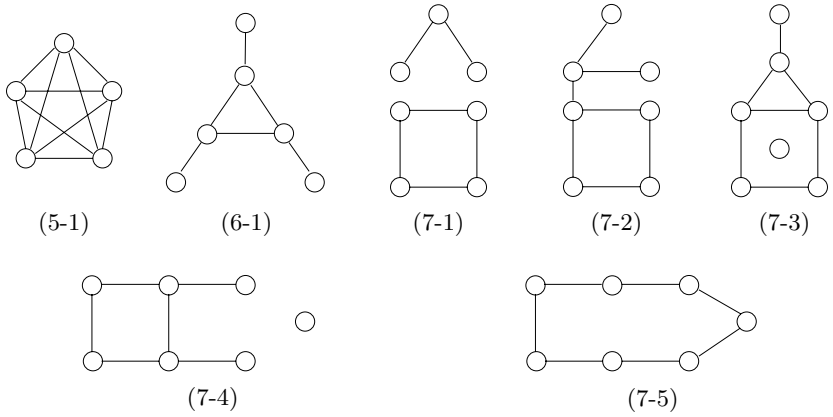


FIG. 3.1. *The critically cyclic graphs of order* 5, 6, *and* 7.

For doubleton selectors, $\sigma = \{x_1, x_2\}$, we can reason in the same way. The number of edges that change is $|\sigma| \cdot (7 - |\sigma|) = 10$. We must make sure that every selector makes at most five edges disappear. The only possible way, knowing that the maximum degree is three, is to take $\sigma = \{2, 6\}$, but in that case only four edges are removed, because one edge occurs inside the subgraph induced by σ.

For selectors of size 3, twelve edges will change. Hence we must look for selectors σ which create fewer than six edges, that is, σ makes more than six edges of G disappear. For this, the selector must contain a vertex of degree three, say $\{2\}$. If also $6 \in \sigma$, then the number of edges to be removed is four. Since there are no other vertices of degree three in G, we conclude that $6 \notin \sigma$. If σ has two vertices of degree two, then the subset σ has at most six edges going to its complement, because either the two of them are adjacent, or one of them is adjacent to 2.

Note that C_n for $n \leq 6$ has an acyclic switch: take an independent set of cardinality $\lfloor n/2 \rfloor$. However, the following was already proved by Acharya [1].

LEMMA 3.3. *The cycle C_n is critically cyclic for each $n \geq 7$.*

Proof. Let $G = C_n$ with $n \geq 7$. First, removing any vertex from $G = C_n$ gives us an acyclic graph P_{n-1}. Hence we need to prove only that all switches of G have a cycle. Let $\{x_1, x_2, \ldots, x_n\}$ be the vertices of G, where $G(x_i x_{i+1}) = 1 = G(x_n x_1)$. Suppose that G^σ is acyclic.

Assume first that σ has the same value for two adjacent vertices, say $\sigma(x_1) = 1 = \sigma(x_2)$. Now $\sigma(x_i) = 1$ for each $4 \leq i \leq n-1$, since $\{x_1, x_2, x_i\}$ does not induce a triangle C_3 in G^σ. Also $\sigma(x_3) = 1$ and $\sigma(x_n) = 1$ because otherwise $\{x_3, x_{n-1}, x_{n-2}\}$ or $\{x_n, x_4, x_5\}$ induces a triangle in G^σ. However, now σ is constant and $G^\sigma = G$; a contradiction. This takes care of all C_n where $n \geq 7$ is odd.

There remains the case where σ contains every other vertex of G. In this case, it is easy to see that if $n = 8$, then G^σ is again a cycle C_n, and if $n \geq 10$ and n is even, then the subset $\{x_1, x_2, x_4, x_5, x_7, x_8\}$ induces a cycle C_6 in C_n^σ. These contradictions prove the claim. □

We now state the result of our computer search for critically cyclic graphs.

THEOREM 3.4. *There are* 27 *switching classes of critically cyclic graphs of order $n \leq 9$. Representatives of these switching classes are given in Figures* 3.1, 3.2, *and* 3.3.

The main theorem proved in this paper is the following.

THEOREM 3.5. *The switching classes* $[C_n]$ *are the only critically cyclic switching classes of order* $n \geq 10$.

In the following proofs we shall refer to the graphs from Figures 3.1–3.4. The black vertices in Figure 3.4 indicate how these graphs can be switched into the corresponding graphs from the former three figures.

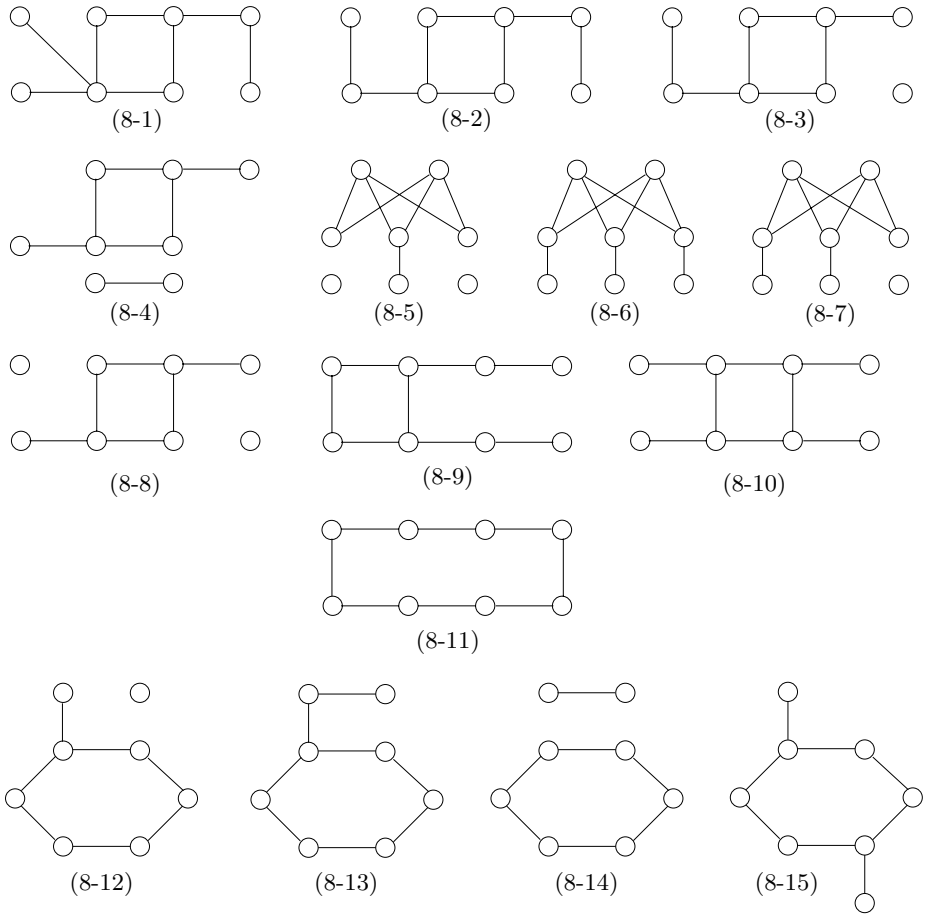


FIG. 3.2. *Critically cyclic graphs of order* 8.

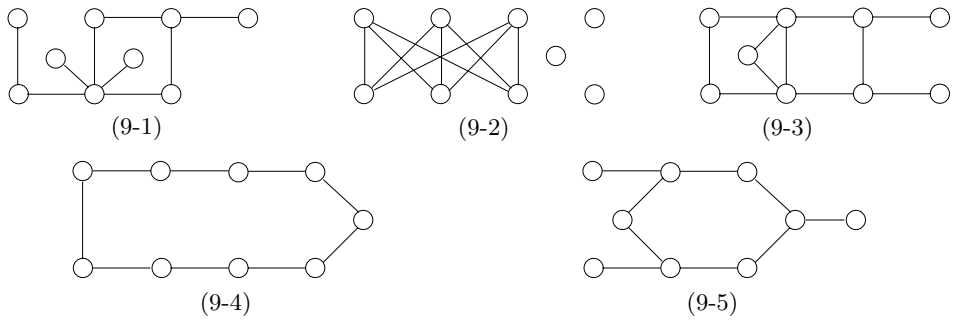


FIG. 3.3. *Critically cyclic graphs of order* 9.

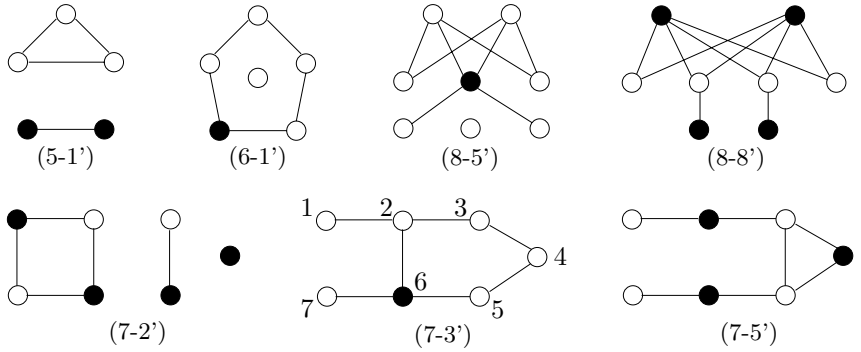


FIG. 3.4. *Switches of known critically cyclic graphs that are used in the proofs.*

4. The special acyclic graphs. We shall now describe the special acyclic graphs of [6] (see Figure 4.1), which will often be referred to in the rest of the paper. These acyclic graphs S have the property that they have an acyclic switch S^σ different from S.

Type (1s). The graph in Figure 4.1(1s) is denoted by $S_{k,m,l}$. It is obtained from the graph $K_{1,k+m}$ by substituting k leaves by an edge and by adding l isolated vertices. To be precise, $S = S_{k,m,l}$ consists of the induced subgraphs H, I, and M as defined in (S1)–(S3).

(S1) $H = \{z\} \cup \{y_i, x_i \mid i = 1, 2, \ldots, k\}$ consists of vertices for which $G(zy_i) = 1 = G(y_i x_i)$ for each i. The vertex z is called the *center* of S.

(S2) $I = \{u_1, u_2, \ldots, u_l\}$ consists of isolated vertices.

(S3) $M = \{v_1, v_2, \ldots, v_m\}$ consists of leaves such that $G(zv_i) = 1$ for each i.

For the types (As) with $A = 2, 3, \ldots, 8$, the corresponding special acyclic graphs will be denoted by $S_A(k, m)$, where k and m indicate the number of leaves of the (black) vertices z_1 and z_2. Because of the symmetry in k and m in each of these graphs, we may assume that $k \geq m$.

Denote by $P_t(m, k)$ the tree that is obtained from the path P_t of t vertices when the leaves are substituted by $K_{1,m}$ and $K_{1,k}$ (see Figure 4.1(4s) for $P_3(k, m)$).

Type (2s). $S_2(k, m) = K_{1,k} \cup K_{1,m}$.

Type (3s). $S_3(k, m) = K_{1,k} \cup K_{1,m} \cup K_1$.

Type (4s). $S_4(k, m) = P_3(k, m)$.

Type (5s). $S_5(k, m) = P_3(k, m) \cup K_1$.

Type (6s). $S_6(k, m) = P_2(k, m)$.

Type (7s). $S_7(k, m)$ is equal to $K_{1,3}(k, m)$, which is $K_{1,3}$ with two leaves substituted by $K_{1,k}$ and $K_{1,m}$ (see Figure 4.1(7s)).

Type (8s). $S_8(k, m) = P_4(k, m)$.

Types (9s)–(12s). The acyclic graphs of these types are P_7, T_7, P_6, and $P_4 \cup P_2$. These are listed in Figures 4.1(9s)–(12s).

A small acyclic graph can be of several of the above types. The role of the small acyclic graphs of the types (9s)–(12s) is strictly limited in this paper, because of their low order. Notice that P_6 equals $P_4(1, 1)$ of the type (8s), but we treat this small instance independently.

In [6] we proved the following theorem.

THEOREM 4.1. *Every switching class contains at most three acyclic graphs up to isomorphism. The acyclic graphs G that have an acyclic switch G^σ by a nonconstant selector σ are the special graphs of the types* (1s)–(12s).

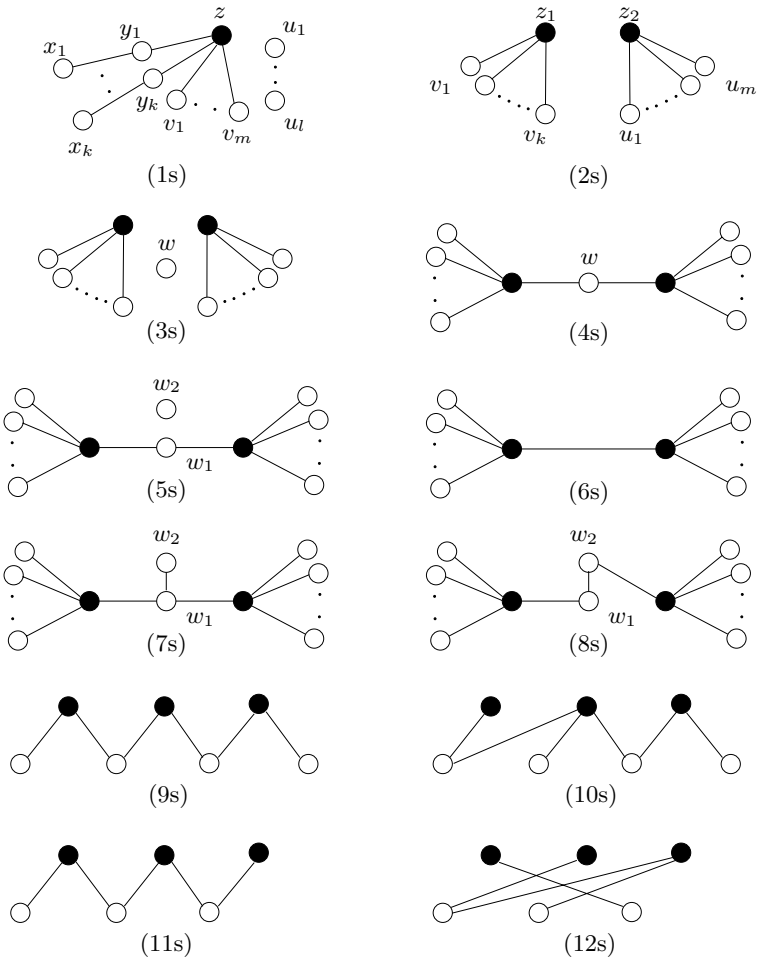


FIG. 4.1. *The special acyclic graphs* (1s)–(12s).

The blackened vertices in Figure 4.1 constitute the *centers* of the special acyclic graphs S. We denote by $Z(S)$ the set of these centers.

Let $\sigma = Z(S)$ for a special acyclic graph S. We observe that S^σ is of the same type as S except for a few of the cases: A graph of the type (3s) switches into a graph of the type (4s) (and vice versa); the graphs (11s) and (12s) switch into each other.

We shall often want to use the fact that a certain special acyclic graph S has a singular acyclic switch, that is, a unique acyclic switch by a nonconstant selector. We shall now give, in Lemma 4.2, a list of small special acyclic graphs that have such a switch. We omit the proof, because the graphs in it are small and the claim can be easily checked even by hand although some work is required.

LEMMA 4.2. *The following special acyclic graphs S have a singular acyclic switch:*

(i) $S_{1,2,2}$ *and* $S_{0,3,3}$ *of the type* (1*s*),
(ii) $S_A(2,2)$ *of the types* (*As*) *for* $A = 2, 3, 4$,
(iii) $S_A(2,1)$ *of the types* (*As*) *for* $A = 5, 7, 8$, *and*
(iv) $S_6(3,3)$ *of the type* (6*s*).

In these cases the singular acyclic switch S^σ is obtained by the selector $\sigma = Z(S)$ and its complement $\bar{\sigma}$.

We turn now to special acyclic graphs that are more general in their structure.

LEMMA 4.3. *The special acyclic graph $S = S_A(k, m)$ has a singular acyclic switch if*

(i) $k, m \geq 2$ *for the types* $(2s)$, $(3s)$, *and* $(4s)$,
(ii) $k \geq 2, m \geq 1$ *for the types* $(5s)$, $(7s)$, *and* $(8s)$,
(iii) $k, m \geq 3$ *for the type* $(6s)$.

In these cases the singular acyclic switch S^σ is obtained by the selector $\sigma = Z(S)$ and its complement $\bar{\sigma}$.

Proof. In each of the cases under consideration the special acyclic graph $S = S_A(k, m)$ has two centers, say $Z(S) = \{z_1, z_2\}$. Let $S^\sigma \neq S$ for a nonconstant selector σ such that S^σ is acyclic. We may assume that $\sigma(z_1) = 1$. In each of these cases every pair $x, y \notin Z(S)$ of distinct vertices of S belongs to an induced subgraph $R_{x,y}$ from Lemma 4.2 such that $Z(S) \subseteq R_{x,y}$ and $R_{x,y}$ is of the same type as S. When σ is restricted to $R_{x,y}$, we have that the switch $R^\sigma_{x,y}$ is acyclic. By Lemma 4.2, $\sigma(x) = \sigma(y)$, from which the claim follows. ∎

LEMMA 4.4. *The special acyclic graph $S = S_{k,m,l}$ has a singular acyclic switch in the following cases:*

(i) $k \geq 3$,
(ii) $k = 2$ *and* $m + l \geq 2$,
(iii) $k = 1$ *and* $m, l \geq 2$, *and*
(iv) $k = 0$ *and* $m, l \geq 3$.

In these cases the singular acyclic switch S^σ is obtained by the singleton selector $\sigma = Z(S)$ and its complement $\bar{\sigma}$.

Proof. Let $S = S_{k,m,l}$ with $Z(S) = \{z\}$ be as described in (S1)–(S3), where k, m, and l satisfy the requirements of the claim. Let σ be a nonconstant selector such that S^σ is acyclic. Since $S^\sigma = S^{\bar{\sigma}}$, we may assume that $\sigma(z) = 1$. We shall show that $\sigma = \{z\}$, from which the claim follows. Note that the induced subgraph $S - z$ of S equals the disjoint union $k \cdot P_2 \cup (m + l) \cdot K_1$.

Suppose that $k \geq 3$. Now $S - z$ is not special, and by Theorem 4.1, σ must be constant on $S - z$. Therefore $\sigma = \{z\}$.

Suppose then that $k = 2$ and $m + l \geq 2$. In this case, $S - z$ equals $2 \cdot P_2 \cup 2 \cdot K_1$, which is not special. As in the above, we have $\sigma = \{z\}$.

For the rest of the cases where $k \leq 1$, the claim follows from Lemma 4.2 as in the proof of Lemma 4.3, since now every pair of vertices of S belongs to an induced subgraph $S_{1,2,2}$ or $S_{0,3,3}$ that contains the center z of S. ∎

5. Isolated vertices. In this section we give constraints for the isolated vertices in critically cyclic graphs. In particular, we prove our main tool for the final proof: if G is critically cyclic and is such that $G - x$ is acyclic for a vertex x, then $G - x$ has no isolated vertices.

LEMMA 5.1. *A critically cyclic graph G has at most two isolated vertices, or else $G = K_{3,3} \cup 3 \cdot K_1$ (see* (9-2) *in Figure* 3.3*).*

Proof. Let $I = \{x_1, x_2, \ldots, x_m\}$ be the set of the isolated vertices in G. We assume that $m \geq 3$. The graph G is critically cyclic, and hence $G - x_1$ is not acyclic but has an acyclic switch $(G - x_1)^\tau$. The induced subgraph $G - I$ has a cycle, and therefore τ is not constant on $G - I$, say $\tau(v_0) = 0$ and $\tau(v_1) = 1$ for some $v_0, v_1 \in V(G) - I$.

We show that τ has different values on the elements of $I - \{x_1\}$. From this it follows that $m = 3$. For this purpose, suppose that there are two vertices, say x_2 and x_3, in $I - \{x_1\}$ having the same value. Without loss of generality we may assume that $\tau(x_2) = 1 = \tau(x_3)$. If $\tau(v) = 0$ for a vertex $v \in V - \{v_0, x_1, x_2, x_3\}$, then (x_2, v_0, x_3, v)

is a cycle in $(G - x_1)^\tau$, which contradicts the choice of τ. Therefore, $\tau(v) = 1$ for all $v \notin \{x_1, v_0\}$. Whatever we choose for $\tau(x_1)$, all cycles in G^τ should go through x_1. However, if $\tau(x) = 1$ for all $x \in I - \{x_1, x_2, x_3\}$, then x_1 is a leaf at v_0 in G^τ when we choose $\tau(x_1) = 1$. This proves that the vertices of $I - \{x_1\}$ have different values in τ. Hence $m = 3$ with $\tau(x_2) \neq \tau(x_3)$.

Since $(G - x_1)^\tau$ is acyclic, $G^\tau(x_2x_3) = 1$, and every vertex of $V - I$ is adjacent to either x_2 or x_3, it follows that $V - I$ is independent in $(G - x_1)^\tau$. The switching class of a discrete graph of order n consists of the complete bipartite graphs of order n (see [7]), and therefore $G = K_{r,s} \cup 3 \cdot K_1$, where $r, s \geq 2$ since G is not acyclic. Since $K_{3,3} \cup 3 \cdot K_1$ is a critically cyclic graph, and each $K_{2,s} \cup 3 \cdot K_1$, for $s \geq 4$, has an acyclic switch (by switching one of the vertices in the part of size 2 of $K_{2,s}$), the claim follows. □

The following lemma is an immediate corollary to the previous result.

LEMMA 5.2. *Let G be critically cyclic of order $n \geq 10$. Then no vertex $v \in V$ is adjacent to more than two leaves of G.*

Proof. Let L be a set of leaves of G such that $G(uv) = 1$ for all $u \in L$. Then the vertices of L are isolated in G^σ for the selector $\sigma = \{v\}$. By Lemma 5.1, L has at most two elements, since the graph $K_{3,3} \cup 3 \cdot K_1$ is of order 9. □

In the proof of the following result we require knowledge of the small critically cyclic graphs. We shall say that a graph G *avoids (c-i)* if G does not contain (as an induced subgraph) the graph in Figure 3.1(c-i) if $c = 5, 6, 7$, Figure 3.2(c-i) if $c = 8$, or Figure 3.3(c-i) if $c = 9$.

Recall that every proper induced subgraph of a critically cyclic graph can be switched to a graph with no cycles. In particular, we have the following corollary that is used often in the rest of this paper.

LEMMA 5.3. *Let G be a critically cyclic graph of order $n \geq 10$. Then G avoids the graphs in Figures* 3.1, 3.2, *and* 3.3.

The next lemma is our first general tool concerning isolated vertices.

LEMMA 5.4. *Let G be a critically cyclic graph of order $n \geq 10$. Then G has at most one isolated vertex.*

Proof. By Lemma 5.1, G has at most two isolated vertices. Suppose that G has exactly two isolated vertices, say $I = \{x_1, x_2\}$. By assumption, G is critically cyclic, and therefore there exists a selector τ such that $(G - x_1)^\tau$ is acyclic. We can assume without restriction that $\tau(x_2) = 0$.

The set τ is independent in G, as well as in $(G - x_1)^\tau$, for otherwise, there is a triangle containing x_2 in $(G - x_1)^\tau$. In fact, τ contains at most one vertex from each connected component of $(G - I)^\tau$, since x_2 is adjacent to each $x \in \tau$ in the switch $(G - x_1)^\tau$. Notice that these connected components are trees, because $(G - x_1)^\tau$ is acyclic.

We extend the domain of τ by setting $\tau(x_1) = 0$. Let $\tau = \{z_1, \cdots, z_r\}$. In Figure 5.1 we have depicted the graph

$$G^\tau = (H + (T_1 \cup T_2 \cup \ldots \cup T_r)) \cup F,$$

where

- $H = K_{2,r}$ has the bipartition $(\{x_1, x_2\}, \{z_1, \ldots, z_r\})$,
- the induced subgraphs T_i are disjoint trees with $H \cap T_i = \{z_i\}$, and
- F is an acyclic induced subgraph or is empty.

In the following we let $C^{(i,j)}$ denote the cycle (x_1, z_i, x_2, z_j) for different i and j. Since G^τ is not acyclic, we must have $r \geq 2$, and thus $C^{(1,2)}$ always exists in G^τ.

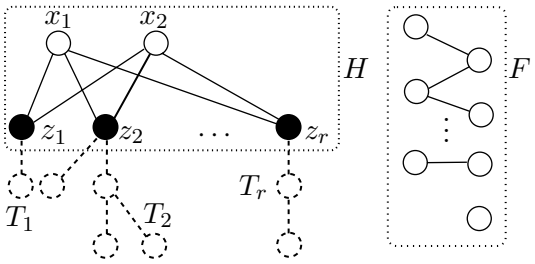


FIG. 5.1. *The graph* $G^\tau = (H + (T_1 \cup T_2 \cup \cdots \cup T_r)) \cup F$.

CLAIM 1. *We have* $|F| \leq 2$. *Hence* F *is either empty, discrete, or a path* P_2.

To see this, suppose that $F \neq \emptyset$. To avoid (7-1) and (7-2') in $C^{(1,2)} \cup F$, F must be either discrete or a path P_2. If F is discrete, then $|F| \leq 2$ by Lemma 5.1.

CLAIM 2. *At most two of the trees* $T_1, \ldots, T_r$ *have more than one vertex.*

Indeed, x_1, x_2 together with different $z_i, z_j, z_k \in \tau$ induce a $K_{2,3}$ in G^τ. The graph (8-6) then implies the present claim.

CLAIM 3. *Each nonsingleton tree* T_i *has the form*

$$T_i = S_{k_i,s_i,0} \quad \text{or} \quad T_i = P_4(s_i, 0),$$

where $k_i \geq 0$, $s_i \leq 2$ *and* z_i *is the center of* $S_{k_i,s_i,0}$ *or the center of* $P_4(s_i, 0)$ *adjacent to the* s_i *leaves.*

For this, let T_i be a nonsingleton tree. By considering the subgraph of G^τ induced by $C^{(i,j)}$ and T_i (for any $j \neq i$) we deduce that

- to avoid (7-1) the longest path of T_i starting from z_i has length at most 3,
- to avoid (7-1) each $v \neq z_i$ in T_i with $v \notin N_{T_i}(z_i)$ satisfies $d_{T_i}(v) \leq 2$,
- to avoid (7-2) each $v \in N_{T_i}(z_i)$ satisfies $d_{T_i}(v) \leq 2$,
- to avoid (7-2') T_i cannot have edge disjoint paths P_4 and P_3 with the common end z_i.

Hence T_i has the required form. By Lemma 5.2, we have $s_i \leq 2$.

We shall divide our considerations according to the number N of nonsingleton trees T_i.

Case $N = 0$. In this case G^τ equals $K_{2,r} \cup F$, where $|F| \leq 2$. All these graphs have an acyclic switch. Indeed, in $K_{2,r}$ and $K_{2,r} \cup K_1$ we can take the selector $\{x_1, x_2\}$, and in $K_{2,r} \cup 2 \cdot K_1$ and $K_{2,r} \cup P_2$ we can take $\{x_1, x_2, v\}$, where $v \in F$. These contradict the assumption that G has no acyclic switches.

Case $N = 1$. Suppose G^τ has a unique nonsingleton tree, say T_1, among the trees $T_1, \ldots, T_r$.

(1) Suppose first that $T_1 = P_4(s_1, 0)$. We have $r = 2$, for if $r \geq 3$, then $C^{(2,3)}$ together with $T_1 - z_1$ does not avoid (7-1). Also, necessarily $F = \emptyset$ to avoid the graph (7-2') in the subgraph induced by $C^{(1,2)}$ and F together with the edge at the leaf of T_1. However, now $n \leq 9$ contradicts our assumption on the order n.

(2) Suppose then that $T_1 = S_{k_1,s_1,0}$ with $k_1 > 0$.

(2.1) Assume that $r \geq 3$. Now $F = \emptyset$, $s_1 = 0$, and $k_1 = 1$, for otherwise the graph (7-2') is in the subgraph induced by $C^{(2,3)}$, F, and $T_1 - z_1$. Hence T_1 is a path P_3. However, now G^τ has an acyclic switch for all $r \geq 3$ (select all z_i's and the other end of the path T_1); a contradiction.

(2.2) Let $r = 2$. To avoid (7-2'), F cannot be P_2, and hence it is discrete or empty. However, now G^τ has an acyclic switch (select z_1); a contradiction.

(3) Finally, suppose that $T_1 = S_{0,s_1,0}$. Since $1 \leq s_1 \leq 2$ and $|F| \leq 2$, we must have $r \geq 4$. By considering the graph induced by $C^{(2,3)}$, F, and $T_1 - z_1$, we notice that to avoid (7-2') the induced subgraph F cannot be P_2. To avoid (8-5) ($K_{\{x_1,x_2\},\{z_1,z_2,z_3\}}$ with a leaf of T_1 and F) necessarily $|F| \leq 1$. If $|F| = 1$, then $s_1 = 1$ to avoid the graph (8-5'). Hence T_1 is a path P_2. In both of these cases ($s_1 = 1$ and $|F| = 1$, and $s_1 = 2$ and $F = \emptyset$), we have an acyclic switch (select x_1, x_2 and a leaf adjacent to z_1); a contradiction.

Case $N = 2$. Suppose that G^τ has exactly two nonsingleton trees, say T_1 and T_2, among the trees $T_1, \ldots, T_r$. Assume also without loss of generality that $|T_1| \geq |T_2|$.

To avoid (8-4) and (8-8) in the subgraph induced by $C^{(1,2)}$ and F, necessarily $|F| \leq 1$. Since $\{x_1, x_2, z_1, \ldots, z_4\}$ forms a $K_{2,4}$, we must have $r \leq 3$ in order to avoid (8-8'). Moreover, if $r = 3$, then $F = \emptyset$ to avoid (8-7). In any case we have $r + |F| \leq 3$. Since $n \geq 10$, it follows that $|T_1| + |T_2| \geq (10 - 2) - (r - 2) - |F| \geq 7$.

Let t be the length of the longest path in T_1 starting from z_1. By Claim 3 we know that $t \leq 3$. First suppose that $t = 1$ and hence that $T_1 = S_{0,s_1,0}$. We know $s_1 \leq 2$ and hence $|T_1| \leq 3$ and we have $|T_2| \geq 4$, which contradicts the assumption $|T_1| \geq |T_2|$. Therefore $t \geq 2$. To avoid (8-1) in the subgraph induced by $C^{(1,2)}$, T_1, and T_2, necessarily $d_{T_2}(z_2) = 1$. Moreover, to avoid (8-2), T_2 must be a path P_2. Consequently $|T_1| \geq 5$. If $T_1 = S_{k_1,s_1,0}$, then $k_1 = 1$. Indeed, if $k_1 \geq 2$, we remove the middle vertex from one of the paths P_3 in T_1, to obtain the graph (8-3) in G^τ. Now in all cases $|T_1| = t + 1 + s_1 \geq 5$. However, the case where $t \geq 2$ and $s_1 = 2$ is excluded by (9-1), and the cases $t = 3$ with $1 \leq s_1 \leq 2$ are excluded by (8-4) (remove the neighbor of z_1 on the longest path of T_1 starting from z_1). □

Analogously to Lemma 5.2, we obtain the following result.

LEMMA 5.5. *Let G be a critically cyclic graph of order $n \geq 10$. Then no vertex $v \in V$ is adjacent to more than one leaf of G.*

We consider next isolated vertices in the induced subgraphs $G - x$ for critically cyclic graphs G.

LEMMA 5.6. *Let G be a critically cyclic graph of order $n \geq 10$ and let $x \in V$.*

(i) *$G - x$ can have at most two isolated vertices. Moreover, if $G - x$ has two isolated vertices, then x is adjacent to exactly one of these in G.*

(ii) *If a vertex $v \neq x$ is adjacent to m leaves of $G - x$, then $m \leq 2$. Moreover, if $m = 2$, then x is adjacent to exactly one of these.*

Proof. For (i) we only need to observe that if $G - x$ has three isolated vertices, then in either G^x or G at least two of these are isolated and we can apply Lemma 5.4. The same holds if the number of isolated vertices is two, but x is not adjacent to exactly one of them in G.

For (ii) assume that there is a vertex $v \neq x$ adjacent to more than two leaves. The vertex x is nonadjacent to at least two of these in either G or G^x, and the result then follows from Lemma 5.5. □

Let G be a graph and x and y be vertices of G such that $G - x$ is acyclic. We say that y is *compatible* with x if $G - y$ and $G^x - y$ are not acyclic.

LEMMA 5.7. *Let G be a critically cyclic graph such that $G - x$ is acyclic.*

(i) *If y is compatible with x, then $G - \{x, y\}$ is a special acyclic graph.*

(ii) *If G is of order $n \geq 8$ and $G \notin [C_n]$, then there exists a vertex $y \in V$ that is compatible with x.*

Proof. Since G is critically cyclic, $G - y$ has an acyclic switch $(G - y)^\tau$. Let $S = G - \{x, y\}$. Because S and S^τ are both acyclic graphs, it follows that either S is special or τ is constant on S.

Suppose τ is constant on S, and thus either $\tau = S \cup \{x\}$ or $\tau = S$. In the former case $(G-y)^\tau = G-y$, a contradiction, because $(G-y)^\tau$ is acyclic and $G-y$ is not. In the second case we have $(G-y)^\tau = (G^x-y)^{S\cup\{x\}} = G^x - y$. Again we have a contradiction, since $G^x - y$ is supposed to have a cycle. This proves that S is special.

For (ii), suppose $G \notin [C_n]$. Since G has no acyclic switches, there are cycles in G and G^x, and they all pass through x, because $G-x$ is acyclic. Moreover, since C_k is critically cyclic for $k \geq 7$, the induced cycles of G and G^x have length at most 6.

If G or G^x has an induced cycle C_5 or C_6, let y be a vertex that is not on such a cycle. It is clear that $G-y$ and G^x-y both contain a cycle, and therefore each such vertex y is compatible with x.

If G and G^x both have an induced cycle of length at most 4, then these two cycles have altogether at most 7 vertices (since they share the vertex x). Since $n \geq 8$, there exists a vertex y that is not on these cycles. For such a vertex y, both $G-y$ and G^x-y are not acyclic. This proves the claim. □

The next lemma is our second main tool for isolated vertices.

LEMMA 5.8. *Let G be a critically cyclic graph of order $n \geq 10$ such that $G-x$ is acyclic. Then $G-x$ has no isolated vertices.*

Proof. Assume to the contrary that there is an isolated vertex u in $G-x$. Now u is either a leaf adjacent to x in G (and hence isolated in G^x) or it is isolated in G (and hence a leaf adjacent to x in G^x). Hence no cycle goes through u in G or G^x. It follows that $G-u$ and G^x-u are not acyclic, and by Lemma 5.7(i), $S = G-\{x,u\}$ is special.

Let τ be a selector for which $(G-u)^\tau$ is acyclic. Since $G-u$ and G^x-u both have a cycle, τ is not constant on S.

If $S = S_A(k,m)$ is of the type (As) for $A \in \{2,\dots,8\}$, then by Lemma 5.6(ii), the centers of S are adjacent to at most two leaves of $G-x$ and hence $k,m \leq 2$. In this case, S has at least eight vertices (since $n \geq 10$), and this rules out the types (2s), (3s), (4s), and (6s). Thus S is of the type (1s) or it is one of the types (5s), (7s), (8s) with $k = 2 = m$.

In the cases (5s), (7s), and (8s), by Lemma 4.3, S has a singular acyclic switch with respect to its centers $Z(S) = \{z_1,z_2\}$ and therefore $\tau = \{z_1,z_2\}$ or $\tau = \{x,z_1,z_2\}$. However, by Lemma 5.6(ii), x is adjacent to two leaves of S, one leaf being adjacent to z_1 and one to z_2. The same holds for S^τ.

This means that $(G-u)^\tau$ is not acyclic, which is a contradiction.

Consider then the case $S = S_{k,m,l}$ and adopt the notation (S1)–(S3) for it. Without restriction we can assume that $\tau(z) = 1$ for the center z of S, since $(G-u)^{\bar\tau} = (G-u)^\tau$. Extend τ to the whole domain by setting $\tau(u) = 0$.

We have $n = (2k+1)+m+l+2 \geq 10$, and thus $k \geq \frac{1}{2}(7-(m+l))$. By Lemma 5.6, $m \leq 2$ and $l \leq 1$. (Recall that u is isolated in $G-x$.) In particular, $k \geq 2$, and if $k = 2$, then $m = 2$, $l = 1$, and $n = 10$. (In Figure 5.2 we have depicted the graph $S_{k,m,l}$ obtained so far. In solid lines we indicate what must be there, in dotted lines what may be there.)

In these cases, by Lemma 4.4, the special acyclic graph S has a singular acyclic switch S^ρ for $\rho = \{z\}$. Now $(G-u)^\tau$ is acyclic, and hence so is S^τ. Since τ is not constant on S, the uniqueness of ρ implies that $\rho(v) = \tau(v)$ for all $v \notin \{x,u\}$. Also, the only vertices in G that can be adjacent to u in G^τ are x and z, and because G^τ is not acyclic, both must be adjacent to u. Moreover, x is adjacent in G^τ to exactly one vertex $v \in H \cup I$, since G^τ has a cycle but $G^\tau - u$ $(= (G-u)^\tau)$ is acyclic.

Suppose that v is a leaf of the part H of S, say $v = x_1$. If $l \geq 1$, then

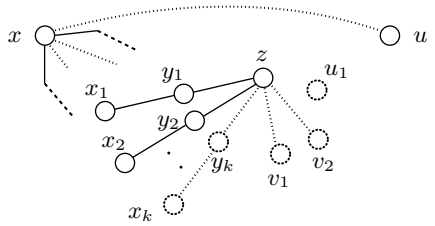


FIG. 5.2. *The case of the special graph* $S_{k,m,l}$.

$\{x, x_1, z, u, y_1, u_1, y_2\}$ induces a graph (7-4) in G^τ. Similarly, if $m \geq 1$, then we find that $\{x, x_1, z, u, y_1, w, v_1\}$ induces a graph (7-4) in G^τ, where $w = x_2$ if $G^\tau(xv_1) = 1$ and $w = y_2$ otherwise. Therefore we have $m = 0 = l$. Now $k \geq 4$, and G^τ contains an induced graph (7-4) obtained by removing x_2.

If v is a middle vertex in H, say $v = y_1$, then $\{x, y_1, x_1, z, u\}$ induces a cycle C_5 in G^τ, and hence G^τ has an induced graph (6-1') obtained by removing x_2.

If $v \in I$ is isolated in S, say $v = u_1$, then to avoid (8-3) as being induced by the set $\{x, u_1, z, u, x_1, y_1, y_2, v_i\}$ (for any $v_i \in M$), we must have $G^\tau(xv_i) = 0$ (provided that $m > 0$). However, now $(G^\tau)^z$ is acyclic; a contradiction.

Finally, if $v = z$, then again $(G^\tau)^z$ is acyclic. This contradiction completes the proof of the lemma. □

6. The proof of Theorem 3.5. In the following we shall consider every type of special acyclic graph in turn and show that each case leads to a contradiction, thereby proving our main theorem, that besides the graphs in $[C_n]$ there are no critically cyclic graphs of order $n \geq 10$.

Throughout this section we let G be a critically cyclic graph of order $n = |V| \geq 10$ such that $G \notin [C_n]$. Also, let $x \in V$ be a fixed vertex.

Since G is critically cyclic, there exists an acyclic switch $(G - x)^\rho$ of the induced subgraph $G - x$. Since the switches of critically cyclic graphs are critically cyclic, we can assume that ρ is constant on V, and therefore that $G - x$ is acyclic already.

By Lemma 5.7(ii), there exists a vertex y that is compatible with x, that is, $G - y$ and $G^x - y$ are not acyclic. Since G is critically cyclic, there is a nonconstant selector σ such that $(G - y)^\sigma$ is acyclic. By Lemma 5.7(i), $S = G - \{x, y\}$ is a special acyclic graph, and it is of one of the types (1s)–(8s), since its order is at least 8.

In the following proofs a number of simple properties are often used, and we note them here: first, the vertex y is adjacent to at most one vertex of each component of S; if not, $G - x$ would not be acyclic. Also, there must be a cycle in G that does not contain y, because $G - y$ is not acyclic. This also holds for $G^x - y$.

We shall now formulate a few conditions that hold for the special acyclic graphs (1s)–(8s). For any graph G' and vertex v, let $L_{G'}(v)$ be the set of leaves adjacent to v in G', and let $I_{G'}$ denote the set of isolated vertices in G'.

LEMMA 6.1. *In the above notation, we have*

(i) $I_S \subseteq N_G(y)$;

(ii) *for all* $v \in S$, $|L_S(v)| \leq 3$. *Moreover,* $|L_S(v)| = 3$ *implies* $|N_G(x) \cap L_S(v)| \geq 1$ *and* $|N_G(y) \cap L_S(v)| = 1$.

Proof. By Lemma 5.8, $G - x$ does not have isolated vertices, and hence (i) follows.

For (ii), we have $|N_G(y) \cap L_S(v)| \leq 1$, since $G - x$ is acyclic. If $|L_S(v)| \geq 3$, then, by Lemma 5.6(ii), $|L_S(v) - N_G(y)| \leq 2$, and x is adjacent to at most one vertex of $L_S(v) - N_G(y)$. Hence, in this case, we must have $|L_S(v)| = 3$ and in this case x and

y are each adjacent to at least one vertex in $L_S(v)$. Because $G - x$ is acyclic, y is adjacent to exactly one vertex in $L_S(v)$. □

Note how the previous Lemma restricts the values of k and m for the types (2s)–(8s) and m for (1s). On the other hand, $n \geq 10$ gives a lower bound on these values for most types.

6.1. The case (1s). We shall now consider first the most difficult case, $S = S_{k,m,l}$. We adopt the notation of (S1)–(S3) for it. Without restriction we may assume that $\sigma(z) = 1$ for the center z of S. Also, we can assume that $\sigma(x) = 0$, by the symmetry in the definition of compatibility, i.e., by the fact that both $G - y$ and $G^x - y$ are not acyclic. We extend σ to the whole domain by setting $\sigma(y) = 0$. Note that $(G - y)^\sigma = G^\sigma - y$.

LEMMA 6.2. *We have*

(i) $k = 2$,
(ii) $1 \leq l \leq 2$, $1 \leq m \leq 2$, *and* $m + l \geq 3$,
(iii) $M \subseteq N_G(x)$,
(iv) *if* $l = 2$, *then* $|N_G(x) \cap I| = 1$,
(v) *if* $m = 2$, *then* $|N_G(y) \cap M| = 1$,
(vi) $|N_G(x) \cap (H \cup I) - \{z\}| \leq 1$.

Proof. By Lemma 5.4, G has at most one isolated vertex; therefore $|N_G(x) \cap I| \leq 1$. Otherwise, switching by $\{x, y\}$, we obtain two isolated vertices, because y is connected to all vertices in I by Lemma 6.1. When Lemma 5.6(ii) is applied to the vertex y, we have that $l \leq 2$ and also obtain part (iv).

If $k = 0$, then $m + l \geq 7$, since $n \geq 10$. This contradicts the bound $m \leq 3$ of Lemma 6.1(ii) and the above bound $l \leq 2$. Hence $k \geq 1$.

If $k = 1$, then $m + l \geq 5$. In this case, $l = 2$ and $m = 3$. If $k = 2$, then $m + l \geq 3$, using reasoning similar to that for $k = 1$. By Lemma 4.4, in both cases S^z is the singular acyclic switch of S and thus $\sigma = \{z\}$. Now $M \subseteq N_G(x)$; otherwise, there is an isolated vertex of M in the acyclic graph $G^\sigma - y$, contradicting Lemma 5.8 (recall that $\sigma(x) = 0 = \sigma(y)$). Therefore part (iii) is true. Moreover, when Lemma 5.6(ii) is applied to G^σ, it follows that $m \leq 2$, and as a consequence $k \geq 2$, because as was shown above, if $k = 1$, then we must have $m = 3$. Part (v) now follows from Lemma 5.6(ii).

Part (vi) follows from the fact that the subgraph of G^σ induced by $H \cup I$ is connected and $G^\sigma - y$ is acyclic.

Suppose then that $k \geq 3$. By (vi) it follows that there are at least two pairs $x_i y_i$ such that $G(xx_i) = 0 = G(xy_i)$, say for $i = 1, 2$. For $i = 1, 2$ let τ_i be such that $(G - x_i)^{\tau_i}$ is acyclic, where we may choose $\tau_i(z) = 1$. The special acyclic graph $S - x_i$, which is $S_{k-1,m+1,l}$, has a singular acyclic switch $(S - x_i)^z$: the reason is that $n = 2k + m + l + |\{x, y, z\}| \geq 10$ and either $k \geq 4$, or else $k = 3$ and hence $m + l \geq 1$, so that in the case of $S_{k-1,m+1,l}$ we may apply Lemma 4.4.

Clearly $\tau_i = \sigma$ when we set $\tau_i(x_i) = 0$. By Lemma 5.8, the vertex y_i is not isolated in $G^{\tau_i} - x_i$, and therefore $G^{\tau_i}(yy_i) = 1 = G(yy_i)$ for $i = 1, 2$ (since $G(xy_i) = 0 = G^{\tau_i}(xy_i)$) and we have a cycle in $G - x$. This contradiction proves parts (i) and (ii). □

In Figure 6.1 we have depicted part of the situation for the special graph $S_{k,m,l}$: the requirements $I_S \subseteq N_G(y)$, $k = 2$, $1 \leq m \geq 2$, $1 \leq l \leq 2$, $M \subseteq N_G(x)$ are all implied by the drawing (again solid means what must be there, dotted means what might be there). Notice that in the case at hand, Lemma 6.2 implies that $n \leq 11$.

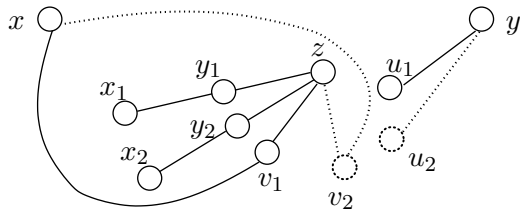


FIG. 6.1. *The case of the special graph $S_{k,m,l}$ after Lemma* 6.2.

We shall now finish the case $S = S_{k,m,l}$. In the following we consider the adjacencies of x and y to vertices in S and each other.

(1) Assume first that x is adjacent to a vertex u in I. By parts (ii) and (iv) of Lemma 6.2, u is the only neighbor of x in I. Then also $G(xz) = 1$, since otherwise, (x, u, z) is a triangle in $G^\sigma - y$, contradicting the fact that $G^\sigma - y$ is acyclic. Moreover, $G(xx_i) = 0 = G(xy_i)$ for $i = 1, 2$, because $G^\sigma - y$ is acyclic. We have $G(xy) = 0$, for otherwise, (x, y, u) is a triangle in G, and to avoid (5-1) with the edges $G(x_iy_i) = 1$, we would have to have that y is adjacent to two vertices in $H - \{z\}$, giving a cycle to $G - x$. Now we know that $N_G(x) = M \cup \{z, u\}$, by Lemma 6.2(iii).

For any vertex $v \in M$, (x, z, v) is a triangle in G. Consider the subgraph of G induced by $\{x, u, y, v, z, x_1, y_1\}$. To avoid (7-5') we must have that either $G(yz) = 1$ or $G(yv) = 1$. Since $G^\sigma - y$ is acyclic, y is adjacent in G to no other vertices of $H \cup M \cup \{x\}$.

(1.1) Suppose $G(yz) = 1$. By Lemma 6.2(v), we have $m = 1$ and so $M = \{v\}$. Then Lemma 6.2(ii) implies that $l = 2$. But now $\{x, u_1, y, z, y_1, x_2, u_2\}$ induces the graph (7-4) in G; a contradiction.

(1.2) Suppose $G(yv) = 1$. Then $\{u_1, y, x, v_1, z, y_1, x_2\}$ induces a (7-3); also a contradiction.

(2) Therefore $G(xu) = 0$ for all $u \in I$, and thus by Lemma 6.2(iv), $l = 1$, $I = \{u_1\}$. We have from Lemma 6.2(ii) that $m = 2$ (and hence $M = \{v_1, v_2\}$) and from part (v) that y is adjacent to exactly one vertex of M, say $G(yv_1) = 1$. In conclusion, so far we know that $G(xv_1) = 1 = G(xv_2)$, $G(yv_1) = 1$, $G(yv_2) = 0$, and $G(yu_1) = 1$, $G(xu_1) = 0$. Also, $G(yw) = 0$ for all $w \in S - \{u_1, v_1\}$, since $G - x$ is acyclic.

Since $G^\sigma - y$ is acyclic, we must have $G(xx_i) = 0 = G(xy_i)$ for $i = 1$ or 2, say $i = 1$. There are two cases to be considered here.

(2.1) Suppose $G(xz) = 0$. Now $G(xy) = 1$, since otherwise, $\{x, v_1, v_2, z, y, x_1, y_1, u_1\}$ induces a graph (8-9) in G.

(2.2) Suppose $G(xz) = 1$. To avoid $\{x, z, v_1, y_1, x_1, y, u_1\}$ inducing a (7-5') we must have $G(xy) = 1$.

In both of these cases, we have $G(xy) = 1$. But now $\{x, y, v_1, x_1, y_1\}$ induces a graph (5-1'). This contradiction proves that the special graph $S = G - \{x, y\}$ is not of the type (1s).

6.2. The other cases. Let $S = S_A(k, m)$ for $A \in \{2, \ldots, 8\}$, where we assume that $k \geq m$. Let z_1 and z_2 be the two centers of S, and $L = \{v_1, v_2, \ldots, v_k\}$ and $M = \{u_1, u_2, \ldots, u_m\}$ be the sets of leaves of S adjacent to z_1 and z_2, respectively.

We may assume that $\sigma(z_1) = 1$ and, as in the previous case, we can assume that $\sigma(x) = 0$. Again we extend σ to the whole domain by setting $\sigma(y) = 0$.

LEMMA 6.3. *We have $k \leq 3$. Moreover,*

(i) *if S is of the types* (3*s*)–(8*s*)*, then $m \leq 2$;*

(ii) *if $k = 3$ and if S^σ is the singular acyclic switch of S, then x and y are each adjacent to exactly one vertex in L, but not the same vertex.*

Proof. The fact $k \leq 3$ is already stated in Lemma 6.1(ii).

For (i), assume that $m \geq 3$, and thus that both z_1 and z_2 have three leaves adjacent to them in S. By Lemma 6.1(ii), y is adjacent to a leaf in both L and M. For the types (4s)–(8s) (where S is connected) $G - x$ has a cycle; a contradiction. For the type (3s) we apply the same argument but taking y in G^σ instead of x in G: we first observe that $S^{Z(S)}$ has a singular acyclic switch by Lemma 4.3, and therefore $\sigma = Z(S)$. Now x is adjacent in $G^\sigma - y$ to a leaf in both L and M. It follows then that $G^\sigma - y$ has a cycle; a contradiction.

For (ii), we first observe that x is adjacent to exactly one vertex in L, since $G^\sigma - y$ is acyclic. The claim then follows from Lemma 5.6(ii) and Lemma 6.1(ii). □

Note that, by Lemma 6.1(ii), Lemma 6.2(i) and (ii), Lemma 6.3(i), it already follows that there are no critically cyclic graphs of order at least 12 (unless they are in $[C_n]$).

6.3. The cases (2s)–(4s). Let S be of the type (2s), (3s), or (4s). Since $n \geq 10$, Lemma 6.1(ii) implies that $k = 3$ and $2 \leq m \leq 3$. By Lemma 4.3, S has a singular acyclic switch which means that $\sigma = \{z_1, z_2\}$. (Recall that $\sigma(z_1) = 1$ and $\sigma(x) = 0 = \sigma(y)$.)

By Lemma 6.3(ii), x is adjacent to one vertex in L, say $G(xv_1) = 1$, and y is adjacent to another vertex of L, say $G(yv_3) = 1$. Furthermore, $G(xv_2) = 0 = G(yv_2)$ for the third vertex v_2 of L. Since $G(z_2v_1) = 0$ and $G(xv_1) = 1$, we must have that $G(xz_2) = 1$, for otherwise there is a cycle in $G^\sigma - y$. We now run through the cases one by one.

Case (2s). Let $S = S_2(k, m)$. By Lemma 6.3 and the fact that $n \geq 10$, we know that $k = 3 = m$. Also, by Lemma 6.3(ii), x and y are both adjacent to one but not the same vertex in M, say $G(xu_1) = 1$ and $G(yu_3) = 1$. By the uniqueness of σ, x must be adjacent to z_1 to ensure that $G^\sigma - y$ is acyclic (or else $\{z_1, x, u_1\}$ would be a triangle). Because y can only be connected to one vertex in every component of S, the only remaining unknown is $G(xy)$. If $G(xy) = 0$, then $\{x, v_1, z_1, u_3, y\}$ induces the graph (5-1), and if $G(xy) = 1$, then $\{u_1, x, y, v_3, z_1, v_2, u_2\}$ induces the graph (7-4). These contradictions show that S is not of the type (2s).

Case (3s). Let $S = S_3(k, m)$. In this case S^σ is of the type (4s).

By the above, $G(xz_2) = 1$ and $G(xv_2) = 0 = G(xv_3)$. To avoid a cycle in $G^\sigma - y$, necessarily $G(xz_1) = 1$. $G(xw) = 0$ (for the isolated vertex w of S) and $G(xu) = 0$ for all $u \in M$. The reason is that x is adjacent to v_1 in S^σ, and the above choices prevent x from being adjacent to any other vertex of the connected graph S^σ.

By Lemma 5.6(ii), $m = 2$ and y is adjacent to one vertex in M, say $G(yu_2) = 1$. By Lemma 6.1(i), y is connected to w. The only unknown is the value $G(xy)$. If $G(xy) = 0$, then $\{v_1, x, z_2, u_1, y, v_3, z_1\}$ induces the graph (7-5'), and if $G(xy) = 1$, then $\{v_1, x, y, u_2, z_2, v_2, u_1\}$ induces the graph (7-4). Hence S is not of the type (3s).

Case (4s). Let $S = S_4(k, m)$. Because S is connected, y is not adjacent to any other vertex of S (except v_3). Hence, $m = 2$, and x is adjacent to one vertex of M, say $G(xu_1) = 1$ (see Lemma 5.6(ii)). Since $G^\sigma - y$ is acyclic, x must be adjacent to z_1. If $G(xy) = 0$, then $\{x, u_1, z_2, v_3, y\}$ induces the graph (5-1). If $G(xy) = 1$, then $\{v_1, z_1, v_2, v_3, x, u_1, y\}$ induces the graph (7-4) in G^x. Hence S is not of the type (4s).

6.4. The cases (5s)–(8s). We shall first consider the type (6s).

Case (6s). Let $S = S_6(k, m)$. In this case, $n \geq 10$ implies that $k, m \geq 3$. But this contradicts Lemma 6.3(i). Hence S is not of the type (6s).

For the remaining cases (5s), (7s), and (8s), let w_1 be the neighbor of z_1 of degree at least 2. Let w_2 be the vertex that is not adjacent to z_1 and z_2 in the types (5s) and (7s), and which is adjacent to z_2 in (8s) (see Figure 4.1(5s), (7s), and (8s)).

By Lemma 6.1(ii) and $n \geq 10$, $2 \leq k \leq 3$, $m \geq 1$, and $k + m \geq 4$. In all these cases S has a singular acyclic switch by Lemma 4.3 and therefore $\sigma = \{z_1, z_2\}$.

We can assume that x is adjacent to a vertex in L, say $G(xv_1) = 1$. This follows from Lemma 6.3(ii) if $k = 3$. On the other hand, if $k = 2$, then necessarily $m = 2$, since $k + m \geq 4$ and by the general assumption that $k \geq m$. Then $N_G(y) \cap L = \emptyset$ or $N_G(y) \cap M = \emptyset$ in order to avoid a cycle in $G - x$. By Lemma 5.6(ii), $N_G(x) \cap M \neq \emptyset$ or $N_G(x) \cap L \neq \emptyset$, respectively. Because k equals m, we may interchange L and M, if necessary, to obtain $G(xv_1) = 1$.

CLAIM 1. *The following adjacencies for x exist: $G(xz_1) = 1 = G(xz_2)$, and $G(xu) = 0$ for all $u \notin \{v_1, z_1, z_2, w_2, y\}$. Moreover, $G(xw_2) = 0$ if $d_S(w_2) \neq 0$ (that is, excepting the case* (5s)*).*

Proof. Recall that $\sigma = \{z_1, z_2\}$ (and $\sigma(x) = 0$). In the cases under consideration the centers z_1 and z_2 belong to the same connected component as v_1 in both of the acyclic graphs S and S^σ. Since $G^\sigma - y$ is acyclic, x can be adjacent in $G^\sigma - y$ only to v_1. Hence $G(xz_1) = 1 = G(xz_2)$. Also, if $d_S(w_2) > 0$, then w_2 is in the same connected component, from which it follows that $G(xw_2) = 0$ as required. □

CLAIM 2. *The following adjacencies for y exist: $G(yv) = 1$ holds for exactly one vertex $v \in S - \{w_2\}$, and either*

(i) *$v \in L$, say $G(yv_3) = 1$, in which case $k = 3$ and $m = 1$,*

(ii) *$v \in M$, say $G(yu_2) = 1$, in which case $k = 2$, $m = 2$.*

Moreover, $G(yw_2) = 1$ holds only in the case (5s).

Proof. For the first statement we observe that S is connected in the cases (7s) and (8s) and $S - w_2$ is connected in the case (5s). Hence y is adjacent to at most one vertex in $S - w_2$, since $G - x$ is acyclic. By Lemma 5.8, $G - x$ does not have isolated vertices, and therefore $G(yv) = 1$ for a unique $v \in S - \{w_2\}$ as required.

Now if y is not adjacent to a vertex of M, then $|M| = 1$ by Lemma 5.6(ii) and the fact that $G(xu) = 0$ for all $u \in M$. It follows that $k = 3$, and, consequently, y is adjacent to a vertex of L. On the other hand, if $G(yu) = 1$ for a $u \in M$, then $G(yv) = 0$ for all $v \in L$ to avoid a cycle in $G - x$, and in this case, $k = 2$ by Lemma 5.6. That $G(yw_2) = 1$ in the case (5s) follows from Lemma 6.1(i). In the other two cases, $G(yw_2) = 1$ would result in a cycle in $G - x$. □

These two claims together determine G with the exception of the value for $G(xy)$. We are ready to exclude the remaining cases.

Case (5s). Let $S = S_5(k, m)$. Now x is not adjacent to w_1 in G and neither is y. Hence in $G^\sigma - y$ the vertex w_1 is isolated, which contradicts Lemma 5.8.

Case (7s). Let $S = S_7(k, m)$. In both cases (i) and (ii) of Claim 2, $G(xy) = 1$ to avoid (7-4) being the subgraph induced by the vertices $\{x, z_1, w_1, z_2, v_2, w_2, y\}$. Now G contains a switch of the graph (7-4) if $k = 3$ and $m = 1$ (this is $G^{\{z_1\}} - \{w_1, w_2, u_1\}$), and G contains the graph (7-5') if $k = 2 = m$ (this is $G - \{u_1, v_2, z_2\}$).

Case (8s). Let $S = S_8(k, m)$. In both cases of Claim 2, $G(xy) = 1$ to avoid (6-1) being the subgraph induced by the vertices $\{x, z_1, w_1, w_2, z_2, y\}$. Now the set $\{x, z_1, w_1, w_2, z_2, y, u_1\}$ induces the graph (7-3').

This proves Theorem 3.5. □

7. Concluding remarks. Finding the critically cyclic graphs was done as follows: a program was written in `C` that listed, for a number n of vertices, a representative of each switching class that did not contain any acyclic switches. In a later phase,

when we were looking for critically cyclic graphs on n vertices, we only had to make sure that all critically cyclic graphs of lower order could not occur anymore in these graphs. The program was run in this way for up to 12 vertices. We used here the files from [9] which list generators for the switching classes up to isomorphism and up to complementation for up to 10 vertices.

A computer program in the functional language `Scheme` verified that the critically cyclic graphs found were in fact critically cyclic. Also, the authors verified this by hand.

In our proofs, not all of the critically cyclic graphs were used. The graphs that were not used are (8-10)–(8-15) and (9-3)–(9-5). Lemma 5.7 excludes the cycles C_8 and C_9. For the other graphs, except (8-12), the reason is that if they are induced subgraphs of any graph of order at least 10, then this graph also contains one of the cyclic graphs from Figures 3.1, 3.2, and 3.3 or it contains (8-12). The graph (8-12) does not occur in our proofs, because it is overruled by Lemmas 5.7 and 5.8 in the following sense. Consider a graph G of order 10 that does not have any acyclic switches. If G has two isolated vertices and is such that $G - x$ is acyclic and $G - \{x, y\}$ is special, then G contains an induced critically cyclic graph that was used in the proofs.

As an aside we note that our program found that the graphs (8-9) and (8-12) have a similar property: adding two vertices to either of these graphs in any way always results in a graph that contains a switch of one of the other critically cyclic graphs.

Acknowledgment. The authors thank the anonymous referee for many useful corrections and suggestions.

REFERENCES

[1] D. Acharya, *On characterizing graphs switching equivalent to acyclic graphs*, Indian J. Pure Appl. Math, 12 (1981), pp. 1187–1191.
[2] P. J. Cameron, *Cohomological aspects of two-graphs*, Math. Z., 157 (1977), pp. 101–119.
[3] D. G. Corneil and R. A. Mathon, eds., *Geometry and Combinatorics: Selected Works of J. J. Seidel*, Academic Press, Boston, 1991.
[4] A. Ehrenfeucht, T. Harju, and G. Rozenberg, *The Theory of* 2*-Structures*, World Scientific, Singapore, 1999.
[5] J. Hage, *Structural Aspects Of Switching Classes*, Ph.D. thesis, Leiden Institute of Advanced Computer Science, 2001; available online at http://www.cs.uu.nl/people/jur/2s.html.
[6] J. Hage and T. Harju, *Acyclicity of switching classes*, European J. Combin., 19 (1998), pp. 321–327.
[7] J. J. Seidel, *A survey of two-graphs*, in Intern. Coll. Teorie Combinatorie (Roma, 1973), Vol. I, Rome, 1976, Accad. Naz. Lincei, pp. 481–511; reprinted in [3].
[8] J. J. Seidel and D. E. Taylor, *Two-graphs, a second survey*, in Algebraic Methods in Graph Theory (Proceedings of the International Colloquium, Szeged, 1978), L. Lovasz and V. Sós, eds., Vol. II, North-Holland, Amsterdam, 1981, pp. 689–711; reprinted in [3].
[9] E. Spence, *Tables of Two-Graphs*, http://gauss.maths.gla.ac.uk/~ted/.
[10] J. H. van Lint and J. J. Seidel, *Equilateral points in elliptic geometry*, Proc. Konink. Nederl. Akad. Wetensch., 69 (1966), pp. 335–348; reprinted in [3].
[11] T. Zaslavsky, *A Mathematical Bibliography of Signed and Gain Graphs and Allied Areas*, Electronic J. Combin., Dynamic Survey No. DS8, http://combinatorics.org/Surveys/index.html.

SIAM J. DISCRETE MATH.
Vol. 18, No. 1, pp. 177–194

ON THE NUMBER OF MINIMUM CUTS IN A GRAPH*

L. SUNIL CHANDRAN† AND L. SHANKAR RAM‡

Abstract. We relate the number of minimum cuts in a weighted undirected graph with various structural parameters of the graph. In particular, we provide upper bounds for the number of minimum cuts in terms of the radius, diameter, minimum degree, maximum degree, chordality, girth, and some other parameters of the graph.

Key words. minimum cuts, circular partition

AMS subject classifications. 05C35, 05C99

DOI. 10.1137/S0895480103427138

1. Introduction. Let $G = (V, E)$ be a graph or a multigraph with positive weights on its edges. In this paper, we will use n to denote $|V|$. By an unweighted graph, we mean that all the edges have unit weight. Let $(A, \overline{A})$ denote a cut of G, defined by the subsets $A \subset V$ and $\overline{A} = V - A$. We denote by $E(A, \overline{A})$ the set of edges in the cut, i.e., $E(A, \overline{A}) = \{(u, v) \in E : u \in A \text{ and } v \in \overline{A}\}$. The weight of the cut $(A, \overline{A})$ is defined as the sum of weights on all the edges in $E(A, \overline{A})$, and will be denoted by $w(A, \overline{A})$. A minimum cut $(S, \overline{S})$ is one with the minimum weight over all cuts in G. (Some authors use the words *global minimum cuts* or *connectivity cuts* instead of minimum cuts). We will denote the weight of the minimum cut in G by $\lambda(G)$. Note that if G is unweighted, $\lambda(G)$ is the same as the edge connectivity of the graph, i.e., the minimum number of edges whose removal disconnects the graph.

Note that the minimum cut in a graph may not be unique. We use $\Lambda(G)$ to denote the number of minimum cuts in G. The problem of counting the number of minimum cuts in a weighted undirected graph arises in various aspects of network reliability, like testing the super-λ-ness of a graph [8], estimating the probabilistic connectedness of a stochastic graph in which edges are subject to failure with probability p [4, 5, 6, 30], and other areas [29]. For example, for a sufficiently small p, the probabilistic connectedness of G can be approximated as $P(G, p) \approx 1 - \Lambda(G)p^{\lambda(G)}(1-p)^{|E|-\lambda(G)}$, suggesting the importance of counting and bounding $\Lambda(G)$.

It is well known that for any weighted (positive weights) graph G, $\Lambda(G) \leq \binom{n}{2}$, and this upper bound is achieved if G is a cycle C_n of n nodes with each edge having weight $\frac{\lambda(G)}{2}$ [12, 7, 22]. It is interesting to explore whether there exist tighter bounds for $\Lambda(G)$ when the graph satisfies various properties. For example, Bixby [7] studies $\Lambda(G)$ in terms of the weight of the minimum cuts $\lambda(G)$ in the special case where all the edge weights are positive integers and $\lambda(G)$ is an odd integer. For this case, Bixby [7] shows that $\Lambda(G) \leq \lfloor \frac{3n}{2} \rfloor - 2$. In the case of unweighted simple graphs it is shown by Lehel, Maffray, and Preissmann [23] that if $\lambda(G) = k$, where $k \geq 4$ is an even positive

*Received by the editors May 1, 2003; accepted for publication (in revised form) December 17, 2003; published electronically August 19, 2004. A preliminary version of this paper appeared in *Proceedings of the 8th International Computing and Combinatorics Conference*, Lecture Notes in Comput. Sci. 2387, Springer-Verlag, New York, 2002.

http://www.siam.org/journals/sidma/18-1/42713.html

†Max-Planck Institute for Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany (sunil@mpi-sb.mpg.de). Most of this research was done while this author was in the Indian Institute of Science, Bangalore, and was supported in part by the Infosys Fellowship.

‡Department of Computer Science, Swiss Federal Institute of Technology, Haldeneggsteig 4, CH-8092, Zürich, Switzerland (lshankar@inf.ethz.ch).

integer, then $\Lambda(G) \leq \frac{2n^2}{(k+1)^2} + \frac{(k-1)n}{k+1}$. When $k > 5$ is an odd integer, they show that $\Lambda(G) \leq (1 + \frac{4}{k+5})n$. The inherent structural difference between graphs with odd and even edge connectivity was pointed out by Kanevsky [21] also.

In this paper, we provide upper bounds for $\Lambda(G)$ in terms of many other important parameters of graphs. We assume weighted graphs, unless otherwise specified. Multigraphs, as far as the results here are concerned, can be considered as a special case of weighted graphs, since the multiedges can be replaced by a single edge of appropriate weight without affecting the value of $\Lambda(G)$. Our only assumption about the weights is that they are positive. Note that, for the purposes of this paper, this assumption is equivalent to the assumption that the weights are at least 1, since multiplying the weights on every edge by the same constant will not change $\Lambda(G)$. While our upper bounds are valid for weighted undirected graphs and multigraphs, in most cases, the properties in terms of which the upper bounds are stated depend only on the structure of those graphs. In other words, the radius or minimum degree in terms of which we describe the upper bounds are those of the underlying unweighted simple graph and do not depend on the weights of the edges.

There is an abundance of literature regarding the determination of $\lambda(G)$ and finding a minimum cut in G. The problem of enumerating all the minimum cuts is considered by many authors [12, 28, 14, 15], and various data structures are invented to efficiently represent all the minimum cuts in a graph. (The currently fastest deterministic algorithm for computing all minimum cuts in a nonnegative, real weighted graph is due to Nagamochi, Nakamura, and Ishii [26].) The fact that the performance of some of these algorithms depends on the number of minimum cuts in the graph also makes it interesting to look for tighter upper bounds for $\Lambda(G)$ when G satisfies certain properties. (For example, a randomized algorithm due to Karger builds a data structure that represents all minimum cuts in $O(\Lambda(G) + n \log n)$ space.) See [14] for a brief survey of results regarding the enumeration of all minimum cuts.

The slightly different question of determining upper bounds for the number of approximate minimum cuts, i.e., those cuts having weight at most $f\lambda(G)$, where $f > 1$ is a constant, is considered in [31, 22, 27, 20]. For example, Karger [22], uses probabilistic analysis to show that there are at most $O(n^{2f})$ cuts of the above kind in a graph of n nodes. Nagamochi, Nishimura, and Ibaraki [27] show that the number of cuts of weight at most $\frac{4}{3}\lambda(G)$ is bounded above by $\binom{n}{2}$. Henzinger and Williamson [20] show an upper bound of $O(n^2)$ for the number of cuts of weight at most $\frac{3}{2}\lambda(G)$, extending the arguments of [27].

1.1. Our results. *Radius and diameter.* If $G = (V, E)$ is a connected graph, the *eccentricity* of a node $v \in V$ is defined as $e(v) = \max distance(v, u)$ over all the nodes $u \in V$. We define the radius of the graph G as $r(G) = \min_{v \in V} e(v)$. A vertex v is a central node if $e(v) = r(G)$. We define the diameter of G as $d(G) = \max_{v \in V} e(v)$. (Note that, in this paper, by "distance" we mean only the distances in the underlying unweighted graph. Thus radius, eccentricity, and diameter have nothing to do with the weights.) We show that the number of minimum cuts $\Lambda(G) \leq (r+1)n-(2r+1) \leq (d+1)n-(2d+1)$, where G is a weighted graph (positive weights) and r, d are the radius and diameter of G. As a special case, we observe that if there is a node which is a neighbor of every other node in the graph, i.e., if $r(G) = 1$, then $\Lambda(G) \leq 2n - 3$. We illustrate the tightness of this bound by constructing a weighted clique $\mathcal{K}_n$ for which $\Lambda(\mathcal{K}_n) = 2n - 3$.

Minimum and maximum degree. Let the minimum degree and maximum degree of G be δ and Δ, respectively. (Note that minimum and maximum degrees have nothing

to do with the weights, i.e., $\delta = \min_{u \in V} |N(u)|$ and $\Delta = \max_{u \in V} |N(u)|$, $N(u)$ being the set of neighbors of the node u). We show that $\Lambda(G) \leq (\frac{3n}{2(\delta+1)} + 1.5)n - (\frac{3n}{\delta+1} + 2)$ and $\Lambda(G) \leq \frac{(n-\Delta+3)n}{2} - (n - \Delta + 2)$. Note that these bounds become significant when the involved parameters are reasonably large. Also it is easy to get an upper bound involving both δ and Δ, by extending the techniques discussed in the paper.

Chordality. Let C be a simple cycle of a weighted undirected graph G. Any edge in the induced subgraph on the nodes of C, $G[C]$, other than the cycle edges themselves, is called a chord of C. C is called an induced cycle (or chordless cycle)[1] if and only if C does not have any chords. The length of the largest induced cycle in a graph G is called chordality of G. A graph G is called k-chordal if and only if the chordality of G is at most k. We show that $\Lambda(G) \leq \frac{(k+1)n}{2} - k$, where k is the chordality of the underlying unweighted (simple) graph corresponding to G. We also show the tightness of the bound by exhibiting a k-chordal graph G for arbitrarily large n such that $\Lambda(G) = \frac{(k+1)n}{2} - k$.

The word "chordality" originates from the well-known subclass of perfect graphs, the chordal graphs. A graph G is chordal if and only if there is no induced cycle of length 4 or more in G. We define the chordality of a chordal graph to be 3. All graphs other than chordal graphs have chordality ≥ 4. Some other important classes of graphs with low chordality value are the cocomparability graphs, chordal bipartite graphs, and weakly chordal graphs, all of which are known to be 4-chordal. It can be easily shown that asteroidal triple-free (AT-free) graphs have chordality at most 5. Thus, by substituting the appropriate values for chordality in the above upper bound, we obtain a list of results for various special classes of graphs.

Note that C_n (the cycle on n nodes) is the graph with maximum chordality amongst all graphs on n nodes. Also, it is a graph which contains the maximum number of minimum cuts possible, namely, $\binom{n}{2} = \frac{(n+1)n}{2} - n$. (In fact, our bound given above shows that C_n with each edge having weight $\frac{\lambda}{2}$ is the *only* graph which contains $\binom{n}{2}$ minimum cuts, the weight of the minimum cut being λ). The fact that the maximum value of $\Lambda(G)$ is achieved by the graph of largest chordality motivates a study of the influence of chordality on $\Lambda(G)$.

Girth. Girth is the length of the smallest cycle in G. We show that if G is an unweighted graph with girth g and minimum degree δ, then $\Lambda(G) < (\frac{n}{x+1} + 1)n - (\frac{2n}{x+1} + 1)$, where x is an integer greater than $e^{-2}(2(\delta - 1)^{\frac{g-2}{2}} - 2)$. Note that this is in contrast with the bound in terms of chordality, the length of the largest induced cycle.

The Fiedler value. The Laplacian matrix of a graph G is defined as $L = D - A$, where A is the adjacency matrix and D is the diagonal matrix whose (i, i)th entry is the degree of the ith vertex in G. The smallest eigenvalue of L can be shown to be equal to 0. The second smallest eigenvalue μ of L is sometimes known as the Fiedler value of G. This is a well-studied graph parameter. It can be easily shown that if G is a regular graph, then μ is equal to the gap between the two highest eigenvalues of the adjacency matrix A of G. Various structural parameters of a graph (like diameter, vertex connectivity, vertex and edge expansion, and bisection width) are known to be related to μ and in general to the eigenvalues of A or L. (See [13, 3, 25, 1].)

We observe that if μ is above the threshold value $1 + \frac{\delta}{n-\delta}$, where δ is the minimum

[1] An induced cycle or a chordless cycle is often called a "hole" in the perfect graph literature. Recall that the strong perfect graph *theorem* characterizes perfect graphs in terms of odd holes and antiholes.

degree, then all the minimum cuts in an unweighted graph G are single vertex cuts. In general, if μ is the Fiedler value and λ is the edge connectivity of G, we show that $\Lambda(G) \leq \frac{(\lfloor \frac{2\lambda}{\mu} \rfloor + 3)}{2} n - (\lfloor \frac{2\lambda}{\mu} \rfloor + 2)$ provided $\lfloor \frac{2\lambda}{\mu} \rfloor < \frac{n}{3}$.

2. Preliminaries. Consider an undirected graph $G = (V, E)$ with a weight function $w : E \to \Re^+$. Let U and W be disjoint subsets of V. Let $E(U, W) = \{(u, v) \in E : u \in U,\ v \in W\}$ be the set of edges between the vertices in U and the vertices in W. Also, let $w(U, W)$ be the sum of the weights on the edges in $E(U, W)$. As mentioned in the introduction, $\lambda(G)$ denotes the weight of a minimum cut, and $\Lambda(G)$ denotes the number of minimum cuts in G. Let $X \subset V$. We will denote the induced subgraph on X by $G[X]$.

LEMMA 2.1. *If $(S, \overline{S})$ is a minimum cut of a connected undirected graph G, then $G[S]$ and $G[\overline{S}]$ are connected.*

Proof. Suppose that $G[S]$ is not connected. Let $G[S_1]$ be a connected component of $G[S]$, where $S_1 \subset S$. Clearly $(S_1, \overline{S}_1)$ is a cut of G and $w(S_1, \overline{S}_1) < w(S, \overline{S})$ since $E(S_1, \overline{S}_1) \subset E(S, \overline{S})$. But this is a contradiction since $(S, \overline{S})$ is assumed to be a minimum cut. ∎

DEFINITION 2.2. *Let $(X, \overline{X})$ and $(Y, \overline{Y})$ be two cuts in a weighted undirected graph. $(X, \overline{X})$ and $(Y, \overline{Y})$ are said to cross each other if and only if all the four sets $X \cap Y$, $X \cap \overline{Y}$, $\overline{X} \cap Y$, and $\overline{X} \cap \overline{Y}$ are nonempty. Then $(X, \overline{X})$ and $(Y, \overline{Y})$ are called a crossing pair of cuts.*

LEMMA 2.3. *A pair of cuts $(S, \overline{S})$ and $(P, \overline{P})$ do not cross if and only if S (or $\overline{S}$) is a subset of P or $\overline{P}$. (That is, $S \subseteq P$, $S \subseteq \overline{P}$, $\overline{S} \subseteq P$, or $\overline{S} \subseteq \overline{P}$.)*

Proof. The proof follows from the definition of a crossing pair of cuts. ∎

LEMMA 2.4 (Bixby [7] and Dinic, Karzanov, and Lomosonov [12]). *Let $(X, \overline{X})$ and $(Y, \overline{Y})$ be a crossing pair of minimum cuts in a weighted undirected graph G. Let $A = X \cap Y$, $B = \overline{X} \cap Y$, $C = X \cap \overline{Y}$, and $D = \overline{X} \cap \overline{Y}$. Then,*

1. *$w(A, B) = w(B, D) = w(D, C) = w(C, A) = \frac{\lambda(G)}{2}$;*
2. *$w(A, D) = w(B, C) = 0$. That is, $E(A, D) \cup E(B, C) = \emptyset$.*

LEMMA 2.5. *If $(P, \overline{P})$ and $(S, \overline{S})$ are a crossing pair of minimum cuts, then $E(P, \overline{P}) \cap E(S, \overline{S}) = \emptyset$.*

Proof. $E(P, \overline{P}) \cap E(S, \overline{S}) = E(S \cap P, \overline{S} \cap \overline{P}) \cup E(\overline{S} \cap P, S \cap \overline{P}) = \emptyset$ by Lemma 2.4. ∎

DEFINITION 2.6. *A circular partition $\mathcal{C} = (U_0, U_1, U_2, \ldots, U_{k-1})$ (where $k \geq 4$) of the vertices of a graph G is a partition of the set of vertices V of G into disjoint nonempty subsets $U_0, U_1, \ldots, U_{k-1}$ such that the following hold:*

1. *$w(U_i, U_{i+1 \bmod k}) = \frac{\lambda(G)}{2}$ for $0 \leq i \leq k-1$.*
2. *If $i \neq j+1 \bmod k$ or $i \neq j-1 \bmod k$, then $w(U_i, U_j) = 0$; i.e., $E(U_i, U_j) = \emptyset$.*
3. *For $0 \leq i \leq k-1$, the cut $(U_i, \overline{U}_i)$—which is a minimum cut by conditions 1 and 2—does not cross with any other minimum cut $(A, \overline{A})$ in G.*

DEFINITION 2.7. *A cut $(A, \overline{A})$ is called a union cut with respect to a circular partition $\mathcal{C} = (U_0, U_1, \ldots, U_{k-1})$ if and only if there exists some i, $0 \leq i \leq k-1$, such that $A = \bigcup_{j=i}^{i+b-1 \bmod k} U_j$, where $2 \leq b \leq k-2$. (Note that both A and $\overline{A}$ contain at least two subsets in $\mathcal{C}$). The cut $(A, \overline{A})$ is called a subset cut with respect to $\mathcal{C}$ if and only if $A \subseteq U_i$ or $\overline{A} \subseteq U_i$ for some i.*

LEMMA 2.8. *Let $\mathcal{C} = (U_0, U_1, \ldots, U_{k-1})$ be a circular partition of G. Then any minimum cut $(S, \overline{S})$ of G is either a union cut or a subset cut with respect to $\mathcal{C}$. Moreover, every union cut with respect to $\mathcal{C}$ is a minimum cut in G.*

Proof. By the definition of a circular partition, $(S, \overline{S})$ does not cross with any of the minimum cuts $(U_i, \overline{U}_i)$. Therefore, by Lemma 2.3 $S \subseteq U_i$, $\overline{S} \subseteq U_i$, $S \subseteq \overline{U}_i$, or $\overline{S} \subseteq \overline{U}_i$. Suppose $(S, \overline{S})$ is not a subset cut. Then, we have $U_i \subseteq \overline{S}$ or $U_i \subseteq S$ for all i. Since by Lemma 2.1, $G[S]$ and $G[\overline{S}]$ are connected, we infer that $(S, \overline{S})$ is a union cut. Also, that a union cut is a minimum cut follows easily from conditions 1 and 2 of the definition of a circular partition (Definition 2.6). □

LEMMA 2.9. *Let G be a weighted undirected graph. Then G has a circular partition $\mathcal{C} = (U_0, U_1, \ldots, U_{k-1})$, where $k \geq 4$, if and only if there exists a crossing pair of minimum cuts in G.*

Proof. If there is a circular partition $\mathcal{C} = (U_0, U_1, \ldots, U_{k-1})$ with $k \geq 4$, clearly the minimum cuts $(U_0 \bigcup U_1, \overline{U_0 \bigcup U_1})$ and $(U_1 \bigcup U_2, \overline{U_1 \bigcup U_2})$ cross with each other. On the other hand, if there is a crossing pair of minimum cuts in G, namely, $(S_1, \overline{S}_1)$ and $(S_2, \overline{S}_2)$, due to a theorem of Bixby [7] and Dinic, Karzanov, and Lomosonov [12], there exists a circular partition $\mathcal{C} = (U_0, U_1, \ldots, U_{k-1})$ such that each of $S_1 \cap S_2$, $\overline{S}_1 \cap S_2$, $S_1 \cap \overline{S}_2$, and $\overline{S}_1 \cap \overline{S}_2$ equals $\bigcup_{i=a}^{b-1} U_i$ for appropriate choices for a and b. The "if" part of the Lemma follows immediately from this. □

For a circular partition $\mathcal{C}$ of G, let the *partition number* $p(\mathcal{C})$ be defined as the number of subsets in $\mathcal{C}$. We define the *partition number of the graph* G as follows.

DEFINITION 2.10. *The partition number $p(G)$ of a graph G is defined as $p(G) = 3$ if there is no circular partition for G. Otherwise, $p(G) = \max p(\mathcal{C})$, over all circular partitions $\mathcal{C}$ of G.*

Note that if there is a crossing pair of minimum cuts in G, then $p(G) \geq 4$, by Lemma 2.9. Otherwise, $p(G) = 3$.

DEFINITION 2.11. *By contraction of a subset of vertices $X \subset V$, we mean replacing all the vertices in X by a single vertex x and adding the edges (y, x) for each $y \in N(X)$, where $N(X)$ is the set of neighbors[2] of X. The weight of the edge (y, x) (where $y \in N(X)$) is assigned to be $w(y, x) = \sum_{z \in X} w(y, z)$, where $(y, z) \in E(G)$. We denote the graph obtained after the contraction operation by G/X. We will refer to the operation of undoing the effect of a contraction (i.e., restoring G from G/X) by putting back X in the place of x, as expanding the node x.*

LEMMA 2.12. *If $(S, \overline{S})$ is a minimum cut in a weighted undirected graph G such that no other minimum cut $(A, \overline{A})$ crosses with $(S, \overline{S})$, then $\Lambda(G) = \Lambda(G/S) + \Lambda(G/\overline{S}) - 1$.*

Proof. Note that since $(S, \overline{S})$ is a minimum cut, the value of the minimum cut in G/S and $G/\overline{S}$ will be the same as that in G. First we claim that $\Lambda(G) \leq \Lambda(G/S) + \Lambda(G/\overline{S}) - 1$. This can be seen by observing that corresponding to each minimum cut in G there is a minimum cut in either G/S or $G/\overline{S}$. This follows from the assumption that no minimum cut $(A, \overline{A})$ of G crosses with $(S, \overline{S})$ and so exactly one of the four cases $A \supset S$, $A \supset \overline{S}$, $\overline{A} \supset S$, or $\overline{A} \supset \overline{S}$ is true by Lemma 2.3. Thus the minimum cut $(A, \overline{A})$ remains intact either in G/S or $G/\overline{S}$. Also, $(S, \overline{S})$ appears in both G/S and $G/\overline{S}$, which accounts for subtracting 1. To see $\Lambda(G) \geq \Lambda(G/S) + \Lambda(G/\overline{S}) - 1$, observe that any minimum cut $(A, \overline{A})$ in G/S or in $G/\overline{S}$ has a corresponding minimum cut in G. For example, consider a minimum cut $(A, \overline{A})$ in G/S. Without loss of generality, let the node s in G/S (which corresponds to the contraction of S) be in A. When we expand s, clearly the minimum cut $(A \bigcup S - \{s\}, \overline{A \bigcup S - \{s\}})$ of G corresponds to the minimum cut $(A, \overline{A})$ of G/S. Moreover, it can be easily verified that the cuts of G which correspond to the cuts of G/S are distinct from the cuts of G which correspond

[2] $N(X) = \{u \in V - X : \text{There exists a node } v \in X \text{ such that } (u, v) \in E\}$.

to the cuts of $G/\overline{S}$ except for $(S, \overline{S})$, which is accounted for by subtracting 1. Hence the result follows. □

LEMMA 2.13. *If there are no crossing pairs of minimum cuts in G, then $\Lambda(G) \leq 2n-3$. Moreover, there exists a graph on n nodes, G_n (for every $n \geq 2$), such that $\Lambda(G_n) = 2n-3$.*

Proof. If $n = 2$, clearly $\Lambda(G) = 1$, and the lemma is true. Assume that for all graphs with number of nodes $< n$ (where $n \geq 3$), the lemma is true. Consider a graph G on n nodes with no crossing pairs of minimum cuts. If all the minimum cuts of G are singlenode cuts (i.e., of the form $(\{u\}, \overline{\{u\}})$), then clearly there are at most n minimum cuts. Then, $\Lambda(G) \leq n \leq 2n-3$. Otherwise, there is a minimum cut $(S, \overline{S})$ such that $|S| \geq 2$ and $|\overline{S}| \geq 2$. Let $G_1 = G/S$ and $G_2 = G/\overline{S}$. Also, let the number of nodes in G_1 and G_2 be n_1 and n_2, respectively. Since any minimum cut $(A, \overline{A})$ of G does not cross with $(S, \overline{S})$, by Lemma 2.12 we have $\Lambda(G) = \Lambda(G_1) + \Lambda(G_2) - 1$. Also, it can be easily verified that there will not be any crossing pair of minimum cuts in G_1 or G_2, since such a pair will give rise to a corresponding pair of crossing minimum cuts in G also, which is a contradiction. Thus since G_1 and G_2 have $< n$ vertices, we have $\Lambda(G) \leq 2n_1 - 3 + 2n_2 - 3 - 1 = 2(n+2) - 7 = 2n - 3$, since $n_1 + n_2 - 2 = n$. In Theorem 5.2, we show a way to assign weights to the edges of a clique K_n such that $\Lambda(K_n) = 2n-3$, illustrating the tightness of the bound given by this Lemma. □

3. Partition number, $p(G)$.

LEMMA 3.1. *Let G be a weighted undirected graph. If $(X, \overline{X})$ is a minimum cut of G such that no other minimum cut crosses with $(X, \overline{X})$, then $p(G/X) \leq p(G)$.*

Proof. Suppose $p(G/X) = p' > p(G) = p$. Clearly $p' \geq 4$ (since by definition of the partition number, $p(G) = p \geq 3$). Consider a circular partition $\mathcal{C}' = (U_0, U_1, \ldots, U_{p'-1})$ of G/X. Without loss of generality, assume that the node x obtained by contracting X is present in U_0. We claim that $\mathcal{C} = (W_0, \ldots, W_{p'-1})$, where $W_0 = (U_0 - \{x\}) \cup X$ and $W_i = U_i$ for $0 < i \leq p'-1$, is a circular partition for G. (This will clearly contradict the assumption that $p(G/X) = p' > p(G)$, proving the lemma).

Suppose $\mathcal{C} = (W_0, W_1, \ldots, W_{p'-1})$ is not a circular partition for G. Then, by definition of circular partition, there exists a minimum cut $(A, \overline{A})$ of G which crosses with $(W_i, \overline{W}_i)$ for some i.

Case 1. *$(A, \overline{A})$ does not cross with $(W_0, \overline{W}_0)$ but it crosses with $(W_i, \overline{W}_i)$ for some $i > 0$.* Since $(A, \overline{A})$ does not cross with $(W_0, \overline{W}_0)$ by Lemma 2.3, we have (1) $A \subseteq W_0$, (2) $\overline{A} \subseteq W_0$, (3) $A \subseteq \overline{W}_0$, or (4) $\overline{A} \subseteq \overline{W}_0$.

> *Case* 1.1. *$A \subseteq W_0$ or $\overline{A} \subseteq W_0$.* Since $W_0 \subseteq \overline{W}_i$, we have $A \subseteq \overline{W}_i$ or $\overline{A} \subseteq \overline{W}_i$, respectively. So, in both cases $(A, \overline{A})$ does not cross with $(W_i, \overline{W}_i)$ by Lemma 2.3, contradicting the assumption of Case 1.
>
> *Case* 1.2. *$A \subseteq \overline{W}_0$ or $\overline{A} \subseteq \overline{W}_0$*, i.e., $W_0 \subseteq \overline{A}$ or $W_0 \subseteq A$, respectively. Since $X \subseteq W_0$, $X \subseteq \overline{A}$ or $X \subseteq A$, which means that all the nodes in X are on the same side of the cut $(A, \overline{A})$. Thus, the cut $(A', \overline{A'})$ of G/X corresponding to $(A, \overline{A})$ is a minimum cut of G/X. Since $X \subseteq W_0 \subseteq \overline{W}_i$, $X \subseteq \overline{W}_i \cap A$ or $X \subseteq \overline{W}_i \cap \overline{A}$; i.e., all the nodes in X are either in $\overline{W}_i \cap A$ or $\overline{W}_i \cap \overline{A}$. But since $(W_i, \overline{W}_i)$ crosses with $(A, \overline{A})$, the sets $W_i \cap A$, $\overline{W}_i \cap A$, $W_i \cap \overline{A}$, and $\overline{W}_i \cap \overline{A}$ are nonempty. Therefore, clearly when we contract X to get G/X, the corresponding four sets $U_i \cap A'$, $\overline{U}_i \cap A'$, $U_i \cap \overline{A'}$, and $\overline{U}_i \cap \overline{A'}$ are also nonempty. This means that in G/X, $(A', \overline{A'})$ crosses with $(U_i, \overline{U}_i)$, contradicting the assumption that $\mathcal{C}' = (U_0, U_1, \ldots, U_{p'-1})$ is a circular partition for G/X.

Case 2. *$(A, \overline{A})$ crosses with $(W_0, \overline{W}_0)$.* Remember that by assumption $(A, \overline{A})$ does not

cross with $(X, \overline{X})$. We have the following four possibilities by Lemma 2.3: (1) $A \subseteq X$, (2) $\overline{A} \subseteq X$, (3) $A \subseteq \overline{X}$, or (4) $\overline{A} \subseteq \overline{X}$.

Case 2.1. $A \subseteq X$ *or* $\overline{A} \subseteq X$. Since $X \subseteq W_0$, we have $A \subseteq W_0$ or $\overline{A} \subseteq W_0$, respectively, which means that by Lemma 2.3, $(A, \overline{A})$ does not cross with $(W_0, \overline{W}_0)$ in both cases, contradicting the assumption of Case 2.

Case 2.2. $A \subseteq \overline{X}$ *or* $\overline{A} \subseteq \overline{X}$; i.e., $X \subseteq \overline{A}$ or $X \subseteq A$, which means that all the nodes in X are on the same side of $(A, \overline{A})$. Thus, the cut $(A', \overline{A'})$ of G/X corresponding to $(A, \overline{A})$ is a minimum cut of G/X. Since $X \subseteq W_0$, we have $X \subseteq W_0 \cap A$ or $X \subseteq W_0 \cap \overline{A}$, i.e., all the nodes in X are completely in $W_0 \cap A$ or $W_0 \cap \overline{A}$. But since $(W_0, \overline{W}_0)$ crosses with $(A, \overline{A})$, the sets $W_0 \cap A$, $\overline{W}_0 \cap A$, $W_0 \cap \overline{A}$, and $\overline{W}_0 \cap \overline{A}$ are nonempty. Therefore, clearly when we contract X to get G/X, the corresponding four sets $U_0 \cap A'$, $\overline{U}_0 \cap A'$, $U_0 \cap \overline{A'}$, and $\overline{U}_0 \cap \overline{A'}$ also are nonempty. This means that in G/X, $(A', \overline{A'})$ crosses with $(U_0, \overline{U}_0)$, contradicting the assumption that $\mathcal{C}' = (U_0, U_1, \ldots, U_{p'-1})$ is a circular partition for G/X.

Thus, we infer that no minimum cut $(A, \overline{A})$ can cross with any cut $(W_i, \overline{W}_i)$ in the circular partition $\mathcal{C} = (W_0, W_1, \ldots, W_{p'-1})$ for G. But $p(\mathcal{C}) = p' > p = p(G)$, a contradiction. Thus we have $p(G/X) \leq p(G)$. ∎

In the following lemma, we provide an upper bound for $\Lambda(G)$ in terms of the partition number. The tightness of the lemma will be established in Theorem 9.1.

LEMMA 3.2. *Let $G = (V, E)$ be a weighted undirected graph, where $|V| = n \geq 2$, and let the partition number $p(G) \leq p$. Then, $\Lambda(G) \leq \frac{(p+1)n}{2} - p$.*

Proof. The proof is by induction on n. If $n = 2$, by definition $p(G) = 3$, and it is easy to verify that $\Lambda(G) \leq \frac{(p+1)n}{2} - p$. Now, assume that for all graphs with number of nodes $< n$ (where $n \geq 3$), the lemma is true. Let G be a graph on n nodes ($n \geq 3$) with $p(G) = p$. If $p = 3$, then by Lemma 2.9 there are no crossing pairs of minimum cuts in G, and hence by Lemma 2.13 $\Lambda(G) \leq 2n - 3 = \frac{(p+1)n}{2} - p$. Now, let $p \geq 4$. By Lemma 2.9 there exists a circular partition $\mathcal{C} = (U_0, U_1, \ldots, U_{p-1})$ of G. If $p = n$, then $|U_i| = 1$ for each i and clearly $G = C_n$, a cycle of n nodes with each edge having weight $\frac{\lambda(G)}{2}$. Therefore, it has $\binom{n}{2} = \frac{n(n-1)}{2} = \frac{(n+1)n}{2} - n$ minimum cuts. If $p < n$, then there exists a $U_i \in \mathcal{C}$ such that $|U_i| \geq 2$. Let $G_1 = (V_1, E_1) = G/U_i$ and $G_2 = (V_2, E_2) = G/\overline{U}_i$ be the graphs obtained by contracting U_i and $\overline{U}_i$, respectively. Since $|U_i| \geq 2$ and $|\overline{U}_i| \geq 3$ (note that this follows from $p(G) \geq 4$), clearly $n > |V_1| \geq 4$ and $n > |V_2| \geq 3$. Let $p_1 = p(G_1)$ and $p_2 = p(G_2)$. Note that by the definition of a circular partition, no minimum cut $(A, \overline{A})$ of G crosses with $(U_i, \overline{U}_i)$. Hence by Lemma 2.12 we have $\Lambda(G) = \Lambda(G_1) + \Lambda(G_2) - 1$. Now, by the induction assumption, $\Lambda(G_j) \leq \frac{(p_j+1)n_j}{2} - p_j$ (for $j = 1, 2$) and we have

$$\Lambda(G) \leq \frac{(p_1+1)n_1}{2} - p_1 + \frac{(p_2+1)n_2}{2} - p_2 - 1. \tag{3.1}$$

By Lemma 3.1 we have $p_1 = p(G/U_i) \leq p$ and $p_2 = p(G/\overline{U}_i) \leq p$ since the minimum cut $(U_i, \overline{U}_i)$ does not cross with any other minimum cut in G. For $n_i \geq 2$ and $p_i \leq p$ ($i = 1, 2$), it is easy to verify that $\frac{(p_i+1)n_i}{2} - p_i \leq \frac{(p+1)n_i}{2} - p$. Substituting in inequality (3.1) and noting that $n_1 + n_2 - 2 = n$, we get $\Lambda(G) \leq \frac{(p+1)n}{2} - p$. ∎

In the rest of the paper, we show that various structural parameters of a graph can influence the partition number $p(G)$. Thus by means of Lemma 3.2 we relate the number of minimum cuts, $\Lambda(G)$, with many seemingly unrelated properties of the graph.

Remark. Please note that if $n \geq 2$ and $x \geq p$, then $\frac{(x+1)n}{2} - x \geq \frac{(p+1)n}{2} - p$. In most of the theorems below, we show that p is bounded above by a function $f(y)$ of y, where y is some parameter of G, thereby showing that $\Lambda(G) \leq \frac{(f(y)+1)n}{2} - f(y)$.

4. Radius and diameter. If $G = (V, E)$ is a connected graph, the *eccentricity* of a node $v \in V$ is defined as $e(v) = \max distance(v, u)$ over all the nodes $u \in V$. We define the radius of the graph G as $r(G) = \min_{v \in V} e(v)$. A vertex v is a central node if $e(v) = r(G)$. We define the diameter of G as $d(G) = \max_{v \in V} e(v)$. (Note that by "distance" we mean only the distances in the underlying unweighted graph. Thus radius, eccentricity, and diameter have nothing to do with the weights.)

THEOREM 4.1. *If r is the radius of a weighted undirected graph G, then $\Lambda(G) \leq (r+1)n - (2r+1)$ (where $n \geq 2$).*

Proof. Suppose there are no crossing pairs of minimum cuts in G. It follows by Lemma 2.13 that $\Lambda(G) \leq 2n - 3$. Since the radius is at least 1, it is easy to verify that $\Lambda(G) \leq (r+1)n - (2r+1)$ in this case. Otherwise, by Lemma 2.9 there exists a circular partition $\mathcal{C} = (U_0, U_1, \ldots, U_{p-1})$ for G, where $p = p(G) \geq 4$. Let $x \in U_i$ be a central node of G. Let $y \in U_{i+\lfloor \frac{p}{2} \rfloor \ mod\ p}$. Clearly $distance(x, y) \geq \lfloor \frac{p}{2} \rfloor$. That is, $r \geq \lfloor \frac{p}{2} \rfloor$ or $p \leq 2r + 1$. Now, by Lemma 3.2 we get $\Lambda(G) \leq (r+1)n - (2r+1)$. □

We note that the bound given by the above theorem can be tight. For example, consider C_{2n+1}, the cycle on $2n+1$ nodes. Clearly the radius of C_{2n+1} is n, and the number of minimum cuts $= \binom{2n+1}{2} = (n+1)(2n+1) - (2n+1)$.

Observe that similar arguments as given for the case of the radius also hold well for the diameter. Thus,

$$\Lambda(G) \leq (d+1)n - (2d+1).$$

This can also be verified from $\Lambda(G) \leq (r+1)n - (2r+1) \leq (d+1)n - (2d+1)$ by noting that $d \geq r$ and $n \geq 2$.

5. Universal node. An interesting special case of Theorem 4.1 occurs when $radius(G) = 1$. Then, there exists a node which is adjacent to every other node of the graph. (Such a node is called a *universal* node.) Thus, if there is a universal node in the graph, then $\Lambda(G) \leq 2n - 3$ by Theorem 4.1. In fact, a stronger statement is true.

THEOREM 5.1. *If there is a universal node u in G, then there cannot be any crossing pairs of minimum cuts in G.*

Proof. If there is a crossing pair of minimum cuts, then by Lemma 2.9 there is a circular partition $\mathcal{C} = (U_0, U_1, \ldots, U_{k-1})$ $(k \geq 4)$. Without loss of generality let $u \in U_0$. Clearly u cannot be adjacent to any node in U_2, by the definition of circular partition, contradicting the assumption that u is a universal node. □

Note that in a complete graph, K_n, every node is a universal node. Thus, there are no crossing pairs of minimum cuts in a clique. Below, we show a way to assign weights to the edges of K_n such that the number of minimum cuts $\Lambda(K_n) = 2n - 3$, thus illustrating that the bound of Lemma 2.13 is tight. Moreover, since the radius of a clique is 1, this is a tight example for Theorem 4.1 too. Since a complete graph is a chordal graph, the example below also illustrates the tightness of Theorem 7.2.

THEOREM 5.2. *For any $n \geq 2$ and $\lambda > 0$, there exists a weighted complete graph $\mathcal{K}_n$ such that $\lambda(\mathcal{K}_n) = \lambda$ and $\Lambda(\mathcal{K}_n) = 2n - 3$. Moreover, every node x of $\mathcal{K}_n$ defines a minimum cut $(\{x\}, \overline{\{x\}})$ of $\mathcal{K}_n$.*

Proof. For $n = 2$, it is trivial. For $n = 3$, let $\mathcal{K}_3$ be the triangle with each edge having weight $\frac{\lambda}{2}$. Clearly $\Lambda(\mathcal{K}_3) = 2 \cdot 3 - 3 = 3$. Also, note that every node x

in $\mathcal{K}_3$ defines a minimum cut $(\{x\}, \overline{\{x\}})$. Now inductively assume that there exists a weighted complete graph on $n-1$ nodes $\mathcal{K}_{n-1}$ $(n \geq 4)$ such that $\lambda(\mathcal{K}_{n-1}) = \lambda$, $\Lambda(\mathcal{K}_{n-1}) = 2(n-1) - 3 = 2n - 5$, and every node x of $\mathcal{K}_{n-1}$ defines a minimum cut $(\{x\}, \overline{\{x\}})$ in $\mathcal{K}_{n-1}$. We show how to construct a weighted complete graph $\mathcal{K}_n$ from $\mathcal{K}_{n-1}$ such that $\lambda(\mathcal{K}_n) = \lambda$, $\Lambda(\mathcal{K}_n) = 2n - 3$, and every node x of $\mathcal{K}_n$ defines a minimum cut $(\{x\}, \overline{\{x\}})$ in $\mathcal{K}_n$.

Let u be any node of $\mathcal{K}_{n-1}$. We remove u from $\mathcal{K}_{n-1}$ (along with the edges incident on it) and then add two other nodes u', u'' in its place. From each node y in $\mathcal{K}_{n-1}$ $(y \neq u)$, we add the edges (y, u') as well as (y, u'') to the new nodes and assign weights $w(y, u') = w(y, u'') = \frac{w(y,u)}{2}$. We also add the edge (u', u'') with $w(u', u'') = \frac{\lambda}{2}$.

It is easy to see that the new graph $\mathcal{K}_n$ is a complete graph. Let $S = \{u', u''\}$. Since $(\{u\}, \overline{\{u\}})$ is a minimum cut of $\mathcal{K}_{n-1}$, $w(S, \overline{S}) = \lambda(\mathcal{K}_{n-1}) = \lambda$. We claim that $\lambda(\mathcal{K}_n) = \lambda$. If not, there exists a cut $(A, \overline{A})$ in $\mathcal{K}_n$ such that $w(A, \overline{A}) < \lambda$. If both the nodes of $S = \{u', u''\}$ are present on the same side of the cut $(A, \overline{A})$, then the corresponding cut in $\mathcal{K}_{n-1}$ obtained by contracting S will also have weight $< \lambda$, contradicting the assumption that $\lambda(\mathcal{K}_{n-1}) = \lambda$. Therefore, without loss of generality, we can assume that $u' \in A$ and $u'' \in \overline{A}$. Then clearly $(u', u'') \in E(A, \overline{A})$. Also, for each y in $\mathcal{K}_n$ ($y \neq u'$ and $y \neq u''$), exactly one of the edges (u', y) or (u'', y) belongs to $E(A, \overline{A})$. Now recall that $w(y, u') = w(y, u'') = \frac{w(y,u)}{2}$ and $\sum_{y \neq u} w(y, u) = \lambda$ in $\mathcal{K}_{n-1}$. So we have $w(A, \overline{A}) \geq w(u', u'') + \frac{1}{2} \sum_{y \neq u', u''} (w(y, u') + w(y, u'')) = \frac{\lambda}{2} + \frac{\lambda}{2} = \lambda$, which is a contradiction to the assumption that $w(A, \overline{A}) < \lambda$. Hence, $\lambda(\mathcal{K}_n) = \lambda$ and $(S, \overline{S})$ is a minimum cut of $\mathcal{K}_n$. Also, since $\mathcal{K}_n$ is a clique, by Theorem 5.1 no minimum cut $(A, \overline{A})$ of $\mathcal{K}_n$ crosses with $(S, \overline{S})$. By Lemma 2.12 we have $\Lambda(\mathcal{K}_n) = \Lambda(\mathcal{K}_n/S) + \Lambda(\mathcal{K}_n/\overline{S}) - 1$. But $\mathcal{K}_n/S = \mathcal{K}_{n-1}$ and $\mathcal{K}_n/\overline{S} = \mathcal{K}_3$. Thus we have $\Lambda(\mathcal{K}_n) = 2n - 5 + 3 - 1 = 2n - 3$. Also it is easy to verify that every node x of $\mathcal{K}_n$ defines a minimum cut $(\{x\}, \overline{\{x\}})$ of $\mathcal{K}_n$. □

6. Maximum and minimum degree. The maximum degree $\Delta(G)$ (when it is reasonably high) can also constrain the number of minimum cuts $\Lambda(G)$.

THEOREM 6.1. *If Δ is the maximum degree of a weighted undirected graph G, then $\Lambda(G) \leq \frac{(n-\Delta+3)n}{2} - (n - \Delta + 2)$, where $n \geq 2$.*

Proof. Suppose there are no crossing pairs of minimum cuts in G; then by Lemma 2.13, $\Lambda(G) \leq 2n - 3 \leq \frac{(n-\Delta+3)n}{2} - (n - \Delta + 2)$, which will be true if $0 \leq n^2 - (\Delta + 3)n + (2\Delta + 2)$ or $0 \leq (n - \Delta - 1)(n - 2)$, which is true since $n \geq 2$ and $\Delta \leq n - 1$. Now, if there is a crossing pair of minimum cuts in G, then by Lemma 2.9 there is a circular partition $\mathcal{C} = (U_0, U_1, \ldots, U_{p-1})$ $(p = p(G) \geq 4)$. Without loss of generality, let the maximum degree node $u \in U_1$. Then, $|U_0 \cup U_1 \cup U_2| \geq \Delta + 1$ since every neighbor of u must be in U_0, U_1, or U_2. Thus, $p \leq 3 + (n - \Delta - 1) = n - \Delta + 2$ since each U_i of the circular partition must contain at least one node. By Lemma 3.2, $\Lambda(G) \leq \frac{(n-\Delta+3)n}{2} - (n - \Delta + 2)$. □

Interestingly, the minimum degree of the graph can also control the number of minimum cuts.

THEOREM 6.2. *If δ is the minimum degree of a weighted undirected graph G, then $\Lambda(G) \leq (\frac{3n}{2(\delta+1)} + 1.5)n - (\frac{3n}{\delta+1} + 2)$, where $n \geq 2$.*

Proof. If there are no crossing pairs of minimum cuts in G, it can be easily verified that $\Lambda(G) \leq 2n - 3 \leq (\frac{3n}{2(\delta+1)} + 1.5)n - (\frac{3n}{\delta+1} + 2)$ for $n \geq 2$. Otherwise consider a circular partition $\mathcal{C} = (U_0, U_1, \ldots, U_{p-1})$ $(p = p(G) \geq 4)$. Group the subsets in $\mathcal{C}$ into $\lfloor \frac{p}{3} \rfloor$ triplets $(U_{3i}, U_{3i+1}, U_{3i+2})$ for $0 \leq i \leq \lfloor \frac{p}{3} \rfloor - 1$. $|U_{3i}| + |U_{3i+1}| + |U_{3i+2}| \geq \delta + 1$ since

each neighbor of a node $u \in U_{3i+1}$ must be in one of the three sets in the corresponding triplet. Thus, $\lfloor \frac{p}{3} \rfloor (\delta + 1) \leq n$, and the result follows by Lemma 3.2. □

7. Chordality. Recall that the chordality of a graph is the length of the longest induced cycle in the graph. We provide an upper bound for $\Lambda(G)$ in terms of chordality in the following theorem. Its tightness is established in Theorem 9.1.

THEOREM 7.1. *If G is a weighted undirected graph with chordality k, then $\Lambda(G) \leq \frac{(k+1)n}{2} - k$, where $n \geq 2$.*

Proof. If there are no crossing pairs of minimum cuts in G, then by Lemma 2.13, $\Lambda(G) \leq 2n - 3 \leq \frac{(k+1)n}{2} - k$, since k is at least 3 by definition and $n \geq 2$. Otherwise, consider a circular partition $\mathcal{C}$ for G such that $p(\mathcal{C}) = p(G)$. If $p(\mathcal{C}) > k$, clearly there is an induced cycle in G with length $> k$, contradicting the k-chordality of G. It follows that $p(G) \leq k$. Therefore, by Lemma 3.2, $\Lambda(G) \leq \frac{(k+1)n}{2} - k$. □

THEOREM 7.2. *If G is a weighted chordal graph, then $\Lambda(G) \leq 2n - 3$, where $n \geq 2$. Moreover, there are no crossing pairs of minimum cuts in G. Also, there exists a weighted chordal graph G, for every $n \geq 2$, such that $\Lambda(G) = 2n - 3$.*

Proof. Since for chordal graphs $k = 3$ (by definition), $\Lambda(G) \leq 2n - 3$ follows from Theorem 7.1. If there is a crossing pair of minimum cuts in G, then there is a circular partition $\mathcal{C}$ for G with $p(\mathcal{C}) \geq 4$ by Lemma 2.9. This immediately implies an induced cycle of length ≥ 4, contradicting the fact that G is chordal. Finally, since complete graphs are chordal graphs, the construction of Theorem 5.2 establishes the tightness of this bound. □

There are some interesting special classes of graphs which can be shown to have low chordality value. We list below a few results which immeditately follow from Theorem 7.1.

Cocomparability graphs consist of graphs whose complements are comparability graphs. See [17] for the definition of a comparability graph. It can be shown that the chordality of cocomparability graphs is at most four; see, for example, [16]. Thus by Theorem 7.1 we have the following theorem.

THEOREM 7.3. *If G is a cocomparability graph on n vertices with positive edge weights, $\Lambda(G) \leq 2.5n - 4$.*

The class of weakly chordal graphs was introduced by Hayward in [19]. G is defined as a weakly chordal graph if and only if neither G nor the complement of G contains a chordless cycle of length at least 5. It follows from this definition that the chordality of weakly chordal graphs is at most 4. The class of weakly chordal graphs is quite a large one, as it contains the classes of cochordal graphs, chordal bipartite graphs, permutation graphs, trapezoid graphs, tolerance graphs, 2-threshold graphs, and others. Applying Theorem 7.1, we have the following result.

THEOREM 7.4. *If G is a weakly chordal graph on n vertices with positive weights on its edges, then $\Lambda(G) \leq 2.5n - 4$.*

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 AT-free if it contains no asteroidal triples. AT-free graphs provide a common generalization of interval, permutation, trapezoid, and cocomparability graphs.

THEOREM 7.5. *If G is an AT-free graph on n nodes with positive weights on the edges, then $\Lambda(G) \leq 3n - 5$.*

Proof. In view of Theorem 7.1, we just have to show that an AT-free graph doesn't contain a chordless cycle of length at least 6. Suppose it contains a chordless cycle of length 6 or more. Then clearly we can pick three points from this cycle such that they form an independent set and any two of them has a path between them, which avoids

the neighborhood of the third. But this is not possible since the graph is assumed to be AT-free. ∎

In fact there are many more special classes of graphs with low chordality value. The interested reader is referred to [10].

8. The stability number. The stability number α is defined as the size of the maximum independent set in the graph.

THEOREM 8.1.

$$\Lambda(G) \leq (\alpha+1)n - (2\alpha+1).$$

Proof. Since α is at least 1, if there is no crossing pair of minimum cuts in G, the theorem is clearly true. Otherwise it is easy to see that the partition number $p \leq 2\alpha+1$. ∎

9. A tight construction. We establish the tightness of Theorem 7.1 and Lemma 3.2 by the following construction.

THEOREM 9.1. *For each $k \geq 3$ and $\lambda > 0$, there exists an infinite family $\mathcal{G}$ of weighted undirected k-chordal graphs such that for each graph $G_n \in \mathcal{G}$ with n nodes (n being an integer of the form $k+q(k-2)$ for $q = 0, 1, \ldots$), $\Lambda(G_n) = \frac{(k+1)n}{2} - k$, weight of the minimum cut $= \lambda$, and $p(G_n) = k$. Moreover, every node u of G_n defines a minimum cut $(\{u\}, \overline{\{u\}})$.*

Proof. When $k = 3$, the family of cliques constructed in Theorem 5.2 has the desired properties; i.e., $\lambda(\mathcal{K}_n) = \lambda$, $\Lambda(\mathcal{K}_n) = 2n-3$, $p(\mathcal{K}_n) = 3$, and every node x of $\mathcal{K}_n$ defines a minimum cut $(\{x\}, \overline{\{x\}})$. In the rest of the proof we assume that $k \geq 4$. First note that $G_k = C_k$ and that the cycle on k nodes with each edge of weight $\frac{\lambda}{2}$ is a k-chordal graph with the desired properties, i.e., $\lambda(C_k) = \lambda$, $\Lambda(C_k) = \binom{k}{2} = \frac{(k+1)k}{2} - k$, and $p(C_k) = k$. Also every node $x \in C_k$ defines a minimum cut.

Now we show how to inductively construct the desired family. Let $G_n = (V, E)$ be a k-chordal graph on n nodes such that $\Lambda(G_n) = \frac{(k+1)n}{2} - k$, $p(G_n) = k$, and $\lambda(G_n) = \lambda$. Also assume that each node x in G_n defines a minimum cut. We describe how to construct a k-chordal graph $G_{n'} = (V', E')$ from G_n, where $|V'| = n' = n+k-2$, such that $\Lambda(G_{n'}) = \frac{(k+1)n'}{2} - k$, $\lambda(G_{n'}) = \lambda$, $p(G_{n'}) = k$, and every node of $G_{n'}$ defines a minimum cut, thereby proving the existence of the desired family.

Construction of $G_{n'}$ from G_n. Let u be any node in G_n. Then, let $V' = (V - \{u\}) \cup P$, where $P = \{y_1, y_2, \ldots, y_{k-1}\}$ are not already present in V. Let $N(u) = \{z_1, z_2, \ldots, z_l\}$ be the neighbors of u in G_n. Let $E' = (E - \{(u, z_i) : 1 \leq i \leq l\}) \cup \{(y_1, z_i) : 1 \leq i \leq l\} \cup \{(y_{k-1}, z_i) : 1 \leq i \leq l\} \cup \{(y_j, y_{j+1}) : 1 \leq j \leq k-2\}$, where the weights $w(z_i, y_1) = w(z_i, y_{k-1}) = \frac{w(z_i,u)}{2}$ for $1 \leq i \leq l$ and $w(y_j, y_{j+1}) = \frac{\lambda}{2}$ for $1 \leq j \leq k-2$. Thus, to get $G_{n'}$, we remove u from G_n along with the edges incident on it and add a path $(y_1, y_2, \ldots, y_{k-1})$ with each edge of weight $\frac{\lambda}{2}$. Also each neighbor z_i of u in G_n is now connected to y_1 and y_{k-1}. Moreover, the weight of (z_i, y_1) and (z_i, y_{k-1}) will be assigned half the weight of the edge (z_i, u) in G_n. It may be noted that the contracted graphs $G_{n'}/P = G_n$ and $G_{n'}/\overline{P} = C_k$ with each edge having weight $\frac{\lambda}{2}$.

CLAIM 9.2. *Let $(S, \overline{S})$ be a minimum cut of $G_{n'}$ which crosses with the cut $(P, \overline{P})$, where $P = \{y_1, y_2, \ldots, y_{k-1}\}$. Then, exactly one of the two edges (z_i, y_1) or (z_i, y_{k-1}) $(1 \leq i \leq l)$ will belong to $E(S, \overline{S})$. (Recall that $\{z_1, z_2, \ldots, z_l\}$ are the nodes in $G_{n'}$ which correspond to the neighbors of u in G_n.)*

Proof. First, we claim that both the nodes y_1 and y_{k-1} cannot be on the same side of the minimum cut $(S, \overline{S})$. Suppose for example, $\{y_1, y_{k-1}\} \subseteq S$. Because all

the edges from P to $\overline{P}$ are incident on either y_1 or y_{k-1}, $E(\overline{S} \cap P, \overline{S} \cap \overline{P}) = \emptyset$. (Note that $\overline{S} \cap P$ and $\overline{S} \cap \overline{P}$ will be nonempty since $(S, \overline{S})$ is assumed to cross with $(P, \overline{P})$.) Therefore the induced subgraph on $\overline{S}$ will be disconnected, which is a contradiction of Lemma 2.1 since $(S, \overline{S})$ is assumed to be a minimum cut. Now without loss of generality assume that $y_1 \in S$ and $y_{k-1} \in \overline{S}$. Then clearly one of the two edges (z_i, y_1) or (z_i, y_{k-1}) (since both these edges exist by construction) will belong to $E(S, \overline{S})$.

CLAIM 9.3. *$\lambda(G_{n'}) = \lambda(G_n) = \lambda$ and $(P, \overline{P})$ is a minimum cut of $G_{n'}$.*

Proof. First note that the cut $(P, \overline{P})$ in $G_{n'}$ has weight $w(P, \overline{P}) = \lambda$. This is easily seen from the fact that if we contract P, replacing the set P with the node u, we will get G_n (i.e., $G_{n'}/P = G_n$), and the cut $(P, \overline{P})$ in $G_{n'}$ will correspond to the single vertex minimum cut $(\{u\}, \overline{\{u\}})$ in G_n. Now we will show that every cut in $G_{n'}$ has weight at least λ, thereby establishing that $\lambda(G_{n'}) = \lambda$ and $(P, \overline{P})$ is a minimum cut of $G_{n'}$. Suppose $\lambda(G_{n'}) < \lambda$. Then let $(S, \overline{S})$ be a minimum cut of $G_{n'}$. If S (or $\overline{S}$) is a subset of P or $\overline{P}$, then one of the contracted graphs $G_{n'}/P = G_n$ or $G_{n'}/\overline{P} = C_k$ will contain a corresponding cut with the same value, which clearly will be a contradiction since $\lambda(G_n) = \lambda$ and $\lambda(C_k) = \lambda$. Thus by Lemma 2.3, $(S, \overline{S})$ must cross with $(P, \overline{P})$ in $G_{n'}$, which means $S \cap P, S \cap \overline{P}$, $\overline{S} \cap P$, and $\overline{S} \cap \overline{P}$ are nonempty. Now, by Claim 9.2, exactly one of the two edges (z_i, y_1) or (z_i, y_{k-1}) $(1 \leq i \leq l)$ will belong to $E(S, \overline{S})$. Recall that $w(z_i, y_1) = w(z_i, y_{k-1})$ and $\sum_{i=1}^{i=l} w(z_i, y_1) + \sum_{i=1}^{i=l} w(z_i, y_{k-1}) = w(\{u\}, \overline{\{u\}}) = \lambda(G_n) = \lambda$. Therefore, the total contribution to the weight of $(S, \overline{S})$ due to the edges of the form (y_j, z_i) $(j = 1, k-1$, and $1 \leq i \leq l)$ is $\frac{\lambda}{2}$. Also, since each edge in the path defined by the nodes in P has weight $\frac{\lambda}{2}$, it is clear that $w(S \cap P, \overline{S} \cap P) \geq \frac{\lambda}{2}$. Thus, considering both contributions, we infer that $w(S, \overline{S}) \geq \lambda$, contradicting the assumption that $w(S, \overline{S}) < \lambda$. So we have established that $\lambda(G_{n'}) = \lambda$, and therefore $(P, \overline{P})$ is a minimum cut of $G_{n'}$.

CLAIM 9.4. *No minimum cut $(S, \overline{S})$ of $G_{n'}$ crosses with the minimum cut $(P, \overline{P})$ and $\Lambda(G_{n'}) = \frac{(k+1)n'}{2} - k$.*

Proof. Suppose a minimum cut $(S, \overline{S})$ crosses with the minimum cut $(P, \overline{P})$ in $G_{n'}$. Then by Claim 9.2, exactly one of the two edges (z_i, y_1) or (z_i, y_{k-1}) $(1 \leq i \leq l)$ will belong to $E(S, \overline{S})$. Clearly both (z_i, y_1) and (z_i, y_{k-1}) are in $E(P, \overline{P})$. Thus $E(S, \overline{S}) \cap E(P, \overline{P}) \neq \emptyset$, contradicting Lemma 2.5. Therefore we infer that no minimum cut $(S, \overline{S})$ of $G_{n'}$ can cross with $(P, \overline{P})$. Now by applying Lemma 2.12 and noting that $n' = n + k - 2$, we have

$$\begin{aligned}
\Lambda(G_{n'}) &= \Lambda(G_n) + \Lambda(C_k) - 1 \\
&= \frac{(k+1)n}{2} - k + \frac{k(k-1)}{2} - 1 \\
&= \frac{(k+1)(n+k-2)}{2} - k \\
&= \frac{(k+1)n'}{2} - k
\end{aligned}$$

CLAIM 9.5. *Each node $x \in V'$ defines a minimum cut $(\{x\}, \overline{\{x\}})$ of $G_{n'}$.*

Proof. It is easy to check that the sum of weights on edges incident on the nodes of $\overline{P}$ has not changed from what it was in G_n. Also, it is clear that for $2 \leq i \leq k-2$, $w(\{y_i\}, \overline{\{y_i\}}) = \lambda$. Finally $w(\{y_1\}, \overline{\{y_1\}}) = w(y_1, y_2) + \sum_{i=1}^{i=l} w(y_1, z_i)$ $= \frac{\lambda}{2} + \frac{w(\{u\}, \overline{\{u\}})}{2} = \frac{\lambda}{2} + \frac{\lambda}{2} = \lambda$. The same argument also holds for y_{k-1}.

CLAIM 9.6. *$G_{n'}$ is k-chordal.*

Proof. Suppose there is an induced cycle C of length $> k$ in $G_{n'}$. We consider two cases, and show contradictions in both cases.

Case 1. C contains a node y_i from P other than y_1 or y_{k-1}. In this case, clearly C also must contain the nodes y_1 and y_{k-1}. Let z_i be the neighbor of y_1 in C from $\overline{P}$. Then z_i must also be the neighbor of y_{k-1} in C, since otherwise the edge (z_i, y_{k-1}) will form a chord for C. (Note that this edge exists by construction of $G_{n'}$.) Thus C will be $(z_i, y_1, y_2, \ldots, y_{k-1}, z_i)$, a cycle of length k, contradicting the assumption that $|C| > k$.

Case 2. C does not contain any node y_i from P other than y_1 or y_{k-1}. In this case clearly C is an (induced) subgraph of $G[\overline{P} \bigcup \{y_1, y_{k-1}\}]$. We claim that C must contain both the nodes y_1 and y_{k-1}. Otherwise, if, for example, C does not contain y_1, $|C| > k$ cannot be true, since the structure of $G[\overline{P} \bigcup \{y_{k-1}\}]$ is the same as that of G_n (except for the weights) and G_n is assumed to be k-chordal. Thus we infer that $\{y_1, y_{k-1}\} \subset C$. Now let z_i and z_j be the neighbors of y_1 in the cycle C. (Note that $z_i \neq y_{k-1}$ and $z_j \neq y_{k-1}$ since y_1 and y_{k-1} are not adjacent in $G_{n'}$.) Then z_i and z_j also must be the neighbors of y_{k-1}, in C. Otherwise, for example, if z_i is not a neighbor of y_{k-1} in C, clearly the edge (z_i, y_{k-1}) (which exists by construction) will form a chord for C, contradicting the fact that C is a chordless cycle. Now if z_i and z_j are the neighbors of both y_1 and y_{k-1} in C, clearly $C = (y_1, z_i, y_{k-1}, z_j, y_1)$, a cycle of length 4, contradicting the assumption that $|C| > k$, since $k \geq 4$ by assumption.

CLAIM 9.7. *$p(G_{n'}) = k$.*

Proof. By Claim 9.6, the chordality of $G_{n'}$ is k. Since the partition number is upper-bounded by the chordality (see the proof of Theorem 7.1), we have $p(G_{n'}) \leq k$. Now, since by Claim 9.4, the minimum cut $(P, \overline{P})$ does not cross with any other minimum cut in $G_{n'}$, by applying Lemma 3.1 and the induction assumption that $p(G_n) = k$, we get $k = p(G_n) = p(G_{n'}/P) \leq p(G_{n'}) \leq k$. Therefore, it follows that $p(G_{n'}) = k$. □

10. Girth and minimum degree. In this section we give an upper bound for $\Lambda(G)$ in terms of the girth and minimum degree in the case of *unweighted graphs*. The following classical results are not very difficult to prove.

LEMMA 10.1. *If $(S, \overline{S})$ is a minimum cut of an unweighted undirected graph G, then $|S| = 1$ or $|S| \geq \delta$, where δ is the minimum degree of G.*

LEMMA 10.2 (see Harary [18]). *If δ is the minimum degree and $\lambda(G) = \lambda$ is the size of a minimum cut (i.e., edge connectivity) in an unweighted undirected graph G, then $\delta \geq \lambda$.*

LEMMA 10.3. *Let $(X, \overline{X})$ be a minimum cut of an unweighted undirected graph G with girth g and minimum degree $\delta \geq 3$. Then $|X| = 1$ or $|X| \geq g$.*

Proof. Suppose $|X| > 1$. If the induced subgraph $G[X]$ on X is acyclic, then its average degree $d_x < 2$. Then clearly $|E(X, \overline{X})| \geq (\delta - d_x)|X| > |X|$, since $\delta \geq 3$. By Lemma 10.1, $|X| \geq \delta$. Thus $|E(X, \overline{X})| > \delta$ and this is a contradiction in view of Lemma 10.2, since $(X, \overline{X})$ is a minimum cut. Thus, $G[X]$ contains a cycle. Clearly the cycle contains at least g nodes since g is the girth of G. It follows that $|X| \geq g$. □

The following is a recent result from Alon, Hoory, and Linial [2]. (The reader may recall that the average degree of a graph is defined as the sum of degrees of the vertices divided by the total number of vertices in the graph.)

LEMMA 10.4 (see [2]). *The number of vertices n in a graph of girth g and average degree at least $d \geq 2$ satisfies $n \geq N(d, g)$, where*

$$N(d, 2r+1) = 1 + d\sum_{i=0}^{r-1}(d-1)^i,$$

$$N(d, 2r) = 2\sum_{i=0}^{r-1}(d-1)^i$$

for integer $r \geq 1$.

Remark. For $d > 2$, $N(d, g) \geq 2(d-1)^{\lfloor \frac{g-1}{2} \rfloor} - 2$.

LEMMA 10.5. *Let* $G(V, E)$ *be an unweighted undirected graph with girth* g *and minimum degree* $\delta \geq 3$. *If* $(X, \overline{X})$ *is a minimum cut of* G, *then either* $|X| = 1$ *or* $|X| > e^{-2}N(\delta, g)$, *where* $N(\delta, g)$ *is as defined in Lemma* 10.4.

Proof. Suppose that $(X, \overline{X})$ is a minimum cut and $|X| > 1$. We need to show that $|X| > e^{-2}N(\delta, g)$. Note that $e^{-2}N(\delta, 3) = e^{-2}(\delta+1) < \delta$ and $e^{-2}N(\delta, 4) = e^{-2}2\delta < \delta$. Therefore in view of Lemma 10.1, the present lemma is true for $g = 3$ and $g = 4$. Thus we can assume that $g \geq 5$.

Case 1. $g = 2r + 1$. Note that since $g \geq 5$, $r \geq 2$. Assume for contradiction that $|X| \leq e^{-2}N(\delta, 2r+1)$. We claim that the average degree d_x of the induced subgraph $G[X]$ on X is less than $\left(\delta - \frac{\delta-1}{r}\right)$. Suppose not. Then $d_x \geq \left(\delta - \frac{\delta-1}{r}\right)$. Note that since $\delta \geq 3$ and $r \geq 2$, $d_x \geq 2$. Also note that $G[X]$ is not acyclic (since $d_x \geq 2$) and its girth is at least $g = 2r + 1$. Thus, Lemma 10.4 is applicable and we have

$$\begin{aligned}
|X| &\geq N\left(\delta - \frac{\delta-1}{r}, 2r+1\right) \\
&= 1 + \left(\delta - \frac{\delta-1}{r}\right)\sum_{i=0}^{r-1}\left(\delta - \frac{\delta-1}{r} - 1\right)^i \\
&> 1 + \left(\delta - \frac{\delta}{r}\right)\sum_{i=0}^{r-1}\left(\delta - 1 - \frac{\delta-1}{r}\right)^i \\
&> e^{-2} + \delta\left(1 - \frac{1}{r}\right)\sum_{i=0}^{r-1}(\delta-1)^i\left(1 - \frac{1}{r}\right)^i \\
&> e^{-2} + \left(1 - \frac{1}{r}\right)^r \delta\sum_{i=0}^{r-1}(\delta-1)^i \\
&\geq e^{-2}N(\delta, 2r+1),
\end{aligned}$$

which contradicts the assumption that $|X| \leq e^{-2}N(\delta, 2r+1)$. (Note that the final step follows from the inequality $(1-x) \geq e^{-\frac{x}{1-x}}$ for $x < 1$. Thus $(1 - \frac{1}{r})^r \geq e^{-\frac{r}{r-1}} \geq e^{-2}$ since $r \geq 2$.)

Thus we infer that $d_x < \delta - \frac{\delta-1}{r}$. It follows that $|E(X, \overline{X})| > \frac{\delta-1}{r}|X|$. By Lemma 10.3 we have $|X| \geq g$ and thus $|E(X, \overline{X})| > \frac{(\delta-1)(2r+1)}{r} > \delta$, which is a contradiction in view of Lemma 10.2 since $(X, \overline{X})$ is a minimum cut. We infer that $|X| > e^{-2}N(\delta, 2r+1)$.

Case 2. $g = 2r$. Since $g \geq 5$, $r \geq 3$. Assume for contradiction that $|X| \leq e^{-2}N(\delta, 2r)$. We claim that the average degree d_x of the induced subgraph $G[X]$ on X is less than $\delta - \frac{\delta-1}{r-1}$. Suppose not. Then $d_x \geq \delta - \frac{\delta-1}{r-1} \geq 2$ since $\delta, r \geq 3$. Also $G[X]$ is not acyclic (since $d_x \geq 2$), and its girth is at least $g = 2r$. By applying Lemma 10.4,

we get

$$\begin{aligned}
|X| &\geq N\left(\delta - \frac{\delta-1}{r-1}, 2r\right) \\
&= 2\sum_{i=0}^{r-1}\left(\delta - \frac{\delta-1}{r-1} - 1\right)^i \\
&= 2\sum_{i=0}^{r-1}(\delta-1)^i\left(1 - \frac{1}{r-1}\right)^i \\
&> 2\left(1 - \frac{1}{r-1}\right)^{r-1}\sum_{i=0}^{r-1}(\delta-1)^i \\
&\geq e^{-2}N(\delta, 2r),
\end{aligned}$$

which is a contradiction to the assumption that $|X| \leq e^{-2}N(\delta, 2r)$. We infer that $d_x < \delta - \frac{\delta-1}{r-1}$. It follows that $|E(X, \overline{X})| > \frac{\delta-1}{r-1}|X|$. Applying Lemma 10.3, we get $|E(X, \overline{X})| > \frac{\delta-1}{r-1}2r > \delta$, since $\delta \geq 3$. This is a contradiction in view of Lemma 10.2, since $(X, \overline{X})$ is a minimum cut. We conclude that $|X| > e^{-2}N(\delta, 2r)$. □

THEOREM 10.6. *If G is an unweighted undirected graph with minimum degree δ (at least 3) and girth g, then $\Lambda(G) < (\frac{n}{x+1} + 1)n - (\frac{2n}{x+1} + 1)$, where $x = e^{-2}N(\delta, g)$ with $N(\delta, g) \geq 2(\delta-1)^{\lfloor \frac{g-2}{2} \rfloor} - 2$, is as defined in Lemma 10.4.*

Proof. Suppose there is a circular partition $\mathcal{C} = (U_0, U_1, \ldots, U_{p-1})$ for G. By Lemma 10.5, $|U_i| = 1$ or $|U_i| > x$. If $|U_i| = 1$, then we claim that $|U_{i-1 \bmod p}| > x$ and $|U_{i+1 \bmod p}| > x$. This is because if $|U_i| = 1$ (i.e., $(U_i, \overline{U}_i)$ defines a single-node minimum cut), then the size of the minimum cut $\lambda = \delta$. Now by the definition of circular partition, $|E(U_{i+1 \bmod p}, U_i)| = |E(U_{i-1 \bmod p}, U_i)| = \frac{\lambda}{2} = \frac{\delta}{2} > 1$, since $\delta \geq 3$. Thus, $|U_{i-1 \bmod p}| > 1$ and $|U_{i+1 \bmod p}| > 1$ since G is an *unweighted simple* graph and it follows from Lemma 10.5 that $|U_{i-1 \bmod p}| > x$ and $|U_{i+1 \bmod p}| > x$. Now, for each $0 \leq i \leq \lfloor \frac{p}{2} \rfloor - 1$, $|U_{2i} \bigcup U_{2i+1}| > x + 1$. Therefore, $\lfloor \frac{p}{2} \rfloor (x+1) < n$, and so $p < \frac{2n}{x+1} + 1$ and the theorem follows by Lemma 3.2. □

We don't know whether the bound given by Theorem 10.6 is tight. But in view of the following well-known conjecture,[3] it is not too bad.

CONJECTURE 10.7. *There exists a constant c such that for all integers $g, \delta \geq 3$, there is a graph $G(g, \delta)$ of minimum degree at least δ and girth at least g whose order (number of vertices) is at most $c(\delta-1)^{\lfloor \frac{g-1}{2} \rfloor}$.*

Let $x = c(\delta-1)^{\lfloor \frac{g-1}{2} \rfloor}$. If Conjecture 10.7 is true, let $G_1, G_2, \ldots, G_k$ be k copies of $G(g, \delta)$ for $k \geq g$. Now select a representative node from each of these copies. Let x_i be the representative node from G_i. Now connect together $G_1, G_2, \ldots, G_k$ by a cycle of length k passing through the representative nodes $x_1, x_2, \ldots, x_k$. Let the weight of the cycle edges (x_i, x_{i+1}) for $i = 1, \ldots, k-1$ be $\frac{\lambda}{2}$, where λ is the edge connectivity of G_i. Let this new graph be called G'. Clearly the number of minimum cuts in the new graph is at least $\binom{k}{2}$. Thus if $n = kx$ is the total number of nodes, $\lambda(G') = \Omega(\frac{n^2}{x^2})$ while the upper bound given by Theorem 10.6 is $O(\frac{n^2}{x})$.

[3] See, for example, [9, p. 164]. Also, see [11] for a brief history of the work toward constructing such families of graphs—the so-called high girth graphs—as proposed by the conjecture. The current best result is $O((\delta-1)^{\frac{3g}{4}})$, achieved, for example, by the Ramanujan graphs of [24].

11. Spectral bounds for $\Lambda(G)$.

11.1. A bound in terms of $\frac{\lambda}{\mu}$. Let X and Y be two disjoint subsets of V. Let $x = \frac{|X|}{n}$ and $y = \frac{|Y|}{n}$, and let ρ be the distance in G between X and Y; i.e., $\rho = \min_{u \in X, v \in Y} \mathit{distance}(u, v)$. Also let E_X denote the set of edges with both end points in the subset X. Let μ denote the Fiedler value, i.e., the second smallest eigenvalue of the Laplacian matrix of the graph. (See the introduction for the definition of Fiedler value.) Alon and Millman [3] proved the following result.

LEMMA 11.1. $\mu n \leq \frac{1}{\rho^2}(\frac{1}{x} + \frac{1}{y})(|E| - |E_X| - |E_Y|)$.

Let $(X, \overline{X})$ be a cut of G. Then, the following corollary to the above lemma is useful for us.

COROLLARY 11.2. *If $(X, \overline{X})$ is any cut in G, then $\mu \leq (\frac{1}{|X|} + \frac{1}{|\overline{X}|})|E(X, \overline{X})|$, where $E(X, \overline{X})$ is the set of edges between the disjoint sets X and $\overline{X}$.*

LEMMA 11.3. *If μ is the second smallest eigenvalue of the Laplacian matrix L of an unweighted undirected graph G, then for any minimum cut $(S, \overline{S})$, $|S| > \frac{n}{2}$ or $|S| \leq \lfloor \frac{2\lambda}{\mu} \rfloor$, where $\lambda = \lambda(G)$ is the edge connectivity of G.*

Proof. Suppose $|S| \leq \frac{n}{2}$. By Corollary 11.2, $\mu \leq (\frac{1}{|S|} + \frac{1}{n-|S|})|E(S, \overline{S})| \leq \frac{2\lambda}{|S|}$ since $(S, \overline{S})$ is a minimum cut. Since $|S|$ is an integer, we have $|S| \leq \lfloor \frac{2\lambda}{\mu} \rfloor$. ∎

COROLLARY 11.4. $\lfloor \frac{2\lambda}{\mu} \rfloor \geq 1$.

Proof. Let $(S, \overline{S})$ be a minimum cut with $|S| \leq |\overline{S}|$. Thus $|S| \leq \frac{n}{2}$. So, by Lemma 11.3, $1 \leq |S| \leq \lfloor \frac{2\lambda}{\mu} \rfloor$. ∎

THEOREM 11.5. *Let G be an unweighted undirected graph with $\lambda(G) = \lambda$ and let μ be the second smallest eigenvalue of the Laplacian matrix of G. If $\lfloor \frac{2\lambda}{\mu} \rfloor < \frac{n}{3}$, then* $\Lambda(G) \leq \frac{(\lfloor \frac{2\lambda}{\mu} \rfloor + 3)}{2} n - (\lfloor \frac{2\lambda}{\mu} \rfloor + 2)$.

Proof. Let $\gamma = \lfloor \frac{2\lambda}{\mu} \rfloor$. Suppose there are no crossing pairs of minimum cuts in G. Then by Lemma 2.13, $\Lambda(G) \leq 2n - 3 \leq \frac{(\gamma+3)}{2} n - (\gamma + 2)$, since $\gamma \geq 1$ (by Corollary 11.4). Otherwise, by Lemma 2.9, there exists a circular partition $\mathcal{C} = (U_0, U_1, \ldots, U_{p-1})$ of G with $p = p(G) \geq 4$. We claim that $p \leq \gamma + 2$. Suppose $p > \gamma + 2$. We will show a contradiction. Let U_j be the subset in $\mathcal{C}$ such that $|U_j| = \max_i |U_i|$. We will show first that $|U_i| \leq \gamma$ for all $i \in \{(j+1) \mod p, (j+2) \mod p, \ldots, (j+\gamma+1) \mod p\}$. If $|U_j| \leq \gamma$, this is clearly true since $|U_j| = \max_i |U_i|$ by assumption. Remembering that $(U_j, \overline{U}_j)$ is a minimum cut, by Lemma 11.3, if $|U_j| > \gamma$, then $|U_j| > \frac{n}{2}$. Therefore, $|V - U_j| < \frac{n}{2}$. Thus, for $i \in \{(j+1) \mod p, (j+2) \mod p, \ldots, (j+\gamma+1) \mod p\}$, $|U_i| \leq |V - U_j| < \frac{n}{2}$ and since $(U_i, \overline{U}_i)$ is a minimum cut, by Lemma 11.3, $|U_i| \leq \gamma$, as required.

Now let k be the smallest integer such that $|\bigcup_{i=j+1}^{j+k} U_i| > \gamma$. (Note that $\gamma + 1 \geq k \geq 2$.) Let $X = \bigcup_{i=j+1}^{j+k} U_i$. Note that since $|X - U_{j+k}| \leq \gamma$ and $|U_{j+k}| \leq \gamma$, we have

$$|X| \leq 2\gamma < \frac{2n}{3}. \tag{11.1}$$

But $(X, \overline{X})$ is a minimum cut by Lemma 2.8. Since $|X| > \gamma$, we have by Lemma 11.3 that $|X| > \frac{n}{2}$. That is, $|\overline{X}| < \frac{n}{2}$. It follows from inequality (11.1) that $\gamma < \frac{n}{3} < |\overline{X}| < \frac{n}{2}$, which is again a contradiction by Lemma 11.3 since $(\overline{X}, X)$ is a minimum cut. Thus we infer that $p \leq \gamma + 2$ and hence by Lemma 3.2, the theorem follows. (Please note that if $n \geq 2$ and $f \geq p$, then $\frac{(f+1)n}{2} - f \geq \frac{(p+1)n}{2} - p$.) ∎

The above bound is interesting for certain ranges of λ and μ, for example, when λ is relatively small and μ is not too small, say, not $O(\frac{1}{n})$. It may be noted that restricting λ to be bounded above by a constant doesn't imply that the value of μ also will be small. In fact there are δ-regular graphs for which the value of μ can be as high as $\Omega(\sqrt{\delta})$.

11.2. When μ is large. We observe that if μ is above a *threshold value*, then all the minimum cuts are single-vertex cuts, i.e., cuts of the form $(\{x\}, \overline{\{x\}})$, where x is a node. This is captured in the following theorem.

THEOREM 11.6. *Let μ be the second smallest eigenvalue of the Laplacian L of an unweighted undirected graph G. If $\mu > 1 + \frac{\delta}{n-\delta}$, where δ is the minimum degree, then all the minimum cuts in G are single-vertex minimum cuts.*

Proof. If there is a minimum cut $(S, \overline{S})$ with $|S| \leq |\overline{S}|$ and $|S| \neq 1$, then from Lemma 10.1, we have $|S| \geq \delta$. By Corollary 11.2, $\mu \leq (\frac{1}{|S|} + \frac{1}{|\overline{S}|})\lambda$. But this is a contradiction when $\mu > 1 + \frac{\delta}{n-\delta}$ since $\lambda \leq \delta$ by Lemma 10.2. □

The threshold given by the above theorem is tight. For example, it can be verified that for the graph C_4, the cycle graph on four nodes, $n = 4, \delta = 2$, and $\mu = 2$. Thus for C_4, $1 + \frac{\delta}{n-\delta} = \mu$, but it has minimum cuts which are not single-vertex cuts. In fact it is also possible to construct such examples with a larger number of nodes.

REFERENCES

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

[2] N. ALON, S. HOORY, AND N. LINIAL, *The Moore bound for irregular graphs*, Graphs Combin., 18 (2002), pp. 53–57.

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

[4] M. BALL AND J. PROVAN, *Calculating bounds on reachability and connectedness in stochastic networks*, Networks, 13 (1983), pp. 253–278.

[5] J. PROVAN AND M. BALL, *The complexity of counting cuts and of computing the probability that a graph is connected*, SIAM J. Comput., 12 (1983), pp. 777–788.

[6] M. BALL AND J. PROVAN, *Computing network reliability in time polynomial in the number of cuts*, Oper. Res., 32 (1984), pp. 516–521.

[7] R. BIXBY, *The minimum number of edges and vertices in a graph with edge connectivity n and m n-bonds*, Networks, 5 (1975), pp. 253–298.

[8] F. BOESCH, *Synthesis of reliable networks—a survey*, IEEE Trans. Reliability, R-35 (1986), pp. 240–246.

[9] B. BOLLOBÁS, *Extremal Graph Theory*, London Math. Soc. Monogr. 11, Academic Press, London, New York, 1978.

[10] A. BRANDSTÄDT, V. B. LE, AND J. P. SPINRAD, *Graph Classes: A Survey*, SIAM, Philadelphia, 1999.

[11] L. S. CHANDRAN, *A high girth graph construction*, SIAM J. Discrete Math., 16 (2003), pp. 366–370.

[12] E. DINIC, A. KARZANOV, AND M. LOMOSONOV, *The structure of a system of minimal edge cuts of a graph*, in Studies in Discrete Optimization, Nauka, Moscow, 1976, pp. 290–306 (in Russian).

[13] M. FIEDLER, *The algebraic connectivity of graphs*, Czechoslovak Math J., 23 (1973), pp. 298–305.

[14] L. FLEISCHER, *Building chain and cactus representations of all minimum cuts from Hao–Orlin in the same asymptotic run time*, J. Algorithms, 33 (1999), pp. 51–72.

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

[16] T. GALLAI, *Transitiv orientierbare graphen*, Acta. Math. Acad. Sci. Hungar., 18 (1967), pp. 25–66.

[17] M. C. GOLUMBIC, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[18] F. HARARY, *Graph Theory*, Addison–Wesley, Reading, MA, 1969.

[19] R. B. Hayward, *Weakly triangulated graphs*, J. Combin. Theory Ser. B, 39 (1985), pp. 200–208.
[20] M. Henzinger and D. Williamson, *On the number of small cuts*, Inform. Process. Lett., 59 (1996), pp. 41–44.
[21] A. Kanevsky, *Graphs with Odd and Even Edge Connectivity Are Inherently Different*, Tech. Rep. TAMU-89-10, Texas A & M University, College Station, TX, 1989.
[22] D. Karger, *Random sampling in cut, flow, and network design problems*, Math. Oper. Res., 24 (1999), pp. 383–413.
[23] J. Lehel, F. Maffray, and M. Preissmann, *Graphs with largest number of minimum cuts*, Discrete Appl. Math., 65 (1996), pp. 387–407.
[24] A. Lubotzky, R. Phillips, and P. Sarnak, *Ramanujan graphs*, Combinatorica, 8 (1988), pp. 261–271.
[25] B. Mohar, *Eigenvalues, diameter, and mean distance in graphs*, Graphs. Combin., 7 (1991), pp. 53–64.
[26] H. Nagamochi, S. Nakamura, and T. Ishii, *Constructing a cactus for minimum cuts of a graph in $o(mn+n^2\log n)$ time and $o(m)$ space*, Inst. Electron. Inform. Comm. Eng. Trans. Fundamentals, E86–D (2003), pp. 179–185.
[27] H. Nagamochi, K. Nishimura, and T. Ibaraki, *Computing all small cuts in an undirected network*, SIAM J. Discrete Math., 10 (1997), pp. 469–481.
[28] D. Naor and V. V. Vazirani, *Representing and enumerating edge connectivity cuts in RNC*, in Proceedings of the Second Workshop on Algorithms and Data Structures, Lecture Notes in Comput. Sci. 519, Springer-Verlag, New York, 1991, pp. 273–285.
[29] J. Picard and M. Queyranne, *On the structure of all minimum cuts in a network and applications*, Math. Programming Stud., 13 (1980), pp. 8–16.
[30] J. Provan, *Bounds on the reliability of networks*, IEEE Trans. Reliability, R-35 (1986), pp. 260–268.
[31] V. V. Vazirani and M. Yannakakis, *Suboptimal cuts: Their enumeration, weight, and number*, in Proceedings of the 19th International Colloquium on Automata, Languages and Programming, Lecture Notes in Comput. Sci. 623, Springer-Verlag, New York, 1992, pp. 366–377.

SIAM J. DISCRETE MATH.
Vol. 18, No. 1, pp. 195–206

ON THE BAND-, TREE-, AND CLIQUE-WIDTH OF GRAPHS WITH BOUNDED VERTEX DEGREE*

V. LOZIN† AND D. RAUTENBACH‡

Abstract. The band-, tree-, and clique-width are of primary importance in algorithmic graph theory due to the fact that many problems that are NP-hard for general graphs can be solved in polynomial time when restricted to graphs where one of these parameters is bounded. It is known that for any fixed $\Delta \geq 3$, all three parameters are unbounded for graphs with vertex degree at most Δ. In this paper, we distinguish representative subclasses of graphs with bounded vertex degree that have bounded band-, tree-, or clique-width. Our proofs are constructive and lead to efficient algorithms for a variety of NP-hard graph problems when restricted to those classes.

Key words. band-width, clique-width, tree-width, hereditary class, forbidden induced subgraph

AMS subject classifications. 05C78, 05C99

DOI. 10.1137/S0895480102419755

1. Introduction.

The band-, tree-, and clique-width are of primary importance in algorithmic graph theory due to the fact that many problems which are NP-hard for general graphs can be solved in polynomial time when restricted to graphs where one of these parameters is bounded. In fact, all problems expressible in monadic second-order logic become polynomial time solvable when restricted to graphs of bounded tree-width [7, 16], and all problems expressible in monadic second-order logic using quantifiers on vertices but not on edges become polynomial time solvable when restricted to graphs of bounded clique-width [10, 11]. This includes, for example, *maximum clique*, *independent set*, *minimum dominating* or *independent dominating set* problems as well as *k-colorability* (for fixed k), *maximum induced matching*, *induced path*, etc. Furthermore, graphs of bounded band-width are easily seen to be of bounded tree- and clique-width (cf. Proposition 1 below) and the mentioned problems are thus equally tractable.

It is known that for any fixed $\Delta \geq 3$, all three parameters are unbounded for graphs with vertex degree at most Δ. This is the case, for instance, for the so-called *walls* which are planar graphs of maximum vertex degree 3 (cf. [16]) and have arbitrarily large band-, tree-, and clique-width [9, 16].

The objective of the present paper is to distinguish subclasses of graphs of bounded vertex degree that have bounded band-, tree-, or clique-width. To this end, we study hereditary classes defined by forbidding large induced subgraphs. Two particular forbidden graphs play the critical role in our study. For integers $i, j, k \geq 0$, these are the two graphs $S_{i,j,k}$ and $T_{i,j,k}$ depicted in Figure 1(a) and 1(b), respectively. (Note that $S_{0,j,k}$ is a chordless path on $j + k + 1$ vertices and $T_{0,0,0}$ is simply a triangle.) By $\mathcal{S}$ we shall denote the class of graphs whose components are all of the form $S_{i,j,k}$, and by $\mathcal{T}$ the class of graphs whose components are all of the form $T_{i,j,k}$.

*Received by the editors December 16, 2002; accepted for publication (in revised form) July 17, 2003; published electronically August 19, 2004. This research was carried out while Dieter Rautenbach was visiting at Rutgers University. The assistance of RUTCOR and DIMACS is gratefully acknowledged.

http://www.siam.org/journals/sidma/18-1/41975.html

†RUTCOR, Rutgers University, 640 Bartholomew Rd., Piscataway, NJ 08854-8003 (lozin@rutcor.rutgers.edu).

‡Forschungsinstitut für Diskrete Mathematik, Universität Bonn, Lennestraße 2, 53113 Bonn, Ger-

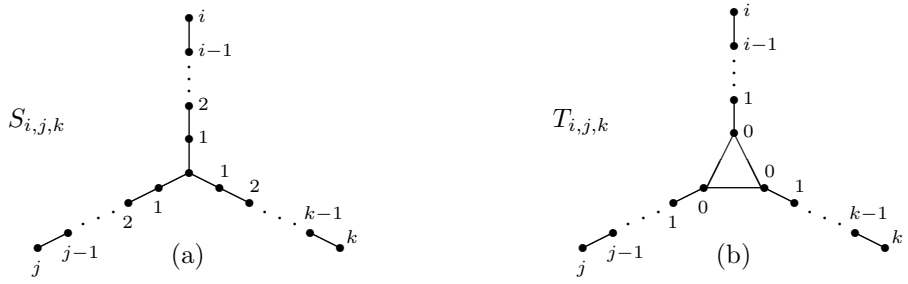


FIG. 1. *The graphs* (a) $S_{i,j,k}$ *and* (b) $T_{i,j,k}$.

We prove that in connected graphs of bounded vertex degree that do not contain S_{i_1,j_1,k_1} and T_{i_2,j_2,k_2} (for arbitrary values of the indices) as induced subgraphs we can find in polynomial time an induced path such that all vertices are within bounded distance of this path. This implies, in particular, that for any hereditary class of graphs $\mathcal{X}$ with bounded vertex degree that contains neither $\mathcal{S}$ nor $\mathcal{T}$, the tree-width and the clique-width are bounded by a constant. Under the assumption that $P \neq NP$, this result is best possible for the classes of graphs defined by finitely many forbidden induced subgraphs. This is a consequence of the following two facts: first, if a class of graphs $\mathcal{X}$ defined by finitely many forbidden induced subgraphs contains $\mathcal{S}$ or $\mathcal{T}$, then the independent dominating set problem is NP-hard in the class $\mathcal{X}$ [3], and second, this problem is polynomially solvable for graphs with bounded tree-width or clique-width. For the band-width, we discover some stronger conditions under which this parameter is bounded. All our results are algorithmical in the sense that they not only prove boundedness but also imply the existence of simple polynomial time algorithms for the respective width problems.

The outline of our presentation is as follows. In the next section we provide formal definitions and prove preliminary results. In section 3 we establish structural properties of graphs of bounded vertex degree that do not contain $S_{k,k,k}$ and $T_{k,k,k}$ as induced subgraphs. Section 4 contains our main results. In section 5 we illustrate the results by applying them to asteroidal triple-free graphs (these include comparability, permutation, and interval graphs) and chordal bipartite graphs.

2. Definitions and preliminary results. For standard graph-theoretical terminology, the reader is referred to any of the classical textbooks on graph theory. The set of vertices and set of edges of a graph G will be denoted by $V(G)$ and $E(G)$, respectively. Given a vertex $v \in V(G)$, $N_G(v)$ stands for the neighborhood of v in the graph G, i.e., the set of vertices of G adjacent to v, and $N_G[v]$ stands for the closed neighborhood of v, i.e. $N_G[v] = N_G(v) \cup \{v\}$. For a subset of vertices $U \subseteq V(G)$, $G - U$ is the subgraph of G obtained by deleting the vertices of U, $N_G(U)$ is the neighborhood of U, i.e., the set of vertices of G outside U that have a neighbor in U, and $N_G[U]$ is the closed neighborhood of U, i.e., $N_G[U] = N_G(U) \cup U$. A vertex v in a graph G is a *cutvertex* if the number of connected components of $G - \{v\}$ is strictly greater than that of G. A *block* in G is a maximal connected induced subgraph without cutvertices.

A graph G will be called H*-free* if G does not contain H as an induced subgraph. The class of all graphs containing no induced subgraphs in a set M will be denoted

many (rauten@or.uni-bonn.de).

by Free(M). We shall write Free$_k(M)$ to denote the graphs with vertex degree at most k in Free(M). It is well known that a class of graphs $\mathcal{X}$ is *hereditary* (i.e., closed under deletion of vertices) if and only if $\mathcal{X} = Free(M)$ for a certain set M (possibly infinite).

For a class of graphs $\mathcal{Y}$, we denote by $[\mathcal{Y}]_k$ the class of graphs G such that $G - U$ belongs to $\mathcal{Y}$ for a subset $U \subseteq V(G)$ of cardinality at most k. Also, given a class of graphs $\mathcal{Y}$, $[\mathcal{Y}]_B$ denotes the class of graphs whose blocks all belong to $\mathcal{Y}$.

We now define the three width parameters that we consider in the paper. Let L be an ordering of the vertices of a graph G; i.e., L is a bijection from $V(G)$ to $\{1, 2, \ldots, |V(G)|\}$. The *width* of the ordering is defined as

$$\max\{|L(u) - L(v)| \mid uv \in E(G)\}.$$

The *band-width* bw(G) of G [12] is the minimum integer k for which there exists an ordering L of width k for G.

A *tree decomposition* [2, 5, 15, 16, 17, 18] of a graph G is a pair $(T, \mathcal{W})$ where T is a tree and $\mathcal{W}$ assigns a set $W_t \subseteq V(G)$ to each vertex t of T such that

(i) $V(G) = \bigcup_{t \in V(T)} W_t$,
(ii) for every edge $uv \in E(G)$, there is some $t \in V(T)$ such that $u, v \in W_t$, and
(iii) for every vertex $u \in V(G)$, the set $\{t \in V(T) \mid u \in W_t\}$ induces a subtree of T.

The *width* of a tree decomposition $(T, \mathcal{W})$ is $\max_{t \in V(T)} |W_t| - 1$, and the *tree-width* tw(G) of G is the minimum width of a tree decomposition of G.

Finally, the *clique-width* cw(G) [8] of a graph G is the minimum number of labels needed to construct G using the following four operations:

(i) Create a new vertex v with label i (denoted $i(v)$).
(ii) Form the disjoint union of two labeled graphs G and H (denoted $G \oplus H$).
(iii) Join all vertices with label i to all vertices with label j ($i \neq j$, denoted by $\eta_{i,j}$).
(iv) Change the label of all vertices with label i to j (denoted by $\rho_{i \to j}$).

Every graph can be defined by an algebraic expression using these four operations. For instance, the cycle on five consecutive vertices a, b, c, d, and e can be defined as follows:

$$\eta_{4,1}(\eta_{4,3}(4(e) \oplus \rho_{4 \to 3}(\rho_{3 \to 2}(\eta_{4,3}(4(d) \oplus \eta_{3,2}(3(c) \oplus \eta_{2,1}(2(b) \oplus 1(a)))))))).$$

Such an expression is called a *k-expression* if it uses at most k different labels. Thus the clique-width of G is the minimum k for which there exists a k-expression defining G.

After these definitions we proceed to some auxiliary results.

PROPOSITION 1. *Let G be a graph and $U \subseteq V(G)$. Given an ordering of width k of the vertices of $G - U$, one can construct in polynomial time*

(i) *a tree decomposition of G of width at most $k + |U|$ and*
(ii) *a $(k + 2 + |U|)$-expression defining G.*

Proof. Let $n = |V(G) \setminus U|$. Let L be the ordering of width k of the vertices of $G - U$ and let $v_i \in V(G) \setminus U$ be such that $L(v_i) = i$ for $1 \leq i \leq n$.

Let T be such that $V(T) = V(G) \setminus U$ and

$$E(T) = \{v_i v_{i+1} \mid 1 \leq i \leq n - 1\}.$$

For $1 \leq i \leq n$, let

$$W_{v_i} = U \cup \{v_j \mid i \leq j \leq n,\ j - i \leq k\}.$$

Now it is easy to check that $(T, \mathcal{W})$ is a tree decomposition of G of width at most $k + |U|$ which implies (i).

To prove (ii) we describe a procedure to construct G using the four operations described above and $k + 2 + |U|$ different labels.

First, for all vertices $u \in U$, apply $i_u(u)$ such that $i_u \neq i_v$ for all $u, v \in U$ with $u \neq v$. Next, for all edges $uv \in E(G)$ with $u, v \in U$, apply η_{i_u, i_v}.

Let I be a set of $(k+1)$ different labels such that $I \cap \{i_u \mid u \in U\} = \emptyset$. Let i_0 be a label not in $I \cup \{i_u \mid u \in U\}$.

For r from 1 up to n, apply the following sequence of operations: Let i_s be the label presently assigned to vertex v_s with $\max\{1, r-k\} \leq s \leq r$. Let

$$i_r \in I \setminus \{i_s \mid \max\{1, r-k\} \leq s \leq r\}.$$

Apply $i_r(v_r)$. For all $u \in N_G(v_r) \cap U$, apply η_{i_r, i_u}. For all $v_s \in N_G(v_r) \cap \{v_1, v_2, \ldots, v_{r-1}\}$, apply η_{i_r, i_s}. Finally, apply $\rho_{i_{r-k} \to i_0}$ and proceed to $r+1$.

It is obvious that this procedure constructs G using $k + 2 + |U|$ different labels, which completes the proof. □

COROLLARY 1. *If G is a graph and $U \subseteq V(G)$, then*

(i) $\mathrm{tw}(G) \leq \mathrm{bw}(G - U) + |U|$ *and*

(ii) $\mathrm{cw}(G) \leq \mathrm{bw}(G - U) + 2 + |U|$.

PROPOSITION 2. *If the tree-width of graphs in a class $\mathcal{Y}$ is bounded by p, and a tree decomposition of width at most p for any graph in $\mathcal{Y}$ can be constructed in polynomial time, then one can construct in polynomial time a tree decomposition of width at most $p+k$ for any graph in $[\mathcal{Y}]_k$ and a tree decomposition of width at most p for any graph in $[\mathcal{Y}]_B$.*

Proof. For the graphs in $[\mathcal{Y}]_k$, one can proceed as in the proof of Proposition 1. Now let $G \in [\mathcal{Y}]_B$ and let $B_1, B_2, \ldots, B_l$ be the blocks of G. Let $(T(i), \mathcal{W}(i))$ be a tree decomposition of B_i for $1 \leq i \leq l$. We assume that all graphs G, $T(1)$, $T(2)$, $\ldots$, $T(l-1)$, and $T(l)$ are vertex disjoint. Let C be the set of cutvertices of G. For every cutvertex $u \in C$ and every block B_i such that $u \in V(B_i)$, we fix an arbitrary vertex $t(u,i) \in V(T(i))$ such that $u \in W(i)_{t(u,i)}$.

Let T be the tree such that $V(T) = C \cup (\bigcup_{i=1}^{l} V(T(i)))$, $\bigcup_{i=1}^{l} E(T(i)) \subseteq E(T)$, and $E(T)$ contains an edge $ut(u,i)$ for every cutvertex $u \in C$ and every block B_i such that $u \in V(B_i)$. Let $W_t = W(i)_t$ for all $t \in V(T(i))$, $1 \leq i \leq l$, and let $W_u = \{u\}$ for all $u \in B$.

It is straightforward to verify that $(T, \mathcal{W})$ is a tree decomposition of width at most p of G, and thus the proof is completed. □

PROPOSITION 3. *If the clique-width of graphs in a class $\mathcal{Y}$ is bounded by p, and a p-expression for any graph in $\mathcal{Y}$ can be constructed in polynomial time, then one can construct in polynomial time a $(2^k(p+1)-1)$-expression for any graph in $[\mathcal{Y}]_k$ and a $(p+2)$-expression for any graph in $[\mathcal{Y}]_B$.*

Proof. To prove the result for graphs in $[\mathcal{Y}]_k$, it suffices to consider the case $k = 1$. Let $G \in [\mathcal{Y}]_1$ and $G - \{v\} \in \mathcal{Y}$. Given a p-expression F for $G - \{v\}$, we first construct a $2p$-expression F' for $G - \{v\}$ in such a way that every labeled vertex $l(u)$ in F changes its label either to $l'(u)$ or to $l''(u)$ in F' depending on whether or not it is adjacent to v. Now we need only one additional label to obtain a $(2p+1)$-expression for G from F'. Thus, $\mathrm{cw}(G) \leq 2\mathrm{cw}(G - \{v\}) + 1$, and the proposition holds for graphs in $[\mathcal{Y}]_k$ by induction.

To prove the result for graphs in $[\mathcal{Y}]_B$, consider a graph $G \in [\mathcal{Y}]_B$. By assumption, the clique-width of every block of G is bounded by p. We show by induction on the

number of blocks that $\mathrm{cw}(G) \leq p+2$. Two additional labels needed to construct a $(p+2)$-expression for G will be denoted α and β. First, let G be a block and v be an arbitrary vertex in G. Any p-expression $F(G)$ defining G can trivially be modified into a $(p+1)$-expression $F'_v(G)$ in which v is the only vertex labeled with α. We then transform $F'_v(G)$ into a $(p+2)$-expression $F_v(G)$ by relabeling with β every vertex different from v. Now let H be a graph with $b>1$ blocks and let G be a block in H with a single cutvertex v. By deleting from H all the vertices of G except v, we obtain a graph with $b-1$ blocks. Such a graph can be defined by a $(p+2)$-expression T due to the inductive hypothesis. Assume the vertex v is created in T with label j. Then substituting $j(v)$ with $\rho_{\alpha\to j}(F_v(G))$ in T, we obtain a $(p+2)$-expression defining H. To see this, it is enough to notice that the label β is never renamed or used in any η-operation in T by the inductive hypothesis. This completes the proof. □

3. Properties of $(S_{k,k,k}, T_{k,k,k})$-free graphs of bounded vertex degree. In this section we establish a sequence of useful structural and algorithmical properties of graphs that do not contain $S_{k,k,k}$ and $T_{k,k,k}$ as induced subgraphs and have bounded vertex degree. Throughout this section we let k and Δ be fixed positive integers.

LEMMA 1. *If G is an $S_{k,k,k}$-free and $T_{k,k,k}$-free graph of vertex degree at most Δ, then G does not contain two vertex disjoint induced paths*

$$P : x_0 x_1 \dots x_{4\Delta k}$$

and

$$Q : y_0 y_1 \dots y_k$$

such that $y_0 x_{2\Delta k} \in E(G)$ and all edges between vertices of P and vertices of Q are incident to y_0 (see Figure 2).

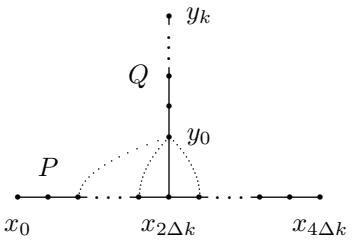


FIG. 2. *P and Q.*

Proof. For contradiction, we assume that two paths P and Q exist as in the statement. If $x_{2\Delta k}$ is the only neighbor of y_0 on P, then G is obviously not $S_{k,k,k}$-free, which is a contradiction. Hence, for some $2 \leq l \leq \Delta - 1$, let

$$N_G(y_0) \cap V(P) = \{x_{i_1}, x_{i_2}, \dots, x_{i_l}\}$$

such that $i_j < i_{j+1}$ for each $j = 1, \dots, l-1$. If $i_{j+1} - i_j \geq 2k$ for some $1 \leq j \leq l-1$, then the subgraph of G induced by the set

$$V(Q) \cup \{x_{i_j}, x_{i_j+1}, \dots, x_{i_j+(k-1)}\} \cup \{x_{i_{j+1}}, x_{i_{j+1}-1}, \dots, x_{i_{j+1}-(k-1)}\}$$

is isomorphic to $S_{k,k,k}$, which is a contradiction. Hence $i_{j+1} - i_j < 2k$ for each $1 \leq j \leq l-1$. Since $x_{2\Delta k} \in N_G(y_0)$, this implies that

$$i_1 > 2\Delta k - (\Delta - 2)2k = 4k,$$
$$i_l < 2\Delta k + (\Delta - 2)2k = 4\Delta k - 4k,$$

and hence the subgraph induced by the set

$$V(Q) \cup \{x_{i_1}, x_{i_1-1}, \ldots, x_0\} \cup \{x_{i_l}, x_{i_l+1}, \ldots, x_{4\Delta k}\}$$

contains either $S_{k,k,k}$ or $T_{k,k,k}$ as an induced subgraph. This contradiction completes the proof. □

LEMMA 2. *Let G be a connected $(S_{k,k,k}, T_{k,k,k})$-free graph of vertex degree at most Δ, and let*

$$P : x_0 x_1 \ldots x_{\Delta k}$$

be an induced path in G. Then the subgraph $G - N_G[V(P)]$ does not contain an induced cycle

$$C : y_1 y_2 \ldots y_l y_1$$

of length $l \geq 2(\Delta - 1)(k + 1)$.

Proof. For contradiction, we assume that $G - N_G[V(P)]$ contains an induced cycle C as in the statement. Since G is connected, we may consider a shortest path

$$Q : z_0 z_1 \ldots z_r$$

connecting P to C such that $z_0 \in V(P)$ and $z_r \in V(C)$. Note that $r \geq 2$.

The vertex z_1 has at most $(\Delta - 1)$ neighbors in $V(P)$. These neighbors divide P into at most Δ edge-disjoint segments, one of which has length at least $\frac{\Delta k}{\Delta} = k$. Without loss of generality, we assume that

$$N_G(z_1) \cap \{x_i, x_{i+1}, \ldots, x_{i+k-1}\} = \{x_i\}$$

for some $0 \leq i \leq \Delta k - k$.

If z_r is the only neighbor of z_{r-1} in $V(C)$, then G is obviously not $S_{k,k,k}$-free, which is a contradiction. Therefore, we assume that z_{r-1} has at least two neighbors in $V(C)$. Clearly, z_{r-1} has at most $(\Delta - 1)$ neighbors in $V(C)$. These neighbors divide C into at least two and at most $(\Delta - 1)$ edge-disjoint segments, one of which has length at least $\frac{2(\Delta-1)(k+1)}{\Delta-1} = 2k + 2$. Without loss of generality, we assume that

$$N_G(z_{r-1}) \cap \{y_j, y_{j+1}, \ldots, y_{j+s}\} = \{y_j, y_{j+s}\}$$

for some $1 \leq j \leq 2(\Delta - 1)(k + 1) - s$ and some $s \geq 2k + 2$. Note that $y_j y_{j+s} \in E(G)$ is possible ($s = l - 1$). Now the subgraph of G induced by the set (see Figure 3)

$$\{z_1, \ldots, z_{r-1}\} \cup \{x_i, x_{i+1}, \ldots, x_{i+k-1}\} \cup \{y_j, y_{j+1}, \ldots, y_{j+k}\} \cup \{y_{j+s}, y_{j+s-1}, \ldots, y_{j+s-k}\}$$

contains either $S_{k,k,k}$ or $T_{k,k,k}$ as an induced subgraph, which is a contradiction, and the proof is completed. □

COROLLARY 2. *If G is a connected $(S_{k,k,k}, T_{k,k,k})$-free graph of vertex degree at most Δ, then there is a set $U \subseteq V(G)$ of at most $c_1 = c_1(k, \Delta)$ vertices such that $G - U$ contains no induced cycle of length at least $2(\Delta - 1)(k + 1)$. Furthermore, such a set can be found in polynomial time.*

Proof. Let $c_1' = c_1'(k, \Delta) = 1 + \sum_{i=0}^{\Delta k - 1} (\Delta - 1)^i \Delta$. If $|V(G)| \leq c_1'$, then let $U = V(G)$. If $|V(G)| \geq c_1' + 1$, then from the degree constraint we derive that for any vertex u in the graph, there must exist a vertex v of distance Δk from u. Therefore, by considering the pairwise distances of the vertices of G, we can find in polynomial

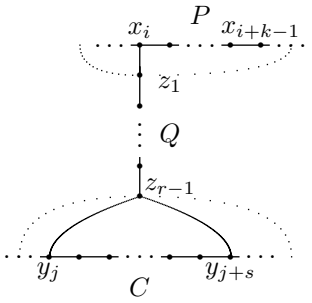


FIG. 3. *Parts of P, Q, and C.*

time an induced path P of length Δk. Since G has vertex degree at most Δ, the set $N_G[V(P)]$ contains at most $c_1'' = c_1''(k, \Delta) = (\Delta k + 1)(\Delta - 1) + 1$ vertices. Let $U = N_G[V(P)]$. By Lemma 2, the graph $G - U$ contains no induced cycle of length at least $2(\Delta - 1)(k + 1)$. The desired result now follows with $c_1 = \max\{c_1', c_1''\}$. ∎

LEMMA 3. *If G is a connected $(S_{k,k,k}, T_{k,k,k})$-free graph of vertex degree at most Δ, then G contains an induced path P such that for every vertex $u \in V(G)$,*

$$\mathrm{dist}_G(u, V(P)) = \min\{\mathrm{dist}_G(u, v) \mid v \in V(P)\} \leq 4\Delta k.$$

Moreover, such a path can be found in polynomial time.

Proof. A procedure that computes a path with the desired property can be roughly described as follows:

(0) Find an arbitrary induced path P which is maximal under inclusion.

(1) Denote by l the length of P, and by $x_0 x_1 \dots x_l$ the vertices of P. For each vertex $u \in V(G)$, compute $\mathrm{dist}_G(u, V(P))$. If all these distances are at most $4\Delta k$, then STOP: P is the path we sought for.

(2) Let $u \in V(G)$ be a vertex of G with $\mathrm{dist}_G(u, V(P)) > 4\Delta k$. If $l \leq 4\Delta k$, then a shortest path P' from u to P is longer than P. In this case, set $P := P'$ and go to step (1).

(3) By considering the pairwise distance between u and the vertices of P, determine the set

$$D = \{v \in V(P) \mid \mathrm{dist}_G(u, v) = \mathrm{dist}_G(u, V(P))\}.$$

Up to symmetry, we let r be the smallest index such that $x_r \in D$ and $r \geq l - 2\Delta k + 1$ (if there is no such r, define r to be the largest index such that $x_r \in D$ and $r \leq 2\Delta k - 1$). Let $Q : u = y_0 y_1 \dots y_s = x_r$ be a shortest path connecting u to x_r, where $s = \mathrm{dist}_G(u, V(P)) > 4\Delta k$. Set

$$P := x_{2\Delta k} x_{2\Delta k+1} \dots x_r y_{s-1} y_{s-2} \dots y_1 u$$

and go to step (1).

Obviously, every step of the procedure can be implemented in polynomial time. Now let us show that the loop (1)–(3) repeats at most $|V(G)|$ times. First, notice that since $l > 4\Delta k$ in step (3),

$$D \cap \{x_{2\Delta k}, x_{2\Delta k+1}, \dots, x_{l-2\Delta k}\} = \emptyset;$$

otherwise G would contain two vertex disjoint paths as described in Lemma 1, which is a contradiction. Therefore,

$$D \subseteq \{x_0, x_1, \ldots, x_{2\Delta k-1}\} \cup \{x_l, x_{l-1}, \ldots, x_{l-2\Delta k+1}\}.$$

Clearly, $D \neq \emptyset$, and hence in step (3) of the procedure we find a new induced path of length

$$r - 2\Delta k + s > l - 2\Delta k + 1 - 2\Delta k + 4\Delta k + 1 > l.$$

Summarizing, in both steps (2) and (3) we go to step (1) with a larger induced path. Therefore, after at most $|V(G)|$ loops the procedure terminates. This proves the correctness of the procedure and a polynomial time bound. □

4. Main results. We now proceed to our main results.

THEOREM 1. *If G is an $S_{k,k,k}$-free and $T_{k,k,k}$-free graph of vertex degree at most Δ that does not contain an induced cycle of length at least $2(\Delta-1)(k+1)$, then the band-width of G is bounded by a constant $c_2 = c_2(k, \Delta)$, and an ordering of the vertices of G of width at most c_2 can be determined in polynomial time.*

Proof. Clearly, we can deal with different components of G separately. Hence we may assume that G is connected.

By Lemma 3, we can find in polynomial time an induced path $P : x_0x_1 \ldots x_l$ such that $\mathrm{dist}_G(u, V(P)) \leq 4\Delta k$ for every vertex $u \in V(G)$.

We will now define a partition

$$V(G) = \bigcup_{i=0}^{l} V_i$$

assigning every vertex of G to some vertex of the path P (see Figure 4). For each $i = 0, \ldots, l$, the set V_i will consist of the vertices assigned to $x_i \in V(P)$.

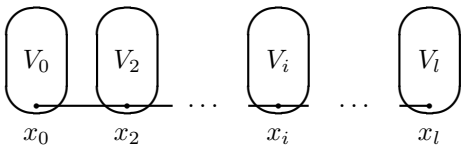


FIG. 4. *The partition $V(G) = V_1 \cup V_2 \cup \ldots \cup V_l$.*

Let $u \in V(G)$. As in the proof of Lemma 3, we determine in polynomial time the set

$$D(u) = \{v \in V(P) \mid \mathrm{dist}_G(u, v) = \mathrm{dist}_G(u, P)\}.$$

Since G has vertex degree at most Δ and $\mathrm{dist}_G(u, V(P)) \leq 4\Delta k$, we have

$$|D(u)| \leq c_2' = c_2'(k, \Delta) = 1 + \sum_{\nu=0}^{4\Delta k} (\Delta - 1)^\nu \Delta.$$

We choose an arbitrary vertex $x_i \in D(u)$ and assign u to x_i, i.e., $u \in V_i$. The main property to establish the boundedness of the band-width is expressed in the following claim.

CLAIM 1. *If $u \in V_i$ and $v \in V_j$ for some $0 \le i < j \le l$ such that*

$$j - i \ge 2(\Delta - 1)(k + 1) + 4(\Delta - 1)(k + 1)c_2',$$

then $uv \notin E(G)$.

Proof of the claim. For contradiction, we assume that $uv \in E(G)$.

Let $x_{i'}$ be the vertex in $D(u)$ with the maximum index; then

$$i' - i \le 2(\Delta - 1)(k + 1)c_2'.$$

To show this, consider two consecutive vertices x_s and x_t in $D(u)$, i.e., for each $r \in \{s+1, \ldots, t-1\}$, $x_r \notin D(u)$. Then $t - s < 2(\Delta-1)(k+1)$, since otherwise the vertices of two shortest paths connecting u to x_s and x_t, together with the vertices $x_s, \ldots, x_t$, would induce a graph containing a chordless cycle of length at least $2(\Delta - 1)(k + 1)$.

Analogously, if $x_{j'}$ is the vertex in $D(v)$ with the minimum index, then

$$j - j' \le 2(\Delta - 1)(k + 1)c_2'.$$

Consequently,

$$j' - i' \ge j - i - 4(\Delta - 1)(k + 1)c_2' \ge 2(\Delta - 1)(k + 1).$$

But now the vertices of two shortest paths connecting u and v, respectively, to $x_{i'}$ and $x_{j'}$, together with the vertices $x_{i'}, x_{i'+1}, \ldots, x_{j'}$, induce in G a graph containing a chordless cycle of length at least $2(\Delta - 1)(k + 1)$. This contradiction completes the proof of the claim.

Since G has vertex degree at most Δ, by Lemma 3, we have $|V_i| \le c_2' = c_2'(k, \Delta)$ for every $0 \le i \le l$. Let $v_1 v_2 \ldots v_n$ be an ordering of the vertices of G in which u comes before v whenever $u \in V_s$ and $v \in V_t$ with $s < t$. By the above claim, it is obvious that $v_i v_j \in E(G)$ implies

$$|j - i| \le c_2 = (2(\Delta - 1)(k + 1) + 4(\Delta - 1)(k + 1)c_2')c_2'$$

and the proof is completed. □

Whereas band-width is very sensitive to the deletion of a bounded number of vertices from the graph (consider, e.g., the star $K_{1,n-1}$), the tree-width and the clique-width are not. Combining Proposition 1, Corollary 2, and Theorem 1, we obtain the following corollary.

COROLLARY 3. *If G is an $S_{k,k,k}$-free and $T_{k,k,k}$-free graph of vertex degree at most Δ, then*

(i) $\mathrm{tw}(G) \le c_2 + c_1$ *and*

(ii) $\mathrm{cw}(G) \le c_2 + 2 + c_1$.

Furthermore, a tree decomposition of G of width at most $c_2 + c_1$ and a $(c_2 + 2 + c_1)$-expression of G can be found in polynomial time.

Note that given the conclusion (i) of Corollary 3, it follows from [1] or [14] that a tree decomposition of small width can be found efficiently. Nevertheless, the procedure we proposed in the proof of Proposition 1 is much simpler.

With the help of Propositions 2 and 3, a stronger result can be derived from Corollary 3 for the tree- and clique-width of graphs with bounded vertex degree. Recall that $\mathcal{S}$ denotes the class of graphs whose components are all of the form $S_{i,j,k}$, and $\mathcal{T}$ the class of graphs whose components are all of the form $T_{i,j,k}$.

THEOREM 2. *Let $\mathcal{X}$ be a hereditary class of graphs with bounded vertex degree. If $\mathcal{S} \not\subseteq \mathcal{X}$ and $\mathcal{T} \not\subseteq \mathcal{X}$, then*

(i) *the tree-width of graphs in $\mathcal{X}$ is bounded by a constant c_3, and a tree decomposition of width at most c_3 can be constructed in polynomial time, and*
(ii) *the clique-width of graphs in $\mathcal{X}$ is bounded by a constant c_4, and a c_4-expression can be constructed in polynomial time.*

Proof. If $\mathcal{S} \not\subseteq \mathcal{X}$ and $\mathcal{T} \not\subseteq \mathcal{X}$, we may consider two graphs $H \in \mathcal{S} \setminus \mathcal{X}$ and $J \in \mathcal{T} \setminus \mathcal{X}$ such that every component of H is of the form $S_{k,k,k}$ and every component of J is of the form $T_{k,k,k}$ for some fixed $k \geq 1$.

Denote by s and t the number of components of H and J, respectively. Also, let H_i and J_i denote, respectively, the subgraphs of H and J induced by the vertices of exactly i components. In particular, $H_1 = S_{k,k,k}$, $J_1 = T_{k,k,k}$, $H_s = H$, and $J_t = J$.

For every graph G with vertex degree at most Δ, there are at most $p = p(k, \Delta)$ vertices in the closed neighborhood of the set of vertices that induces a subgraph isomorphic to either $S_{k,k,k}$ or $T_{k,k,k}$. Therefore, the following inclusion holds:

$$\begin{aligned} \mathrm{Free}_\Delta(H_s, J_t) \quad \subseteq \quad & \mathrm{Free}_\Delta(H_1, J_1) \cup [\mathrm{Free}_\Delta(H_{s-1}, J_1)]_p \\ & \cup [\mathrm{Free}_\Delta(H_1, J_{t-1})]_p \cup [\mathrm{Free}_\Delta(H_{s-1}, J_{t-1})]_{2p}. \end{aligned}$$

From this inclusion and Propositions 2 and 3 we conclude by induction on s and t that the tree-width and clique-width of graphs in the class $\mathcal{X} \subseteq \mathrm{Free}_\Delta(H, J)$ are bounded. The basis of induction is provided by Corollary 3. Notice that s and t are constants associated with the class $\mathcal{X}$. □

5. Examples. An *asteroidal triple* in a graph is an independent set of three vertices such that for any two vertices in the set, there is a path between these two vertices avoiding the neighborhood of the third vertex [6]. A graph is asteroidal triple-free, or AT-free for short, if it contains no asteroidal triple. Asteroidal triple-free graphs include several well-known graph classes such as comparability graphs, permutation graphs, or interval graphs. All three graph parameters studied in this paper are unbounded in the class of AT-free graphs (since they are unbounded even for permutation graphs [13]), and several fundamental graph problems remain NP-hard in that class. However, according to our main results, the situation changes crucially whenever we deal with AT-free graphs with bounded vertex degree. Formally, we have the following.

PROPOSITION 4. *The band-, tree-, and clique-width of AT-free graphs of vertex degree at most Δ are bounded by a constant $c_5 = c_5(\Delta)$.*

Proof. The proposition follows immediately from Theorem 1, Proposition 1, and the observation that $S_{2,2,2}$, $T_{2,2,2}$, and induced cycles of length at least 6 contain an asteroidal triple. □

As another important example, let us mention chordal bipartite graphs, i.e., bipartite graphs without induced cycles of length at least 6. This class has applications in the study of linear programming, as the bipartite adjacency matrices of chordal bipartite graphs are totally balanced. Note that the class of chordal bipartite graphs is not a subclass of AT-free graphs, since $S_{2,2,2}$ is not forbidden in this class.

The clique-width (and hence the band- and tree-width) of chordal bipartite graphs is unbounded, since it is unbounded even for bipartite permutation graphs [4], a proper subclass of chordal bipartite graphs. We use Propositions 2 and 3 and Theorem 2 in order to show that chordal bipartite graphs with vertex degree at most 3 have bounded tree- and clique-width.

PROPOSITION 5. *The tree- and clique-width of chordal bipartite graphs with vertex degree at most* 3 *are bounded by a constant.*

Proof. We shall prove the proposition by showing that every chordal bipartite graph G with vertex degree at most 3 containing $S_{2,2,2}$ as an induced subgraph has a cutvertex. We denote by x the center of $S_{2,2,2}$; by y_1, y_2, and y_3 the vertices of degree 2; and by z_1, z_2, and z_3 the respective vertices of degree 1 in $S_{2,2,2}$.

Assume y_1 is not a cutvertex in G, and consider a shortest path $P : p_0p_1 \dots p_k$ connecting the vertex $z_1 = p_0$ to a vertex $p_k \in \{y_2, y_3, z_2, z_3\}$ in the subgraph $G-\{y_1\}$. If $p_k = z_2$, then k is even. Notice that the even-indexed vertices of P different from p_k are not adjacent to y_2, since otherwise P is not a shortest path. Let j be the largest index with $y_1p_j \in E(G)$ (possibly $j = 0$). But now $y_1p_j \dots p_ky_2xy_1$ is an induced cycle of length at least 6 in G, a contradiction. Analogously, $p_k \neq z_3$.

If $p_k = y_2$, then $k = 3$. Indeed, because of the degree constraint the cycle $C : p_1 \dots p_kxy_1p_1$ contains at most one chord, and this chord is of the form y_1p_j with an even j. Therefore, if $k > 3$, then the vertices of C induce a chordless cycle of length at least 6. Thus $k = 3$, and to avoid an induced cycle C_6, $y_1p_2 \in E(G)$ and $p_1z_2 \notin E(G)$. But now y_3 is a cutvertex in G. Indeed, if not, there must exist a path in $G - \{y_3\}$ connecting z_3 to a vertex in the set $\{z_1, p_1, z_2\}$ missing the vertices of the set $B = \{x, y_1, y_2, p_2\}$ (since all vertices in B have degree 3 in G), but then the vertices of the path together with some vertices in B and y_3 would induce a forbidden cycle.

Thus, the chordal bipartite graphs with vertex degree at most 3 form a subclass of the class $[\mathrm{Free}_3(S_{2,2,2}, T_{0,0,0})]_B$. In the latter class the tree- and clique-width are bounded according to Propositions 2 and 3 and Theorem 2. □

REFERENCES

[1] H. L. BODLAENDER, *A linear-time algorithm for finding tree-decompositions of small treewidth*, SIAM J. Comput., 25 (1996), pp. 1305–1317.

[2] H. L. BODLAENDER, *Treewidth: Algorithmic techniques and results*, in Mathematical Foundations of Computer Science, Lecture Notes in Comput. Sci. 1295, Springer-Verlag, Berlin, 1997, pp. 19–36.

[3] R. BOLIAC AND V. LOZIN, *Independent domination in finitely defined classes of graphs*, Theoret. Comput. Sci., 301 (2003), pp. 271–284.

[4] A. BRANDSTÄDT AND V. V. LOZIN, *On the linear structure and clique-width of bipartite permutation graphs*, Ars Combin., 67 (2003), pp. 273–281.

[5] Y. COLIN DE VERDIERE, *Multiplicities of eigenvalues and tree-width of graphs*, J. Combin. Theory Ser. B, 74 (1998), pp. 121–146.

[6] D. G. CORNEIL, S. OLARIU, AND L. STEWART, *Asteroidal triple-free graphs*, SIAM J. Discrete Math., 10 (1997), pp. 399–430.

[7] B. COURCELLE, *The monadic second-order logic of graphs.* III. *Tree-decompositions, minors and complexity issues*, RAIRO Inform. Théor. Appl., 26 (1992), pp. 257–286.

[8] B. COURCELLE, J. ENGELFRIET, AND G. ROZENBERG, *Handle-rewriting hypergraph grammars*, J. Comput. System Sci., 46 (1993), pp. 218–270.

[9] B. COURCELLE AND S. OLARIU, *Upper bounds to the clique-width of a graph*, Discrete Appl. Math., 101 (2000), pp. 77–114.

[10] B. COURCELLE, J. A. MAKOWSKY, AND U. ROTICS, *Linear time solvable optimization problems on graphs of bounded clique-width*, Theory Comput. Syst., 33 (2000), pp. 125–150.

[11] B. COURCELLE, J. A. MAKOWSKY, AND U. ROTICS, *On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic*, Discrete Appl. Math., 108 (2001), pp. 23–52.

[12] J. DÍAZ, J. PETIT, AND M. J. SERNA, *A survey of graph layout problems*, ACM Computing Surveys, 34 (2002), pp. 313–356.

[13] M. C. GOLUMBIC AND U. ROTICS, *On the clique-width of some perfect graph classes*, Internat. J. Found. Comput. Sci., 11 (2000), pp. 423–443.

[14] L. Perković and B. Reed, *An improved algorithm for finding tree decompositions of small width*, Internat. J. Found. Comput. Sci., 11 (2000), pp. 365–371.
[15] S. Ramachandramurthi, *The structure and number of obstructions to treewidth*, SIAM J. Discrete Math., 10 (1997), pp. 146–157.
[16] B. Reed, *Tree width and tangles: A new connectivity measure and some applications*, in Surveys in Combinatorics, London Math. Soc. Lecture Note Ser. 241, Cambridge University Press, Cambridge, UK, 1997, pp. 87–162.
[17] N. Robertson and P. D. Seymour, *Graph minors.* II. *Algorithmic aspects of tree-width*, J. Algorithms, 7 (1986), pp. 309–322.
[18] N. Robertson, P. D. Seymour, and R. Thomas, *Quickly excluding a planar graph*, J. Combin. Theory Ser. B, 62 (1994), pp. 323–348.

SIAM J. DISCRETE MATH.
Vol. 18, No. 1, pp. 207–217

IMPROVED COMBINATORIAL APPROXIMATION ALGORITHMS FOR THE k-LEVEL FACILITY LOCATION PROBLEM*

ALEXANDER AGEEV†, YINYU YE‡, AND JIAWEI ZHANG‡

Abstract. In this paper we present improved combinatorial approximation algorithms for the k-level facility location problem. First, by modifying the path reduction developed in [A. A. Ageev, *Oper. Res. Lett.*, 30 (2002), pp. 327–332], we obtain a combinatorial algorithm with a performance factor of 3.27 for any $k \geq 2$, thus improving the previous bound of 4.56 achieved by a combinatorial algorithm. Then we develop another combinatorial algorithm that has a better performance guarantee and uses the first algorithm as a subroutine. The latter algorithm can be recursively implemented and achieves a guarantee factor $h(k)$, where $h(k)$ is strictly less than 3.27 for any k and tends to 3.27 as k goes to ∞. The values of $h(k)$ can be easily computed with an arbitrary accuracy: $h(2) \approx 2.4211$, $h(3) \approx 2.8446$, $h(4) \approx 3.0565$, $h(5) \approx 3.1678$, and so on. Thus, for the cases of $k = 2$ and $k = 3$ the second combinatorial algorithm ensures an approximation factor substantially better than 3, which is currently the best approximation ratio for the k-level problem provided by the noncombinatorial algorithm due to Aardal, Chudak, and Shmoys [*Inform. Process. Lett.*, 72 (1999), pp. 161–167].

Key words. facility location, approximation algorithm, performance guarantee, polynomial-time reduction

AMS subject classifications. 68W25, 90B80, 68W40, 68Q25, 90C27

DOI. 10.1137/S0895480102417215

1. Introduction.

In the k-level facility location problem (k-LFLP) we are given a complete $(k+1)$-partite graph $G = (D \cup \mathcal{F}_1 \cup \cdots \cup \mathcal{F}_k; E)$ whose node set is the union of $k+1$ disjoint sets $D, \mathcal{F}_1, \ldots, \mathcal{F}_k$ and the edge set E consists of all edges between these sets. The nodes in D are called demand sites and the nodes in $\mathcal{F} = \mathcal{F}_1 \cup \cdots \cup \mathcal{F}_k$ are facilities (of level $1, \ldots, k$, respectively). We are given edge costs $c \in \mathbb{R}_+^E$ and opening costs $f \in \mathbb{R}_+^{\mathcal{F}}$ (i.e., opening a facility $i \in \mathcal{F}$ incurs a cost $f_i \geq 0$).

The objective is to open a set of facilities $X_t \subseteq F_t$ on each level $t = 1, \ldots, k$ and to connect each demand site $j \in D$ to a path (or chain) $\varphi(j) = (i_1(j), i_2(j), \ldots, i_k(j))$ along open facilities $i_1(j) \in X_1, i_2(j) \in X_2, \ldots, i_k(j) \in X_k$ so that the total cost of opening and connecting

$$\sum_{i \in X_1 \cup \cdots \cup X_k} f_i + \sum_{j \in D} \Big(c(j, i_1(j)) + c(i_1(j), i_2(j)) + \cdots + c(i_{k-1}(j), i_k(j)) \Big)$$

is minimized.

In this paper we consider the metric case of the problem where c is induced by a metric on the whole set of nodes $V = D \cup \mathcal{F}_1 \cup \cdots \cup \mathcal{F}_k$. Recent applications of metric facility location problems include finding product clustering, cost-effective placement of servers on the internet, and optimized supply chains [6].

*Received by the editors November 6, 2002; accepted for publication (in revised form) December 19, 2003; published electronically August 19, 2004.
http://www.siam.org/journals/sidma/18-1/41721.html

†Sobolev Institute of Mathematics, Novosibirsk, Russia (ageev@math.nsc.ru). The research of this author was supported in part by the Russian Foundation for Basic Research, project codes 01-01-00786, 02-01-01153; by INTAS, project code 00-217; by the Programme "Universities of Russia," project code UR.04.01.012; and by Russian Leading Schools grant 313.2003.1.

‡Department of Management Science and Engineering, Stanford University, Stanford, CA 94305 (yinyu-ye@stanford.edu, jiazhang@stanford.edu). The research of these authors was supported in part by NSF grant DMI-0231600.

Since the metric k-LFLP is NP-hard, most research work is concentrated on designing approximation algorithms. We say that an algorithm for a minimization problem with nonnegative objective function is a *ρ-approximation algorithm* if it runs in polynomial time and for any instance outputs a solution of cost at most ρ times the optimum.

The special case of k-LFLP where $k = 1$ (1-LFLP) is nothing but the well-known (metric) uncapacitated facility location problem (UFLP). It is known that the existence of a 1.463-approximation algorithm for solving UFLP would imply $NP \notin DTIME[n^{O(\log \log n)}]$ [7]. In recent years quite a number of approximation algorithms have been developed for solving UFLP. The current best approximation algorithm, due to Mahdian, Ye, and Zhang [11], achieves a factor of 1.517. See Shmoys [13] and [11] for a detailed survey on approximation algorithms for UFLP.

Obviously, the approximability lower bound 1.463 also applies to k-LFLP. On the positive side, it is known that k-LFLP can be solved within a factor of 3 by an LP rounding algorithm due to Aardal, Chudak, and Shmoys [1]. A drawback of this algorithm is that it includes a phase of solving a linear relaxation with exponential number of variables. Although this relaxation can be solved by the ellipsoid method in polynomial time, the algorithm would be inefficient in practice. For this reason, several combinatorial approximation algorithms have been developed to solve this problem. These algorithms run in strongly polynomial time but with a sacrifice in the performance guarantee. The first such algorithm, by Meyerson, Munagala, and Plotkin [12], had an approximation factor of $O(\ln |D|)$. A constant factor of 9.2 was later obtained by Guha, Meyerson, and Munagala [8]. Bumb and Kern [3] developed a dual ascent algorithm that had a performance guarantee of 6. Ageev [2] established that any ρ-approximation algorithm for UFLP could be translated to a 3ρ-approximation algorithm for k-LFLP. Thus, the algorithm in [11] yields a combinatorial 4.56-approximation algorithm for k-LFLP. We will refer to this approach as the *path reduction* technique. It should be noted that Edwards [4] proposed a reduction similar to that in [2] but his construction requires running time exponential in k.

None of the above algorithms has a performance guarantee better than 3. Whether or not k-LFLP can be approximated in polynomial time by a factor less than 3 has become a challenging open question in this field.

In this paper we present improved combinatorial approximation algorithms for the k-level facility location problem.

First, by modifying the path reduction of the k-level problem to the 1-level case developed in [2], we obtain a combinatorial algorithm with a performance guarantee of 3.27 for any k, thus improving the previous bound of 4.56. The algorithm runs in time $O(m_1^3 n^3 + m^2 n)$, where $m = |\mathcal{F}|$, $m_1 = |\mathcal{F}_1|$, and $n = |D|$. Note that the approximation ratio of this path reduction algorithm is fairly close to a factor of 3 provided by the LP rounding algorithm [1]. Furthermore, this theoretical result in some sense explains why in computational experiments the path-reduction-based algorithms perform better than the LP rounding algorithm, as has been observed by Edwards [4].

Although intuition suggests that k-LFLP for small values of $k \geq 2$ may be better approximable than the general problem, our path reduction algorithm, as all the previous algorithms, has the same approximation factor for each k. This drawback motivated our work on a better algorithm whose performance factor would be an increasing function of k with values strictly less than 3.27. Our efforts resulted in a recursive combinatorial algorithm for k-LFLP, which is presented in the second part

of this paper. It is based on a combination of our path reduction algorithm and a recursive reduction of k-LFLP to $(k-1)$-LFLP and UFLP. The algorithm runs in time $O(k(m_1^3n^3 + m^2n))$ and achieves an approximation factor $h(k)$, where $h(k)$ is strictly less than 3.27 for any $k \geq 1$ and tends to 3.27 as k tends to ∞. The values of $h(k)$ can be easily computed with an arbitrary accuracy. In particular, $h(2) \leq 2.4211$, $h(3) \leq 2.8446$, $h(4) \leq 3.0565$, $h(5) \leq 3.1678$. Thus, for 2-LFLP and 3-LFLP, the second algorithm achieves an approximation factor substantially better than 3.

2. The path reduction algorithm. In this section we present a parameterized version of the path reduction, which in combination with the greedy algorithm developed in [11] yields a 3.27-approximation algorithm for solving k-LFLP.

2.1. Definitions and notation. Denote by $\mathcal{P}$ the set of all paths of length $k-1$ connecting a node in $\mathcal{F}_1$ to a node in $\mathcal{F}_k$. For a path $p = (i_1, i_2, \ldots, i_k) \in \mathcal{P}$, let $c(p) = \sum_{t=2}^{k} c(i_{t-1}, i_t)$. For any subset $X \subseteq \mathcal{F}$, let $f(X) = \sum_{i \in X} f_i$, and let $\mathcal{P}(X)$ denote the subset of paths in $\mathcal{P}$ passing through facilities in X.

Let $\mathcal{M}$ be an instance of k-LFLP and SOL be a solution of it. Recall that SOL is a pair (X, φ), where X is a set of open facilities and φ is an assignment mapping D to $\mathcal{P}(X)$. We call a path in $\varphi(D)$ a *service path*.

For our analysis it would be convenient to represent the total cost of any solution SOL for k-LFLP in the split form $F^{SOL} + C^{SOL}$, where F^{SOL} and C^{SOL} stand for the facility and connection costs, respectively. To break down C^{SOL} further, for any $t = 2, \ldots, k$, let C_t^{SOL} denote the total connection cost between open facilities on level $t-1$ and open facilities on level t. Hence $C^{SOL} = \sum_{t=1}^{k} C_t^{SOL}$, where C_1^{SOL} stands for the total connection cost between demand sites and facilities on level 1. Similarly, let F_t^{SOL} denote the total cost to open facilities on level t, and thus $F^{SOL} = \sum_{t=1}^{k} F_t^{SOL}$.

To exploit the cost-split character of the objective function in k-LFLP we modify the standard definition of performance guarantee in the split way, as follows.

DEFINITION 1. *A feasible solution SOL of a k-LFLP is called* (a, b)-approximate *if for any other feasible solution SOL^* of the problem, the cost of SOL is at most $aF^{SOL^*} + bC^{SOL^*}$. An algorithm for a k-LFLP is an* (a, b)-approximation algorithm *if the solution found by the algorithm is (a, b)-approximate.*

Our path reduction algorithm was inspired by the observation that the path reduction developed in [2] admits a slight modification implying that any (a, b)-approximation algorithm for UFLP can be translated into a $(a, 3b)$-approximation algorithm for k-LFLP. Therefore, to obtain a good approximation factor for k-LFLP, we have to solve the reduced UFLP in such a way that the performance guarantee pair (a, b) approximately satisfies $a = 3b$. To this point we apply the algorithm of Mahdian, Ye, and Zhang [11] to obtain a guarantee pair $(3.27, 1.09)$ for UFLP, which then implies a 3.27-approximation for k-LFLP.

2.2. Parameterized path reduction. We now describe a path reduction with positive parameters a, b that generalizes the reduction in [2] (corresponding to the case $a = b = 1$).

Path reduction with parameters (a, b). Let $\mathcal{M}$ be an instance of k-LFLP. For each $i_1 \in \mathcal{F}_1$ and $t \in \{1, \ldots, |D|\}$, compute a path $p(i_1, t)$ that has the minimum value of $t \cdot bc(p) + af(p)$ over all paths $p \in \mathcal{P}$ starting from i_1. (Note that the problem of finding such paths can be easily reduced to the shortest path problem and there are total $|\mathcal{F}_1| \cdot |D|$ of such paths.) Then, associate with $\mathcal{M}$ an instance $\mathcal{S}$ of UFLP in which the set of demand sites is D and the set of "facilities" is the set of all pairs (i_1, t), where $i_1 \in \mathcal{F}_1$ and $t \in \{1, \ldots, |D|\}$. In $\mathcal{S}$, for any demand site $j \in D$ and "facility"

(i_1, t), the cost of connecting j to (i_1, t) is defined to be $c(j, i_1) + c(p(i_1, t))$, and the cost of opening (i_1, t) is defined to be $f(p(i_1, t))$ (i.e., equal to the cost of opening all facilities on path $p(i_1, t)$). Given a solution $SOLS$ of $\mathcal{S}$, we construct back a solution $SOLM$ of $\mathcal{M}$ as follows: for any $j \in D$, connect j to the service path $p(i_1(j), t)$ such that $(i_1(j), t)$ is the "facility" serving j in $SOLS$, and open the facilities on all such service paths.

The main result of this subsection is the following theorem.

THEOREM 1. *If $SOLS$ is an (a, b)-approximate solution of $\mathcal{I}$, then $SOLM$ is an $(a, 3b)$-approximate solution of $\mathcal{M}$. Furthermore, for any solution SOL of $\mathcal{M}$,*

$$F^{SOLM} + C^{SOLM} \le aF^{SOL} + bC_1^{SOL} + 3b\sum_{i=2}^{k} C_i^{SOL}. \tag{1}$$

Therefore, we have the following.

COROLLARY 1. *Any (a, b)-approximation algorithm for solving UFLP yields an $(a, 3b)$-approximation algorithm for solving k-LFLP.* □

Our proof of the theorem is based on Lemmas 1 and 2 below. The first lemma is nothing but Lemma 2 in [2].

LEMMA 1.

$$F^{SOLM} \le F^{SOLS} \quad \textit{and} \quad C^{SOLM} = C^{SOLS}. \qquad \square$$

The second lemma is an improvement of Lemma 4 in [2].

LEMMA 2. *For any solution SOL of $\mathcal{M}$, there exists a corresponding solution SOL^* of the reduced $\mathcal{S}$ such that*

$$aF^{SOL^*} + bC^{SOL^*} \le aF^{SOL} + bC_1^{SOL} + 3b\sum_{t=2}^{k} C_t^{SOL}. \tag{2}$$

We first deduce Theorem 1 from the above lemmas.

Proof of Theorem 1. Let SOL^* be any solution of $\mathcal{M}$. By Lemma 2, there exists a corresponding solution SOL of $\mathcal{S}$ such that

$$aF^{SOL} + bC^{SOL} \le aF^{SOL^*} + bC_1^{SOL^*} + 3b\sum_{t=2}^{k} C_t^{SOL^*} \le aF^{SOL^*} + 3bC^{SOL^*}.$$

On the other hand, by using Lemma 1 and the fact that $SOLS$ is an (a, b)-approximate solution of $\mathcal{S}$, we have

$$F^{SOLM} + C^{SOLM} \le F^{SOLS} + C^{SOLS} \le aF^{SOL} + bC^{SOL},$$

which proves (1). □

To prove Lemma 2 we need the following easy statement, which, being a bit stronger than Lemma 3 in [2], has an almost identical proof.

LEMMA 3. *Let $\mathcal{I}$ be an instance of k-level FLP and $\overline{SOL}$ be a solution of $\mathcal{I}$. Then $\mathcal{I}$ has a solution $SOL = (X, \varphi)$ such that*

(i) *if in paths $\varphi(j') = (i'_1, \dots, i'_k)$ and $\varphi(j'') = (i''_1, \dots, i''_k)$ $i'_l = i''_l$ for some l, then $i'_r = i''_r$ for all $r \ge l$;*

(ii) $C_1^{SOL} = C_1^{\overline{SOL}}$, $\sum_{l=2}^{k} C_l^{SOL} \le \sum_{l=2}^{k} C_l^{\overline{SOL}}$, $F^{SOL} \le F^{\overline{SOL}}$. □

Lemma 3 implies that any solution $\overline{SOL}$ of k-LFLP can be replaced by a solution SOL satisfying (ii) whose service paths constitute a forest consisting of trees rooted at level k.

Proof of Lemma 2. Let $SOL = (X, \varphi)$ be a solution of $\mathcal{M}$. For any $j \in D$, let $\varphi(j) = \big(i_1(j), \dots, i_k(j)\big)$. By Lemma 3 we may assume that SOL satisfies property (i) and thus the service paths of SOL constitute a forest consisting of trees rooted at open facilities in $\mathcal{F}_k$.

For every open facility $u \in X_k = X \cap \mathcal{F}_k$ lying on level k, let D_u be the set of demand sites assigned, by φ, to a path finishing in u, and let $p(u)$ be a path having minimum value of $c(p)$ among all service paths p ending in u. Also, let $\mu(u)$ be the starting facility of $p(u)$ lying on level 1.

Define a new solution $SOLP = (X, \varphi')$ by reassigning each $j \in D_u$ to the path $p(u)$, i.e., by setting $\varphi'(j) = p(u)$ for all $u \in X_k$. Thus, by definition, $SOLP$ satisfies

$$F^{SOLP} \le F^{SOL} \tag{3}$$

and

$$C^{SOLP} = \sum_{u \in X_k} \sum_{j \in D_u} \Big(c(j, \mu(u)) + c(p(u)) \Big).$$

By the triangle inequality and the definitions of $p(u)$ and $\mu(u)$,

$$\begin{aligned} c(j, \mu(u)) + c(p(u)) &\le \big(c(j, i_1(j)) + c(\varphi(j)) + c(p(u)) \big) + c(p(u)) \\ &\le c(j, i_1(j)) + 3c(\varphi(j)). \end{aligned}$$

Thus we have

$$\begin{aligned} C^{SOLP} &\le \sum_{u \in X_k} \sum_{j \in D_u} \big(c(j, i_1(j)) + 3c(\varphi(j)) \big) \\ &= C_1^{SOL} + 3 \sum_{t=2}^{k} C_t^{SOL}. \end{aligned} \tag{4}$$

Now, by (3) and (4), it suffices to show that there exists a solution SOL^* of $\mathcal{S}$ such that

$$aF^{SOL^*} + bC^{SOL^*} \le aF^{SOLP} + bC^{SOLP}. \tag{5}$$

Since the service paths of $SOLP$ are disjoint, we have

$$\begin{aligned} aF^{SOLP} + bC^{SOLP} &= \sum_{u \in X_k} \left(af(p(u)) + b \sum_{j \in D_u} \Big(c(j, \mu(u)) + c(p(u)) \Big) \right) \\ &= \sum_{u \in X_k} \left(af(p(u)) + b|D_u| \cdot c(p(u)) + b \sum_{j \in D_u} c(j, \mu(u)) \right) \\ &= \sum_{u \in X_k} \Big(af(p(u)) + b|D_u| \cdot c(p(u)) \Big) + bC_1^{SOLP}. \end{aligned}$$

Now we define a solution SOL^* of $\mathcal{S}$ by declaring open all facilities lying on the paths $p(\mu(u), |D_u|)$, $u \in X_k$, and by connecting j to the path $p(\mu(u), |D_u|)$ whenever $j \in D_u$.

Then we have

$$aF^{SOL^*} + bC^{SOL^*} = \sum_{u \in X_k} \Big(af(p(\mu(u), |D_u|)) + b|D_u| \cdot c(p(\mu(u), |D_u|)) \Big) + bC_1^{SOLP}$$
$$\leq aF^{SOLP} + bC^{SOLP}.$$

The last inequality holds because for each $u \in X_k$, by the construction of paths $p(i_1, t)$ in the parameterized path reduction,

$$af(p(\mu(u), |D_u|)) + b|D_u| \cdot c(p(\mu(u), |D_u|)) \leq af(p(u)) + b|D_u| \cdot c(p(u)). \qquad \square$$

The next subsection analyzes particular values of parameters (a, b) to establish our final result.

2.3. Algorithm Path Reduction&Greedy. To solve the instance $\mathcal{S}$ of UFLP we use the greedy algorithm developed in [11], referred to as Greedy. For completeness, we sketch the algorithm Greedy below; it is essentially a combination of the algorithms of Jain, Mahdian, and Saberi [9] and Guha and Khuller [7].

Algorithm Greedy.

Phase 1. Given an instance $\mathcal{S}$ of the UFLP, scale up the opening costs of all facilities by a factor of δ (≥ 1) (which is a constant that will be fixed later). Then do the following:

1. At the beginning, all demand sites are *unconnected*, all facilities are *unopened*, and the *budget* of every city j, denoted by B_j, is initialized to 0. At every moment, each demand site j offers some money from its budget to each *unopened* facility i. The amount of this offer is equal to $\max(B_j - c_{ij}, 0)$ if j is unconnected or $\max(c_{i'j} - c_{ij}, 0)$ if it is already connected to some other facility i'.
2. While there is an unconnected demand site, increase the budget of each *unconnected* demand site at the same rate, until one of the following two events occurs:
   (a) For some unopened facility i, the total offer that it receives from demand sites is equal to the (scaled) cost of opening i. In this case, we open facility i and connect j to i for every demand site j (connected or unconnected) which has a positive offer to i,
   (b) For some unconnected demand site j, and some facility i that is already open, the budget of j is equal to the connection cost c_{ij}. In this event, we connect j to i.

Phase 2. Scale down the opening costs of facilities back to their original values all at the same rate. If at any point during this process, a facility could be opened without increasing the total cost (i.e., if the opening cost of the facility equals the total amount that the demand sites can save by switching their "service provider" to that facility), then we open the facility and connect each demand site to its closest open facility.

In what follows we need only the following result from [11].

Let $\gamma_f(\delta) = \gamma_f + \ln \delta$ and $\gamma_c(\delta) = 1 + \frac{\gamma_c - 1}{\delta}$, where $\gamma_f = 1.11$, $\gamma_c = 1.78$.

Lemma 4. *Algorithm* Greedy *is an* $(\gamma_f(\delta), \gamma_c(\delta))$*-approximation algorithm for any* $\delta \geq 1$.

By this lemma, the path reduction algorithm produces a $(\gamma_f(\delta), 3\gamma_c(\delta))$-approximation algorithm for k-LFLP, where δ is an arbitrary number ≥ 1. By taking $\delta = 8.67$, one can see that our algorithm, which we will further refer to as Path Reduction&Greedy, finds a solution within a factor of 3.27 of the minimal cost.

Note that the paths $p(i_1, t)$ in the parameterized path reduction can be computed in $O(m^2n)$ time. On the other hand, the total number of demand sites and facilities in the reduced $\mathcal{S}$ is $n + m_1 n$ and thus GREEDY requires $O(m_1^3 n^3)$ time to solve it. Therefore, the overall running time of PATH REDUCTION&GREEDY is $O(m_1^3 n^3 + m^2 n)$.

We remark that the bound 3.27 cannot be improved much by just using Corollary 1 as a tool box. It is known [9, 10] that for any $x \geq 1$, the existence of $(x, 1+2e^{-x})$-approximation algorithm for UFLP would imply $NP \subseteq DTIME[n^{O(\log \log n)}]$. Therefore, the best we could get by using Corollary 1 is 3.236 since $x + 3(1 + 2e^{-x}) \geq 6.472$ for any $x \geq 1$.

3. The recursive path reduction algorithm. A drawback of algorithm PATH REDUCTION&GREEDY is that the approximation factor of 3.27 it provides does not depend on the number of levels k, whereas 1-LFLP admits a 1.52-approximation and, intuition suggests that k-LFLP for small values of k must be approximable within a smaller ratio than the general problem.

In this section, we present an improved combinatorial algorithm for k-LFLP, which we refer to as SPLIT&RECURSION. It is based on a combination of PATH REDUCTION&GREEDY and a recursive reduction of k-LFLP to $(k-1)$-LFLP and UFLP. Algorithm SPLIT&RECURSION runs in time $O(km_1^3 n^3 + km^2 n)$ and achieves an approximation factor of $h(k)$, where $h(k) < 3.27$ for any $k \geq 1$ and tends to 3.27 as k tends to ∞. The values of $h(k)$ can be easily computed with an arbitrary accuracy. In particular, $h(2) \approx 2.4211$, $h(3) \approx 2.8446$, $h(4) \approx 3.0565$, $h(5) \approx 3.1678$.

3.1. Definitions and high-level description. We first give a few definitions.

For any instance $\mathcal{M}$ of k-LFLP, we define an instance $\mathcal{M}_{k-1}$ of $(k-1)$-LFLP and an instance $\mathcal{S}$ of UFLP (1-LFLP) in the following way:

1. $\mathcal{M}_{k-1}$ is obtained from $\mathcal{M}$ by deleting all facilities on level 1 (or, by opening for free all facilities on level 1). Thus, in $\mathcal{M}_{k-1}$ the set of facilities lying on level r is $\mathcal{F}_{r+1}$, and the connection cost between $j \in D$ and $i_2 \in \mathcal{F}_2$ is
$$\min_{v \in \mathcal{F}_1} \{c(j, v) + c(v, i_2)\}.$$
2. $\mathcal{S}$ is obtained from $\mathcal{M}$ by deleting all facilities on levels greater than 1 (and all edges incident with these facilities) and by doubling all the edge costs between D and $\mathcal{F}_1$.

We are now ready to proceed to a high-level description of the algorithm.

In the case $k = 2$, $\mathcal{M}_1$ and $\mathcal{S}$ are both instances of UFLP and we solve them by GREEDY. Note that each $j \in D$ is assigned to a facility $i_2(j) \in \mathcal{F}_2$ by the solution for $\mathcal{M}_1$ and to a facility $i_1(j) \in \mathcal{F}_1$ by the solution for $\mathcal{S}$. On the basis of these solutions we construct a solution for $\mathcal{M}$, denoted by $SOLMS$, by connecting each j to the path $(i_1(j), i_2(j))$.

Note that the straightforward variant of the above construction where the connection costs coincide with the original ones in both instances of UFLP yields a simple factor-3 reduction of 2-LFLP to UFLP. This reduction was first observed by Gimadi [5].

When $k \geq 3$ our algorithm solves $\mathcal{S}$ by applying GREEDY and calls itself to solve $\mathcal{M}_{k-1}$. Now we have that the solution of $\mathcal{S}$ assigns each $j \in D$ to a facility $i_1(j) \in \mathcal{F}_1$ while the solution to $\mathcal{M}_{k-1}$ assigns each $j \in D$ to a path $(i_2(j) \in \mathcal{F}_2, \ldots, i_k(j) \in \mathcal{F}_k)$. In this case the solution $SOLMS$ for $\mathcal{M}$ is constructed by connecting each j to the composite path $(i_1(j), i_2(j), \ldots, i_k(j))$.

However, the constructed solution $SOLMS$ is not yet the output of the algorithm. In addition, we find another solution $SOLPG$ for $\mathcal{M}$ by applying PATH REDUCTION&GREEDY and finally output a solution having lower cost among the two.

By unfolding this recursive description one can easily obtain a conventional implementation as follows. The algorithm applies GREEDY to solve k instances of UFLP obtained from the original instance $\mathcal{M}$ by deleting the facilities on all levels except a fixed one. It then applies PATH REDUCTION&GREEDY to solve $k-1$ instances of k-LFLP obtained from $\mathcal{M}$ by deleting the facilities on all levels smaller than a fixed one. Finally, in $k-1$ steps, on the basis of the retrieved solutions, it constructs an output solution.

From the above implementation it is clear that SPLIT&RECURSION can be implemented in $O(k(m_1^3n^3+m^2n))$ time.

3.2. Algorithm SPLIT&RECURSION. Now we proceed to a formal description and analysis of the algorithm.

ALGORITHM SPLIT&RECURSION.
Input: An instance $\mathcal{M}$ of k-LFLP.
Output: A solution SOL for $\mathcal{M}$.
`if` $k=1$ `then`
$SOL :=$ the solution obtained by applying GREEDY to $\mathcal{M}$;
`endif`
`if` $k \geq 2$ `then`
Apply SPLIT&RECURSION to find a solution $SOLM$ for $\mathcal{M}_{k-1}$ and GREEDY to find a solution $SOLS$ for $\mathcal{S}$;
Construct a solution $SOLMS$ for $\mathcal{M}$ by connecting each $j \in D$ to the path

$$(i_1(j), i_2(j), \dots, i_k(j))$$

whenever j connects to $i_1(j)$ in $SOLS$ and to the path $(i_2(j), \dots, i_k(j))$ in $SOLM$;
Apply PATH REDUCTION&GREEDY to find a solution $SOLPG$ of $\mathcal{M}$;
$SOL :=$ a solution having lower cost among $SOLMS$ and $SOLPG$.
`endif`

The following theorem is the main result of this section.

THEOREM 2. *Let $k \geq 2$. For any solution SOL^* of $\mathcal{M}$ and any $\delta \geq 1$, the solution SOL retrieved by* SPLIT&RECURSION *satisfies*

$$F^{SOL} + C^{SOL} \leq \gamma_f(\delta) F^{SOL^*} + \theta(k)\gamma_c(\delta) C^{SOL^*}, \tag{6}$$

where

$$\theta(k) = 3\left(1 - \frac{1}{2^{k-2}}\right) + \frac{1}{2^{k-3}}.$$

Since $\gamma_f(\delta)$ is a strictly increasing function of δ on the interval $[1,\infty)$ whereas $\theta(k)\gamma_c(\delta)$ is strictly decreasing, the minimum value of

$$\rho_k(\delta) = \max(\gamma_f(\delta), \theta(k)\gamma_c(\delta))$$

is attained at a unique root δ_k of the transcendent equation

$$\gamma_f(\delta) = \theta(k)\gamma_c(\delta).$$

Thus we derive the following.

COROLLARY 2. SPLIT&RECURSION *is a* $\rho_k(\delta_k)$*-approximation algorithm for* k*-LFLP.* □

By using a binary search, it is easy to compute δ_k approximately for every k. This gives

$$\begin{aligned}
\rho_2(\delta_2) &\le \rho_2(3.71) < 2.4211,\\
\rho_3(\delta_3) &\le \rho_3(5.66) < 2.8446,\\
\rho_4(\delta_4) &\le \rho_4(7.0) < 3.0565,\\
\rho_5(\delta_5) &\le \rho_5(7.66) < 3.1678.
\end{aligned}$$

One can also see that as $k \to \infty$, $\theta(k)$ tends to 3 and the performance factor tends to 3.27 as in algorithm PATH REDUCTION&GREEDY.

Proof of Theorem 2. We proceed by induction on k. Let SOL^* be any solution of $\mathcal{M}$. Then, by Theorem 1,

$$F^{SOLPG} + C^{SOLPG} \le \gamma_f(\delta)F^{SOL^*} + \gamma_c(\delta)C_1^{SOL^*} + 3\gamma_c(\delta)\sum_{t=2}^{k} C_t^{SOL^*}. \tag{7}$$

Observe that SOL^* induces a solution, $SOLS^*$, to $\mathcal{S}$ and a solution, $SOLM^*$, to $\mathcal{M}_{k-1}$, as SOL^* assigns every demand site j to a facility, say, $i_t^*(j) \in \mathcal{F}_t$ for each $t = 1, \dots, k$. That is, j in $SOLS^*$ is assigned to $i_1^*(j)$ of $\mathcal{S}$ with connection cost $2c(j, i_1^*(j))$, and j in $SOLM^*$ is assigned to $(i_2^*(j), \dots, i_k^*(j))$ of $\mathcal{M}_{k-1}$ with connection cost at most

$$c(j, i_1^*(j)) + c(i_1^*(j), i_2^*(j)) + c((i_2^*(j), \dots, i_k^*(j))).$$

More precisely,

$$C^{SOLS^*} = 2C_1^{SOL^*}, \tag{8}$$

$$F^{SOLM^*} = \sum_{t=2}^{k} F_t^{SOL^*}, \tag{9}$$

$$C_1^{SOLM^*} \le C_1^{SOL^*} + C_2^{SOL^*}, \tag{10}$$

$$C_t^{SOLM^*} = C_{t+1}^{SOL^*} \text{ for } t = 2, \dots, k-1. \tag{11}$$

Recall that the connections costs in $\mathcal{S}$ are doubled from the edge costs between D and $\mathcal{F}_1$ in $\mathcal{M}$. Hence, by Lemma 4 and (8)

$$\begin{aligned}
F^{SOLS} + C^{SOLS} &\le \gamma_f(\delta)F^{SOLS^*} + \gamma_c(\delta)C^{SOLS^*}\\
&= \gamma_f(\delta)F_1^{SOL^*} + 2\gamma_c(\delta)C_1^{SOL^*}.
\end{aligned} \tag{12}$$

Assume now that $k = 2$. In this case $\mathcal{M}_{k-1}$ is an instance of UFLP, and thus by using Lemma 4 and (9) and (10) we obtain

$$\begin{aligned}
F^{SOLM} + C^{SOLM} &\le \gamma_f(\delta)F^{SOLM^*} + \gamma_c(\delta)C^{SOLM^*}\\
&\le \gamma_f(\delta)F_2^{SOL^*} + \gamma_c(\delta)(C_1^{SOL^*} + C_2^{SOL^*}).
\end{aligned} \tag{13}$$

By the construction of $SOLMS$ and the triangle inequality,

$$\begin{aligned} F^{SOLMS} + C^{SOLMS} &= F^{SOLS} + F^{SOLM} + \frac{1}{2}C^{SOLS} + \sum_{j\in D} c(i_1(j), i_2(j)) \\ &\le F^{SOLS} + F^{SOLM} + \frac{1}{2}C^{SOLS} + \left(\frac{1}{2}C^{SOLS} + C^{SOLM}\right) \\ &= F^{SOLS} + C^{SOLS} + F^{SOLM} + C^{SOLM}, \end{aligned}$$

and thus, according to (12) and (13), we have

$$F^{SOLMS} + C^{SOLMS} \le \gamma_f(\delta)F^{SOL^*} + 3\gamma_c(\delta)C_1^{SOL^*} + \gamma_c(\delta)C_2^{SOL^*}. \tag{14}$$

Since the cost of SOL is at most half as great as the sum of costs of $SOLMS$ and $SOLPG$, (7) and (14) imply

$$F^{SOL} + C^{SOL} \le \gamma_f(\delta)F^{SOL^*} + 2\gamma_c(\delta)C_1^{SOL^*} + 2\gamma_c(\delta)C_2^{SOL^*},$$

which is nothing but (6) for $k = 2$.

Now, assume that (6) is true for each instance of the r-LFLP with the number of levels r at most $k-1$. Thus, by applying the induction hypothesis to $\mathcal{M}_{k-1}$, which is an instance of $(k-1)$-LFLP, we obtain

$$\begin{aligned} F^{SOLM} + C^{SOLM} &\le \gamma_f(\delta)F^{SOLM^*} + \theta(k-1)\gamma_c(\delta)C^{SOLM^*} \\ &\le \gamma_f(\delta)\sum_{t=2}^{k} F_t^{SOL^*} + \theta(k-1)\gamma_c(\delta)(C_1^{SOL^*} + C_2^{SOL^*}) \\ &\quad + \theta(k-1)\gamma_c(\delta)\sum_{t=3}^{k} C_t^{SOL^*}, \end{aligned} \tag{15}$$

where the second inequality follows from (9)–(11). Again, by the construction of $SOLMS$ and the triangle inequality,

$$F^{SOLMS} + C^{SOLMS} \le F^{SOLS} + C^{SOLS} + F^{SOLM} + C^{SOLM},$$

and thus, by (12) and (15),

$$\begin{aligned} F^{SOLMS} + C^{SOLMS} \le \gamma_f(\delta)F^{SOL^*} &+ \big(\theta(k-1)+2\big)\gamma_c(\delta)C_1^{SOL^*} \\ &+ \theta(k-1)\gamma_c(\delta)\sum_{t=2}^{k} C_t^{SOL^*}. \end{aligned}$$

Together with (7), this yields

$$F^{SOL} + C^{SOL} \le \gamma_f(\delta)F^{SOL^*} + \frac{\theta(k-1)+3}{2}\gamma_c(\delta)C^{SOL^*}.$$

Since

$$\theta(k) = \frac{\theta(k-1)+3}{2},$$

(6) follows. □

Finally, we remark that the approximation factors of our algorithms seem to be insensitive to the particular choice of $(\gamma_f, \gamma_c) = (1.11, 1.78)$ used in the above analysis. For example, if one makes use of the pair $(\gamma_f, \gamma_c) = (1, 2)$ (whose correctness was proved for the algorithm presented by Jain et al. [9, 10]), then $h(k)$ is strictly less than 3.301 for any $k \geq 1$ and tends to 3.301 as k tends to ∞. In particular, $h(2) \leq 2.462$, $h(3) \leq 2.882$, $h(4) \leq 3.091$, $h(5) \leq 3.197$.

REFERENCES

[1] K. Aardal, F. A. Chudak, and D. B. Shmoys, *A 3-approximation algorithm for the k-level uncapacitated facility location problem*, Inform. Process. Lett., 72 (1999), pp. 161–167.

[2] A. A. Ageev, *Improved approximation algorithms for multilevel facility location problems*, Oper. Res. Lett., 30 (2002), pp. 327–332.

[3] A. F. Bumb and W. Kern, *A simple dual ascent algorithm for the multilevel facility location problem*, in Proceedings of the 4th International Workshop on Approximation Algorithms for Combinatorial Optimization, Lecture Notes in Comput. Sci. 2129, Springer, Berlin, 2001, pp. 55–62.

[4] N. Edwards, *Approximation Algorithms for the Multi-Level Facility Location Problem*, Ph.D. thesis, School of Operations Research and Industrial Engineering, Cornell University, Ithica, NY, 2001.

[5] E. Kh. Gimadi, *private communication*.

[6] S. Guha, *Approximation Algorithms for Facility Location Problems*, Ph.D. thesis, Stanford University, Stanford, CA, 2000.

[7] S. Guha and S. Kuller, *Greedy strikes back: Improved facility location algorithms*, J. Algorithms, 31 (1999), pp. 228–248.

[8] S. Guha, A. Meyerson, and K. Munagala, *Hierarchical placement and network design problems*, in Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science, Redondo Beach, CA, 2000, pp. 603–612.

[9] K. Jain, M. Mahdian, and A. Saberi, *A new greedy approach for facility location problems*, in Proceedings of the 34th ACM Symposium on Theory of Computing, Montreal, QC, Canada, 2002, pp. 731–740.

[10] K. Jain, M. Mahdian, E. Markakis, and A. Saberi, *Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP*, J. ACM, 50 (2003), pp. 795–824.

[11] M. Mahdian, Y. Ye, and J. Zhang, *Improved approximation algorithms for metric facility location problems*, in Proceedings of the 5th International Workshop on Approximation Algorithms for Combinatorial Optimization, Lecture Notes in Comput. Sci. 2462, Springer, Berlin, 2002, pp. 229–242.

[12] A. Meyerson, K. Munagala, and S. Plotkin, *Cost-distance: Two-metric network design*, in Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science, Redondo Beach, CA, 2000, pp. 624–630.

[13] D. B. Shmoys, *Approximation algorithms for facility location problems*, in Proceedings of the 3rd International Workshop on Approximation Algorithms for Combinatorial Optimization, Lecture Notes in Comput. Sci. 1913, Springer, Berlin, 2000, pp. 27–33.

SIAM J. DISCRETE MATH.
Vol. 18, No. 2, pp. 219–225

APPROXIMATING MAXIMUM CLIQUE BY REMOVING SUBGRAPHS*

URIEL FEIGE†

Abstract. We show an algorithm that finds cliques of size $(\log n/\log\log n)^2$ whenever a graph has a clique of size at least $n/(\log n)^b$ for an arbitrary constant b. This leads to an algorithm that approximates max clique within a factor of $O(n(\log\log n)^2/(\log n)^3)$, which matches the best approximation ratio known for the chromatic number. The previously best approximation ratio known for max clique was $O(n/(\log n)^2)$.

Key words. approximation algorithm, clique, independent set

AMS subject classifications. 05C69, 05C85, 68W25

DOI. 10.1137/S089548010240415X

1. Introduction.

Max clique is the problem of finding a clique of maximum size in an input graph. This problem is NP-hard. An algorithm is said to have approximation ratio ρ for max clique if, on every graph, it is guaranteed to find a clique whose size is at most a factor of ρ smaller than that of the maximum clique. We allow ρ to grow as a function of n (the number of vertices in the input graph). Håstad [8] shows that for every $\epsilon > 0$ there is no polynomial algorithm that approximates max clique within a ratio of $n^{1-\epsilon}$, unless NP has expected polynomial time algorithms. (See [10] for additional information in this respect.) The best approximation ratio known for max clique was $O(n/(\log n)^2)$ by Boppana and Halldorsson [4].

The problem of max independent set is strongly related to the max clique problem (by complementing the graph), and hence shares the same approximation ratio. The chromatic number of a graph is the smallest number of independent sets that cover all vertices of the graph. It shares essentially the same hardness of approximation results as max clique [5, 10] (this is an empirical observation rather than a theorem). In terms of approximation algorithms, Halldorsson [6] shows that the chromatic number can be approximated within a ratio of $O(n(\log\log n)^2/(\log n)^3)$, which is better than the best approximation ratio known for max clique.

In this paper we show an algorithm that approximates max clique within a ratio of $O(n(\log\log n)^2/(\log n)^3)$, matching the known approximation ratio for the chromatic number. The technically new ingredient in our result is an algorithm that finds cliques of size $(\log n/\log\log n)^2$ whenever a graph has a clique of size at least $n/(\log n)^b$ for an arbitrary constant b. This algorithm is based on ideas which can be viewed as natural extensions of ideas used by Boppana and Halldorsson [4] and by Berger and Rompel [2].

In section 2 we describe our new algorithm. In section 3 we explain how (combined with ideas from [6]) it leads to an $O(n(\log\log n)^2/(\log n)^3)$ approximation ratio for max clique. In section 4 we discuss possible future research directions.

*Received by the editors March 18, 2002; accepted for publication (in revised form) February 2, 2004; published electronically October 1, 2004. This research was supported by Israel Science Foundation grant 263/02.

http://www.siam.org/journals/sidma/18-2/40415.html

†Department of Computer Science and Applied Mathematics, The Weizmann Institute, Rehovot 76100, Israel (uriel.feige@weizmann.ac.il).

2. The new algorithm. Let $G(V,E)$ be an input graph with n vertices which contains a clique of size n/k. In this graph we wish to find a large as possible clique. For a parameter $t \ll n/k$, we shall give an algorithm that finds a clique of size at least $t(\log_{3k}(n/t) - 3)$. The running time of the algorithm is $O(\binom{2kt}{t} n^c)$, where c is some universal constant. For every $b > 0$, whenever $k < (\log n)^b$, we can choose $t = \Theta(\log n / \log\log n)$, and then our algorithm finds a clique of size $\Omega((\log n/\log\log n)^2)$ in polynomial time.

In the course of our algorithm, we shall consider vertex induced subgraphs of G.

DEFINITION 2.1. *Let G be a graph with a clique of size n/k. A vertex induced subgraph S is called* poor *if it does not contain a clique of size $|S|/2k$.*

LEMMA 2.2. *Let G be a graph with a clique of size n/k. Let $S_1, S_2, \ldots$ be arbitrary disjoint poor subgraphs of G (with no clique of size $|S_i|/2k$, respectively). Let $G'(V',E')$ be the vertex induced subgraph of G that remains after removing the poor subgraphs. Then $|V'| \geq n/2k$, and G' contains a clique of size at least $|V'|/k$.*

Proof. The union of disjoint poor subgraphs is itself a poor subgraph. Any subgraph of G contains at most n vertices. Hence the poor subgraph cannot contain a clique larger than $n/2k$. As G has a clique of size n/k, at least $n/2k$ of the clique vertices must remain in G', proving $|V'| \geq n/2k$.

Removing a poor subgraph from G increases the relative density of the maximum clique in the remaining graph. Hence G' contains a clique of size at least $|V'|/k$. ∎

Our algorithm works in phases. The input to a phase is a vertex induced subgraph $G'(V',E')$ of G. (The input to the first phase is the graph G itself.) This subgraph contains a clique of size $|V'|/k$. A phase is completed when one of the following two conditions hold:

1. A clique of size $t\log_{3k}(|V'|/6kt)$ is found.
2. A poor subgraph is found.

If upon finishing a phase the first condition holds, then the algorithm terminates. If upon finishing a phase the second condition holds, then the poor subgraph is removed from G' and a new phase begins with the resulting graph. From Lemma 2.2 it follows that the invariant that the input graph contains a clique of size $|V'|/k$ is maintained when moving from phase to phase. Moreover, by removing poor subgraphs, $|V'|$ cannot drop below $n/2k$, and hence eventually the algorithm must terminate and output a clique of size at least $t\log_{3k}(n/12k^2t) > t(\log_{3k}(n/t) - 3)$.

It remains to show how a single phase is performed. Recall that the input to a phase is a graph $G'(V',E')$ which has a clique of size at least $|V'|/k$. The location of the clique is unknown to the algorithm, but the value of k is known. The algorithm has a parameter t. The larger the t, the larger the size of the clique eventually found. However, the running time of the algorithm also increases with t, and eventually we shall choose $t = \log n/\log\log n$ to balance these two factors.

Each phase has several iterations. The input to an iteration is a subgraph $G''(V'',E'')$ of G' and a set of vertices C that form a clique in $V' \setminus V''$. When the iteration ends, either the set C grows (and V'' shrinks), or V'' is declared as poor. In the first iteration $G'' = G'$ and C is empty. We now describe a single iteration:

1. If $|V''| < 6kt$, end the phase and output C.
2. Partition V'' into disjoint parts, each with $2kt$ vertices. (For simplicity we assume that $2kt$ divides $|V''|$. The algorithm can easily be modified to handle the case that this is not so with negligible effect on the size of the final clique output by the algorithm.)

3. In each part P_i, consider all possible subsets S_{ij} of vertices of cardinality t. (Namely, for every part P_i, for every subset j, $|S_{ij}| = t$.)
4. Let $N(S_{ij})$ be the set of vertices in $V'' \setminus S_{ij}$ that are neighbors in G'' to every vertex in S_{ij}. (Hence $S_{ij}, N(S_{ij})$ form the two sides of a complete bipartite graph in G''.) Call S_{ij} *good* if the subgraph of G'' induced on S_{ij} is a clique and $|N(S_{ij})| \geq |V''|/2k - t$.
5. If some set S_{ij} described above is good, then $C = C \bigcup S_{ij}$, and go to the next iteration with the subgraph induced on $N(S_{ij})$ serving as the new G''.
6. If no set S_{ij} is good, then declare V'' poor, and end the phase.

We first analyze the running time of a phase. The number of iterations in a phase is clearly bounded by $|V'|/t$, as each iteration removes at least t vertices from V'. The number of parts considered in an iteration is $|V''|/2kt$. In each part there are $\binom{2kt}{t}$ subsets to consider. For each subset, the test of whether it is good or not takes polynomial time. Hence the whole phase takes polynomial time if $\binom{2kt}{t}$ is polynomial in n. This condition governs the choice of t. We shall be interested in the case where $k \leq (\log n)^b$ for some constant $b > 0$, in which case we can take $t = \log n / \log\log n$, ensuring a polynomial running time.

We now analyze the output of a phase.

LEMMA 2.3. *If a phase declares a set V'' poor, then indeed the subgraph of G induced on V'' does not contain a clique of size $|V''|/2k$.*

Proof. Assume that the subgraph induced on V'' contains a clique of size $|V''|/2k$. Then by the pigeon-hole principle, at least one of the parts P_i will contain at least t vertices from this clique. The subset that corresponds to these t vertices must be good (it is a clique and has the rest of the clique vertices as its neighbors), and hence V'' will not be declared poor. □

Note that Lemma 2.3 does not claim an if and only if relation. Step 5 of an iteration may succeed even if the subgraph induced on V'' does not contain a clique of size $|V''|/2k$, and then the algorithm does not declare V'' poor.

LEMMA 2.4. *If a phase ends by outputting the set C, then this set contains at least $t \log_{3k}(n'/6kt)$ vertices, and these vertices form a clique in G'.*

Proof. Each iteration of the phase adds t vertices to C. To lower bound the number of iterations in a phase, let n'' denote the number of vertices in the beginning of an iteration. Then the next iteration starts with at least $n''/2k - t$ vertices. When $t < n''/6k$, then this number is at least $n''/3k$. Hence the number of iterations needed to reduce $|V''|$ from $|V'|$ to $6kt$ is at least $\log_{3k}(|V'|/6kt)$. This gives the desired lower bound on the number of vertices in C.

The fact that the vertices of C form a clique in G' (and hence also in G) follows from the fact that each subset of vertices that is added into C is a clique and makes a complete bipartite graph with all vertices added after it. □

3. An $O(n(\log\log n)^2/(\log n)^3)$ approximation ratio. Without loss of generality we assume that the approximation algorithm for max clique knows the size of the maximum clique in the input graph. (There are only n possible values to try out, or even only $\log n$, as it suffices for our purpose to know the size within a factor of 2.) We divide possible maximum clique sizes into three ranges, applying a different algorithm in each case.

If the maximum clique size is below $n/(\log n)^3$, simply output a single vertex, achieving an $O(n/(\log n)^3)$ approximation ratio. If the maximum clique size is above $n/(\log n)^3$, the algorithm presented in section 2 finds in polynomial time a clique of size $\Omega((\log n/\log\log n)^2)$. This gives an $O(n(\log\log n)^2/(\log n)^3)$ approximation

ratio for max clique whenever the size of the maximum clique is $O(n/\log n)$. If the maximum clique size is above $n/\log n$, we use a modified version of our algorithm, as described below, so as to find cliques of size larger than $(\log n/\log\log n)^2$.

The key to the improvement is the use of a specialized algorithm for finding large cliques in graphs that have cliques of size larger than $2n\log\log n/\log n$. For this purpose we shall use the algorithm of Boppana and Halldorsson [4]. (Potentially, the more complicated algorithm of Alon and Kahale [1] can be used here instead of [4].)

The algorithm of [4] is based on the known fact from Ramsey theory that any graph on $n = \binom{s+r-2}{s-1}$ vertices contains either an independent set of size r or a clique of size s. Moreover, there is an efficient algorithm for finding one of the two. In the context of approximating clique, finding a clique of size s may be the desirable event of the algorithm, whereas finding an independent set of size r can serve as the event of discovering a poor subgraph (in the terminology of our paper, provided that the input graph has a clique of size greater than n/r), and this subgraph can be removed. We shall use the following proposition regarding the performance guarantee of the algorithm of [4]. (For a proof, see [6], for example.)

PROPOSITION 3.1. *In a graph that has a clique larger than* $2n\log\log n/\log n$*, the algorithm of* [4] *produces a clique of size at least* $(\log n)^3/6\log\log n$.

The above immediately implies an $O(n(\log\log n/\log n)^3)$ approximation algorithm for max clique. If the input graph has a clique larger than $2n\log\log n/\log n$, use the algorithm of [4]. Otherwise, use our algorithm from section 2.

We can save an $\Omega(\log\log n)$ factor in the approximation ratio by adapting the approach of Halldorsson [6] (which he used to save an $\Omega(\log\log n)$ factor in the approximation ratio for the chromatic number) to our context.

Recall the notion of a good subgraph S_{ij} from section 2. It required in particular that $|N(S_{ij})| \geq n''/2k - t$. Modify the definition of good to require that $|N(S_{ij})| > n_{\text{test}} - t$, where n_{test} is the largest value still satisfying

$$n_{\text{test}} \leq \left(\frac{\log n_{\text{test}}}{2\log\log n_{\text{test}}}\right)\cdot\left(\frac{n''}{2k}\right).$$

Include also the following test which is done in the case that $\frac{n''}{2k} - t \leq |N(S_{ij})| \leq n_{\text{test}} - t$. Run the algorithm of [4] on the subgraph induced on $S_{ij} \cup N(S_{ij})$. If it finds a clique of size at least $(\log n_{\text{test}})^3/6\log\log n_{\text{test}}$, join this clique to C and end the algorithm. Otherwise, do not consider S_{ij} to be good (and if no subset of size t is found to be good in the new sense, declare V'' poor).

The analysis of the modified algorithm is similar in many respects to that of the algorithm of section 2. We present here the changes to the proofs of Lemmas 2.3 and 2.4.

LEMMA 3.2. *If a phase of the modified algorithm declares a set* V'' *poor, then indeed the subgraph of* G *induced on* V'' *does not contain a clique of size* $|V''|/2k$.

Proof. Assume that the subgraph induced on V'' contains a clique of size $|V''|/2k$. Then by the pigeon-hole principle, at least one of the parts P_i will contain at least t vertices from this clique. Let S_{ij} be such a subset. Then $|N(S_{ij})| \geq n''/2k - t$. If $|N(S_{ij})| > n_{\text{test}} - t$, then S_{ij} is good, and V'' will not be declared poor. If $|N(S_{ij})| \leq n_{\text{test}} - t$, then the subgraph induced on $S_{ij} \cup N(S_{ij})$ contains n_{test} vertices (if it contains fewer vertices, add to it vertices arbitrarily) and a clique of size $n''/2k = n_{\text{test}}2\log\log n_{\text{test}}/\log n_{\text{test}}$. Then by Proposition 3.1, the algorithm of [4] finds a clique of size $(\log n_{\text{test}})^3/6\log\log n_{\text{test}}$, and the phase ends without declaring V'' poor. ∎

LEMMA 3.3. *Let k and t be such that $\log n/2\log\log n < k < \log n$ and $t = \log n/\log\log n$. If a phase ends by outputting the set C, then this set is a clique on at least $\Omega(t\log_b n')$ vertices, where $b = \Theta(k\log\log n'/\log n')$. In particular, if a graph has a clique of size $\Theta(n\log\log n/\log n)$, the algorithm finds a clique of size $\Omega((\log n)^2/\log\log n)$.*

Proof. The proof is a modification of the proof of Lemma 2.4. We present the differences. The reader is advised to recall the new definition of a good subgraph (that appears prior to Lemma 3.2).

Consider iterations only up to the point where $n'' < \sqrt{n'}$ (ensuring that $\log n'' = \Theta(\log n')$, a fact that simplifies our computations). If before that point the new test finds a clique of size $(\log n_{\text{test}})^3/6\log\log n_{\text{test}}$, then we are done, because n_{test} is large enough to make this clique size $\Omega((\log n')^3/\log\log n')$. If the new test does not find such a clique, then in every iteration the good set S_{ij} that was found had $|N(S_{ij})| > n_{\text{test}}$ (where the value of n_{test} depends on the particular iteration). This means that n'' decreases by a factor of $O(k\log\log n'/\log n')$ between iterations, rather than $O(k)$. The number of iterations becomes at least $\log_b\sqrt{n'}$, where $b = \Theta(k\log\log n'/\log n')$. □

Summing up, for every value of k we can approximate a clique within a ratio of $O(n(\log\log n)^2/(\log n)^3)$, in graphs with cliques of size n/k. For $k \le \log n/2\log\log n$, use the algorithm of [4]; for $\log n/2\log\log n < k < \log n$, use the algorithm of this section; for $\log n \le k \le (\log n)^3$, use either the algorithm of this section or that of section 2; and for $k > (\log n)^3$, just output a single vertex.

4. Discussion. Extending ideas from [4, 2, 6], an $O(n(\log\log n)^2/(\log n)^3)$ approximation ratio is obtained for max clique. This matches the best approximation ratio for the chromatic number. The fact that the two approximation ratios are essentially the same is a consequence of a general framework that we explain below.

Some algorithms for approximating the chromatic number (including [2, 6]) are based on repeatedly finding large independent sets (which serve as color classes). To find a large independent set, they use the fact that every subgraph of a k-colorable graph is itself k-colorable. Hence every subgraph S has an independent set of size at least $|S|/k$.

This principle cannot be applied directly when approximating maximum independent set or max clique. It is not true that in a graph with a clique of size n/k every subgraph S has a clique of size $|S|/k$. The new idea of our paper is to ignore this fact. We run our approximation algorithms for max clique under the assumption that every subgraph does have a clique of size $|S|/k$ or, in fact, slightly smaller. (We chose $|S|/2k$, but the constant 2 is arbitrary and can be replaced by any other constant greater than 1.) For some subgraphs encountered by the algorithm, this assumption is incorrect. However, then one of two things happens: either the algorithm works anyway, or it gets stuck. The point is that, in any case, we make progress. If the algorithm works, we do not care that the assumption was incorrect. If the algorithm gets stuck, then we deduce that the subgraph on which the algorithm got stuck is poor and remove it from the input graph. In the graph that remains the relative size of the maximum clique increases, making the task of finding a large clique easier.

Some other principles that are used in algorithms for approximate coloring also have an analogue in the context of max clique (or max independent set). An instructive example is the algorithm of Alon and Kahale [1] for finding independent sets of size roughly $n^{3/4}$ in graphs that have independent sets of size somewhat larger than

$n/3$. This algorithm is based on the approach of Karger, Motwani, and Sudan [9] for coloring 3-colorable graphs with roughly $n^{1/4}$ colors. The approach of [9] uses semidefinite programming to obtain a so-called vector 3-coloring of the graph and also uses the idea of Wigderson [12] that the neighbors of a vertex in a 3-colorable graph make a bipartite subgraph. Interestingly, both these principles have their analogues in the algorithm of [1]. On the other hand, it is not clear to what extent the principles used in the algorithms of [3, 7] can be used in the context of finding large independent sets in graphs that are not k-colorable but do have an independent set of size roughly n/k.

Let us note that approximate coloring can be performed by repeatedly approximating maximum independent set. This combined with the known hardness of approximation results for maximum independent set implies that the approximation ratio for max clique can be at most a constant factor better than that of min coloring. (See [6] for more details.) This leads to the following interesting question.

- Are the best possible approximation ratios for max clique and the min chromatic number the same (up to multiplicative constant factors)?

Boppana and Halldorsson [4] pointed out connections between approximating max clique and Ramsey theory. In an approach similar to our algorithm (in fact, their algorithm inspired ours), they remove "poor" subgraphs from the input graph. In their case, the poor subgraphs are large enough independent sets, whose existence (if the graph has no large clique) is guaranteed by Ramsey theory. Moreover, their nonexistence suggests an efficient algorithm for finding relatively large cliques (because the relevant arguments in Ramsey theory are constructive). In our case, we define a poor subgraph in such a liberal way that Ramsey theory becomes unnecessary in order to argue about its existence. Specifically, a poor subgraph S is one that does not contain a clique of size $|S|/2k$, whereas our approximation algorithm is satisfied by finding a clique which is very much smaller than $n/2k$. Clearly, either such a clique exists, or the whole graph is poor. Hence unlike Ramsey theory, existence is not an issue here. The only issue is to have an *efficient* algorithm that finds either a clique or a poor subgraph.

Nevertheless, there are connections between our algorithm and Ramsey theory, and we point them out as they may prove fruitful in the future. There is a more general version of the classical Ramsey numbers. Given parameters r and s, let $f(r, s, n)$ denote the minimum over all n vertex graphs that have no s-clique of the maximum cardinality of a subgraph that has no r-clique. In our context of approximating clique, we could use lower bounds on $f(r, s, n)$, provided that certain conditions hold:

1. $f(r, s, n) > kr$.
2. The lower bound is constructive: there is an efficient algorithm for finding either an s-clique or a subgraph on $f(r, s, n)$ vertices without an r-clique.

Using an algorithm similar to that of section 2 we could then find cliques of size roughly s in graphs that have cliques of size n/k. The current bounds known for the function $f(r, s, n)$ [11] are too weak to offer improved approximation ratios for max clique. Let us remark that previously published work on $f(r, s, n)$ dealt only with the case that $r < s$, which is the only case that makes sense in the context of Ramsey theory. However, in our context, where we seek a constructive version, the case $r \geq s$ also makes sense.

Acknowledgments. The author thanks Shimon Kogan for helpful discussions and thanks Magnus Halldorsson, Robert Krauthgamer, and Michael Langberg for useful comments on earlier versions of this manuscript.

REFERENCES

[1] N. Alon and N. Kahale, *Approximating the independence number via the θ-function*, Math. Programming, 80 (1998), pp. 253–264.

[2] B. Berger and J. Rompel, *A better performance guarantee for approximate graph coloring*, Algorithmica, 5 (1990), pp. 459–466.

[3] A. Blum and D. Karger, *An $\tilde{O}(n^{3/14})$-coloring algorithm for 3-colorable graphs*, Inform. Process. Lett., 61 (1997), pp. 49–53.

[4] R. Boppana and M. Halldorsson, *Approximating maximum independent sets by excluding subgraphs*, BIT, 32 (1992), pp. 180–196.

[5] U. Feige and J. Kilian, *Zero knowledge and the chromatic number*, J. Comput. System Sci., 57 (1998), pp. 187–199.

[6] M. Halldorsson, *A still better performance guarantee for approximate graph coloring*, Inform. Process. Lett., 45 (1993), pp. 19–23.

[7] E. Halperin, R. Nathaniel, and U. Zwick, *Coloring k-colorable graphs using relatively small palettes*, J. Algorithms, 45 (2002), pp. 72–90.

[8] J. Håstad, *Clique is hard to approximate within $n^{1-\epsilon}$*, Acta Math., 182 (1999), pp. 105–142.

[9] D. Karger, R. Motwani, and M. Sudan, *Approximate graph coloring by semidefinite programming*, J. ACM, 45 (1998), pp. 246–265.

[10] S. Khot, *Improved inapproximability results for maxclique, chromatic number and approximate graph coloring*, in Proceedings of the 42nd Annual Symposium on Foundations of Computer Science, Las Vegas, NV, 2001, IEEE Computer Society Press, Los Alamitos, CA, 2001, pp. 600–609.

[11] B. Sudakov, *A new lower bound for a Ramsey-type problem*, Combinatorica, to appear.

[12] A. Wigderson, *Improving the performance guarantee of approximate graph coloring*, J. ACM, 30 (1983), pp. 729–735.

SIAM J. DISCRETE MATH.
Vol. 18, No. 2, pp. 226–240

OPTIMIZING BULL-FREE PERFECT GRAPHS*

CELINA M. H. DE FIGUEIREDO† AND FRÉDÉRIC MAFFRAY‡

Abstract. A bull is a graph with five vertices a, b, c, d, e and five edges ab, ac, bc, da, eb. Here we present polynomial-time combinatorial algorithms for the optimal weighted coloring and weighted clique problems in bull-free perfect graphs. The algorithms are based on a structural analysis and decomposition of bull-free perfect graphs.

Key words. graph algorithms, perfect graphs, analysis of algorithms and problem complexity, combinatorial optimization

AMS subject classifications. 05C85, 05C60, 68Q25, 90C27

DOI. 10.1137/S0895480198339237

1. Introduction.

A graph G is called *perfect* if the vertices of every induced subgraph G' of G can be colored with $\omega(G')$ colors, where $\omega(G')$ is the maximum clique size in H. Berge [1] introduced perfect graphs and conjectured the following characterization: *A graph is perfect if and only if it contains no odd hole and no odd antihole as an induced subgraph,* where a *hole* is a chordless cycle with at least five vertices, and an *antihole* is the complement of a hole. Graphs with no odd hole and no odd antihole have become known as *Berge* graphs. This conjecture, known as the strong perfect graph conjecture, was proved recently by Chudnovsky et al. [5]; thus *every Berge graph is perfect.* One problem that is not yet solved in this context is the existence of a combinatorial algorithm to compute the chromatic number of a perfect graph. Here we will give such an algorithm for bull-free Berge graphs, i.e., graphs with no induced subgraph isomorphic to a bull, where a *bull* is a graph with five vertices a, b, c, d, e and five edges ab, bc, cd, be, ce (see Figure 1). Our algorithm is based on specific properties of these graphs. Let us recall that Chvátal and Sbihi [3] proved the validity of the strong perfect graph conjecture for bull-free graphs, and subsequently Reed and Sbihi [18] gave a polynomial algorithm for recognizing bull-free Berge graphs.

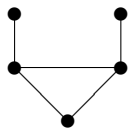


FIG. 1. *The bull.*

In this paper, we present polynomial-time algorithms for solving the following optimization problems for bull-free perfect graphs: find a largest clique, a largest stable

*Received by the editors May 27, 1998; accepted for publication (in revised form) February 8, 2004; published electronically October 1, 2004. This research was partially supported by CNPq and CAPES/COFECUB.

http://www.siam.org/journals/sidma/18-2/33923.html

†Instituto de Matemática, Universidade Federal do Rio de Janeiro, Caixa Postal 68530, 21945-970 Rio de Janeiro, RJ, Brazil (celina@cos.ufrj.br).

‡Laboratoire Leibniz-IMAG, CNRS, 46 avenue Félix Viallet, 38031 Grenoble Cedex, France (frederic.maffray@imag.fr).

set, a minimum coloring, and a minimum clique covering. We actually present algorithms which solve the weighted versions of these problems, defined as follows. We are given a graph G with vertices $v_1, \ldots, v_n$ and positive integer weights $w(v_1), \ldots, w(v_n)$.

Maximum weighted clique problem. Find a clique K of G, such that the weight of K, defined as the sum of the weights of the vertices of K, $w(K) = \sum_{x \in K} w(x)$, is maximum over all cliques of G.

Maximum weighted stable set problem. Find a stable set S of G, such that the weight of S, defined as the sum of the weights of the vertices of S, $w(S) = \sum_{x \in S} w(x)$, is maximum over all stable sets of G.

Minimum weighted coloring problem. Find stable sets $S_1, \ldots, S_t$ and integers $W(S_1), \ldots, W(S_t)$, such that

$$\sum_{S_i \ni v_j} W(S_i) \geq w(v_j) \qquad (\forall v_j) \tag{1}$$

and the sum $W(S_1) + \cdots + W(S_t)$ is minimum over all sets of integers that satisfy (1).

Minimum weighted clique covering problem. Find cliques $K_1, \ldots, K_t$ and weights $W(K_1), \ldots, W(K_t)$, such that

$$\sum_{K_i \ni v_j} W(K_i) \geq w(v_j) \qquad (\forall v_j) \tag{2}$$

and the sum $W(K_1) + \cdots + W(K_t)$ is minimum over all sets of integers that satisfy (2).

Recall that if G is a perfect graph, classical polyhedral considerations (see [12]) imply that (a) the optimal value of the maximum weighted clique problem and of the minimum weighted coloring problem are equal; (b) there exists a minimum weighted coloring that satisfies (1) with equality for every vertex. The same facts hold for the maximum weighted stable set problem and the minimum weighted clique covering problem.

It is possible to color every perfect graph optimally and in polynomial time, thanks to the algorithm of Grötschel, Lovász, and Schrijver [12]; but that algorithm is based on the ellipsoid method and may be rather complex and impractical. In contrast, the algorithm we are going to present here exploits the combinatorial structure of bull-free graphs and is fairly transparent. We will find it convenient, however, to use the following argument. Let $\mathcal{C}$ be a self-complementary class of perfect graphs. *If there exists a strongly polynomial-time algorithm A that can compute the weighted clique number of any graph G in $\mathcal{C}$ in time $O(n^k)$ (n being the number of vertices of G), then there exists a strongly polynomial-time algorithm A' that can construct a minimum weighted coloring for any graph G in $\mathcal{C}$ in time $O(n^{k+4})$.* This argument is implicit in [12, section 9.4] and in [19, Proof of Corollary 67.5c and Theorem 67.6], and we do not copy its proof here. It suffices to note that A' consists mainly in at most n^4 calls to A applied to weighted subgraphs of G and $\overline{G}$; this is independent of the method that A is based on. Since the class of bull-free Berge graphs that we consider here is self-complementary, this argument can be applied; this allows us, therefore, to focus on only one of the above four problems, namely, the maximum weighted clique number.

Roughly speaking, our algorithm follows a decomposition procedure for bull-free Berge graphs; with each bull-free Berge graph G a decomposition tree is associated;

our algorithm uses some known polynomial-time algorithms to solve the problem for the leaves of the tree (these indecomposable graphs turn out to belong to well-known classical families); it then recursively combines solutions along the tree, upward from children to parent, up to the root G. A key point in our proofs is the use of decomposition theorems in order to show how to combine the solutions properly from the children to the parent. Another key point is to show that the number of tree nodes is polynomial so that the total running time of our algorithm itself is polynomial.

In order to present this algorithm exactly and to justify it, a number of definitions and results must be recalled; also, some new results will be proved. The algorithm will be described precisely in section 6.

2. Definitions. Apart from standard graph-theoretic terms, we use the verbs "see" and "miss" instead of "be adjacent to" and "not be adjacent to." The neighborhood $N(x)$ of a vertex x in a graph G is the set of all vertices of $G \setminus x$ that see x. A chordless path on k vertices is denoted by P_k. Unless otherwise specified, the phrase "G contains H" means "G contains H as an induced subgraph." Note also that a graph G is bull-free if and only if its complement $\overline{G}$ is bull-free. For any subset X of vertices of a graph G, we let $G[X]$ denote the subgraph of G induced by X.

Weakly triangulated graphs. A graph is called *weakly triangulated* if it does not contain a hole or an antihole. Hayward [13] proved that all weakly triangulated graphs are perfect. Subsequently, Hayward, Hoàng, and Maffray [14] gave polynomial-time algorithms that solve the four optimization problems above for weakly triangulated graphs.

Transitively orientable graphs. A graph is called *transitively orientable* if it admits a *transitive orientation*, i.e., an orientation of its edges with no circuit and with no P_3 abc with the orientation $\vec{ab}$ and $\vec{bc}$. Such graphs are also called *comparability graphs*. A well-known subclass of comparability graphs is the class of P_4-free graphs, also called *cographs* [8]. Indeed, a result Seinsche [20] states is that for every P_4-free graph G on at least two vertices, either G or its complement $\overline{G}$ is disconnected; from this it is easy to derive that every P_4-free graph is transitively orientable.

Partial vertices, homogeneous sets. Given a subset of vertices S in a graph G, a vertex from $G \setminus S$ is *partial on* S or S*-partial* if it has at least one neighbor and at least one nonneighbor in S. A vertex from $G \setminus S$ is *impartial on* S if it either sees all vertices of S or misses all vertices of S.

A *homogeneous set* (or *module*) in a graph $G = (V, E)$ is a subset $S \subseteq V$ such that every vertex from $G \setminus S$ sees either all or none of S. A homogeneous set S is *proper* if $2 \leq |S| \leq |V| - 1$. Note that if S is a homogeneous set of G, then it is also a homogeneous set of the complementary graph $\overline{G}$. A graph is called *prime* if it has no proper homogeneous set.

Homogeneous pairs. A *homogeneous pair* [3] in a graph G is a pair of disjoint subsets of vertices Q_1, Q_2 such that all Q_1-partial vertices are in Q_2; all Q_2-partial vertices are in Q_1; at least one of Q_1, Q_2 includes at least two vertices; and there are at least two vertices in $G \setminus Q_1 \cup Q_2$. Note that if Q_1, Q_2 is a homogeneous pair in G, then it is also a homogeneous pair in $\overline{G}$.

Whenever a graph G admits a homogeneous pair Q_1, Q_2, we will denote by T_i the set of vertices of $G \setminus (Q_1 \cup Q_2)$ that see all of Q_i and miss all of Q_{3-i} $(i = 1, 2)$, by T the set of vertices of $G \setminus (Q_1 \cup Q_2)$ that see all of $Q_1 \cup Q_2$, and by Z the set of vertices of $G \setminus (Q_1 \cup Q_2)$ that miss all of $Q_1 \cup Q_2$. Following [3], we may decompose G along this homogeneous pair into two graphs H and Q defined as follows. The graph H is made from $G \setminus (Q_1 \cup Q_2)$ by adding four vertices u_1, u_2, s_1, s_2 with edges u_1s_1, u_2s_2,

u_1s_2, u_2s_1, s_1s_2 and with edges tu_i, ts_i for every vertex $t \in T_i \cup T$ for each $i = 1, 2$. The graph Q is the subgraph of G induced by $Q_1 \cup Q_2$.

Let us say that a homogeneous pair Q_1, Q_2 is *interesting* if both Q_1, Q_2 induce connected subgraphs of G, $Q_1 \cup Q_2$ contains a square with an edge in Q_1 and an edge in Q_2, $T_1 \neq \emptyset$, $T_2 \neq \emptyset$, and there exists an edge t_1t_2 with $t_1 \in T_1$, $t_2 \in T_2$.

In [7] the following result was proved (although not stated explicitly this way).

THEOREM 2.1. *Let G be a prime bull-free Berge graph G. If G contains an even hole, then G admits a "box partition."*

The *box partition* is a structural concept whose exact definition we defer to section 3. The proof of that theorem in [7] is actually a polynomial-time algorithm which, given a bull-free Berge graph G, produces a proper homogeneous set of G, or asserts that G contains no even hole, or produces a box partition. Our interest in the box partition here is due mainly to the following lemma.

LEMMA 2.2 (the transitive box partition lemma). *Let G be a bull-free Berge graph with no antihole. If G has a box partition, then G admits a transitive orientation.*

This lemma will be proved in section 3.

THEOREM 2.3. *Let G be a prime bull-free Berge graph that contains a hole and an antihole. Then the following hold.*

(I) *The graph G contains an interesting homogeneous pair Q_1, Q_2.*

(II) *If H, Q are the two graphs obtained by decomposing G along an interesting homogeneous pair, then both H and Q are bull-free Berge graphs.*

(III) *It is possible to build a solution of the maximum weighted clique problem on G from a solution of the same problem on H and Q with appropriately defined vertex-weights.*

This theorem will be proved in section 5.

3. Boxes and transitive orientations. For any subset B of vertices in a graph G, we let $M(B)$ denote the set of vertices of $G \setminus B$ that are partial on B.

DEFINITION 3.1 (the box partition). *Let G be a graph with vertex set V. We call box partition any partition of V into disjoint nonempty subsets called the boxes, inducing connected subgraphs which satisfy the following properties:*

(i) *Each box is labeled either "odd" or "even" (each vertex will be labeled odd or even accordingly), and there is no edge between two odd boxes or between two even boxes.*

(ii) *For each box B such that $M(B) \neq \emptyset$, there exist in $V - B$ two auxiliary adjacent vertices a_B and a'_B, such that a_B sees all of B and misses all of $M(B)$, while a'_B sees all of $M(B)$ and misses all of B.*

Remark 1. When G is bull-free, the fact that a'_B sees every vertex of $M(B)$ is a consequence of the other facts given in property (ii).

Indeed, if a'_B missed a vertex x of $M(B)$, then there should exist adjacent vertices u, v in B such that x sees u and misses v, and then a_B, u, v, x, a'_B would be a bull.

Let us note that if a bull-free perfect graph G with no proper homogeneous set and no $\overline{C}_6$ admits a box partition, then two further properties hold. Say that two neighborhoods $N(u), N(v)$ are *comparable* if $N(u) \subseteq N(v)$ or $N(v) \subseteq N(u)$ holds.

(iii) *Every box is P_4-free.*

(iv) *Any two adjacent vertices in B have comparable neighborhoods in $M(B)$.*

To prove (iii), we recall that a *broom* is the graph made up of a P_4, plus a fifth vertex adjacent to all vertices of the P_4, plus a sixth vertex adjacent to the fifth vertex only. We proved the following result.

LEMMA 3.2 (the broom lemma [7]). *If a bull-free, C_5-free graph contains a broom, then it has a proper homogeneous set which contains the P_4 of the broom.*

Now observe that if a box B contains a P_4, then adding the vertices a_B and a'_B we obtain a broom, and then by the broom lemma G should contain a proper homogeneous set, which is a contradiction.

To prove (iv), suppose on the contrary that some two adjacent vertices u, v in a box B have incomparable neighborhoods in $M(B)$. So there exist a vertex x in $M(B) \cap N(u) - N(v)$ and a vertex y in $M(B) \cap N(v) - N(u)$. Recall the auxiliary vertices a_B, a'_B for B, so that a_B sees u, v, and a'_B and misses x and y, while a'_B sees x and y and misses u and v. If xy is an edge in G, then a_B, u, v, x, y, a'_B is a $\overline{C_6}$. If xy is not an edge in G, then a_B, u, v, x, y is a bull. So (iv) is proved.

3.1. Proof of the transitive box partition lemma. Given a box partition, any edge whose endpoints are in different boxes will be called a *vertical edge*. (Necessarily, for any such edge, one endpoint is in an odd box and the other is in an even box.) The other edges will be called *horizontal*; i.e., a horizontal edge is any edge whose two endpoints are in the same box. Recall from (iii) that each box B is P_4-free, and recall that every P_4-free graph admits a transitive orientation [11]. Let $\mathcal{L}(B)$ be a transitive orientation for each box B. All edges xy of G are oriented according to the following rules:

- *Rule* V0. If an edge is vertical, orient it from its even extremity to its odd extremity.
- *Rule* H1. If x, y are in an even (resp., odd) box B and x has strictly more neighbors than y in $M(B)$, then orient xy from x to y (resp., from y to x).
- *Rule* H2. If x, y are in an even (resp., odd) box B and have the same neighborhood in $M(B)$, and if there exists a P_4 $yxvu$ with $u \in M(B)$ and $v \in B$, then orient the edge yx from y to x (resp., from x to y).
- *Rule* H3. If x, y are in a box B and do not satisfy the hypotheses of Rules H1 and H2, then orient xy according to $\mathcal{L}(B)$.

After these rules are applied, every edge of G has received an orientation. We claim that this is a transitive orientation of G. To certify this claim, we have to check that these combined rules are consistent (i.e., noncontradictory) and that they produce no P_3 xyz with orientation $\vec{xy}$ and $\vec{yz}$ and no circuit. Note that a result of Ghouila-Houri [10] shows that if a graph admits an orientation with no directed P_3, then it admits an acyclic transitive orientation.

CLAIM 1. *The rules are consistent.*

Proof. We need only prove that no edge must be oriented by the rules in two opposite ways. Clearly, the vertical edges are oriented consistently. Since Rules H1, H2, and H3 apply to edges of different types, they cannot contradict each other. Rule H1 cannot orient an edge in two opposite ways, by property (iv) of the box partition. Clearly Rule H3 also cannot orient an edge in two opposite ways. So the only case of inconsistency would be the following: some horizontal edge xy ($x, y \in$ box B) must be oriented in one way because there is a P_4 $uvxy$ with $u \in M(B)$ and $v \in B$ (Rule H2) and must also be oriented in the opposite way because there is a P_4 $ztyx$ with $z \in M(B)$ and $t \in B$, and x and y have the same neighbors in $M(B)$. Clearly $v \neq t$ (but $u = z$ is possible). Let a'_B be the auxiliary vertex of B given by property (ii) of the box partition; so a'_B sees u and z and misses all of v, x, y, t. In addition, v must see t or else $vxyt$ is a P_4 in B. Now, either u sees t or z sees v, or else property (iv) is contradicted for v, t. By symmetry we may assume that u sees t, but then v, t, u, a'_B, x is a bull. So Claim 1 is proved. □

CLAIM 2. *The rules produce no P_3 xyz with orientation $\vec{xy}$ and $\vec{yz}$.*

Proof. Suppose the contrary. Rules V0 and H1 imply easily that the vertices x, y, z cannot be in different boxes. So, and by symmetry, we may assume that they lie in one odd box B. Note that one of the edges xy, yz must have been oriented by Rule H1 or by Rule H2.

Case 1. The edge xy was oriented from x to y by Rule H1. This hypothesis means that there exists a vertex u in $M(B) \cap N(y) - N(x)$. If u misses z, then yz should be oriented by Rule H1 from z to y, which is a contradiction; so u sees z. Now x, y, z, u, a'_B is a bull, which is a contradiction.

So xy is not oriented by Rule H1, and then x and y have the same neighborhood in $M(B)$.

Case 2. The edge xy was oriented from x to y by Rule H2. This means that x and y have the same neighborhood in $M(B)$ and that there exists a P_4 $uvxy$ with $u \in M(B)$ and $v \in B$. Because B has no P_4, we have that v sees z. In addition, if u sees z, then u, v, z, y, a'_B is a bull. Moreover, y and z have the same neighborhood in $M(B)$. For, if y has more neighbors than z in $M(B)$, then by Rule H1, we have yz oriented from z to y. If there exists $w \in M(B) \cap N(z) - N(y)$, then w misses x also and Rule H2 orients xy from y to x, which is a contradiction. Thus we can apply Rule H2 to the P_4 $uvzy$ which forces yz to be oriented from z to y, which is a contradiction.

Case 3. The edge yz was oriented from y to z by Rule H1. This hypothesis means that there exists a vertex u in $M(B) \cap N(z) - N(y)$. Recall that x and y have the same neighborhood in $M(B)$. Thus we can apply Rule H2 to the P_4 $uzyx$ which forces xy to be oriented from y to x, which is a contradiction.

Case 4. The edge yz was oriented from y to z by Rule H2. This means that there exists a P_4 $uvyz$ with $u \in M(B)$ and that y and z have the same neighborhood in $M(B)$. Recall that x and y also have the same neighborhood in $M(B)$. Vertex x misses v or else u, v, x, y, z is a bull. Thus we can apply Rule H2 to the P_4 $uvyx$ which forces xy to be oriented from y to x, which is a contradiction.

In all cases a contradiction arises; so Claim 2 is proved. □

CLAIM 3. *The rules produce no circuit.*

Proof. By Rule V0, a circuit may occur only inside a box. Without loss of generality, let us assume that an odd box B contains a circuit $C = c_1 \cdots c_r$. Observe that if an edge xy in B is oriented from x to y, then y has at least as many neighbors as x in $M(B)$ because of Rule H1. Therefore, if somewhere along the circuit two consecutive vertices c_i, c_{i+1} satisfy $N(c_i) \cap M(B) \subset N(c_{i+1}) \cap M(B)$ (where $\subset$ denotes strict inclusion), then necessarily elsewhere on the cycle some two consecutive vertices c_j, c_{j+1} must satisfy $N(c_{j+1}) \cap M(B) \subset N(c_j) \cap M(B)$. But then this inclusion contradicts the fact that the edge $c_j c_{j+1}$ is oriented from c_j to c_{j+1}. So, all vertices along C have the same neighborhood in $M(B)$. Moreover, since $\mathcal{L}(B)$ has no circuit, at least one edge of C must have been oriented by Rule H2. So let us assume that there is a P_4 uvc_1c_2 with $u \in M(B)$ and $v \in B$. Since all the vertices of C have the same neighborhood in $M(B)$, in particular, they all miss u, and v is not one of the c_i's. Let j be the last subscript such that v misses c_j ($j \geq 2$). Then $uvc_{j+1}c_j$ is a P_4 implying that the edge $c_j c_{j+1}$ is oriented from c_{j+1} to c_j, which is a contradiction. So Claim 3 is proved. □

Now the proof of Lemma 2.2 is complete.

4. More about the box partition. Everywhere in this section we reserve the letter G for a prime bull-free Berge graph that contains a hole. We let k denote the

length of a shortest even hole in G. So there are k nonempty sets $V_1, \ldots, V_k$ such that every vertex in V_i sees every vertex in V_{i+1} (modulo k) and there is no other edge between two V_i's. By [7] G admits a box partition built from the V_i's. We will need to use some properties and notation from [7] concerning this box partition. In particular, the boxes of this partition are classified as either "central" or "peripheral" with the following properties that will be used here:

(a) If $k \geq 8$, every central box is a homogeneous set. (To see this, recall from [7] that when $k \geq 8$ the central boxes are the connected components of the k sets $V_1, \ldots, V_k$. If $B \subseteq V_1$, then the proof of [7, Lemma 3] gives $M(B) \subseteq A_2$, where A_2 is the set of vertices that are adjacent to all of $V_1 \cup V_3 \cup \cdots \cup V_{k-1}$; hence every vertex adjacent to B is adjacent to all of B, and B is a homogeneous set.)

(b) If $k = 6$, there are eight sets $D_1, \ldots, D_6, A_1, A_2$ such that the central boxes are exactly the connected components of these eights subgraphs. The sets $D_1, \ldots, D_6$ play symmetrical roles; the sets A_1, A_2 play symmetrical roles. Moreover, if B is a box in D_1 or in A_1, then $M(B) \subseteq D_4 \cup A_2$. There are vertices v_2, v_6 that see all of $D_1 \cup A_1$ and none of $D_4 \cup A_2$; there are vertices v_3, v_5 that see all of $D_4 \cup A_2$ and none of $D_1 \cup A_1$; v_2v_3 and v_5v_6 are the only edges between v_2, v_3, v_5, v_6.

(c) [7, Lemma 4, Property (v)] In a peripheral box B any two adjacent vertices have comparable neighborhoods in $M(B)$.

LEMMA 4.1. *The graph G contains an antihole if and only if it contains a $\overline{C}_6$.*

Proof. The "if" part is trivial. Conversely, suppose that G contains no $\overline{C}_6$. Then Lemma 2.2 implies that G is transitively orientable, and hence it contains no antihole. □

When a C_4 (a "square") is denoted $uvxy$ it is understood that ux and vy are the two nonadjacent pairs.

DEFINITION 4.2 (blocking square). *We say that a square $uvxy$ is blocking if u, v belong to one box B and x, y belong to another box B'. The edges uv and xy are called the blocking edges of the square. Likewise any edge uv with both endpoints in one box is called blocking whenever it is one of the two blocking edges of a blocking square.*

Remark 2. In Definition 4.2, clearly one of B, B' is an even box and the other is an odd box. Clearly too, we have $\{x, y\} \subseteq M(B)$ and $\{u, v\} \subseteq M(B')$.

LEMMA 4.3. *The graph G contains a $\overline{C}_6$ if and only if G contains a blocking square.*

Proof. First suppose that G contains a blocking square $uvxy$ with the notation as in Definition 4.2. Let a_B and a'_B be the auxiliary vertices for B. Then a_B, a'_B, u, v, x, y induce a $\overline{C}_6$.

Conversely, suppose that six vertices $u_1, u_2, \ldots, u_6$ form in G a $\overline{C}_6$, such that the nonadjacent pairs are u_iu_{i+1} (subscripts here are understood modulo 6). In the triangle u_1, u_3, u_5 at least two vertices are on the same side of the box partition; say, u_1 and u_3 are in one even box B. If both u_4, u_6 are in an odd box, then $u_1u_3u_6u_4$ is a blocking square. So let us assume without loss of generality that u_4 is in an even box and hence in B. Then u_2 is in an odd box or else $u_3u_1u_4u_2$ would be a P_4 in B. If u_5 is on the odd side, then $u_1u_4u_2u_5$ is a blocking square. So let us assume that u_5 is on the even side and hence in B. Then u_6 is on the odd side or else $u_5u_3u_6u_4$ would be a P_4 in B. But now $u_3u_5u_2u_6$ is a blocking square. □

LEMMA 4.4. *An edge uv in a box B is a blocking edge if and only if the vertices u, v have incomparable neighborhoods in $M(B)$.*

Proof. The "only if" part of the lemma is trivial. Conversely, suppose that u, v have incomparable neighborhoods in $M(B)$; i.e., there exist a vertex $x \in M(B) \cap N(v) - N(u)$ and a vertex $y \in M(B) \cap N(u) - N(v)$. Recall that the auxiliary vertex a_B sees both u, v and misses both x, y. Then x must see y or else a_B, u, v, x, y would be a bull. Now $uvxy$ is a blocking square and uv is a blocking edge. □

At this point it is useful to recall the graph called H_0 in [3] and featured in Figure 2.

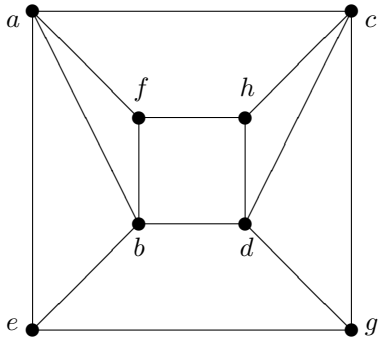


FIG. 2. *The graph* H_0.

LEMMA 4.5. *If* G *contains an antihole, then* G *contains an* H_0.

Proof. By the preceding lemmas we may assume that G admits a blocking square $uvxy$, with blocking edges uv in a box B and xy in a box B'. So the vertices u, v have incomparable neighborhoods in $M(B)$. By [7, Lemma 4, Property (v)] as recalled above, B cannot be a peripheral box. So B is a central box. If $k \geq 8$, then item (a) above implies that B is a proper homogeneous set, which is impossible because u, v have incomparable neighborhoods in $M(B)$. So we have $k = 6$, and we may assume, without loss of generality, that $B \subseteq D_1 \cup A_1$, with the notation of item (b). Now we have

$$u, v \in B \subseteq D_1 \cup A_1, \qquad x, y \in B' \subseteq D_4 \cup A_2.$$

Using the vertices v_2, v_3, v_5, v_6 whose properties are recalled in (b) above, we note that vertex v_2 sees all of $\{u, v, v_3\}$, vertex v_6 sees all of $\{u, v, v_5\}$, vertex v_3 sees all of $\{x, y, v_2\}$, vertex v_5 sees all of $\{x, y, v_6\}$, and there are no other edges between the vertices $u, v, x, y, v_2, v_3, v_5, v_6$. Hence these eight vertices induce an H_0. □

The following result will be useful. Recall that, given a homogeneous pair Q_1, Q_2 in a graph G, we denote by T_i, $i = 1, 2$, the set of vertices in $G \setminus (Q_1 \cup Q_2)$ that see all of Q_i and miss all of Q_{3-i}, by T the set of vertices in $G \setminus (Q_1 \cup Q_2)$ that see all of $Q_1 \cup Q_2$, and by Z the set of vertices in $G \setminus (Q_1 \cup Q_2)$ that see none of $Q_1 \cup Q_2$.

THEOREM 4.6 (see [3]). *Let* G *be a bull-free graph that contains an* H_0 *(with the notation as in Figure* 2*). Then the following hold.*

(i) *G contains a homogeneous pair Q_1, Q_2 such that $a, b \in Q_1$, $c, d \in Q_2$, $e, f \in T_1$, $g, h \in T_2$, and $G[Q_1]$ and $G[Q_2]$ are connected.*

(ii) *If G is connected and prime, then $Z = \emptyset$.*

Proof. Part (i) of the theorem is proved in [3, Theorem 2]; it consists in a polynomial-time algorithm that builds the homogeneous pair Q_1, Q_2 from a given H_0.

To prove part (ii) suppose on the contrary that $Z \neq \emptyset$. Since G is connected, there exists an edge zt with $z \in Z$ and $t \in T \cup T_1 \cup T_2$. If $t \in T_1$, then z, t, a, b, c induce

a bull, which is a contradiction. So we may assume that $t \in T$, and z misses e since $e \in T_1$. Then t sees e, or else z, t, e, a, c would induce a bull. But then z, t, e, a, c, d induce a broom, which is a contradiction to Lemma 3.2. □

5. Proof of Theorem 2.3. Let G be a prime bull-free Berge graph that contains a hole and an antihole. Recall that we want to prove that (I) the graph G contains an interesting homogeneous pair Q_1, Q_2; (II) if H, Q are the two graphs obtained by decomposing G along an interesting homogeneous pair, then both H and Q are bull-free Berge graphs; and (III) it is possible to build a solution of the maximum weighted clique problem on G from a solution of the same problem on H and Q with appropriately defined vertex-weights.

To prove (I), we need only apply Lemma 4.5 and Theorem 4.6 above.

Now let us prove (II). Let $abcd$ be a square with edge ab in Q_1 and edge cd in Q_2. Here again T_i (resp., T) is the set of vertices of $G \setminus (Q_1 \cup Q_2)$ that see all of Q_i and none of Q_{3-i} (resp., all of $Q_1 \cup Q_2$), and $Z = V \setminus (Q_1 \cup Q_2 \cup T_1 \cup T_2 \cup T)$; i.e., no vertex of Z sees any of $Q_1 \cup Q_2$.

Recall that the graph H is obtained from $G \setminus (Q_1 \cup Q_2)$ by adding vertices u_1, u_2, s_1, s_2, edges u_1s_1, u_2s_2, u_1s_2, u_2s_1, s_1s_2, and edges tu_i, ts_i for each i and each vertex $t \in T_i \cup T$.

LEMMA 5.1. *H is perfect and bull-free.*

Proof. Call G^* the subgraph of G induced by $V \setminus ((Q_1 \setminus \{a, b\}) \cup (Q_2 \setminus \{c, d\}))$. Observe that $H \setminus s_1s_2$ is isomorphic to G^*.

First we prove that H is perfect. Consider any induced subgraph H' of H. If H' contains at most one of u_1, s_1 and at most one of u_2, s_2, then H' is isomorphic to one of the subgraphs $G^* \setminus \{a, c\}$, $G^* \setminus \{a, d\}$, so H' is perfect. Suppose now by symmetry that H' contains both u_1 and s_1. Note that s_1 dominates u_1 in H (i.e., $N_H(u_1) \subset N_H(s_1) \cap \{s_1\}$), and thus also in H'. It is well known (see, e.g., [11]) that a minimally imperfect graph cannot contain a pair of vertices such that one dominates the other. So all induced subgraphs of H are perfect, including H itself.

Now suppose that H contains a bull B. It is easy to see that any induced subgraph of H that contains none of the two triangles formed by u_1, s_1, s_2 and u_2, s_1, s_2 is contained in one of the subgraphs $G^* \setminus \{a, c\}$, $G^* \setminus \{a, d\}$ and thus cannot be a bull. So we may assume by symmetry that B contains the triangle u_1, s_1, s_2. Now B must have a vertex adjacent to exactly one of u_1, s_1 and not adjacent to s_2. But H contains no such vertex since u_2 is the only vertex adjacent to exactly one of u_1, s_1. This completes the proof of the lemma. □

Since Q is the subgraph of G induced by $Q_1 \cup Q_2$, the next claim is obvious.

CLAIM 4. *Q is perfect and bull-free.* □

We now prove part (III) of Theorem 2.3. Let us denote by $w(x)$ the weight of a vertex x in G. Define weights for vertices in H as follows. Denote by $\omega(X)$ the maximum weight of a clique in X, and set

$$\begin{aligned} w_H(u_1) = w_H(u_2) &= \omega(Q_1) + \omega(Q_2) - \omega(Q_1 \cup Q_2), \\ w_H(s_1) &= \omega(Q_1 \cup Q_2) - \omega(Q_2), \\ w_H(s_2) &= \omega(Q_1 \cup Q_2) - \omega(Q_1), \\ w_H(x) &= w(x) \qquad \forall x \in G \setminus (Q_1 \cup Q_2). \end{aligned}$$

Say that a set X of vertices in H is of type 0 if $X \cap \{u_1, s_1, u_2, s_2\} = \emptyset$, of type 1 if $X \cap \{u_1, s_1\} \neq \emptyset$ and $X \cap \{u_2, s_2\} = \emptyset$, of type 2 if $X \cap \{u_1, s_1\} = \emptyset$ and $X \cap \{u_2, s_2\} \neq \emptyset$, and of type 3 if $X \cap \{u_1, s_1\} \neq \emptyset$ and $X \cap \{u_2, s_2\} \neq \emptyset$.

Let q be the maximum weight of a clique in H with respect to the weighting w_H, and let C_H be a clique of weight q. We can transform C_H into a clique C_G of weight q in G as follows. If C_H is of type 0, set $C_G = C_H$. If C_H is of type $i \in \{1, 2\}$, let C_G be the union of $C_H \setminus (Q_1 \cup Q_2)$ and of a clique of size $\omega(Q_i)$ in Q_i. If C_H is of type 3, let C_G be the union of $C_H \setminus (Q_1 \cup Q_2)$ and of a clique of size $\omega(Q_1 \cup Q_2)$ in $Q_1 \cup Q_2$.

LEMMA 5.2. *We have $\omega(G) = q$ and C_G is a maximum weighted clique of G.*

Proof. We need only exhibit a q-weighted coloring of G: that will prove both that the clique C_G defined above for G is maximum and that this coloring has minimum weight. The proof of this lemma is essentially the weighted version of the proof of [3, The Homogeneous Pair Lemma].

Recall that $q = \omega(H)$. So there exists a weighted coloring of H of total weight q, that is, a collection of stable sets $S_1^H, \ldots, S_t^H$ of H with corresponding weights $W(S_1^H), \ldots, W(S_t^H)$, such that

$$\sum\{W(S_i^H) \mid S_i^H \ni x\} = w_H(x) \qquad (\forall x \in H)$$

and $W(S_1^H) + \cdots + W(S_t^H) = q$. Split the subscripts $1, 2, \ldots, t$ into sets I_0, I_1, I_2, I_3 by writing $j \in I_i$ if and only if S_j^H is of type i. Thus

$$\begin{aligned} w_H(u_i) &\le \sum\{W(S_j^H) \mid j \in I_i \cup I_3\}, \\ w_H(s_i) &\le \sum\{W(S_j^H) \mid j \in I_i\}. \end{aligned}$$

In addition, since u_i and s_i are adjacent,

$$w_H(u_i) + w_H(s_i) \le \sum\{W(S_j^H) \mid j \in I_i \cup I_3\}.$$

Define a graph F by adding to the subgraph $Q = G[Q_1 \cup Q_2]$ adjacent vertices x_1, x_2 and edges $x_i y$ for all vertices y in Q_i $(i = 1, 2)$. Note that F is isomorphic to the subgraph of G induced by $Q_1 \cup Q_2 \cup \{t_1, t_2\}$, where $t_1 \in T_1$, $t_2 \in T_2$, and $t_1 t_2$ is an edge of G; such vertices exist because Q_1, Q_2 is an interesting homogeneous pair. So F is a perfect graph. Define a weight function W_F on the vertices of F as follows:

$$\begin{aligned} w_F(x_1) &= \sum\{W(S_j^H) \mid j \in I_2\}, \\ w_F(x_2) &= \sum\{W(S_j^H) \mid j \in I_1\}, \\ w_F(y) &= w(y) \qquad (\forall y \in Q_1 \cup Q_2). \end{aligned}$$

We have

$$\begin{aligned} w_F(x_1) + \omega(Q_1) &= w_F(x_1) + w_H(u_1) + w_H(s_1) \\ &\le \sum\{W(S_j^H) \mid j \in I_1 \cup I_2 \cup I_3\}, \end{aligned}$$

and similarly

$$\begin{aligned} w_F(x_2) + \omega(Q_2) &= w_F(x_2) + w_H(u_2) + w_H(s_2) \\ &\le \sum\{W(S_j^H) \mid j \in I_1 \cup I_2 \cup I_3\}. \end{aligned}$$

In addition,

$$\begin{aligned} \omega(Q_1 \cup Q_2) &= w_H(u_1) + w_H(s_1) + w_H(s_2) \\ &\le \sum\{W(S_j^H) \mid j \in I_1 \cup I_2 \cup I_3\}. \end{aligned}$$

Hence each clique C_F has weight at most $q_F = \sum\{W(S_j^H) \mid j \in I_1 \cup I_2 \cup I_3\}$. Since F is perfect, there exist a family of stable sets $S_1^F, \ldots, S_r^F$ of F and weights $W(S_1^F), \ldots, W(S_r^F)$ such that

$$W(S_1^F) + \cdots + W(S_r^F) \leq q_F$$

and

$$\sum\{W(S_j^F) \mid x \in S_j^F\} = w_F(x) \qquad (\forall x \in F).$$

Since x_1 and x_2 are adjacent, no S_j^F contains both x_1, x_2. The definition of $w_F(x_1)$ implies

$$\sum\{W(S_j^F) \mid x_1 \in S_j^F\} = \sum\{W(S_j^H) \mid j \in I_2\},$$

and similarly for x_2.

Now we build a family of stable sets of G by "merging" the families of stable sets of H and of F defined above. This is done as follows: First we merge the family $\{S_i^H \mid i \in I_1\}$ with the family of those stable sets S_j^F that cover x_2. Note that the total weight is the same for both families, by the definition of $w_F(x_2)$, though the individual weights may be different. Also, each set $(S_j^F \cap (Q_1 \cup Q_2)) \cup (S_i^H \setminus (Q_1 \cup Q_2))$ is a stable set, because the choice of j, i is such that S_j^F covers x_2 and $i \in I_1$.

Merging procedure. Take the heaviest set S of the two families (say, the first family), then take the heaviest set T of the second family, and merge them. That is, make the set $S \cup T \setminus \{u_1, s_1, x_2\}$; remove S and T from their respective families; if the weight α of S is strictly larger than the weight β of T, put a copy of S in the first family with weight $\alpha - \beta$; repeat with the remaining families until they are emptied out. Clearly, at the end of each step of the merging subroutine at least one of the two families has one less element, so the merging procedure produces a finite family of stable sets of G (more precisely, the total number of steps, and thus of merged sets that are created, is at most the total size of the two families).

Likewise, we merge the family $\{S_i^H \mid i \in I_2\}$ with the family of stable sets S_j^F covering x_1. Note that these two families have the same total weight, by the definition of $w_F(x_1)$.

Likewise, we merge the family $\{S_j^H \mid j \in I_3\}$ with the remaining family of stable sets S_j^F (i.e., those stable sets S_j^F that do not cover any of x_1, x_2).

Finally, the three families of stable sets produced by the mergings above, plus the family $\{S_j^H \mid j \in I_0\}$, form a family $S_1, \ldots, S_t$ of stable sets of G with weights $W(S_1), \ldots, W(S_t)$, such that

$$\sum_{S_i \ni v_j} W(S_i) = w(v_j) \qquad (\forall v_j). \tag{3}$$

Since the total weight of $S_1, \ldots, S_t$ is q, these stable sets form a minimum weighted coloring of G, and this certifies that C_G is a maximum weighted clique of G.

This completes the proof of Lemma 5.2 and of Theorem 2.3. □

6. The algorithm. We can now present the algorithm BFCLIQUE, which, given a bull-free Berge graph $G = (V, E)$ with a weight $w(x)$ on each vertex x, finds in polynomial time a maximum weighted clique of G. Along with the description of the algorithm it is convenient to maintain a decomposition tree T_G associated with G.

Step 1. In a first phase, we test whether G has any nontrivial homogeneous set. Determining the homogeneous sets of a graph is a problem that is essentially solved by the theory of *modular decomposition*; see, in particular, [4, 6, 17], stemming from the seminal work of Gallai [9, 16]. This theory is rich and complex, and we outline only the aspects that will be used here. Say that a homogeneous set S is *strong* if every homogeneous set S' satisfies $S' \subseteq S$ or $S \subseteq S'$ or $S' \cap S = \emptyset$. It is known [9, 16] that the strong homogeneous sets form a nested family, and so there are at most $2n$ of them, including V and every singleton $\{v\}$ ($v \in V$). One can associate with every graph G a unique rooted tree M_G defined as follows. Let $X_1, \ldots, X_r$ be the (inclusionwise) maximal strong homogeneous sets of G, and let G' be the graph obtained from G by contracting each X_i into one vertex x_i. The root of M_G is G, and the children of node G in M_G are the graphs $G[X_1], \ldots, G[X_r], G'$. For each $i = 1, \ldots, r$, the subtree of M_G rooted at node $G[X_i]$ is the tree $M_{G[X_i]}$ defined recursively. As for G', it follows from the theory of modular decomposition that G' is a clique, or an edgeless graph, or a prime graph, so G' is a leaf of M_G. This tree is called the *modular decomposition tree* of G and can be computed in time linear in the number of edges of G [4, 6, 17].

To obtain a maximum weighted clique for G, we can follow this tree from the bottom up. Assume that we have a maximum weight clique $Q(X_i)$ for each graph $G[X_i]$. We then assign the weight of $Q(X_i)$ to vertex x_i in G'. We then apply the algorithm BFCLIQUE (step 2) on G' and obtain a clique Q' of maximum weight in G'. From Q' we can obtain a clique Q of G by replacing any x_i that lies in Q' by the vertices of $Q(X_i)$. Then Q is a maximum weighted clique of G. (To see this, take a minimum weighted coloring for each of the $G[X_i]$'s, of weight $\omega(G[X_i])$ since $G[X_i]$ is perfect, and for G', of weight $\omega(G')$ since G' is perfect, and merge them in the obvious way; thus a weighted coloring is obtained for G, whose weight is the weight of Q.)

This first step shows that the computation of a maximum clique for G by BFCLIQUE is reduced to calls of BFCLIQUE on at most n graphs (the leaves of the modular decomposition tree). We represent this situation in the associated tree T_G by saying that if G has a nontrivial homogeneous set, then the children of node G in T_G are the leaves of the modular decomposition tree M_G. In that case we say that node G is a modular node of T_G.

Step 2. We are now dealing with a bull-free Berge graph K that is a clique, or an edgeless graph, or a prime graph. We use the algorithm from [7], whose output is one of the following cases.

2.1. K is weakly triangulated. We use the algorithm due to Hayward, Hoàng, and Maffray [14] to produce a maximum weighted clique of K; it is strongly polynomial and its time complexity is $O(n^4m)$. Since K is not subject to a decomposition, node K is a leaf of T_G.

2.2. K contains an even hole and the algorithm produces a box partition for K, and there is no blocking square with respect to this partition. Lemmas 4.1 and 4.3, Theorem 2.1, and Lemma 2.2 imply that K is transitively orientable. A transitive orientation can be found in linear time using the algorithm in [17]. A maximum weighted clique and a minimum weighted coloring can be found using the algorithm in [15], which is strongly polynomial and whose time complexity is $O(nm)$. Here too, node K is a leaf of T_G.

2.3. K contains an even hole and the algorithm produces a box partition for K, and there is a blocking square. Then Lemmas 4.1 and 4.5, and Theorem 2.3 imply that we can decompose K into the two graphs H and Q as above. The proof of part (III) of Theorem 2.3 describes how to obtain a solution to our problem on K from

a solution of the same problems on each of H, Q. Therefore, H and Q are the two children of node K in the tree T_G. We will say that node K is an H_0-node of T_G.

2.4. K contains no even hole, its complementary graph $\overline{K}$ contains an even hole, and the algorithm produces a box partition for $\overline{K}$. Then K is a leaf of T_G. Lemmas 4.3 and 4.1, Theorem 2.1, and Lemma 2.2 imply that $\overline{K}$ is transitively orientable. A transitive orientation of $\overline{K}$ can be found rapidly using the algorithm in [17]. Finding a maximum weighted clique and a minimum weighted coloring for K is equivalent to finding a maximum weighted stable set and a minimum weighted clique covering for the transitively orientable graph $\overline{K}$; this problem can be solved in strongly polynomial time by the algorithm described in [2]. Here too, node K is a leaf of T_G.

7. Complexity analysis. As noted several times, each step of the algorithm can be done in polynomial time. So, in order to prove polynomiality of the whole algorithm, we need only establish the following lemma.

LEMMA 7.1. *There is a polynomial number of nodes in the tree T_G.*

Proof. Let n and m be the number of vertices and edges in G. There are two types of nonleaf nodes in T_G: modular nodes and H_0-nodes. Let $\beta(G), \beta_1(G), \beta_0(G)$ be, respectively, the number of nodes, of modular nodes, and of H_0-nodes in T_G. Note that each node of T_G has no more vertices than its parent. (The "H" child of an H_0-node K may have exactly as many vertices as K; this happens if its sibling the "Q" child of K has exactly four vertices.) Thus every node of T_G has at most n vertices. So each node has at most n children, and $\beta(G) \leq n(\beta_0(G) + \beta_1(G))$. The principle of modular decomposition implies that the parent of a modular node is an H_0-node. Since each H_0-node has exactly two children, we see that $\beta_1(G) \leq 2\beta_0(G) + 1$. Therefore, $\beta(G) \leq n(3\beta_0(G) + 1)$, and we need only prove that $\beta_0(G)$ is a polynomial of n. Our counting argument now focuses on the subgraphs (of nodes of T_G) that induce a $2K_2$, i.e., a graph on four vertices with two nonincident edges. We want to see how the number of $2K_2$'s evolves along T_G.

First, suppose that K is a modular node of T_G. Then it is a routine matter to check that the total number of $2K_2$'s that are induced in its children in M_G is not larger than the number of $2K_2$'s induced in K, and thus that the same holds with respect to the children of K in T_G.

Second, suppose that K is an H_0-node of G, decomposed along an interesting homogeneous pair Q_1, Q_2 (with the notation T_1, T_2, T, Z as usual), and call H, Q the two children of K in the tree. Let us prove that *in total H and Q have strictly fewer $2K_2$'s than K*. For this purpose let us define a one-to-one mapping f that maps every subgraph D that induces a $2K_2$ in H or Q to a subgraph $f(D)$ that induces a $2K_2$ in K. If D is in Q, then set $f(D) = D$. If D is in H, we observe that D does not have an edge with an endvertex in $\{u_1, s_1\}$ and the other in $\{u_2, s_2\}$, for otherwise the remaining two vertices of D should be in Z, which is a contradiction to (ii) of Theorem 4.6. Therefore, if D is in H, we let $f(D)$ be the $2K_2$ of K obtained from D by replacing u_1, s_1, s_2, u_2 (whichever appear in D), respectively, by a, b, c, d. It is a routine matter to check that f is indeed a one-to-one mapping. Moreover, the $2K_2$ of G induced by b, c, e, h is not the image $f(D)$ of any $2K_2$ D of H or Q. This ensures that H and Q have in total strictly fewer $2K_2$'s than K.

Now let T'_G be the tree obtained from T_G by contracting each node that is not an H_0-node with its parent. The number of nodes of T'_G is $\beta_0(G)$ (if G is an H_0-node) or $\beta_0(G) + 1$ (if G is a modular node). The preceding two paragraphs imply that the total number of $2K_2$'s at a given level of T'_G decreases strictly as the level is farther from the root (viewed as level 0), with the only possible exception of the first level

if G is an H_0-node. This implies that the number of nodes in T'_G is bounded by the number of $2K_2$'s in G plus 1, which is $O(m^2)$. □

With this final claim, we obtain that the total number of recursive calls to the algorithm is at most $O(m^2)$. It follows that the algorithm is strongly polynomial, with worst-case complexity $O(n^5m^3)$.

Let us conclude with a remark. The proof of Lemma 5.2 shows how a minimum weighted coloring can be found directly from a minimum weighted coloring of H and of the graph F defined in Lemma 5.2. This method could be the basis for a coloring algorithm that does not involve n^4 calls to the maximum weighted clique algorithm as mentioned in the introduction. However, this method leads to a decomposition algorithm in which a node K that contains an H_0 must be decomposed into three graphs H, Q, F. In that case, note that the vertices of Q also appear in F, and so we cannot guarantee that the total number of nodes of the decomposition tree remains polynomial in the size of the root graph.

Acknowledgment. We thank the referees for their careful reading and valuable suggestions, which helped improve the presentation of this paper.

REFERENCES

[1] C. Berge, *Les problèmes de coloration en théorie des graphes*, Publ. Inst. Statist. Univ. Paris, 9 (1960), pp. 123–160.
[2] K. Cameron, *Antichain sequences*, Order, 2 (1985), pp. 249–255.
[3] V. Chvátal and N. Sbihi, *Bull-free Berge graphs are perfect*, Graphs Combin., 3 (1987), pp. 127–139.
[4] A. Cournier and M. Habib, *A new linear algorithm for modular decomposition*, in Trees in Algebra and Programming—CAAP 1994 (Edinburgh), Lecture Notes in Comput. Sci. 787, Springer-Verlag, Berlin, 1994, pp. 68–84.
[5] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas, *The Strong Perfect Graph Theorem*, manuscript, Princeton University, Princeton, NJ, 2002.
[6] E. Dahlhaus, J. Gustedt, and R. M. McConnell, *Efficient and practical modular decomposition*, in Proceedings of the 8th Annual ACM–SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 1997, pp. 26–35.
[7] C. M. H. de Figueiredo, F. Maffray, and O. Porto, *On the structure of bull-free perfect graphs*, Graphs Combin., 13 (1997), pp. 31–55.
[8] P. Duchet, *Classical perfect graphs*, in Topics on Perfect Graphs, North–Holland Math. Stud. 88, C. Berge and V. Chvátal, eds., North–Holland, Amsterdam, 1984, pp. 67–96.
[9] T. Gallai, *Transitiv orientierbare Graphen*, Acta Math. Acad. Sci. Hung., 18 (1967), pp. 25–66.
[10] A. Ghouila-Houri, *Caractérization des graphes non orientés dont on peut orienter les arêtes de manière à obtenir le graphe d'une relation d'ordre*, C. R. Acad. Sci. Paris, 254 (1962), pp. 1370–1371.
[11] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.
[12] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, New York, 1988.
[13] R. Hayward, *Weakly triangulated graphs*, J. Combin. Theory Ser. B, 39 (1985), pp. 200–209.
[14] R. Hayward, C. T. Hoàng, and F. Maffray, *Optimizing weakly triangulated graphs*, Graphs and Combinatorics, 5 (1989), pp. 339–349. See erratum in 6 (1990), pp. 33–35.
[15] C. T. Hoàng, *Efficient algorithms for minimum weighted colouring of some classes of perfect graphs*, Discrete Appl. Math., 55 (1994), pp. 133–143.
[16] F. Maffray and M. Preissmann, *A translation of Tibor Gallai's article "Transitiv orientierbare Graphen,"* in Perfect Graphs, J. L. Ramírez-Alfonsín and B. A. Reed, eds., John Wiley & Sons, New York, 2001.
[17] R. M. McConnell and J. P. Spinrad, *Linear-time transitive orientation*, in Proceedings of the 8th Annual ACM–SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 1997, pp. 19–25.

[18] B. A. Reed and N. Sbihi, *Recognizing bull-free perfect graphs*, Graphs Combin., 11 (1995), pp. 171–178.
[19] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, Springer-Verlag, New York, 2003.
[20] S. Seinsche, *On a property of the class of n-colorable graphs*, J. Combin. Theory Ser. B, 16 (1974), pp. 191–193.

SIAM J. DISCRETE MATH.
Vol. 18, No. 2, pp. 241–245

PARSIMONIOUS MULTIGRAPHS*

TODD G. WILL† AND HEATHER HULETT†

Abstract. We view a loopless multigraph as a complete graph with nonnegative integer edge weights indicating the multiplicity of each edge. A multigraph realization of a given degree sequence is *parsimonious* if it has the least number of positive weight edges. We characterize the graphs that can appear as components in a parsimonious multigraph and show that minimizing the number of positive weight edges is equivalent to maximizing the number of cycle-free components.

Key words. parsimonious, multigraph, degree sequence

AMS subject classification. 05C12

DOI. 10.1137/S0895480103432477

1. Introduction.

A sequence of integers $d = (d_1, \ldots, d_n)$ is *graphic* if it is the degree sequence of some loopless multigraph G, in which case we say that G is a *realization* of d. We view a loopless multigraph as a complete graph with nonnegative integer edge weights indicating the multiplicity of each edge. The *support graph* for a multigraph is the simple graph consisting of the edges with positive weight. We call a multigraph realization of a degree sequence *parsimonious* if no other realization has fewer edges in its support graph. And we call a simple graph a *parsimonious support graph* if it appears as the support graph of some parsimonious multigraph. Parsimonious multigraphs have also been studied in [1] and [7] with the restriction that edges have a maximum multiplicity of 2.

To construct a parsimonious multigraph one attempts to minimize the number of positive weight edges. The opposite extreme was studied by Kleitman [5] and others. They provided efficient algorithms for producing multigraph realizations that minimize the number of multiple edges, where an edge of weight $w > 1$ is counted as $w - 1$ multiple edges. Since a multigraph with degree sum D and m multiple edges has $\frac{D}{2} - m$ positive weight edges, they were in effect producing realizations with a maximum number of positive weight edges. Multigraph realizations subject to other constraints have been studied in [2], [4], [6].

A potential application of parsimonious multigraphs arises in network design. Suppose a length n degree sequence represents the desired number of connections per node in an n node network. Suppose further that establishing the first connection between a pair of nodes comes at unit cost, but subsequent connections between the same pair come with no cost. For example, digging a trench between two nodes is expensive, but once the trench is open the number of wires laid in the trench makes little difference. In such a context a parsimonious realization yields the cheapest network.

In this paper we show that finding a support graph of minimum size is equivalent to finding a support graph with the maximum number of cycle-free components. We then characterize a component of a parsimonious support graph as either a tree or a tree plus one edge where the unique cycle formed has odd length.

*Received by the editors July 29, 2003; accepted for publication (in revised form) March 29, 2004; published electronically October 1, 2004.
http://www.siam.org/journals/sidma/18-2/43247.html
†Mathematics Department, University of Wisconsin-La Crosse, La Crosse, WI 54601 (will.todd@uwlax.edu, hulett.heat@uwlax.edu).

2. Necessary conditions. We begin by describing a method of producing a new graph with the same degree sequence as a given graph. Let $T = (v_1, \ldots, v_{2n})$ be an even length closed walk in a multigraph G. (We allow edges of weight zero in the walk.) We call edges of the form (v_{2k}, v_{2k+1}) *even* edges and edges of the form (v_{2k-1}, v_{2k}) *odd* edges. Let m be the minimum weight of any even edge in T. If none of the even edges of T is repeated, we perform a *T-interchange* on G by subtracting m from the weight of the even edges and adding m to the weight of the odd edges. If an odd edge is used k times, then a total of km is added to the weight of this edge. After showing that a T-interchange preserves the degree sequence, it will be our primary tool for manipulating realizations of a given degree sequence.

LEMMA 2.1 (interchange lemma). *If T is an even length closed walk in a multigraph G which repeats no even edge, then a T-interchange on G produces a multigraph G' with the same degree sequence as G.*

Proof. Let $T = (v_1, \ldots, v_{2n})$ be an even length closed walk which repeats no even edge in a multigraph G. Since the T-interchange subtracts the minimum weight of any edge, all of the edge weights remain nonnegative. Since any vertex v on T is incident to an equal number of odd and even edges on T, we conclude that G and G' have the same vertex degrees. ∎

LEMMA 2.2. *A parsimonious support graph contains no even length closed trails.*

Proof. Suppose that the support graph of the multigraph G contains an even length closed trail T. By definition of a support graph all of the edge weights in T are positive. A T-interchange on G reduces at least one of these weights to zero, showing that G is not parsimonious. ∎

The following lemma guarantees that each component of a parsimonious support graph is either a tree or a tree plus one edge.

LEMMA 2.3. *A parsimonious support graph has at most one cycle in each component.*

Proof. Suppose that the support graph of a parsimonious multigraph G has two distinct cycles C_1 and C_2 in the same component. By Lemma 2.2, C_1 and C_2 have odd length.

Case 1. Suppose the cycles are not vertex disjoint. Let e be an edge on C_1 but not on C_2. Starting on e, follow cycle C_1 in each direction until reaching vertices u and v on C_2, with u possibly equal to v. If $u = v$, then C_1 and C_2 can be combined at vertex u to give an even length closed trail, contradicting Lemma 2.2. If $u \neq v$, then u and v split C_2 into two paths P_1 and P_2 of opposite parity, one of which can be combined with the uv portion of cycle C_1 containing e to form an even length closed trail, contradicting Lemma 2.2.

Case 2. Suppose the cycles are vertex disjoint. Let P be a shortest (u, v) path from C_1 to C_2. Label $C_1 = (u_1, \ldots, u_{2j+1})$ with $u_1 = u$ and label $C_2 = (v_1, \ldots, v_{2k+1})$ with $v_1 = v$.

Case 2(a). Suppose P has length 1. Applying the interchange lemma (Lemma 2.1) to the closed walk $T = (u_1, v_1, \ldots, v_{2k+1}, v_1, u_1, \ldots, u_{2j+1})$ yields a multigraph with the same degree sequence but fewer edges in its support graph, showing that G is not parsimonious.

Case 2(b). Suppose the path $P = (u_1, w_1, \ldots, w_{2r}, v_1)$ has odd length $2r + 1 > 1$. Apply the interchange lemma to the even length closed walk $(u_1, w_1, \ldots, w_{2r}, v_1)$. Since P is a shortest path, the edge v_1u_1 has weight zero. Consequently the T-interchange adds v_1u_1 to the support graph, producing a realization which falls under Case 2(a).

Case 2(c). Suppose the path $P = (u_1, w_1, \ldots, w_{2r+1}, v_1)$ has even length. Apply the interchange lemma to the even length closed walk $(u_1, w_1, \ldots, w_{2r+1}, v_1, v_2)$. Again using the fact that P has minimum length, the interchange adds the edge v_2u_1 to the support graph. Since G is assumed to be parsimonious, exactly one other edge of the walk must be removed from the support graph. If (v_1, v_2) is removed, then the interchange produces a new cycle $(v_2, \ldots, v_{2k+1}, v_1, w_{2r+1}, \ldots, w_1, u_1)$ incident to C_1, contradicting Case 1. If it is an edge from P that is removed, then again we have a realization which falls under Case 2(a). □

Lemma 2.3 can now be used to show that minimizing the number of positive weight edges is equivalent to maximizing the number of cycle-free components. We state this result in the following theorem and leave the proof to the reader.

THEOREM 2.4. *A parsimonious support graph on n vertices with k cycle-free components has $n - k$ edges.*

3. Sufficient conditions. In the previous section we showed that each component of a parsimonious support graph is either a tree or a tree plus one edge where the unique cycle has odd length. In this section we show that these necessary conditions on the components of a parsimonious support graph are also sufficient. We begin with a simple requirement for a realization to have multiple components.

LEMMA 3.1. *The sequence $d = (d_1, \ldots, d_n)$, where each d_i is even, has a realization with more than one component if and only if there exists a partition of $[n]$ into sets A and B such that $\sum_{i\in A} d_i - 2\max_{i\in A} d_i \geq 0$ and $\sum_{i\in B} d_i - 2\max_{i\in B} d_i \geq 0$.*

Proof. Since each degree is even, all subsets have even sums. So the result follows from the well-known property that a sequence is realizable by a loopless multigraph if and only if the sum is even and the maximum degree is no greater than the sum of the remaining degrees. □

The next lemma shows when a degree sequence has a cycle-free realization.

LEMMA 3.2. *The sequence $d = (d_1, \ldots, d_n)$ has a cycle-free realization if and only if there exists a partition of $[n]$ into sets A and B such that $\sum_{i\in A} d_i = \sum_{i\in B} d_i$.*

Proof. A cycle-free realization of the sequence is a bipartite graph, the two partite sets of which have equal degree sums. In the other direction, given the sets A and B with equal sums, it is easy to form a bipartite realization. There can be no odd length cycles in the bipartite graph, and any even length cycles can be removed by performing T-interchanges. □

In the following three lemmas, we create degree sequences for which the given graph will be parsimonious. First, we construct edge weights to show that any tree can appear in a parsimonious multigraph.

LEMMA 3.3. *Every nontrivial tree is a parsimonious support graph.*

Proof. We prove by induction on the number of vertices n that every tree can be assigned edge weights so that the resulting degree sequence has all even degrees and is not multicomponent realizable. Theorem 2.4 then implies that the tree is parsimonious. For $n = 2$, give the single edge weight 2. Given a tree T on $n > 2$ vertices, let uv be a pendant edge with leaf v. Form T' by deleting v, and apply induction to obtain edge weights for T' and a resulting degree sequence $d' = (d'_1, \ldots, d'_{n-1})$ of even degrees which cannot be realized by more than one component. Without loss of generality, assume d'_{n-1} is the degree of vertex u in T', and for convenience define $d'_n = 0$. In T assign the edge uv weight 2, and assign all other edges three times their weight in T'. This gives T the degree sequence $d = (d_1, \ldots, d_n) = (3d'_1, \ldots, 3d'_{n-2}, 2 + 3d'_{n-1}, 2 + 3d'_n)$. It remains to show that d is not multicomponent realizable.

Let A and B be a partition of $[n]$. Since T' is not multicomponent realizable, we may assume without loss of generality that $\sum_{i\in A} d'_i - 2\max_{i\in A} d'_i \le -2$. Then $\sum_{i\in A} 3d'_i - 2\max_{i\in A} 3d'_i \le -6$, which implies $\sum_{i\in A} d_i - 2\max_{i\in A} d_i \le -2$. Thus T is not multicomponent realizable. ∎

Next, we construct edge weights to show that any odd length cycle can appear in a parsimonious graph.

LEMMA 3.4. *Every odd length cycle is a parsimonious support graph.*

Proof. For $k \ge 1$, let $C = (v_1, \dots, v_{2k+1})$ be an odd length cycle. For $i = 1, \dots, 2k$ assign to edge $v_i v_{i+1}$ weight 2^i, and assign to edge $v_{2k+1}v_1$ weight 2^{2k+1}. These edge weights produce the degree sequence $d = (3\cdot 2^{2k}, 2 + 2^{2k+1}, 3\cdot 2^{2k-1}, \dots, 3\cdot 2^2, 3\cdot 2^1)$. We prove that C is parsimonious by showing that d cannot be realized by a tree or by a graph with more than one component.

If d could be realized by a tree, then d could be realized by a bipartite graph. This would imply that the sequence d could be partitioned into two sets with equal sums. However, the fact that $2 + 2^{2k+1}$ is the only number in the sequence which is not divisible by 3 precludes this possibility.

Note that $3\cdot 2^1 + 3\cdot 2^2 + \cdots + 3\cdot 2^{i-1} = 3\cdot 2^i - 6$. Since the largest degree in a component can be at most the sum of the remaining degrees, no degree of the form $3\cdot 2^i$, with $i \le 2k-1$, can be the largest degree in a component. Moreover, $2 + 2^{2k+1}$ is required to be in the same component as $3\cdot 2^{2k}$, and hence, any realization of d can have only one component. ∎

Finally, we construct edge weights to show that any tree with an extra edge forming an odd cycle can appear in a parsimonious support graph.

LEMMA 3.5. *If G is a tree plus one edge forming an odd length cycle, then G is a parsimonious graph.*

Proof. We prove by induction on the number of edges k not belonging to the cycle that edge weights can be assigned so that the resulting degree sequence has all even degrees, is not multicomponent realizable, and is not cycle-free realizable. Theorem 2.4 then implies that the graph is parsimonious. For $k = 0$, Lemma 3.4 applies. Given a graph G with $k > 0$ edges not on the cycle, let uv be a pendant edge with leaf v. Form G' by deleting v, and apply induction to obtain edge weights for G' and a resulting degree sequence $d' = (d'_1, \dots, d'_{n-1})$ which cannot be realized by more than one component and cannot be realized as a tree. Without loss of generality, assume d'_{n-1} is the degree of vertex u in G', and for convenience define $d'_n = 0$. In G assign the edge uv weight 2 and assign all other edges three times their weight in G'. This gives G the degree sequence $d = (d_1, \dots, d_n) = (3d'_1, \dots, 3d'_{n-2}, 2 + 3d'_{n-1}, 2 + 3d'_n)$. The argument in the proof of Lemma 3.3 again shows that d is not multicomponent realizable. It remains to show that d is not cycle-free realizable.

Let A and B be a partition of $[n]$. Since G' is not cycle-free realizable, Lemma 3.2 implies $|\sum_{i\in A} d'_i - \sum_{i\in B} d'_i| \ge 2$. Then $|\sum_{i\in A} 3d'_i - \sum_{i\in B} 3d'_i| \ge 6$, which implies $|\sum_{i\in A} d_i - \sum_{i\in B} d_i| \ge 2$ and shows again by Lemma 3.2 that G is not cycle-free realizable. ∎

Combining the necessary and sufficient conditions gives the final theorem.

THEOREM 3.6. *A simple graph T can be a component of a parsimonious support graph if and only if T is a tree or T is a tree plus one edge where the unique cycle formed has odd length.*

4. Complexity. As mentioned in the introduction, Kleitman and others found efficient algorithms for maximizing the number of edges in a support graph. We conjecture that there is not an efficient algorithm for minimizing the number of edges

in the support graph.

Our conjecture is based on a connection to the NP-complete problem of NUMERICAL THREE DIMENSIONAL MATCHING (N3DM) [3]. One variant of this problem asks if a collection of $3k$ positive integers can be split into k triples such that in each triple one of the numbers is equal to the sum of the other two.

Let DISTINCT NUMERICAL THREE DIMENSIONAL MATCHING (DN3DM) be the restriction of N3DM to the case where all integers are distinct. If we view the $3k$ integers as a degree sequence d, it is relatively easy to see that there is a successful matching into triples if and only if the support graph of the parsimonious multigraph for d consists of k disjoint length 2 paths.

Of course it is possible that DN3DM is no longer NP-complete. However, intuitively it seems that repeated numbers should make matching easier, and therefore restricting N3DM to the case where all the numbers are distinct would be a restriction to the "hardest case" of N3DM.

REFERENCES

[1] R. A. Brualdi and T. S. Michael, *The class of* 2*-multigraphs with a prescribed degree sequence*, Linear and Multilinear Algebra, 24 (1989), pp. 81–102.

[2] R. B. Eggleton and D. A. Holton, *Simple and multigraphic realizations of degree sequences*, in Combinatorial Mathematics, VIII (Geelong, 1980), Lecture Notes in Math. 884, Springer-Verlag, Berlin, New York, 1981, pp. 155–172.

[3] M. R. Garey and D. S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness*, A Series of Books in the Mathematical Sciences, W. H. Freeman and Co., San Francisco, CA, 1979.

[4] A. J. Goldman and R. H. Byrd, *Minimum-loop realization of degree sequences*, J. Res. Nat. Bur. Standards, 87 (1982), pp. 75–78.

[5] D. J. Kleitman, *Minimal number of multiple edges in realization of an incidence sequence without loops*, SIAM J. Appl. Math., 18 (1970), pp. 25–28.

[6] Z. Majcher, *Alternating cycles and realizations of a degree sequence*, Comment. Math. Univ. Carolin., 28 (1987), pp. 467–480.

[7] T. G. Will and D. B. West, *Parsimonious* 2*-multigraphs*, in Graph Theory, Combinatorics, and Algorithms, Vol. 1, 2 (Kalamazoo, MI, 1992), Wiley-Interscience, Wiley, New York, 1995, pp. 1249–1258.

SIAM J. DISCRETE MATH.
Vol. 18, No. 2, pp. 246–271

CONSTRAINING PLANE CONFIGURATIONS IN CAD: CIRCLES, LINES, AND ANGLES IN THE PLANE*

FRANCO SALIOLA† AND WALTER WHITELEY‡

Abstract. This paper investigates the local uniqueness of designs of m-circles (lines and circles) in the plane up to inversion under a set of angles of intersection as constraints. This local behavior is studied through the Jacobian of the angle measurements in a form analogous to the rigidity matrix for a framework of points with distance constraints. After showing directly that the complete set of angle constraints on v distinct m-circles gives a matrix of rank $3v - 6$, we show that the Jacobian is column equivalent by a geometric correspondence to the rigidity matrix for a bar-and-joint framework in Euclidean 3-space. As a corollary, the complexity of the independence of angle constraints on generic plane circles is the complexity of the old unsolved combinatorial problem of generic rigidity in 3-space. This theory is not known to have a polynomial time algorithm for generic independence that offers a warning about the complexity of general systems of geometric constraints even in the plane.

Our correspondence extends to all dimensions. Angle constraints on spheres in 3-space then match the even more complex first-order theory of frameworks in 4-space. This theory is not predicted to have a polynomial time algorithm for generic points.

Key words. computer aided design, constraint, inversive geometry, circles and angles, generic rigidity, hyperbolic geometry

AMS subject classifications. Primary, 68U07; Secondary, 05C50, 51N05, 52C25

DOI. 10.1137/S0895480100374138

1. Introduction.

A standard problem in computer aided design (CAD) is to find stable combinatorial techniques to decide when a set of constraints is independent, and solvable algebraically, in a reasonable time (or real time in parametric CAD programming) [16, 10]. In the study of combinatorial algorithms for independence, the classical example is points and distance in a plane or, equivalently, generic rigidity of plane frameworks. Given a graph, there is a fast combinatorial algorithm to decide the independence of the edges, as distance constraints, for almost all (generic) placements of the vertices as distinct points in the plane [15, 7, 24]. At the other extreme, there are sample problems, including points and distances in 3-space, for which there is no known polynomial time combinatorial algorithm to decide independence [7, 25, 32].

It is of interest to consider other geometric objects beyond points and other geometric constraints beyond distances (e.g., angles) and to discover the combinatorics and the geometry of their independence structure. In this paper, we consider circles of variable radii in the plane, with the angle of intersection of pairs of circles as the constraint. The case of mutually tangent circles, or even lines tangent to circles, are special cases of this problem, as outlined below. This problem of circles of variable radii with angles of intersection (including tangency) is a real problem in practical CAD programming [17, 12, 13]. We describe some new results and some important

*Received by the editors June 20, 2000; accepted for publication (in revised form) February 26, 2003; published electronically October 8, 2004.

http://www.siam.org/journals/sidma/18-2/37413.html

†Department of Mathematics, 310 Malott Hall, Cornell University, Ithaca, NY 14853-4201 (saliola@math.cornell.edu). This author's work was supported in part by a summer research award from NSERC (Canada), held at York University.

‡Department of Mathematics and Statistics, York University, 4700 Keele Street, North York, ON M3J 1P3, Canada (whiteley@mathstat.yorku.ca). This author's work was supported by a grant from NSERC (Canada).

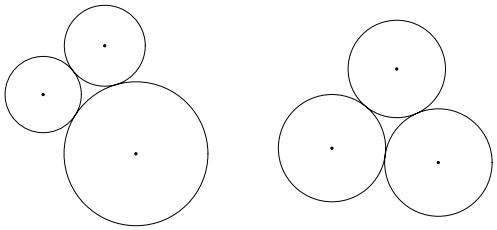


FIG. 1.1. *Are three mutually tangent circles unique up to inversion?*

connections, which we hope will offer insights for CAD programmers working on this problem, embedded in general systems of constraints.

Of course, answering this combinatorial question is only one step in the process. Once we have a combinatorial algorithm, there is still the problem of determining algebraic methods for solving the constraints for maximal independent sets of constraints and the problem of "special position" in which the constraints that are typically independent become dependent, such as the lengths of the third edge of a collinear triangle for distance constraints. We will not address either of these hard problems in detail in this paper.

Consider the two sets of mutually tangent circles in Figure 1.1. The following question arises: Is there an angle preserving map of the plane (a series of inversions) that carries the three circles on the left onto the three circles on the right preserving tangency? The question can be broadened. Suppose that the constraints on circles are not limited to tangency but include a fixed angle for each designated pair of circles. A similar question can be posed for any design: a set of circles and lines under angle constraints at their intersections.

This problem of uniqueness, which is typical of underlying problems which arise in CAD, is hard once we move to larger sets [14, 16, 17]. As an important analogy, recall that for points in the plane, constrained by distances, the general problem of "global" rigidity (uniqueness up to congruence) is unsolved, and we have no general algorithm which applies to all designs [5, 8].

A simpler problem, which is required for parametric CAD programming, is to test "local uniqueness" of the design—a property that can be predicted using the linear algebra of the Jacobian of the constraints for almost all configurations [4, 16, 23, 32]. The rank of this Jacobian matrix for the constraints, often called the rigidity matrix (for distances) or the constraint matrix (for more general constraints), is determined through two layers of analysis:

1. the combinatorial level, which determines the maximum rank by the combinatorics of which constraints are selected (in our case, a graph with vertices for the circles and an edge for each angle constraint);
2. the geometric level, which describes when the rank of the matrix drops below the maximum in (1).

Whenever the rank is maximal, the linear algebra of the Jacobian predicts whether the design is locally unique or has a continuous path of configurations satisfying the same constraints but which is geometrically inequivalent (section 4.3). Since this maximal rank occurs for "almost all" choices of the objects, this combinatorial theory gives a strong prediction of what will happen in a CAD problem for "generic" circumstances. Moreover, even if a lower rank configuration remained locally unique, the

usual numerical algorithms will become ill-determined in these nongeneric situations. In short, the rank of the Jacobian is a critical factor in any analysis of a design.

For angle constraints among circles and lines, the appropriate geometry is inversive geometry. Under inversions, the set of circles and lines goes to the set of circles and lines, and all angles are preserved. Thus, at best, angle constraints can make a design unique, or locally unique, up to inversion. Because inversive geometry may not be familiar to many readers in CAD, in section 2 we provide a brief introduction to inversions in the plane, including both synthetic and analytic representations. This is followed by an introduction to a representation of the circles as points in $\mathbb{R}^3$ (section 3).

We then present the Jacobian of the system of angle constraints, proving a basic theorem on the minimum number of constraints which makes a design unique up to local inversion: $|E| = 3|V| - 6$ (section 4).

Within this initial analysis, we find a striking similarity to the theory of rigid bar-and-joint frameworks in Euclidean 3-space with the same graph [4, 25, 30, 32]. After a brief recollection of the first-order theory of such frameworks (section 5), we present an unexpected isomorphism between the Jacobian for these three-dimensional frameworks and the constraint matrix for circles and lines with angle constraints in the plane (section 6). This permits the complete transfer of known results as well as longstanding conjectures from the first-order theory of frameworks of points and distances into this first-order theory for circles, lines, and angles (section 7). This complete transfer raises the likelihood that there is no polynomial time algorithm for determining the independence of generic patterns of angle constraints on circles in the plane, in contrast to the desirable model of a $O(|E|^2)$ algorithm offered by points and distances in the plane.

These results extend to all dimensions, giving, for example, a transfer between the first-order theory of spheres and planes in 3-space, constrained by angles, and the first-order theory of bar-and-joint frameworks in Euclidean 4-space (section 8).

To this point, the theory assumes circles with variable radii and only angle constraints. In section 9 we introduce additional constraints to fix some or all of the radii of the circles. An extension of the transfer to 3-space confirms that, with all radii fixed, the theory simplifies to the theory of the centers of the circles with distance constraints replacing the angle constraints. This special case does have fast algorithms. However, when we mix fixed and variable radii (with more than two variable circles), the theory returns to at least the complexity of spatial rigidity.

Beneath this first-order isomorphism with Euclidean frameworks lies a stronger isomorphism between the inversive theory of m-circles constrained by angles in the plane and the congruence theory of planes in hyperbolic 3-space constrained by (hyperbolic) angles—or, dually, the theory of ideal points in hyperbolic space, constrained by hyperbolic distance. This correspondence applies beyond the first order to all levels of rigidity and flexibility and is presented in [21]. These isomorphic theories then share a common first-order isomorphism with Euclidean and spherical frameworks, which is explored in detail in [21]. While some aspects of this correspondence of plane circles and hyperbolic geometry are well known, there remain new insights to be extracted. It may be a surprise to people in CAD programming that simple plane and 3-space CAD contains, as a subset, the theory of planes and hyperplanes with angle constraints in hyperbolic 3- and 4-space.

This study continues a series of papers which investigate constraints among points, lines, and circles in the plane [2, 21, 22, 33]. One central problem is appropriate algorithms for "generic" behavior of configurations of objects and constraints. The

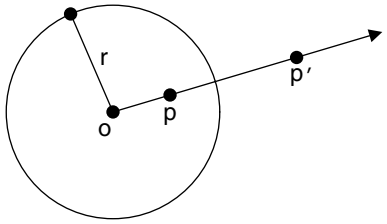


FIG. 2.1. *Inversion in a circle:* $\vec{op}\cdot\vec{op}' = r^2$.

results in this paper emphasize that there are plane problems that, unlike simple distance constraints in the plane, have a complexity that may not be polynomial. We offer further observations and unsolved problems in section 10.

2. Circles in the inversive plane. While people familiar with hyperbolic geometry and inversive geometry will be familiar with all the material in this section, we feel it is important to make this paper more self-contained for workers in CAD programming.

2.1. The plane model. Plane inversive geometry is the underlying geometry in this investigation, so this section will recall the definition and some useful results. Our two primary sources for these sections are [18, 34]. While this paper will focus on the plane, we will indicate how the results generalize directly to n dimensions $n \geq 1$ in section 8. The interested reader can find a nice treatment of inversive geometry in n-dimensional space in [34].

Let C be a circle centered at an arbitrary point o with radius r and p a point different from o. Then to the point p associate the point p' on the ray $\vec{op}$ satisfying $\vec{op}\cdot\vec{op}' = r^2$ (Figure 2.1). One easily finds that this association is a bijection of the plane onto itself, except for the point o. To complete the bijection, we adjoin exactly one point to the plane, the *point at infinity*, denoted by o', and pair it with o under this correspondence. The plane augmented with the point at infinity is the *inversive plane*, denoted Π.

DEFINITION 2.1. *Let C be a circle centered at a point o with radius r, o' the point at infinity, and p a point in the inversive plane. Then the* inverse of p in C *is*

1. *o' if $p = o$,*
2. *o if $p = o'$,*
3. *the point p' on the ray $\vec{op}$ satisfying $\vec{op}\cdot\vec{op}' = r^2$ otherwise.*

The point o is called the center of inversion *and C the* circle of inversion.

Note the points of the circle C are invariant under inversion in C.

2.2. The sphere model. This section describes the sphere model of the inversive plane. We include the sphere model because it sometimes provides a more convenient image for results about inversive geometry.

DEFINITION 2.2. *Let Σ be a sphere, p a point on Σ, and C a circle on Σ. Let t_C be the tip of the cone tangent to Σ at C. Then the* inverse of p in C *is the second point of intersection of the line through p and t_C with Σ (Figure 2.2).*

If the circle C in the above definition is a great circle (the intersection of a plane through the center of Σ with Σ), then t_C is a "point at infinity" in 3-space, and all lines through t_C and a point on Σ are parallel to the normal of the great circle. That

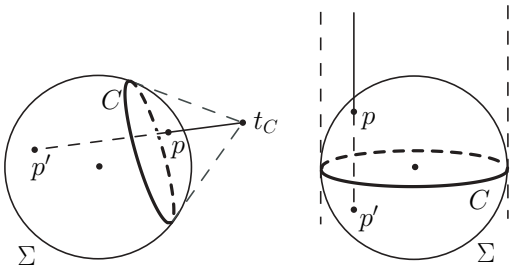


FIG. 2.2. *Inversion in a circle on the sphere model.*

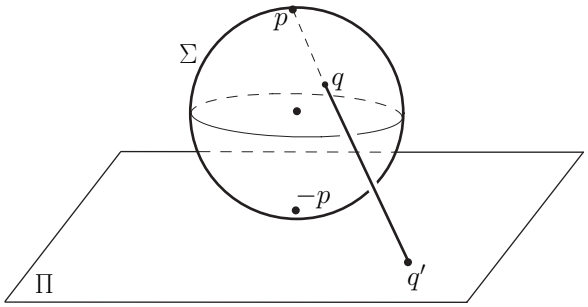


FIG. 2.3. *Stereographic projection from the sphere to the inversive plane.*

is, the inverse of a point p in a great circle is obtained by reflecting p in the great circle. Notice that in this spherical model the bijection of the sphere onto itself is complete; there is no need to augment the sphere with an additional point.

2.3. Connecting the plane model and the sphere model. The plane and sphere models of the inversive plane are connected through *stereographic projection*. Let p be a point on the sphere Σ and $-p$ the antipodal point of p. Let Π be a plane tangent to Σ at $-p$. Then we have a bijection between Σ and Π: $q \in \Sigma$ corresponds to $q' \in \Pi$, where q' is the point of intersection of the line pq with Π, and p corresponds to the point at infinity of Π (Figure 2.3).

Hence, the plane model Π of the inversive plane is obtained from the sphere Σ model by stereographic projection of the sphere from a point p onto a plane tangent to the sphere at the $-p$. Similarly, the sphere model is obtained from the plane model by *lifting* the plane onto the sphere (the inverse mapping of stereographic projection). Note that the point p of stereographic projection corresponds to the point at infinity in the plane model. Also note that circles through the projection point go to lines in the plane. Figure 2.4 illustrates that stereographic projection preserves angles of intersection of circles.

Notice that inversion in a great circle through the north and south poles is a reflection in the vertical plane of this great circle. This corresponds to an "inversion" in the projected line in the plane—now seen as a reflection in the line. Since we had not previously defined inversion in a line, we define this inversion as the reflection. Since all isometries of the plane are products of reflections, all isometries of the plane can be viewed as products of inversions.

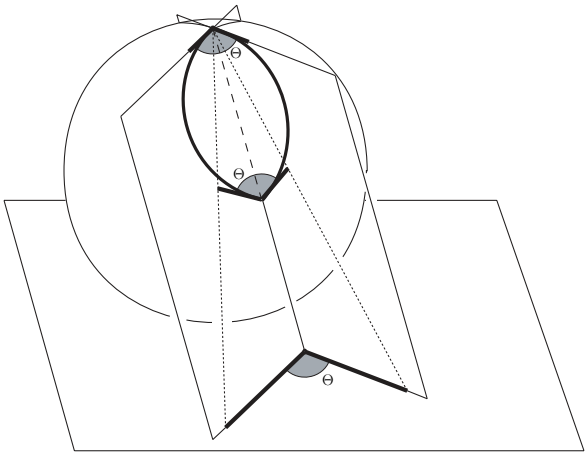


FIG. 2.4. *Stereographic projection preserves angles. The angles on the sphere are equal by symmetry. The vectors at the north pole are in a plane parallel to the inversive plane, so a translation gives the equality with the angle in the plane.*

2.4. Properties of inversions. The choice of the size of the sphere and the point of tangency of the sphere with the plane in the stereographic projection from the previous section is arbitrary. However, there is one such choice that makes many important facts about inversions visually obvious. Suppose C is the circle of inversion in the inversive plane Π centered at o with radius r. Take a sphere Σ of diameter r tangent to Π at o. Lifting Π onto Σ maps the circle of inversion onto the "equator" of the sphere—a great circle. Also, the point at infinity in the inversive plane maps onto the point of projection p on the sphere. The inversion in the sphere model is merely a reflection in this equatorial great circle. Stereographic projection of the reflected sphere back onto a plane through the great circle yields the image of the plane Π under the inversion.

This approach to inversion lends itself well to proving properties of inversions in the plane.

PROPOSITION 2.3. *Properties of inversions.*

(1) *The product of two inversions in the same circle is the identity mapping.*
(2) *Inversion preserves angles.*
(3) *The circle of inversion is invariant under inversion.*
(4) *Lines through the center of inversion are invariant under inversion.*
(5) *Circles orthogonal to the circle of inversion are invariant under inversion.*
(6) *The inverse of a circle through the center of inversion is a line not through the center of inversion.*
(7) *The inverse of a line not through the center of inversion is a circle through the center of inversion.*
(8) *The inverse of a circle not through the center of inversion is a circle not through the center of inversion.*

Proof. (1) follows from the fact that the product of two reflections of the sphere in the same great circle is the identity mapping.

(2). Since stereographic projection and reflections on the sphere preserve angles, it follows that inversion preserves angles.

(3), (4), (5). The mentioned objects lift onto objects invariant under the reflection in the "equator" of the sphere. Therefore, they are invariant under an inversion.

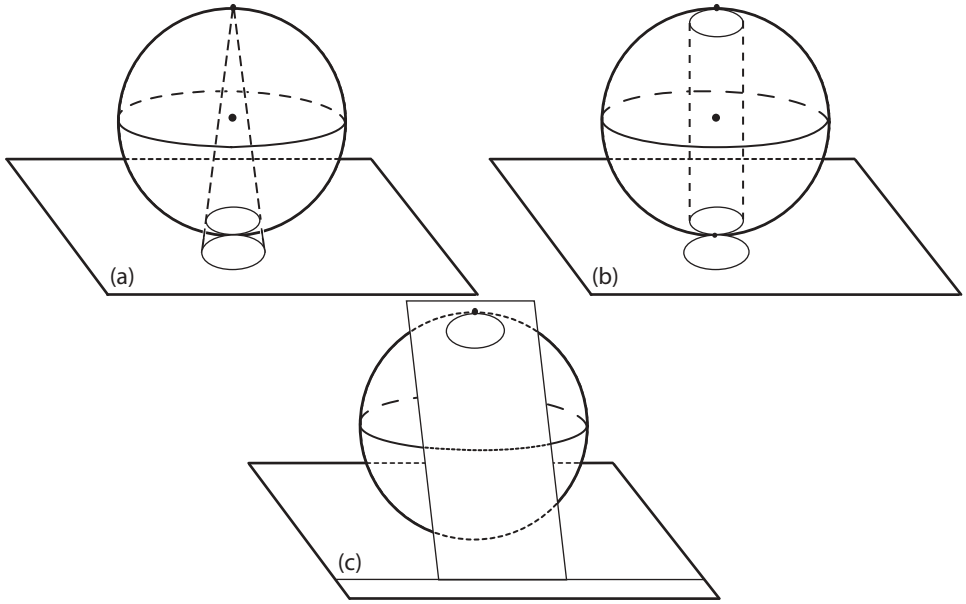


FIG. 2.5. *Proof without words of* (6) *and* (7) *of Proposition* 1: (a) *stereographic lifting of the circle;* (b) *inversion of the lift in the equator;* (c) *stereographic projection of the image.*

(6), (7). See Figure 2.5 for a proof without words of (6) and (7).

(8) follows from (6) and (7) and the fact that stereographic projection carries circles on the sphere onto circles on the plane. □

The following section develops some of these properties of inversions algebraically.

2.5. An algebraic look at inversion. Since we will be working with circles and lines simultaneously, we introduce the following definition.

DEFINITION 2.4. *A* Möbius-Circle *(or* m-circle*) in the plane is a line or circle in the plane.*

Inversion can now be described as a map that carries m-circles to m-circles. For an algebraic representation of m-circles we begin with the equation

$$M \equiv a(x^2 + y^2) - 2bx - 2cy + d = 0,$$

where one of a, b, or c is nonzero ($a^2 + b^2 + c^2 \neq 0$). If $a \neq 0$, then M is a circle; if $a = 0$, then M is a line.

To study the inverse of an m-circle, take the circle of inversion to be the circle centered at the origin of radius k. Then the inverse of the point (x, y) is (x', y'), where

$$x' = k^2 \frac{x}{x^2 + y^2}, \quad y' = k^2 \frac{y}{x^2 + y^2}.$$

The inverse of the m-circle $ax^2 + ay^2 - 2bx - 2cy + d = 0$ is

$$d(x^2 + y^2) - 2bk^2 x - 2ck^2 y + ak^4 = 0.$$

The following is now obvious: the inverse of a circle through the center of inversion ($a \neq 0$ and $d = 0$) is a line not through the center of inversion; the inverse of a line through the center of inversion ($a = 0$ and $d = 0$) is a line through the center of

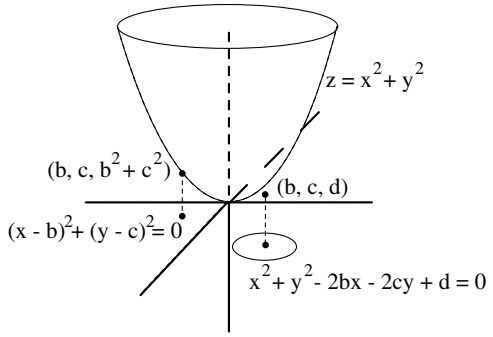


FIG. 3.1. *Representation of circles as points in* $\mathbb{R}^3$.

inversion; the inverse of a line not through the center of inversion ($a = 0$, $d \neq 0$) is a circle through the center of inversion.

An inversion in an arbitrary circle with the center at (x_0, y_0) is obtained by translating the center of inversion to the origin, performing an inversion about the origin, and translating back to (x_0, y_0).

3. Representation of circles as points in $\mathbb{R}^3$. We introduce a third model for m-circles in the inversive plane that will be central to our analysis (Figure 3.1).

3.1. Representation of circles as points in $\mathbb{R}^3$. Properties of a circle may be read from the representation

$$M \equiv a(x^2 + y^2) - 2bx - 2cy + d = 0.$$

If $a \neq 0$, the center of the circle M is $(\frac{b}{a}, \frac{c}{a})$, and the square of the radius is $\frac{b^2+c^2-ad}{a^2}$. If $a = 0$ and $c \neq 0$, then the slope of line M is given by $-\frac{b}{c}$ and the y-intercept by $\frac{d}{2c}$. If $c = 0$, then $b \neq 0$, and the line is $x = \frac{d}{2b}$.

We can represent M by the four-tuple (a, b, c, d). These are homogeneous coordinates of M since $(\lambda a, \lambda b, \lambda c, \lambda d)$ represents the same m-circle as (a, b, c, d) for nonzero λ. It is convenient to normalize to make the coordinates unique.

One choice would be to set $d = 1$. This is equivalent to insisting that the circles (and lines) do not pass through the origin, since the origin is not a point on any circle with $d = 1$. Every m-circle in the plane not through the origin would be represented by a unique four-tuple $(a, b, c, 1)$ such that $(a, b, c) \neq (0, 0, 0)$.

Instead we will take a second normalization, $a = 1$. So now we are dealing directly with circles only, with the circles through the origin corresponding to lines by an inversion in the unit circle. This representation of a circle now takes the form of a triple (b, c, d), insisting that $a = 1$. The center of a circle is (b, c), and the square of the radius is $b^2 + c^2 - d$. This normalization is the inverse of the normalization $d = 1$ above. That is, we are omitting all the circles through the point at infinity (i.e., lines).

In [18] Pedoe presents a nice three-dimensional representation of this model. The three-tuple (b, c, d) corresponds to the circle centered at (b, c) of radius $\sqrt{b^2 + c^2 - d}$, so given a point in $\mathbb{R}^3$, the center of the corresponding circle is the vertical projection of that point onto the xy-plane. If Ω denotes the paraboloid $z = x^2 + y^2$, then a point on Ω represents a circle with radius zero (a point circle), a point above Ω represents a circle with complex radius, and a point below Ω represents a circle with a positive

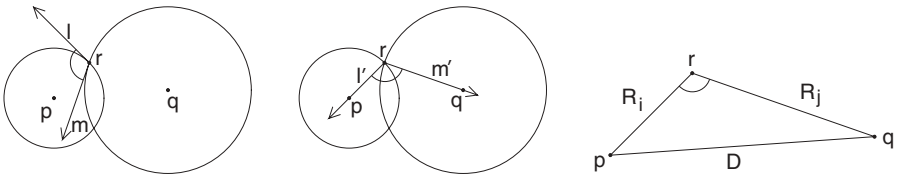


FIG. 3.2. *Measuring the angle of intersection of two circles: rotate by* 90° *and form a triangle with the centers.*

radius. Since we are primarily interested in circles with positive radius, we will focus on the points outside the paraboloid. We refer to this model as the *paraboloid model of the inversive plane*.

3.2. The angle of intersection between two circles. In order to eliminate the ambiguity of the angle of intersection between two circles, we introduce a convention. We orient circles in the counterclockwise direction and measure the angle between the oriented tangents to the circles at the point of intersection. In Figure 3.2, the angle of intersection of the two circles is the angle subtended by l and m in the counterclockwise direction. A rotation of $\frac{\pi}{2}$ sends l, m onto l', m'. Since l, m are tangents to the circles, l', m' pass through the centers of the circles. Therefore, the angle of intersection is $\angle prq$, where p and q are the centers of the circles and r is the point of intersection. (Note that the angle at the second point of intersection of the circles is identical.)

The cosine law applied to the triangle prq in Figure 3.2 yields

$$cos(\angle prq) = \frac{R_i^2 + R_j^2 - D^2}{2R_iR_j}, \tag{3.1}$$

where R_i, R_j are the radii of the circles, and D is the distance $|pq|$. Representing the circles by points (b_i, c_i, d_i) and (b_j, c_j, d_j) in $\mathbb{R}^3$, (3.1) becomes

$$\mathcal{K}_{ij} \equiv cos(\angle prq) = \frac{2b_ib_j + 2c_ic_j - d_i - d_j}{2\sqrt{(b_i^2 + c_i^2 - d_i)(b_j^2 + c_j^2 - d_j)}}. \tag{3.2}$$

As a special case, two circles (b_i, c_i, d_i) and (b_j, c_j, d_j) are orthogonal iff

$$2b_ib_j + 2c_ic_j - d_i - d_j = 0. \tag{3.3}$$

This condition is equivalent to $R_i^2 + R_j^2 = D^2$.

3.3. Coaxal systems and bundles of circles. There is a rich geometry of circles that would be needed to explore examples in these designs and to explore special position configurations. However, for the specific content of this paper, we will just briefly describe two specific families related to linear dependence of m-circles.

Two distinct circles $\mathbf{m}_1 = (b_1, c_1, d_1)$ and $\mathbf{m}_2 = (b_2, c_2, d_2)$ span the family of circles called the *coaxal system of circles generated by* $\mathbf{m}_1$ *and* $\mathbf{m}_2$, obtained by taking affine combinations of $\mathbf{m}_1$ and $\mathbf{m}_2$,

$$\lambda\mathbf{m}_1 + (1-\lambda)\mathbf{m}_2 = (\lambda b_1 + (1-\lambda)b_2, \lambda c_1 + (1-\lambda)c_2, \lambda d_1 + (1-\lambda)d_2).$$

Therefore, a coaxal system of circles is represented by a line l in $\mathbb{R}^3$. This line projects onto a line in the xy-plane; hence the centers of the circles in the coaxal system are

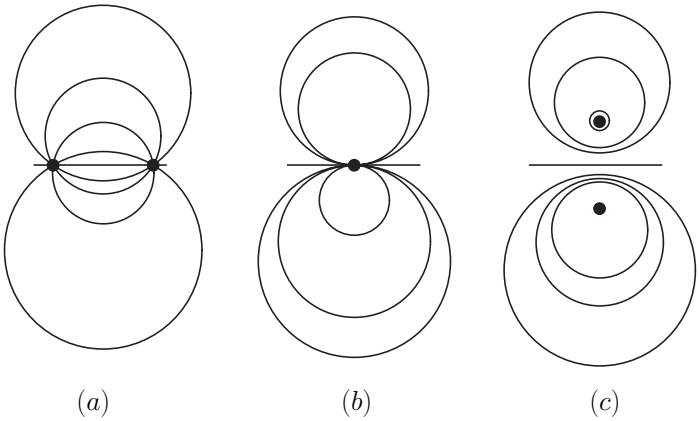


FIG. 3.3. *Coaxal system of circles:* (a) *where the line missed the paraboloid;* (b) *where the line is tangent to the paraboloid;* (c) *where the line intersects the paraboloid.*

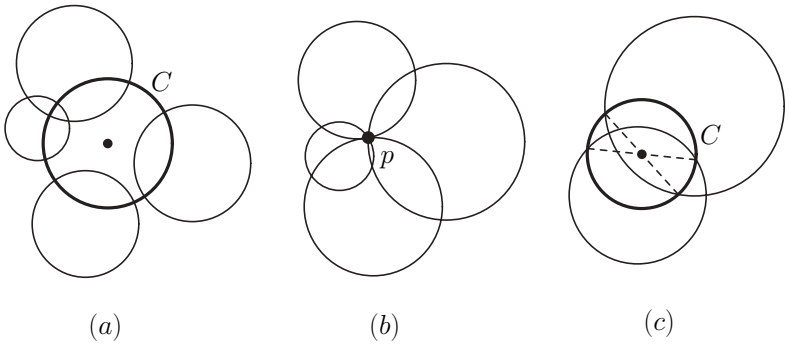


FIG. 3.4. *Bundles of circles:* (a) *circles orthogonal to the fixed circle C;* (b) *circles through the fixed point p;* (c) *circles intersecting antipodal points of the fixed circle C or, equivalently, circles orthogonal to a circle with a complex radius.*

collinear. Visually and algebraically, there are three different types of coaxal systems of circles corresponding to whether l misses, is tangent to, or intersects the paraboloid Ω. The three types of coaxal systems of circles are illustrated in Figure 3.3.

A *bundle of circles* is generated by three affinely independent circles, represented by a plane $\mathbf{P}$ in $\mathbb{R}^3$. There are three types of bundles depending on whether $\mathbf{P}$ misses, is tangent to, or intersects the paraboloid $z = x^2 + y^2$. The three types of bundles are illustrated in Figure 3.4.

4. *m*-circle designs. An *m-circle design* $(G, \mathbf{m})$ is a graph $G = (V, E)$ together with a point $\mathbf{m} \in \mathbb{R}^{3|V|}$, where $\mathbf{m} = (\mathbf{m}_1, \ldots, \mathbf{m}_i, \ldots, \mathbf{m}_{|V|})$, $i \in V$, such that $b_i^2 + c_i^2 - d_i > 0$ for each $\mathbf{m}_i = (b_i, c_i, d_i)$. We wish to track when two m-circle designs are equivalent under inversion, but that problem is too difficult. Instead we will consider a simpler problem: When is the design $(G, \mathbf{m})$ unique, with the given angles, in a neighborhood of $\mathbf{m}$ in $\mathbb{R}^{3|V|}$? In full generality, this local uniqueness is also too hard—but it does have a linearized version that answers the question of local uniqueness for almost all designs $\mathbf{m} \in \mathbb{R}^{3|V|}$. This linearized or *first-order* version is studied in the next few subsections. We will then return to state the standard results about how this first-order analysis demonstrates the local uniqueness.

4.1. The constraint function. The *constraint function* $\mathcal{K}_{i,j}$ for two circles $\mathbf{m}_i$ and $\mathbf{m}_j$ of nonzero radius is

$$\mathcal{K}_{i,j} = \frac{2b_ib_j + 2c_ic_j - d_i - d_j}{2\sqrt{(b_i^2 + c_i^2 - d_i)(b_j^2 + c_j^2 - d_j)}}. \tag{4.1}$$

Note that the constraint function has an obvious geometric interpretation only if $\mathcal{K}_{i,j} \in [-1, 1]$—it measures the cosine of the angle of intersection between the circles $\mathbf{m}_i$ and $\mathbf{m}_j$. However, the constraint function exists for nonintersecting circles as well and can be used for geometric purposes. It takes the value $\cosh\delta$, where δ is the natural logarithm of the ratio (larger to smaller) of the radii of two concentric circles. It can be shown that any two nonintersecting circles can be inverted into two concentric circles and that this ratio is constant. δ is called the *inversive distance* between the two circles [34].

In general, a single inversion in a circle multiplies the value of $\mathcal{K}_{i,j}$ by -1. Since inversion preserves angles and inversive distance, $|\mathcal{K}_{i,j}|$ is invariant under all inversions.

However, for local uniqueness, we will restrict ourselves to the group of *direct circular transformations*: products of an even number of inversions. In general, a single inversion is not local—it takes a configuration to a "faraway" configuration in the appropriate metric for configurations in $\mathbb{R}^3$. This group includes translations, rotations, dilations by a positive factor, etc. The value of $\mathcal{K}_{i,j}$ is invariant under this group for all circles with positive radii. In fact, the constraint function is invariant even for circles with imaginary radii. The function is not defined for point circles (of radius 0).

4.2. Shakes and the constraint matrix. Let $\mathbf{m}_i = (b_i, c_i, d_i)$ and $\mathbf{m}_j = (b_j, c_j, d_j)$ be two m-circles, with nonzero radius and constraint $\mathcal{K}_{i,j} = C$, where C is some constant. If $\mathbf{m}(t) = (\mathbf{m}_i(t), \mathbf{m}_j(t))$ is a path differentiable at $t = 0$ with $\mathbf{m}(0) = (\mathbf{m}_i, \mathbf{m}_j)$, then

$$\left(\frac{\mathrm{d}}{\mathrm{dt}}\mathcal{K}_{i,j}\right)(0) \;=\; \frac{\mathbf{k}_{i,j}}{h}\cdot\mathbf{m}_i' + \frac{\mathbf{k}_{j,i}}{h}\cdot\mathbf{m}_j' \;=\; 0, \tag{4.2}$$

where

$$\begin{aligned}
\frac{\mathbf{k}_{i,j}}{h} &= \left[\frac{\partial}{\partial b_i}\mathcal{K}_{i,j}, \frac{\partial}{\partial c_i}\mathcal{K}_{i,j}, \frac{\partial}{\partial d_i}\mathcal{K}_{i,j}\right], \\
\mathbf{m}_i' &= \left[\frac{\mathrm{d}}{\mathrm{dt}}b_i, \frac{\mathrm{d}}{\mathrm{dt}}c_i, \frac{\mathrm{d}}{\mathrm{dt}}d_i\right], \\
h &= 2\sqrt{(b_i^2 + c_i^2 - d_i)(b_j^2 + c_j^2 - d_j)}.
\end{aligned}$$

Since $h \neq 0$, (4.2) is equivalent to

$$\mathbf{k}_{i,j}\cdot\mathbf{m}_i' + \mathbf{k}_{j,i}\cdot\mathbf{m}_j' \;=\; 0. \tag{4.3}$$

This prompts the following definition.

DEFINITION 4.1. *Let $(G, \mathbf{m})$ be an m-circle design. The vector $\mathbf{m}' \in \mathbb{R}^{3|V|}$ is a* first-order motion *or* shake *of $(G, \mathbf{m})$ if for every $\{i, j\} \in E$,*

$$\mathbf{k}_{i,j}\cdot\mathbf{m}_i' + \mathbf{k}_{j,i}\cdot\mathbf{m}_j' \;=\; 0.$$

This system of linear equations generates the constraint matrix $\mathcal{C}(G,\mathbf{m})$ of the m-circle design $(G,\mathbf{m})$. *The space of shakes of* $(G,\mathbf{m})$ *is precisely the nullspace of* $\mathcal{C}(G,\mathbf{m})$.

Remark. The constraint matrix has one row for each constraint in E and three columns for each m-circle $\mathbf{m}_i$ (one column for each of b_i, c_i, d_i). So $\mathcal{C}(G,\mathbf{m})$ is a $|E| \times 3|V|$ matrix. For reference we note that $\mathbf{k}_{i,j} = [h\frac{\partial}{\partial b_i}\mathcal{K}_{i,j}, h\frac{\partial}{\partial c_i}\mathcal{K}_{i,j}, h\frac{\partial}{\partial d_i}\mathcal{K}_{i,j}]$, where

$$h\frac{\partial}{\partial b_i}\mathcal{K}_{i,j} = \frac{b_i d_j + 2b_j c_i^2 - 2b_j d_i - 2b_i c_i c_j + b_i d_i}{{b_i}^2 + {c_i}^2 - d_i},$$

$$h\frac{\partial}{\partial c_i}\mathcal{K}_{i,j} = \frac{c_i d_j + 2c_j {b_i}^2 - 2c_j d_i - 2c_i b_i b_j + c_i d_i}{{b_i}^2 + {c_i}^2 - d_i},$$

$$h\frac{\partial}{\partial d_i}\mathcal{K}_{i,j} = -\frac{1}{2}\frac{2{b_i}^2 + 2{c_i}^2 - 2b_i b_j - 2c_i c_j - d_i + d_j}{{b_i}^2 + {c_i}^2 - d_i},$$

and

$$\mathcal{K}(G,\mathbf{m}) = \begin{array}{c} \\ \{1,2\} \\ \\ \{i,j\} \\ \\ \end{array} \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & \cdots & i & \cdots & j & \cdots & v \end{array} \\ \begin{pmatrix} \mathbf{k}_{1,2} & \mathbf{k}_{2,1} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{k}_{i,j} & \cdots & \mathbf{k}_{j,i} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \end{pmatrix} \end{array}.$$

4.3. The trivial shakes. The following six vectors, called the *trivial shakes*, are in the nullspace of $\mathcal{C}(G,\mathbf{m})$ for any graph G on the given circles $\mathbf{m}$.

$$\begin{bmatrix} b_1^2 - \frac{1}{2}d_1 \\ b_1 c_1 \\ b_1 d_1 \\ \vdots \\ b_v^2 - \frac{1}{2}d_v \\ b_v c_v \\ b_v d_v \end{bmatrix}, \begin{bmatrix} b_1 c_1 \\ c_1^2 - \frac{1}{2}d_1 \\ c_1 d_1 \\ \vdots \\ b_v c_v \\ c_v^2 - \frac{1}{2}d_v \\ c_v d_v \end{bmatrix}, \begin{bmatrix} \frac{1}{2}c_1 \\ -\frac{1}{2}b_1 \\ 0 \\ \vdots \\ \frac{1}{2}c_v \\ -\frac{1}{2}b_v \\ 0 \end{bmatrix}, \begin{bmatrix} -\frac{1}{2} \\ 0 \\ -b_1 \\ \vdots \\ -\frac{1}{2} \\ 0 \\ -b_v \end{bmatrix}, \begin{bmatrix} 0 \\ -\frac{1}{2} \\ -c_1 \\ \vdots \\ 0 \\ -\frac{1}{2} \\ -c_v \end{bmatrix}, \begin{bmatrix} -b_1 \\ -c_1 \\ -2d_1 \\ \vdots \\ -b_v \\ -c_v \\ -2d_v \end{bmatrix}. \tag{4.4}$$

These six vectors are linearly independent provided $v = |V| \geq 3$ and at least three circles do not lie in a coaxal system.

Remark. The six solution vectors described above correspond to instantaneous velocities experienced by the m-circle design at the beginning of each type of even inversive transformation. The first two vectors correspond to translations in the x- and y-directions, respectively. The third vector corresponds to a rotation about the origin. The fourth and fifth vectors correspond to a reflection in an axis followed by an inversion in a circle tangent to that axis. The sixth vector corresponds to the product of two inversions in concentric circles—a dilation.

4.4. Stiff *m*-circle designs. A configuration $\mathbf{m}$ of m-circles is in *general position* if no set of 4 of the m-circles lie in a bundle, and if $|V| = 3$, then the circles are not coaxal. An m-circle design $(G,\mathbf{m})$ with $|V| \geq 3$ and $\mathbf{m}$ in general position is *inversively stiff*, or just *stiff*, if the kernel of $\mathcal{C}(G,m)$ is generated by the trivial shakes

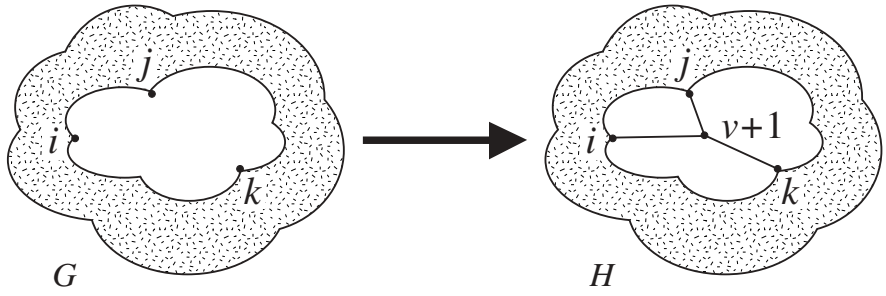


FIG. 4.1. *Extending the graph G of a stiff design to a graph H for a larger stiff design.*

$(\text{rank}(\mathcal{C}(G,m)) = 3|V|-6)$. Otherwise, $(G,\mathbf{m})$ is said to be *inversively shaky*, or just *shaky*.

This section proves the following characterization of stiff m-circle designs (see Theorem 4.5): An m-circle design $(G,\mathbf{m})$ with $|V| \geq 3$ and $\mathbf{m}$ in general position is stiff iff the nullspace of $\mathcal{C}(G,\mathbf{m})$ is equal to the nullspace of $\mathcal{C}(K_{|V|},\mathbf{m})$, where $K_{|V|}$ is the complete graph on $|V|$ vertices.

LEMMA 4.2. *If $|V| \geq 3$, then* $\text{rank}(\mathcal{C}(G,\mathbf{m})) \leq 3|V|-6$.

Proof. If we have at least three inversively independent circles, then the assertion follows directly from the independence of the six trivial motions (see (4.4)). If all of the m-circles are dependent on two m-circles, then it is a simple matter to see that the rank of the matrix can drop only from the maximum dimension achieved for independent m-circles. In fact, in this case, the rank becomes $|V|-1 < 3|V|-6$. ∎

LEMMA 4.3. *For $|V| \geq 3$ and $\mathbf{m} = (\mathbf{m}_1,\ldots,\mathbf{m}_{|V|})$ in general position, there exists a graph $G=(V,E)$ such that* $\text{rank}(\mathcal{C}(G,\mathbf{m})) = 3|V|-6$.

Proof. The proof will induct on the number of vertices of G. For $|V|=3$, take the design $(G,\mathbf{m})$, where G is the complete graph on three vertices K_3 (a triangle) and $\mathbf{m}_1=(1,0,0)$, $\mathbf{m}_2=(0,1,0)$, $\mathbf{m}_3=(0,0,1)$. Note that $\mathbf{m}_1$, $\mathbf{m}_2$, and $\mathbf{m}_3$ are in general position. The constraint matrix for $(G,\mathbf{m})$ is

$$\mathcal{C}(G,\mathbf{m}) = \begin{bmatrix} 0 & 2 & -1 & 2 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & -3/2 & 0 & 0 & 0 & 2 & 0 & -1/2 \\ 0 & 0 & 0 & 0 & 1 & -3/2 & 0 & 2 & -1/2 \end{bmatrix}.$$

The second, seventh, and eighth columns of $\mathcal{C}(G,\mathbf{m})$ yield a 3×3 matrix of rank 3, so $\mathcal{C}(G,\mathbf{m})$ has rank $3 = 3|V|-6$.

Suppose there exists a graph G with v vertices such that $\text{rank}(\mathcal{C}(G,\mathbf{m})) = 3v-6$. Let H be the graph obtained from G by adding a new vertex $v+1$ to G and three new edges, each connecting $v+1$ to the distinct vertices i,j,k of G (Figure 4.1). Therefore, a new circle $\mathbf{m}_{v+1}$ is added to the design $(G,\mathbf{m})$ with three distinct constraints, creating the new design $(H,\mathbf{n})$. So if the constraint matrix of $(G,\mathbf{m})$ is

$$\mathcal{C}(G,\mathbf{m}) = \begin{pmatrix} \mathbf{k}_{1,2} & \mathbf{k}_{2,1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & & \vdots & & \vdots & & & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{k}_{f,g} & \cdots & \mathbf{k}_{g,f} \end{pmatrix},$$

then the constraint matrix $\mathcal{C}(H, \mathbf{n})$ is

$$\begin{array}{c} \\ \\ \\ \\ \\ \{i,v+1\} \\ \{j,v+1\} \\ \{k,v+1\} \end{array} \begin{array}{c} \begin{array}{cccccccc|c} & & i & j & k & & & & & v+1 \end{array} \\ \left(\begin{array}{cccccccc|c} \mathbf{k}_{1,2} & \mathbf{k}_{2,1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & | & \mathbf{0} \\ \vdots & & \vdots & & \vdots & & & \vdots & | & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{k}_{f,g} & \cdots & \mathbf{k}_{g,f} & | & \mathbf{0} \\ - & - & - & - & - & - & - & - & - & - \\ \mathbf{0} & \mathbf{0} & \mathbf{k}_{i,v+1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & | & \mathbf{k}_{v+1,i} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{k}_{j,v+1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & | & \mathbf{k}_{v+1,j} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{k}_{k,v+1} & \mathbf{0} & \cdots & \mathbf{0} & | & \mathbf{k}_{v+1,k} \end{array}\right) \end{array},$$

where $\mathbf{n} = (\mathbf{m}, \mathbf{m}_{v+1})$. The three added columns account for the m-circle $\mathbf{m}_{v+1}$, and the three added rows account for the constraint equations. Now, $\operatorname{rank}(\mathcal{C}(H, \mathbf{n})) = 3(v+1) - 6 = 3v - 3$ iff the three new rows of $\mathcal{C}(H, \mathbf{n})$ add 3 to the rank of $\mathcal{C}(G, \mathbf{m})$. The rank of $\mathcal{C}(G, \mathbf{m})$ increases by 3 iff

$$\begin{vmatrix} \mathbf{k}_{v+1,i} \\ \mathbf{k}_{v+1,j} \\ \mathbf{k}_{v+1,k} \end{vmatrix} \neq 0.$$

This condition is equivalent to

$$\frac{2}{r_{v+1}^2} \begin{vmatrix} b_{v+1} & c_{v+1} & d_{v+1} & 1 \\ b_i & c_i & d_i & 1 \\ b_j & c_j & d_j & 1 \\ b_k & c_k & d_k & 1 \end{vmatrix} \neq 0,$$

where r_{v+1} is the radius of the circle $\mathbf{m}_{v+1}$. Therefore, unless $\mathbf{m}_{v+1}$ is a linear combination of the circles $\mathbf{m}_i$, $\mathbf{m}_j$, and $\mathbf{m}_k$, that is, unless $\mathbf{m}_i$, $\mathbf{m}_j$, $\mathbf{m}_k$, and $\mathbf{m}_{v+1}$ lie in the same bundle, $\operatorname{rank}(\mathcal{C}(H, \mathbf{n})) = 3v - 3$. By assumption, the m-circle design is in general position, so this determinant is nonzero, and the new design has rank $3(|V| + 1) - 6$ as required. □

LEMMA 4.4. *For $|V| \geq 3$ and $\mathbf{m} = (\mathbf{m}_1, \ldots, \mathbf{m}_{|V|})$ in general position, the nullspace of $\mathcal{C}(K_{|V|}, \mathbf{m})$ is generated by the trivial shakes.*

Proof. Let $\mathcal{S}$ denote the span of the trivial shakes. Since the trivial shakes are solutions to the linear system $\mathcal{C}(G, \mathbf{m})\mathbf{m}' = 0$ for any graph G, we have that $\mathcal{S} \subset \operatorname{null}(\mathcal{C}(G, \mathbf{m}))$.

By Lemma 4.3, for any $|V| \geq 3$ and any $\mathbf{m} = (\mathbf{m}_1, \ldots, \mathbf{m}_{|V|})$ in general position, there exists a graph $G = (V, E)$ with $\operatorname{rank}(\mathcal{C}(G, \mathbf{m})) = 3|V| - 6$. Add edges to G to obtain the complete graph $K_{|V|}$ on the vertex set V. Therefore,

$$3|V| - 6 = \operatorname{rank}(\mathcal{C}(G, \mathbf{m})) \leq \operatorname{rank}(\mathcal{C}(K_{|V|}, \mathbf{m}),$$

and Lemma 4.2 gives

$$\operatorname{rank}(\mathcal{C}(K_{|V|}, \mathbf{m})) \leq 3|V| - 6.$$

Therefore, $\operatorname{rank}(\mathcal{C}(K_{|V|}, \mathbf{m})) = 3|V| - 6$ and

$$\dim \operatorname{null}(\mathcal{C}(G, \mathbf{m})) = 3|V| - \operatorname{rank}(\mathcal{C}(K_{|V|}, \mathbf{m})) = 6 = \dim \mathcal{S}.$$

Therefore, $\operatorname{null}(\mathcal{C}(G, \mathbf{m})) = \mathcal{S}$. □

THEOREM 4.5. *An m-circle design $(G, \mathbf{m})$ with $|V| \geq 3$ and $\mathbf{m}$ in general position is stiff iff the nullspace of $\mathcal{C}(G, \mathbf{m})$ is equal to the nullspace of $\mathcal{C}(K_{|V|}, \mathbf{m})$, where $K_{|V|}$ is the complete graph on $|V|$ vertices.*

4.5. Stiffness and local uniqueness. Our original goal was to study whether an m-circle design $(G, \mathbf{m})$ was locally unique, up to inversions. Explicitly, we have a map $f_G : \mathbb{R}^{3|V|} \to \mathbb{R}^{|E|}$ that measures the "cosine of the angle" or the value of $\mathcal{K}_{i,j}$ for each edge in the graph: $f(\mathbf{m}) = (\ldots, k_{i,j}, \ldots)$. Let $I(\mathbf{m})$ be the set of all configurations equivalent to $\mathbf{m}$ by inversions.

DEFINITION 4.6. *A design* $(G, \mathbf{m})$ *is* locally unique *if there is an open neighborhood* $N_{\mathbf{m}}$ *of* $\mathbf{m}$ *such that* $f_G^{-1} f_G(\mathbf{m}) \cap N_{\mathbf{m}} \subset I(\mathbf{m})$.

Our constraint matrix is then the Jacobian $\mathrm{d}f_G$ of the function f_G. Moreover, the function f_G is (up to squaring entries) a polynomial function. There are standard results about how this Jacobian at $\mathbf{m}$ predicts the dimension of the space $f_G^{-1}(f_G(\mathbf{m})) \cap N_{\mathbf{m}}$, provided the point is *regular*, that is, that the Jacobian achieves its maximum rank at the point $\mathbf{m}$ [4, 20].

For polynomial functions, there are also standard results that state the failure of local uniqueness is equivalent to the existence of an analytic path $\mathbf{m}(t)$, $0 \le t < 1$, of inversively inequivalent configurations inside $f_G^{-1}(f_G(\mathbf{m}))$ (that is, all constraints are the same as at $\mathbf{m}$, but some other value of $\mathcal{K}_{h,l}$ is changing for some $\{h, l\} \notin E$). The following is a translation of the analogous result for bar-and-joint frameworks [4, 20]. We say that an m-circle design is *flexible* if there is an analytic path $\mathbf{m}(t)$, $0 \le t \le 1$ of inversively equivalent designs with the same constraints.

THEOREM 4.7. *If an* m*-circle design* $(G, \mathbf{m})$ *is stiff, then the design is locally unique. If* $(G, \mathbf{m})$ *is never stiff for any* m*-circle configuration, then* $(G, \mathbf{m})$ *is flexible for all regular points* $\mathbf{m}$ *at which* $\mathcal{C}(G, \mathbf{m})$ *achieves its maximum rank.*

Proof. For $|V| = 1$, all designs are stiff, and all single circles are inversively unique up to translations (of the center) and dilation (of the radius).

For $|V| = 2$, with distinct circles, there are two cases. If the edge is not present, then the design is not stiff, nor is the design even locally unique (e.g., change the distance between the centers without changing radii). However, if the edge is present, then all the solutions to $\mathcal{C}(G, \mathbf{m}) \times \mathbf{m}' = \mathbf{0}$ are trivial, and the design is stiff. In this case, any two circles with the same value of $\mathcal{K}_{1,2}$ are equivalent under inversion, and the design is, again, unique.

Assume that $(G, \mathbf{m})$ is stiff with $|V| \ge 3$. Therefore, the only inversive shakes are the trivial shakes that are derivatives of direct circular maps.

Assume we have an analytic path $\mathbf{m}(t)$ preserving the constraints; then this can be replaced by an analytic flex. If we take the first derivatives along this path, with the angles fixed, we will find a shake of the design. If this shake is not the derivative of an angle preserving map, then we know that the design was not stiff. This is a contradiction.

However, if a flex is not angle preserving, it may be the kth derivative that is not the derivative of an angle preserving map. By adding some angle preserving map, we can ensure that the first $k - 1$ derivatives of the flex are all zero. (For example, we can assume that an initial circle is fixed and that other circles are fixed as long as the derivatives match angle preserving maps.) With this assumption, it is easy to verify that the kth derivative of the constraint functions gives a (nontrivial) shake to the design. This is the desired contradiction.

On the other hand, assume that $(G, \mathbf{m})$ is never stiff for any m-circle configuration. The inverse function theorem guarantees that, at regular points, the dimension of $f_G^{-1}(f_G(\mathbf{m})) \cap N_{\mathbf{m}}$ is more than 6. There is a sequence $\mathbf{m}(n)$, $n = 1, 2, \ldots$ of designs converging to $\mathbf{m}$ that preserve the angle constraints. By the curve selection theorem of Milnor, which applies to the constraints, if there is such a converging sequence preserving these constraints (which can be written in polynomial form), then there

is a piecewise analytic path preserving the constraints [4, 20]. This gives the desired flex at the selected regular point. □

5. Bar-and-joint frameworks in Euclidean 3-space. So far, our study of circles in the plane under angle constraints is clearly analogous to the study of bar-and-joint frameworks in Euclidean 3-space. The constraints for a bar-and-joint framework take the form of a distance constraint (a bar) between two vertices (joints) and generate graphs and constraint matrices for which the results precisely match the results of the previous sections [24, 25, 32]. We will briefly summarize this theory in order to present a precise isomorphism that underlies this analogy.

As before, we start with a graph $G = (V, E)$ and create a framework by realizing the vertices as points in 3-space.

DEFINITION 5.1. *A* bar-and-joint framework *or* framework $(G, \mathbf{p})$ *in* $\mathbb{R}^3$ *is a graph* $G = (V, E)$ *together with a configuration or point* $\mathbf{p} \in \mathbb{R}^{3|V|}$*, where* $\mathbf{p} = (\mathbf{p}_1, \ldots, \mathbf{p}_i, \ldots, \mathbf{p}_{|V|})$*,* $i \in V$*.*

5.1. First-order motions and the rigidity matrix. A *first-order motion* of the framework $(G, \mathbf{p})$ is a map $\mathbf{u} : V \to \mathbb{R}^3$, where we denote $\mathbf{u}(i)$ by $\mathbf{u}_i$, such that for every edge $\{i, j\} \in E$,

$$(\mathbf{p}_i - \mathbf{p}_j) \cdot (\mathbf{u}_i - \mathbf{u}_j) = 0.$$

This gives rise to the *rigidity matrix* of the bar-and-joint framework $(G, \mathbf{p})$:

$$R(G, \mathbf{p}) \;=\; \begin{matrix} \\ \{i,j\} \end{matrix} \begin{pmatrix} & i & \cdots & j & \\ & \vdots & & \vdots & \\ \cdots & \mathbf{p}_i - \mathbf{p}_j & \cdots & \mathbf{p}_j - \mathbf{p}_i & \cdots \\ & \vdots & & \vdots & \end{pmatrix}$$

(with all other entries zero). The nullspace of the rigidity matrix is the space of first-order motions of $(G, \mathbf{p})$. A first-order motion of a framework is *trivial* if the motion is a restriction of the derivative of a Euclidean motion of $\mathbb{R}^3$, restricted to the vertices of the framework. The framework $(G, \mathbf{p})$ is *first-order rigid* if all the motions of $(G, \mathbf{p})$ are trivial.

5.2. Trivial solutions of the rigidity matrix. The following are six linearly independent vectors in the space of first-order motions for any framework in $\mathbb{R}^3$ with at least 3 noncollinear vertices. They correspond to translations in the x-, y-, and z-directions and rotations about the x-, y-, and z-axes, respectively:

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ -z_1 \\ y_1 \\ \vdots \\ 0 \\ -z_{|V|} \\ y_{|V|} \end{bmatrix}, \begin{bmatrix} -z_1 \\ 0 \\ x_1 \\ \vdots \\ -z_{|V|} \\ 0 \\ x_{|V|} \end{bmatrix}, \begin{bmatrix} -y_1 \\ x_1 \\ 0 \\ \vdots \\ -y_{|V|} \\ x_{|V|} \\ 0 \end{bmatrix}.$$

A framework is first-order rigid if its space of first-order motions is generated by these motions.

6. An isomorphism between frameworks and m-circle designs. It can shown that a first-order rigid framework on $v \geq 3$ joints requires at least $3v - 6$ bars. The proof is identical to the proofs presented in section 4. In fact, the proofs from section 4 were adapted from proofs for the equivalent statements for bar-and-joint frameworks. Therefore, the counts for rigid bar-and-joint frameworks and stiff m-circle designs are identical. This is not a coincidence: there is a geometric isomorphism between the two first-order theories.

The key observation is the following identity relating the determinant of a submatrix of $\mathcal{C}(G, \mathbf{m})$ with the determinant of a submatrix of $R(G, \mathbf{m})$:

$$\begin{vmatrix} \mathbf{k}_{n,i} \\ \mathbf{k}_{n,j} \\ \mathbf{k}_{n,k} \end{vmatrix} = \frac{2}{r_n^2} \begin{vmatrix} b_n & c_n & d_n & 1 \\ b_i & c_i & d_i & 1 \\ b_j & c_j & d_j & 1 \\ b_k & c_k & d_k & 1 \end{vmatrix} = \frac{2}{r_n^2} \begin{vmatrix} b_n - b_i & c_n - c_i & d_n - d_i \\ b_n - b_j & c_n - c_j & d_n - d_j \\ b_n - b_k & c_n - c_k & d_n - d_k \end{vmatrix}.$$

This identity suggests the existence of a linear transformation carrying $\mathcal{C}(G, \mathbf{m})$ onto $R(G, \mathbf{m})$. Indeed, the system

$$\begin{bmatrix} \mathbf{k}_{n,i} \\ \mathbf{k}_{n,j} \\ \mathbf{k}_{n,k} \end{bmatrix} \mathbf{T}_n = \begin{bmatrix} b_n - b_i & c_n - c_i & d_n - d_i \\ b_n - b_j & c_n - c_j & d_n - d_j \\ b_n - b_k & c_n - c_k & d_n - d_k \end{bmatrix}$$

has the solution

$$\mathbf{T}_n = \begin{bmatrix} -\frac{1}{2} & 0 & -b_n \\ 0 & -\frac{1}{2} & -c_n \\ -b_n & -c_n & -2d_n \end{bmatrix}.$$

In general, $\mathcal{C}(G, \mathbf{m})\, \mathbf{T}_\mathbf{m} = R(G, \mathbf{p})$, where $\mathbf{T}_\mathbf{m}$ is the block diagonal matrix, which depends only on the point $\mathbf{m}$,

$$\mathbf{T}_\mathbf{m} = \begin{bmatrix} \mathbf{T}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{T}_2 & \cdots & 0 \\ 0 & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \mathbf{T}_v \end{bmatrix}.$$

We summarize this discussion.

THEOREM 6.1. *Given an m-circle design $(G, \mathbf{m})$, there is an invertible transformation $\mathbf{T}_\mathbf{m}$ such that $\mathcal{C}(G, \mathbf{m}) \times \mathbf{T}_\mathbf{m} = R(G, \mathbf{m})$. In particular, the m-circle design $(G, \mathbf{m})$ is stiff iff the bar-and-joint framework $(G, \mathbf{m})$ is first-order rigid.*

Notice that not all configurations in 3-space for frameworks are m-circle configurations, since we have restricted ourselves to circles of positive radius (points in $\mathbb{R}^3$ below the paraboloid).

Remark. It is important to note that the two theories are only infinitesimally equivalent. That is, on the level of Jacobians, the two theories are equivalent. There is no transformation that carries the constraint equation of one system into the constraint equation of the other system. Rigidity is not equivalent, but first-order rigidity is equivalent to stiffness. The regular points of the two maps for a fixed G are identical (they are defined by the rank of the isomorphic Jacobians). The rigidity or flexibility of frameworks or designs at these regular points (which form an open dense subset of $\mathbb{R}^{3v}$) are also equivalent. The distinctions all occur on the singular points of the two maps, where one configuration may still give a locally unique design for one map but a flexible framework for the other.

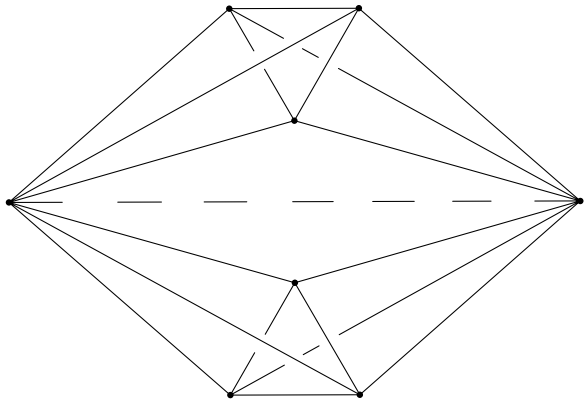


FIG. 7.1. *Although the framework satisfies all the conditions of Proposal* 1*, the framework is not rigid: there is a rotation of one half of the framework about the dotted line.*

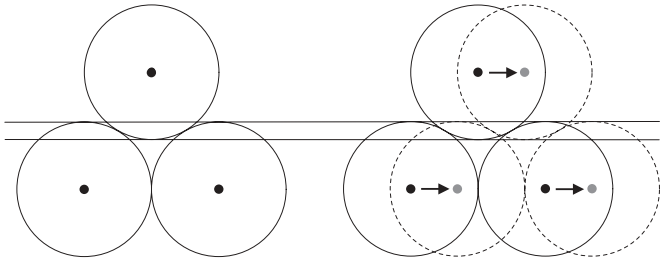


FIG. 7.2. *An m-circle design corresponding to the framework in Figure* 7.1.

7. Transferring between theories. There are many results and conjectures for the widely studied first-order theory of frameworks in 3-space that convert easily into the less studied theory of circles and angles in the inversive plane [9, 29, 30, 32]. However, it is important to emphasize that both equivalent theories are incomplete.

7.1. The failure of the counts. A question that arises naturally is to characterize graphs that result in a stiff design for some configuration of circles. A quick conjecture may take the following form.

PROPOSAL 1. *If a graph G with $|V| \geq 3$ satisfies $|E| = 3|V| - 6$ and all subgraphs G' of G with $|V'| \geq 3$ satisfy $|E'| = 3|V'| - 6$, then the m-circle design is stiff.*

This is the immediate generalization of a theorem of Laman [15] characterizing edge-minimal first-order rigid bar-and-joint frameworks in the plane. This proposal does not hold in the theory of bar-and-joint frameworks, as is evident in Figure 7.1. This framework satisfies the conditions of the proposal but is obviously flexible with a rotation of one piece about the dotted line. If we construct an m-circle design on the same graph, then the resulting design is also not uniquely determined, as is evident in Figure 7.2.

7.2. Projective transformations. Since inversions are angle preserving transformations, if $\mathbf{m}$ is an m-circle configuration and we apply any inversive transformation T, then $T(\mathbf{m})$ gives an isomorphic constraint matrix for any graph. The stiffness, independence, etc. of any design $(G, \mathbf{m})$ is invariant under inversion.

Similarly, it is clear that if $\mathbf{p}$ is a spatial configuration and we apply any congruence map T, then $T(\mathbf{p})$ gives an isomorphic set of distance constraints, and for any graph the rank of the rigidity matrix is unchanged. The first-order rigidity, independence, etc. of any framework $(G, \mathbf{p})$ is invariant under congruence.

Now it is not true that an inversion in the plane induces a congruence of the 3-space paraboloid model. However, there is a family of transformations that includes both the spatial versions of inversions and the congruences of Euclidean 3-space. These are the projective transformations of 3-space.

It is well known that these projective transformations leave the rank of the corresponding rigidity matrices unchanged for any graph G and any configuration $\mathbf{m}$ where the points remain finite. (For projective points at infinity, there is a full projective form of the theory, including a projective rigidity matrix, which incorporates such constraints involving such points [3, 26, 27].)

From our correspondence, it follows that these projective transformations also leave the rank of the constraint matrix unchanged, provided that none of the points of the configuration move onto the paraboloid, where both the constraint matrix and the isomorphism are undefined. In fact, the inversive maps are precisely the projective transformations which preserve the paraboloid.

We close by stating (without proof) this conclusion.

THEOREM 7.1 (see [20]). *Given an m-circle design $(G, \mathbf{m})$ and a projective transformation T of 3-space such that $T(\mathbf{m})$ is another configuration of real circles, the constraint matrices $\mathcal{C}(G, \mathbf{m})$ and $\mathcal{C}(G, T(\mathbf{m}))$ have isomorphic row spaces.*

8. Extensions to other dimensions. Up to this point, we have studied m-circle designs for the plane and their connection to bar-and-joint frameworks in 3-space. However, the basic problems of CAD lie in three dimensions, and frameworks have been studied in all dimensions. It is natural to ask whether our results extend immediately to m-sphere designs with spheres in 3-space constrained by angles of intersection and bar-and-joint frameworks in 4-space. They do.

All the correspondences extend to designs of (hyper)spheres in n dimensions and bar-and-joint frameworks in $n + 1$ dimensions. We will present only this extension explicitly for m-sphere designs and frameworks in 4-space, but the reader will easily see how the general extension works.

As an aside, we note that there is also a correspondence between the first-order theory of plane bar-and-joint frameworks and m-interval designs along the line. While we know of no direct use for these m-interval designs, such an analysis is useful anytime an m-circle design or an m-sphere design contains a substantial piece which lies in a linear family, as this piece will behave as an m-interval design. See section 9.

For spheres in 3-space, we choose a similar normalization to that for circles and represent the sphere with equation

$$x^2 + y^2 + z^2 - 2bx - 2cy - 2dz + e = 0$$

by the four-tuple (b, c, d, e), where the orthogonal projection of (b, c, d, e) onto the xyz-space yields the center of the sphere, (b, c, d). The radius of the sphere is $b^2+c^2+d^2-e$. The cosine of the angle of intersection between two intersecting spheres (b_1, c_1, d_1, e_1) and (b_2, c_2, d_2, e_2) is

$$\mathcal{K}_{1,2} \;=\; \frac{2b_1b_2 + 2c_1c_2 + 2d_1d_2 - e_1 - e_2}{2\sqrt{(b_1^2 + c_1^2 + d_1^2 - e_1)(b_2^2 + c_2^2 + d_2^2 - e_2)}}.$$

From this equation, one has the constraint equation and can easily derive the $|E|\times 4|V|$ constraint matrix $\mathcal{C}(G,\mathbf{m})$ for a design $(G,\mathbf{m})$, $\mathbf{m}\in\mathbb{R}^{4|V|}$.

The first-order theory of bar-and-joint frameworks in 4-space gives the analogous $|E|\times 4|V|$ rigidity matrix $R(G,\mathbf{m})$ for the configuration.

To translate between the first-order theory of bar-and-joint frameworks in 4-space and the m-spheres in 3-space, we use the following transformation:

$$\mathbf{T_m}=\begin{bmatrix}\mathbf{T}_1 & 0 & \cdots & 0\\ 0 & \mathbf{T}_2 & \cdots & 0\\ 0 & \vdots & \ddots & 0\\ 0 & 0 & 0 & \mathbf{T}_{|V|}\end{bmatrix},$$

where

$$\mathbf{T}_i=\begin{bmatrix}-\frac{1}{2} & 0 & 0 & -b_i\\ 0 & -\frac{1}{2} & 0 & -c_i\\ 0 & 0 & -\frac{1}{2} & -d_i\\ -b_i & -c_i & -d_i & -2e_i\end{bmatrix}.$$

As before, $\mathcal{C}(G,\mathbf{m})\times\mathbf{T_m}=R(G,\mathbf{m})$ for all graphs and all configurations with no points on the paraboloid.

At this point, we hope it is obvious how the results continue to generalize into 3-space and also generalize into higher dimensions.

Remark. It is surprising to find a geometric structure in 3-space whose constraints geometrically model the problems of rigidity in 4-space. Previous work [32] provided a structure from bivariate splines in CAGD which was analogous to the generic properties of rigidity in 4-space, but this structure is known to not be even generically equivalent [32]. m-sphere designs in 3-space provide a full 3-space embodiment of the geometric and combinatorial first-order theory of frameworks in 4-space.

The limited studies of first-order rigidity in 4-space suggest major complexity that does not arise simply from the counting (matroidal) level [32]. This complexity is a warning that general geometric constraints in 3-space are much harder than the simple theory of distances in plane frameworks.

9. Circles of fixed radius. Within CAD programming, it is possible that some or all of the circles have a fixed radius. The theory presented offers a simple extension for this case, independent of whether the fixed radii are distinct numbers or all the same size.

9.1. A fixed radius. The radius r of a circle (b,c,d) is given by the equation

$$r^2=b^2+c^2-d.$$

If r is constant, the usual process for finding the Jacobian turns this into the homogeneous linear equation,

$$(2b)b'+(2c)c'-d'=0.$$

This gives a row for our constraint matrix which has only nonzero entries $[2b,2c,-1]$ under the single circle.

If we partition the circles into two classes, indexed by V for variable radii circles and U for fixed radii circles, we simply add such a row for each circle of fixed radius

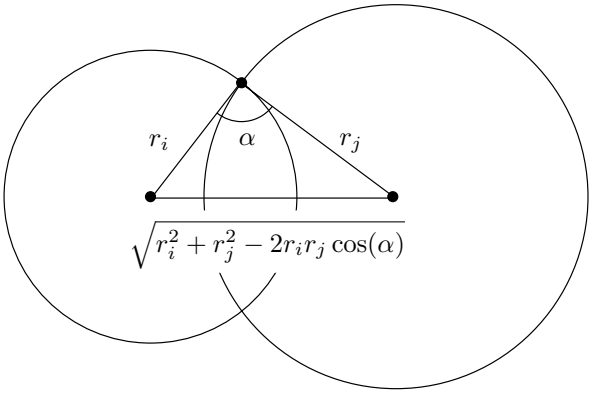


FIG. 9.1. *If we have two circles of fixed radii with a fixed angle of intersection, then the distance between the centers is also fixed.*

to create the *extended constraint matrix* $\mathcal{C}((V, U; E), \mathbf{m})$. While we could analyze this directly, it will be simpler to translate immediately to the corresponding matrix for first-order rigidity: $\mathcal{C}((V, U; E), \mathbf{m}) \times \mathbf{T_m} = R((V, U; E), \mathbf{m})$.

If we examine the transfer multiplication for the added rows for circles in U, we find

$$[2b\ 2c\ -1] \times \begin{bmatrix} -\frac{1}{2} & 0 & -b \\ 0 & -\frac{1}{2} & -c \\ -b & -c & -2d \end{bmatrix} = [b - b\ c - c\ -b^2 - c^2 + d] = [0\ 0\ -r^2].$$

We have an extended rigidity matrix that forces the third coordinates of each of the points in U to have derivative 0 (assuming $r \neq 0$ as we have throughout the entire translation process). Alternatively, we can use these rows to do a row reduction which makes all other entries in the third columns into 0 and sets off a set of $|U|$ independent rows at the bottom.

9.2. All circles of fixed radius. If $V = \emptyset$ and all radii are fixed, the extended rigidity matrix is now the rigidity matrix for a plane framework with vertices at the centers of the circles, plus an added, spread copy of the identity at the bottom. The angle constraints on the circles are isomorphic, at first-order, to distances constraints between the centers.

This equivalence is also evident from elementary geometry. If we have two circles with constant radii r_i and r_j, respectively, then the angle constraint α actually does fix the distance between the centers, and this distance is given by the law of cosines $\sqrt{r_i^2 + r_j^2 - 2r_i r_j \cos\alpha}$ (Figure 9.1). (Note that this is even true when the circles are nonintersecting and the constraint is an inversive distance. The only real limitation in this is that the circles have nonzero, though possibly imaginary, radii.)

As mentioned in section 7.1, this theory of plane points, with distance constraints or, equivalently, plane circles with fixed radii and angle constraints, has a combinatorial theory represented by the counts for independence [15]:

$$|E'| \leq 2|U'| - 3 \text{ for all nonempty subsets } U' \subseteq U.$$

Similarly for all n, if we work with hyperspheres in n-space with angle constraints and all radii fixed, we find the theory is isomorphic to the theory of points in n-space with distance constraints.

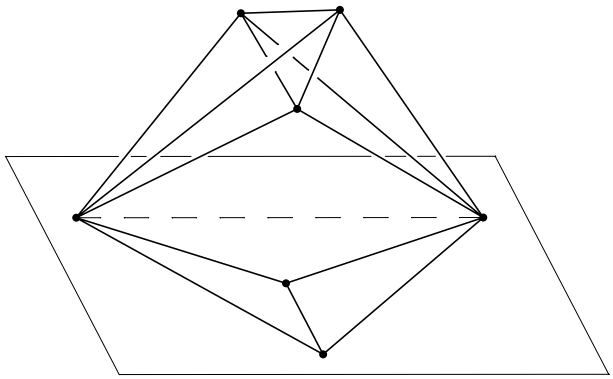


FIG. 9.2. *Although the mixed framework satisfies all the necessary counting conditions for mixed structures, the framework is not rigid: there is a rotation of one half of the framework about the dotted line.*

9.3. Some circles of fixed radius. A more interesting middle situation is where we have some circles of fixed radii and others of variable radii. Equivalently, we may be looking to see which subsets of angles and radii are independent constraints in a design, as new constraints of either type are considered for addition to a currently independent design.

Assume that $|V| \geq 3$ and $|U| \geq 3$. With this assumption on $|U|$, the only trivial motions are generated by the translations parallel to the x- and y-axes and rotation about the z-axis, which occurred for plane rigidity. (The reader can verify this, even in the original circle constraint representation with the initial list of generators for the trivial motions and the initial representation of a fixed radius constraint.) This leaves the following obvious counting condition for independence:

$$|E'| \leq 3|V'| + 2|U'| - 3 \quad \text{for all } |U'| \geq 3.$$

If $|U| < 3$, we have other spaces of trivial motions, as follows:

1. for $|U| = 2$ and $|V| \geq 1$, the space of trivial motions has dimension 4, adding the 3-space rotations about the line through the two points in U;
2. for $|U| = 1$ and $|V| \geq 2$, the space of trivial motions has dimension 5, adding the 2-space of spatial rotations fixing this point to the trivial motions of the plane;
3. for $|U| = 0$ and $|V| \geq 3$, the space of trivial motions has dimension 6.

These observations can be pulled together into necessary counting conditions for independence. However, provided the design is large enough to contain structures such as the two bananas of Figure 7.1 or the adapted mixed version of Figure 9.2, these conditions will not be sufficient for independence.

9.4. Lines among circles of fixed radius. All of this analysis for fixed radii works within our simplifying assumption that our m-circles did not include lines. Of course, with no fixed radii, or only one, we could use inversion (fixing the radius) to pull the lines into circles and continue the analysis of this equivalent configuration containing no lines.

However, with several fixed radii, we are restricted in our transformations, and lines are not incorporated into the simple theory. Intuitively, if we fix the "radii of lines" as circles of infinite radius, then we would restrict them to remain lines. (The

transformations should be reversible, and no fixed finite radius can become infinite.) By an elementary observation, an angle constraint between a circle of fixed radius and a line fixed to remain a line, still fixes the distance from the center of the circle to the line.

A detailed analysis here would again require a fully projective presentation, using homogeneous coordinates for all points in the model (or, equivalently, for lines and circles, including the effects of fixed radii). It can be done if there are specific situations where this would be significant.

9.5. Fixed radius circles in the spherical model. If we work with circles and angle constraints on the sphere, the condition for a fixed radius changes. Recall that a circle is represented by the point at the tip of a cone tangent to the sphere at the circle: $t = (x, y, z)$. To hold the radius of such a circle fixed, we just fix the distance (squared) from this point to the center of the sphere (the origin): $x^2 + y^2 + z^2 = d^2$. In the Jacobian, this gives the row whose only nonzero entries are (x, y, z). This is equivalent, for any analysis of independence or dependence of constraints, to adding the center of the sphere as a vertex and a bar from this center to the circle point, creating an overall 3-space framework for these radial constraints and the angle constraints. This equivalence holds both combinatorially and at the specific geometric level of possible special positions in which a generically independent set of constraints drops in rank and becomes dependent.

If we constrain the radius of each of the circles, then we have the center of the sphere as a vertex joined to all the other vertices. By the cone theorem for frameworks [28, 32], this is equivalent to the constraint matrices for the framework created by projecting from the center onto a projection plane tangent to the sphere at $z = 1$. (This is a general projection, so points below the equatorial plane are joined to the center and the line extended to intersect the projection plane.) It is also equivalent to the spherical framework in which each of the circle vertices is pulled onto the initial sphere (at the center of the original circle), and one studies the framework constrained to remain on the sphere.

It is worth recalling that the central projection to a plane framework is different from the stereographic projection into a plane circle configuration from the top of the sphere. The plane framework here is distinct from the framework in which the plane radii were fixed. In general, these two frameworks for fixed radii in the plane and fixed radii on the sphere are not even projectively equivalent. They do have the same graph and will, generically, have the same independence structure. However, for special positions arising from the geometric placement of the plane vertices, they may have distinct behavior.

10. Concluding remarks. We have analyzed correspondences among circles and lines in the inversive plane, circles on the sphere, and points in Euclidean (and hyperbolic) space. An identical pattern happens for other dimensions. For example, if we study the angle constraints between intersecting (and nonintersecting) spheres in inversive 3-space, these are isomorphic to angle constraints between corresponding hyperplanes in hyperbolic 4-space and the $\mathcal{K}$ constraints among ideal points in hyperbolic 4-space. At first order, they are also identical to the distance constraints between points in Euclidean 4-space, as we noted in section 8.

There are geometric connections between these interconnected problems of circles in the plane or the sphere and Andreev's theorem and its extensions [1]. Via the correspondence offered here and the related correspondences in [21], these results of Andreev are also connected to the rigidity theorems for convex polyhedra of Cauchy

and Alexandrov (see [19, 30]). We will explore these connections more extensively in a forthcoming paper.

By giving a correspondence between angle constraints in the inversive plane and distance constraints in the Euclidean space, we raised the question of a polynomial time algorithm for the generic rank of a configuration of circles and angles. The corresponding unsolved problem for 3-space has been studied, and conjectured about, for over a century. At least one other plane geometric problem, that of bivariate C_2^1-splines, is also conjectured to be isomorphic at a generic (but not a geometric) level for the rank of a corresponding matrix on a given graph [32]. The study of each of these variants has contributed to our store of shared techniques and results, but we need new approaches to solve the shared problem.

A natural question to ask is whether the situation with m-circles can contribute any additional insights. In [21] we describe the equivalence of first-order rigidity in all the Cayley–Klein geometries extracted from the underlying projective geometry, including the hyperbolic, spherical, and Euclidean spaces [6]. This isomorphism suggests that we will not easily find new combinatorial results by switching so transparently among these equivalent theories.

There remain other variants of these problems of plane objects such as points, lines, and circles with geometric constraints in CAD. Many are unsolved, and some are simply unstudied. Consider including points which are assigned to lie on one or several circles in an m-circle design. In its most general form, such an incidence pattern would include the projective configurations of lines and incident points. After all, lines are simply inversive circles which all share a common point (chosen to be the inversive point at infinity). Without some additional restrictions, this problem with circles should be at least as hard as the specific problem of incident lines and points in the plane, which we have previously conjectured to have no polynomial time algorithm [32].

However, if we insist that all circles intersect at one specific point, with fixed angles at that point, then we have a problem which has been solved. This is inversively equivalent to the problem of lines with fixed angles—or, equivalently, parallel drawings of configurations of lines—and has a polynomial time algorithm related to the counting algorithms for the generic rigidity for plane frameworks [22, 31]. Moreover, the analogous problem for parallel drawings of planes in 3-space with fixed angles also has a polynomial time algorithm [31].

If we drop the condition that all angles at the common point of intersection are fixed, we return, once more, to an unsolved (and difficult) problem. In the plane, we have studied lines, incidences, and angles with additional simplifying restrictions that all lines are "short" (have no more than two assigned incident points). These incidence constraints can then be extended by additional selected angle constraints among the points and lines. Even this very special case is hard and is conjectured not to have a polynomial time algorithm [2].

All of these interconnected problems, many unsolved, confirm the complexity of various sets of constraints in plane and spatial CAD. The specific case of points and distance constraints in the plane, plane first-order rigidity, stands out as an exceptional case in which we do have a polynomial time algorithm. This is not typical of constraints in CAD, and strategies based on the assumption of polynomial time algorithms for even generic rigidity or independence of constraints are limited in their applications [10, 11]. The need for fast symbolic algorithms does, in effect, restrict the patterns of constraints that are handled well in CAD programs, before resorting to more brute force (and more unstable) numerical analysis of the constraints [16, 17].

The study of various sets of constraints, even in plane CAD, continues to generate rich connections backward into classical geometry in all its forms, new connections among these classical problems, and new insights. When we started this investigation of circles and angles, we had no expectation that it would lead to hyperbolic 3-space and correspondences to Euclidean space. We look forward, with anticipation, to the next piece of the puzzle and the connections it will bring forward for our geometric play.

Acknowledgments. This work was supported by continuing conversations over several summers within a larger group of summer research students at York University. We particularly want to thank Lily Berenchtein for her contributions, both direct and indirect, to our understanding of these correspondences and the broader correspondences in [21]. The second author also wants to thank John Owen of D-Cubed Limited for continuing conversations on the problems in the geometry in parametric CAD, including our introduction to the problems of constraints on plane circles.

REFERENCES

[1] E. M. Andreev, *Convex polyhedra in Lobačevskiĭ spaces,* Mat. Sb. (N.S.), 81 1970, pp. 445–478.
[2] M. Bousfield, K. Caldwell, L. Duong, D. Moscovitz, and W. Whiteley, *Constraining Plane Configurations in CAD: Angles,* preprint, York University, North York, ON, Canada, 1996.
[3] H. Crapo and W. Whiteley, *Statics of frameworks and motions of panel structures: A projective geometric introduction,* Structural Topology, 6 (1982), pp. 43–82.
[4] R. Connelly and W. Whiteley, *Second-order rigidity and prestress stability for tensegrity frameworks*, SIAM J. Discrete Math., 9 (1996), pp. 453–491.
[5] R. Connelly, *On generic global rigidity,* in Applied Geometry and Discrete Mathematics, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 4, AMS, Providence, RI, 1991, pp. 147–155.
[6] H. S. M. Coxeter, *An Introduction to Non-Euclidean Geometry,* 6th ed., Mathematical Association of America, Washington, D.C., 1998.
[7] J. Graver, B. Servatius, and H. Servatius, *Combinatorial Rigidity*, AMS, Providence, RI, 1994.
[8] B. Hendrickson, *Conditions for unique graph realizations*, SIAM J. Comput., 21 (1992), pp. 65–84.
[9] L. Henneberg, *Die graphische Statik der Starren Systeme*, Leipzig 1911, Johnson Reprint, 1968.
[10] C. Hoffman, A. Lomonosov, and M. Sitharam, *Decomposition plans for geometric constraint systems, part* I*: Performance measures for CAD,* J. Symbolic Comput., 31 (2001), pp. 367–408.
[11] C. Hoffman, A. Lomonosov, and M. Sitharam, *Decomposition plans for geometric constraint systems, part* II*: New algorithms,* J. Symbolic Comput., 31 (2001), pp. 409–428.
[12] C. Hoffman and C.-S. Chiang, *Variable-radius circles in cluster merging, part* I, CAD, 34 (2002), pp. 787–797.
[13] C. Hoffman and C. S. Chiang, *Variable-radius circles in cluster merging, part* II, CAD, 34 (2002), pp. 799–805.
[14] G. Kramer, *Solving Geometric Constraint Systems (A Case Study in Kinematics),* MIT Press, Cambridge, MA, 1992.
[15] G. Laman, *On graphs and the rigidity of plane skeletal structures,* J. Engrg. Math., 4 (1970), pp. 331–340.
[16] J. C. Owen, *Algebraic solutions for geometry from dimensional constraints,* in Proceedings of the First Symposium on Solid Modeling Foundations and CAD/CAM Applications, Austin, TX, 1991, ACM, New York, 1991, pp. 397–407.
[17] J. C. Owen, *Constraints on simple geometry in two and three dimensions,* Internat. J. Comput. Geom. Appl., 6 (1996), pp. 421–434.
[18] D. Pedoe, *Geometry: A Comprehensive Course,* Dover, New York, 1988.
[19] A. V. Pogorelov, *Extrinsic Geometry of Convex Surfaces,* Transl. Math. Monogr. 35, AMS, Providence, RI, 1973.

[20] B. Roth and W. Whiteley, *Tensegrity frameworks,* Trans. Amer. Math. Soc., 177 (1981), pp. 419–446.
[21] F. Saliola and W. Whiteley, *The First-Order Equivalence of Rigidity in Euclidean, Hyperbolic and Spherical Geometries,* preprint, York University, North York, ON, Canada.
[22] B. Servatius and W. Whiteley, *Constraining plane configurations in computer-aided design: Combinatorics of directions and lengths,* SIAM J. Discrete Math., 12 (1999), pp. 136–153.
[23] K. Sugihara, *Detection of structural inconsistency in systems of equations with degrees of freedom,* J. Discrete Appl. Math., 10 (1985), pp. 297–312.
[24] T.-S. Tay, *On the generic rigidity of bar-frameworks,* Adv. in Appl. Math., 23 (1999), pp. 14–28.
[25] T.-S. Tay and W. Whiteley, *Generating all isostatic frameworks,* Structural Topology, 11 (1985), pp. 21–69.
[26] T.-S. Tay, N. White, and W. Whiteley, *Skeletal rigidity of symplicial complexes* I, European J. Combin., 16 (1995), pp. 381–403.
[27] T.-S. Tay, N. White, and W. Whiteley, *Skeletal rigidity of symplicial complexes* II, European J. Combin., 16 (1995), pp. 503–523.
[28] W. Whiteley, *Cones, infinity and one-story buildings,* Structural Topology, 8 (1983), pp. 53–70.
[29] W. Whiteley, *Infinitesimal rigidity of a bipartite framework,* Pacific J. Math., 110 (1984), pp. 233–255.
[30] W. Whiteley, *Infinitesimally rigid polyhedra* I*: Statics of frameworks,* Trans. Amer. Math. Soc., 285 (1984), pp. 431–461.
[31] W. Whiteley, *Some matroids on hypergraphs with applications to scene analysis and geometry,* Discrete Comput. Geom., 4 (1988), pp. 75–95.
[32] W. Whiteley, *Matroids for discrete applied geometry,* in Matroid Theory, Contemp. Math. 197, J. Bonin, J. Oxley, and B. Servatius, eds., AMS, Providence, RI, 1996, pp. 171–311.
[33] W. Whiteley, *Constraining Plane Geometric Configurations in CAD: Directions and Distances,* preprint, Department of Mathematics and Statistics, York University, North York, ON, Canada, 1999.
[34] J. Wilker, *Inversive geometry,* in The Geometric Vein: The Coxeter Festschrift, C. Davis, B. Grunbaum, and F. A. Sherk, eds., Springer, New York, 1981, pp. 379–442.

SIAM J. DISCRETE MATH.
Vol. 18, No. 2, pp. 272–286

DYNAMIC DIGRAPH CONNECTIVITY HASTENS MINIMUM SUM-OF-DIAMETERS CLUSTERING*

SARNATH RAMNATH†

Abstract. Dynamic data structures are presented for directed graphs that maintain (a) transitive closure and (b) decomposition into strongly connected components in a "semionline" situation with perfect deletion lookahead but no lookahead for insertions or queries. These algorithms give us "semionline" algorithms for dynamic 2-SAT, as a consequence of which the best known static algorithms for minimum sum-of-diameters clustering are improved by a $O(\log n)$ factor.

Key words. dynamic digraph connectivity, sum-of-diameters clustering, semionline algorithms

AMS subject classifications. 05C12, 05C20, 05C40, 05C85, 05C90, 68Q25

DOI. 10.1137/S0895480102396099

1. Introduction.

Clustering of data is a very old and well-studied problem that dates back to Aristotle, with applications in the natural sciences, psychology, engineering, and a variety of other fields [10]. A basic problem of cluster analysis is to partition a given set of entities into homogeneous and/or well-separated classes, called *clusters*. Separation is commonly characterized by the dissimilarity between objects, which can be expressed as the *distance* between objects. A measure often used in characterizing the homogeneity of a set is the *diameter*, which is defined as the largest distance between any pair of items in the set [9].

The minimum sum of diameters clustering problem is described as follows:

Input. A set of n items, $a_1, a_2, \ldots, a_n$, and an integer k; associated with each pair (a_i, a_j) is a length l_{ij}, which represents the distance between a_i and a_j.

Output. A partitioning of the set into k subsets such that the sum of the diameters of the subsets is minimized.

The input can be represented by a weighted graph, which we shall call the *cluster graph*, as follows: represent each item a_i by a vertex numbered i; add an edge e_{ij} between vertex i and vertex j with length l_{ij}. The output is a partitioning of the vertex set into k clusters $C_0, C_1, \ldots, C_{k-1}$ with diameters $D_0, D_1, \ldots, D_{k-1}$, respectively.

Minimum diameter or minimum sum-of-diameter partitions are of interest in many situations where homogeneity of the clusters is the main concern of the analyst. Brucker has shown that both these problems are NP-hard when $k \geq 3$ [2]. In [15], Rao showed that when $k = 2$, the minimum diameter clustering problem can be solved in $O(n^2)$ time. However, it is well known that minimum diameter partitions suffer from the *dissection effect:* very similar entities may be assigned to very different clusters, and the requirement that the clusters have fairly equal diameters can cause a natural cluster to be split [2, 4, 5, 15]. Since no such equalizing factor is at play, this effect is usually much less damaging when the sum of diameters is minimized [9]. It also appears that, in practice, a bipartitioning algorithm can be recursively applied to approximate a partitioning into three or more clusters that minimizes the sum of the diameters [9].

*Received by the editors September 5, 2002; accepted for publication (in revised form) December 11, 2003; published electronically October 8, 2004.
http://www.siam.org/journals/sidma/18-2/39609.html
†Department of Computer Science, ECC-139, St. Cloud State University, St. Cloud, MN 56301 (rsarnath@stcloudstate.edu).

In the case of minimum sum-of-diameters clustering, the first approximation algorithms for general k were recently given by Doddi et al. [8]. For the case $k = 2$, Hansen and Jaumard gave an $O(n^6)$ algorithm, which they then improved to $O(n^3 \log n)$ [9]. Monma and Suri have shown that in the case of sparse cluster graphs, this algorithm in fact runs in $O(mn \log n)$, where m is the number of edges [14]. (Here, the diameter of a cluster is defined as the length of the longest edge in the cluster.) We shall assume that $k = 2$ for the rest of this paper.

The algorithm used in [9] to find the best partitioning solves $O(n \log n)$ 2-SAT instances, each of which could take $O(m)$ time in the worst case. Here we present two algorithms that dynamically solve $O(m)$ 2-SAT instances, performing an average of $O(n^3/m)$ and $O(n)$ operations, respectively, for each instance. As a result of these, we obtain algorithms for minimum sum-of-diameters clustering that run in $O(n^3)$ and $O(mn)$, respectively. (Although the second result asymptotically subsumes the first, the data structures and overheads are significantly higher for the second algorithm, in particular due to the need for pointers.) The first algorithm dynamically maintains the transitive closure, and the second one dynamically maintains the partitioning of a graph into strongly connected components (SCCs). Both these approaches use the notion of *perfect deletion lookahead*: at any instant we know all the deletable edges in the graph and the order in which these edges are to be deleted. By doing some additional bookkeeping at insertions, we are able to speed up the deletion process. There is no foreknowledge of insertions and queries, and these can be interleaved with the deletes in any arbitrary order. As has been observed in earlier articles on dynamic digraph connectivity, deletion is an expensive operation when dynamically maintaining the transitive closure. In [3], for instance, it is shown that a series of edge insertions can be done in $O(n)$ amortized time per insert, but a similar result for edge deletions works only if the graph is acyclic. The problem becomes even harder if we consider arbitrarily interleaved sequences of inserts and deletes. Consequently, the update times obtained for digraph connectivity by the earlier researchers [12, 6, 11, 3, 13] do not help reduce the complexity of the clustering problem. In particular, the dynamic approach to the clustering operation requires in the worst case a sequence of $O(m + n)$ updates (inserts and deletes, interleaved) and $O(n)$ queries after each update. To improve the algorithm in [9] would therefore mean an amortized/average update time of $O(n)$. In this paper we use the perfect lookahead available to us to speed up the update operations. The idea of using lookaheads to speed up updates has been explored earlier by Khanna, Motwani, and Wilson in [11]. However, they use partial lookahead, and the resulting update times ($O(n^{2.18})$ with $n^{0.18}$ lookahead) are not good enough to improve the upper bounds for clustering in [9, 14]. Semionline dynamic algorithms (which are equivalent to perfect lookahead on one operation) have been considered previously for geometric problems, most notably in [7]. No such approaches appear to have been tried for graph connectivity. The scheme used in this paper to maintain transitive closure performs insertions in $O(n^2)$ time and deletions in $O(1)$ time; the scheme to maintain a decomposition into SCCs has an amortized cost of $O(m + n)$ for each deletable edge and a cost of $O(n)$ for each nondeletable edge.

The next section presents an overview of the approach in [9] and some background on the relationship between 2-SAT and directed graphs. The two following sections present the $O(n^3)$ and $O(mn)$ algorithms; the last section concludes the paper.

2. Preliminaries. As described earlier, our problem is to partition a given set of items into two clusters with diameters D_0 and D_1 such that the sum of D_0 and

D_1 is minimized. We assume without loss of generality that $D_0 \geq D_1$. We say that an edge belongs to a cluster if both end vertices of the edge belong to the cluster. Since the diameter of a cluster is the length of the longest edge in the cluster, the only candidates for D_0 and D_1 that we need to consider are the edge lengths. We shall assume that the cluster graph is not bipartite (if it were bipartite, we would have a trivial solution). Let S_l denote the set of edge lengths.

The algorithm in [9] works as follows.

ALGORITHM CLUSTER.

Step 1: Identify all edge lengths, d_0, in S_l that are possible candidates for D_0.

Step 2: For each candidate edge d_0, found in Step 1, identify the smallest value d_1 in S_l such that there exists a partitioning of the cluster graph into two sets with diameters not exceeding d_0 and d_1, respectively.

Step 3: Choose D_0 and D_1 to be the pair (d_0, d_1) such that the sum of d_0 and d_1 is minimized.

end Cluster

The following three results are from [9]; short proofs are included here for the sake of completeness.

LEMMA 2.1. *Consider a maximum spanning tree (MST) for the cluster graph, constructed using Kruskal's algorithm. The only edges whose lengths are candidates for D_0 are the edge that completed the first odd cycle and the edges included in the spanning forest before the first odd cycle was encountered. It follows that there are at most n candidates for D_0 and that all these candidates can be found in time $O(m + n \log n)$.*

Proof. Consider any odd cycle in the cluster graph. At least one edge of the cycle must fall inside a cluster; therefore, at least one cluster has a diameter no smaller than the length of the shortest edge in the odd cycle; i.e., D_0 must be at least as large as the length of this edge. Since Kruskal's algorithm for the MST will consider all the edges in nonincreasing order of lengths, the edge that completed the first odd cycle will be a shortest edge in that odd cycle.

Next, consider all the edges that completed even cycles before the first odd cycle was completed. For each such edge, e, there exists an even cycle such that all the other edges of the even cycle have a length no less than the length of e. If in any given partitioning e were to fall inside a cluster, then there must be at least one other edge from this even cycle that also falls inside a cluster; since this other edge has length no less than e, the length of e need not be considered as a candidate for D_0. Since Kruskal's algorithm to find the MST runs in $O(m + n \log n)$ time, we can find all the candidates for D_0 in the time bounds claimed in the lemma. □

The above lemma tells us how to compute Step 1 of Algorithm Cluster in $O(m + n \log n)$ time. Since there are at most n candidates for D_0, Step 3 is trivially done in $O(n)$ time. Step 2 is the most expensive part of the computation for which we describe improved algorithms in the following sections.

LEMMA 2.2. *Let D_{min} denote the length of the edge that completed the first odd cycle. All candidates for D_1 are lesser than or equal to D_{min}.*

Proof. From Lemma 2.1 we know that $D_0 \geq D_{min}$. If we choose $D_0 = D_{min}$, then $D_1 \leq D_{min}$. Now, if we increase the choice for D_0, the value of D_1 cannot increase. □

LEMMA 2.3. *Consider the following assertion: There is a partitioning of the vertices into two clusters with diameters not exceeding d_0 and d_1. This assertion can be represented as a 2CNF expression with n variables and $p + q$ conjuncts, where p*

is the number of edges with length greater than d_0 and q is the number of edges with length greater than d_1.

The construction of the 2CNF expression uses two kinds of constraints. If for some edge e_{ij}, $l_{ij} > d_1$, then we need to add a condition that vertex i and vertex j cannot both be in C_1; if x_i is a boolean variable set to 0 if vertex i falls in C_0 (set to 1 if vertex i falls in C_1), then this condition is expressed by the disjunct (not(x_i) OR not(x_j)). Likewise, if $l_{ij} > d_0$, we add the disjunct (x_i OR x_j). We shall refer to the constraint (x_i OR x_j) as the *Type*0 constraint and the constraint (not(x_i) OR not(x_j)) as the *Type*1 constraint of the edge e_{ij}. It has been shown in [1] that any 2CNF expression with n variables and m conjuncts can be represented as a digraph with $2n$ vertices and $2m$ directed arcs as follows: For each variable x_i, $1 \le i \le n$, add two vertices—u_i, labelled x_i and v_i, labelled not(x_i). For each constraint a OR b, where a and b are literals, add two directed edges—one from the vertex labelled by the simplest expression equivalent to $not(a)$ (since a could itself be negated, we may have to remove the double negation to obtain the vertex label) to the vertex labelled b and another directed edge from the vertex labelled by the simplest expression equivalent to $not(b)$ to the vertex labelled a. The 2CNF expression is unsatisfiable if and only if there is a directed cycle containing both u_i and v_i for some i, $1 \le i \le n$. It was shown in [1] that the satisfiability of a 2CNF expression can be decided by looking for such directed cycles in $O(m + n)$ time.

Hansen and Jaumard [9] compute Step 2 as follows: construct a list of edge lengths, $l_1, l_2, \ldots, l_q$, sorted in nonincreasing order, where $l_p = D_{min}$, $1 \le p \le q$. Further, let l_i, $i \le p$, be some candidate for D_0. If t^i_j is a boolean value that holds the truth value of the assertion "there is a partitioning of the vertices into two clusters with diameters not exceeding l_i and l_j," then the sequence of values $t^i_p, t^i_{p+1}, \ldots, t^i_q$ is monotone, and the largest j for which t^i_j is true can be found by a binary search. Since this would involve verifying the truth of $O(\log n)$ assertions of the above kind, each requiring $O(m)$ time, we can find j for any given i in $O(m \log n)$ time. Since there are no more than n candidates for D_0, the entire process can be completed in $O(mn \log n)$ steps.

Our approach differs from that in [9] in two ways: (1) Instead of carrying out a binary search for the largest j that satisfies t^i_j, we find j by a sequential search. (2) Instead of solving each 2-SAT instance from scratch, we do this incrementally/decrementally for each new value of j or i.

To decide a 2-SAT instance, we construct a directed graph. In this digraph we have to look at $O(n)$ pairs of vertices to check if any of these pairs falls in the same SCC. If we have the transitive closure or the decomposition into SCCs, this check can be performed with $O(n)$ queries, each requiring $O(1)$ time.

There are at most p instances of d_0, namely, $l_1, l_2, \ldots, l_p$. In the process of finding d_1 for each d_0, we can either start with $d_0 = l_1$ and then decrease d_0 all the way down to l_p, or we can start with $d_0 = l_p$ and increase to l_1. If we choose the former, we perform $O(n)$ inserts and $O(m)$ deletes; this is the approach we use in section 3, where we dynamically maintain the transitive closure for a digraph, with each insertion taking $O(n^2)$ time and each deletion taking $O(1)$ time, giving us a cost that is $O(n^3)$. If we choose the latter, we have $O(n)$ deletes and $O(m)$ inserts; i.e., the graph has $O(n)$ deletable edges. The scheme presented in section 4 maintains the decomposition into SCCs in such a way that each deletable edge requires $O(m)$ operations and each nondeletable edge requires $O(n)$ operations. Both these approaches must decide $O(m)$ 2-SAT instances, which would require $O(mn)$

steps. Thus we have an $O(n^3)$ algorithm if we maintain the transitive closure and an $O(mn)$ algorithm if we maintain the decomposition into SCCs.

We designate the directed graph used to represent a 2CNF expression as the *constraint graph.* Looking at the types of constraints imposed by the clustering problem, it is obvious that no two negated literals are connected by a directed arc, and, likewise, no two nonnegated literals are connected by a directed arc. Each Type0 constraint induces two edges that are directed from negated literals to nonnegated ones, and each Type1 constraint induces two edges that are directed from nonnegated literals to negated ones. We designate these edges Type0 and Type1 edges, respectively. In the following sections, we shall dynamically insert these edges into the constraint graph to obtain faster algorithms. Whenever we insert a constraint, it means that both the induced edges are added to the constraint graph. The following lemma tells us something about the edges in the cluster graph that complete even cycles in Kruskal's algorithm.

LEMMA 2.4. *Let $e_1, e_2, \ldots, e_m$ be a valid order in which Kruskal's algorithm considers the edges of the cluster graph in order to construct the MST; let T_i denote the subset of edges from $e_1, e_2, \ldots, e_i$ that are included in the MST by Kruskal's algorithm, and let e_i, $1 \leq i \leq m$, be an edge connecting vertices i_s and i_t that completes an even cycle. Then, in the constraint graph induced by the Type*0 *and Type*1 *constraints of the edges in T_{i-1}, we have the following:*

1. *u_{i_s} and v_{i_t} belong to the same strongly connected component.*
2. *u_{i_t} and v_{i_s} belong to the same strongly connected component.*

Proof. Since e_i completes an even cycle, there is a path $i_s, i_{\alpha_1}, i_{\alpha_2}, \ldots, i_{\alpha_{2j}}, i_t$ such that all edges on this path were considered before e_i by Kruskal's algorithm. If we add the Type0 and Type1 constraints for all the edges on this path, it is easy to verify that we have a path $u_{i_s}, v_{i_{\alpha_1}}, u_{i_{\alpha_2}}, \ldots, u_{i_{\alpha_{2j}}}, v_{i_t}$ in the constraint graph, comprised of alternating Type0 and Type1 edges. Similarly, we can find paths from v_{i_t} to u_{i_s}, from u_{i_t} to v_{i_s}, and from v_{i_s} to u_{i_t}. □

What this implies is that if we have a situation where (1) an edge e_{ij} that completes an even cycle in Kruskal's algorithm and (2) the Type0 and Type1 constraints for all the remaining edges in the even cycle have been added to the constraint graph, then adding the constraints for e_{ij} does not affect the connectivity (i.e., does not modify the transitive closure) of the constraint graph. Therefore, in a situation where we are inserting constraints in decreasing order of edge lengths, the edges of length greater than D_{min} that complete even cycles in Kruskal's algorithm can be ignored. All the constraints that we consider in the course of step 2 can therefore be classified into three kinds:

(1) Type1 constraints of edges of length greater than D_{min},
(2) Type0 constraints of edges of length greater than D_{min},
(3) Type1 constraints of edges of length less than or equal to D_{min}.

Since all candidates for d_1 are less than or equal to D_{min}, the constraints in (1) must always be satisfied. These can therefore be added as nondeletable edges to the constraint graph.

The constraints in the other two categories are related as follows: If we choose a smaller value for d_0, we have more constraints from (2), and consequently we may be forced to remove more constraints from (3) and thus increase the value of d_1. By Lemma 2.4, the number of constraints that we need to consider from (2) is $O(n)$. The algorithm in the next section first inserts all the constraints from (3) and then removes these as constraints from (2) are added. The approach in section 4 is to first

insert all the constraints from (2) and then remove these as constraints from (3) are added.

3. An $O(n^3)$ algorithm. In this section we describe an $O(n^3)$ algorithm for minimum sum-of-diameters clustering. We first present a fully dynamic graph connectivity algorithm and then use this to obtain the clustering algorithm

3.1. A fully dynamic connectivity algorithm. In this section we describe a fully dynamic graph connectivity algorithm with the following characteristics:

- Complete lookahead for deletion; i.e., at any point in time, we know all the edges currently in the graph that are going to be deleted at some future point, and we know the order in which these edges are going to be deleted. There is no knowledge of insertions and queries—these can be interleaved with the deletes in any arbitrary order.
- $O(n^2)$ time for each insertion.
- $O(1)$ time for each deletion.
- $O(1)$ query time. (The query checks if there exists a directed path connecting two specified vertices.)
- $O(n^3)$ precomputation time—given an input graph, the required data structures can be precomputed in $O(n^3)$ time.

We define the following concepts:

- Deletion time stamp (DTS): Associated with each edge is a DTS that gives the order in which the edge will be deleted, the edge with the largest DTS being the next edge to be deleted. For deletable edges, this is a unique integer in the range $[1 \dots n^2]$. Edges that will never be deleted have a $DTS = 0$.
- Current time stamp (CTS): An integer between 0 and n^2. When we delete an edge, the CTS is decremented. A CTS of i indicates that the graph has i deletable edges. When we add a deletable edge, CTS is incremented. At any point, CTS is equal to the largest DTS for all the edges in the graph. If an edge has a DTS that is greater than CTS, then the edge is not in the graph.
- Persistence number (PN): Associated with each directed path in the graph is a PN that is computed as the maximum of all the DTS values of the edges on that path. Intuitively, the PN of a path is a measure of how many deletes it will take to disconnect the path . For a path with a PN of p, this measure is computed as $CTS - p + 1$. Therefore, given two paths, the one with a lower PN is more persistent. If a path has a PN equal to CTS, the next delete will disconnect the path; after the delete, the CTS drops below the PN, i.e., $CTS - PN + 1 \leq 0$, implying that this path no longer exists in the graph.
- Connectivity number (CN): For each pair of vertices (u, v), we have a CN that is computed as the minimum of the PNs over all paths that connect from u to v. Intuitively, the CN gives us a measure of the number of deletes needed to eliminate all paths from u to v. If $CN(u, v) = c$, then this measure is computed as $CTS - c + 1$. Thus, if $CN(u, v) > CTS$, there is no directed path from u to v.

Our data structure for a graph with n vertices is an $n \times n$ matrix, which contains, for each pair of vertices (u, v), the DTS of the edge (u, v) and $CN(u, v)$.

LEMMA 3.1. *Given a graph G, with a DTS for each edge, the required data structure can be computed in $O(n^3)$ time.*

Proof. We need to compute a CN for each pair of vertices (i, j). Let $D^{(k)}(i, j)$ denote the PN for the most persistent path from i to j, such that none of the intermediate vertices has an index greater than k. We have the following dynamic programming recurrence:

$D^{(0)}(i, j) = DTS(i, j)$,

$D^{(k+1)}(i, j) = \min[D^{(k)}(i, j), \max(D^{(k)}(i, k + 1), D^{(k)}(k + 1, j))]$.

Since $CN(i, j) = D^{(n)}(i, j)$, our data structure can be precomputed in $O(n^3)$ time. □

The only work done to update the data structure is at the time of insertion, when the CN values are updated in accordance with the dynamic programming recurrence given in the above proof. If (u, v) is the new edge to be inserted with *DTS* t, then, for each pair (x, y), we have a new potential path from x to y, namely, x to u to v to y, and the PN of this path must be taken into account to determine $CN(x, y)$. There are two other issues we must deal with: (1) Some deletes may have been performed since the last insert; therefore, all the CN values that are greater than the CTS must be set to ∞. (2) If $t > 0$, there may be an existing edge in the graph with $DTS = t$; i.e., the new edge is inserted somewhere in the middle of the deletion sequence. In this case, we must increment the DTS of edges with $DTS \geq t$ and increment all the CNs that are greater than or equal to t. These operations are detailed in $Insert(u, v, t)$.

The data structure supports the following operations:

- $Insert(u, v, 0)$: (Insert a nondeletable edge from u to v). For all pairs of vertices (p, q) such that $CTS < CN(p, q) < \infty$, set $CN(p, q) = \infty$. For each pair of vertices (x, y), $CN(x, y) = \min(CN(x, y), \max(CN(x, u), CN(v, y)))$.
- $Insert(u, v, t)$: (Insert an edge with *DTS* t from u to v). For all pairs of vertices (p, q) such that $CTS < CN(p, q) < \infty$, set $CN(p, q) = \infty$. If $CTS \geq t$, increment the DTS for each edge that has a current DTS value greater than or equal to t. Increment the CTS. For each pair of vertices (p, q) with $CN \geq t$, increment $CN(p, q)$. For each pair of vertices (x, y), $CN(x, y) = \min(CN(x, y), \max(CN(x, u), CN(v, y), t))$.
- Delete(u, v): (Deletes the next edge in sequence). Decrement the CTS; $DTS(u, v) = \infty$.
- Path(u, v): (Returns True if there is a path connecting from u to v; False otherwise). If $CTS < CN(u, v)$, return False; else return True.

THEOREM 3.2. *The data structure described above maintains the transitive closure of a digraph, satisfying the following conditions:*

(1) $O(n^2)$ *time for each insertion.*
(2) $O(1)$ *time for each deletion.*
(3) $O(1)$ *query time.*
(4) *Given an input graph, the required precomputation can be done in* $O(n^3)$ *time.*

Proof. To prove that the data structure does indeed maintain the transitive closure, we observe the following: If an edge has a DTS that is less than the CTS, then this edge is present in the graph; and therefore if a path from u to v has a PN that is less than the CTS, the graph has a directed path from u to v. It follows then that if a pair (u, v) of vertices has a CN that is less than the CTS, then there exists a path from u to v and that the operation $Path(u, v)$ correctly tells us whether or not there is directed path from u to v.

To establish (1), note that any pair of vertices (x, y), $x \neq u$, $y \neq v$, is examined at most four times; pairs of the form (u, y) or (x, v) are examined at most $O(n)$ times each, and there are $O(n)$ such pairs. It follows that both the $Insert$ operations

described above are completed in $O(n^2)$ time. (2) and (3) are obvious, and (4) follows from Lemma 3.1. □

3.2. A $O(n^3)$ clustering algorithm. We compute Step 2 of Algorithm Cluster as follows: Add all the Type1 constraints to the constraint graph in nonincreasing order of edge lengths. Next, add the Type0 constraints for all the edges with length greater than D_{min} in nonincreasing order. As soon as we get a cycle containing a variable and its negation, we remove Type1 constraints in nondecreasing order of lengths until the cycle is removed. By keeping track of the length of the edges whose constraints created the cycle and the length of the edges whose constraints were removed to eliminate the cycle, we can obtain all the (d_0, d_1) pairs needed in Step 2 of Algorithm Cluster. This idea is elaborated below.

Let $l_1, l_2, \ldots, l_q$ be the q distinct edge lengths in the cluster graph sorted in nonincreasing order, and let S_i denote the set of all edges of length l_i in the cluster graph. To simplify the presentation, we define $l_0 = \infty$, $l_{q+1} = 0$, and S_0 and S_{q+1} as the associated (empty) sets. Let $D_{min} = l_p$, $1 \le p \le q$.

ALGORITHM CLUSTER1.

1. Insert all the Type1 constraints for edges with length greater than D_{min} into the constraint graph as follows:
   For $i = 1$ to p-1
   For each edge e_{jk} in S_i,
   $Insert(u_j, v_k, 0)$; $Insert(u_k, v_j, 0)$;
   end for;
   end for;
2. Insert all the Type1 constraints for edges with length less than or equal to D_{min}, into the constraint graph as follows:
   $DTS = 1$;
   For $i = p$ to q
   For each edge e_{jk} in S_i,
   $Insert(u_j, v_k, DTS++)$; $Insert(u_k, v_j, DTS++)$
   end for;
   end for;
3. $j = q + 1$;
   For $i = 0$ to p-1
   For each edge e_{kl} in S_i that does not complete an even cycle in Kruskal's algorithm,
   $Insert(v_k, u_l, 0)$; $Insert(v_l, u_k, 0)$;
   /*insert the Type0 constraints*/
   end for;
   While (constraint graph is unsatisfiable)
   $j = j$-1; Delete all Type1 constraints for edges in S_j;
   end while;
   Record (l_{i+1}, l_j) as a (d_0, d_1) pair.
   end for;

THEOREM 3.3. *Algorithm Cluster*1 *correctly computes Step* 2 *of Algorithm Cluster in* $O(n^3)$ *time.*

Proof. The correctness follows from the way we find the (d_0, d_1) pairs. If $d_0 = l_{i+1}$, then the constraint graph must contain Type0 constraints for all edges in $\bigcup_{k=1}^{i} S_k$. From the "while" condition, it is clear that to attain satisfiability, we need to remove at least some Type1 constraints imposed by the edges in S_j. This implies

that d_1 must be at least l_j; i.e., (l_{i+1}, l_j) form a (d_0, d_1) pair.

To establish the complexity, note that steps 1 and 2 perform a series of edge insertions with no interleaving deletes or queries. By Lemma 3.1, all these operations can be done in $O(n^3)$ time. In step 3, we insert Type0 constraints. From Lemma 2.4, we know that Type0 constraints of the edges that completed even cycles in the MST need not be added. Therefore, we have only $O(n)$ Type0 edges to insert, which can be done in $O(n^3)$ steps. In each iteration of the "while" loop, we check if the constraint graph is satisfiable; i.e., for all i, $1 \leq \mathrm{i} \leq n$, we check if u_i and v_i are on a directed cycle. Since our query time is a constant, each check of satisfiability takes $O(n)$ steps. Since j is initially q ($q < n^2$) and decreases in each iteration of the "while" loop, there are fewer than n^2 checks for satisfiability, and the entire computation can be completed in $O(n^3)$ steps. □

4. A $O(mn)$ algorithm.

4.1. Dynamically maintaining SCCs. The algorithm in the previous section cannot do better than $O(n^3)$ in general. The reason for this is that we have $\Omega(n)$ Type0 constraints to be inserted, each of which takes $O(n^2)$ time in the worst case. We shall now present a different technique to take care of sparse cluster graphs. Unlike the previous algorithm, this technique maintains a collection of all SCCs in the constraint graph. The problem of dynamically maintaining SCCs in directed graphs does not seem to have received much attention from researchers, except for [13], where King and Sagert have used the idea of SCC decomposition to maintain the transitive closure.

Definition (strongly connected subgraph). A strongly connected subgraph (SCS) of a digraph $G(V, E)$ is a set of vertices $C \subseteq V$ such that for every pair of vertices (u, v), u, $v \in C$, there exist directed paths p_f from u to v and p_b from v to u in G, such that neither p_f nor p_b contains an intermediate vertex that does not belong to C. A SCC is a maximal SCS.

The approach used here to maintain the decomposition of the graph into SCCs adapts a basic technique from [3], which works as follows: *Maintain n incomplete breadth first search (BFS) traversals, one starting at each vertex. Whenever an edge (u, v) is added, consider all vertices x, whose BFS has reached u, and restart these traversals by adding v to the queue of each such x. Since there are n BFS traversals and each edge that is inserted can be traversed by a traversal at most once, the amortized cost of insertion is $O(n)$.* Cicerone et al. [3] use this method to maintain the transitive closure of a graph under a series of insertions at an amortized cost of $O(n)$ per insert. This method, however, works well for deletions only in the case of acyclic digraphs; furthermore, it performs very poorly in situations where inserts and deletes are interleaved.

To adapt this technique for our situation (i.e., maintain the decomposition into SCCs), we make the following observations:

- Let it be that each of the vertices in the graph is arbitrarily assigned a unique integer value, called the *priority* of the vertex. (The priority of vertex x is denoted $pr(x)$.) We can then associate an integer with each SCS, which is the highest priority of all the vertices in the SCS. We also associate a SCS with each vertex x, denoted x_{SCS}, consisting of all vertices y such that the digraph contains a directed path from x to y and a directed path from y to x, neither of which pass through a vertex with priority greater than $pr(x)$.
- Consider a situation where we delete a given sequence of edges. Deleting an edge causes the SCC that contained the edge to be partitioned into one

or more SCCs. If we represent the SCCs created by this partitioning as children of the original SCC, we get an inheritance structure. Since the relation $xSCCy$ (read "x and y belong to the same SCC") is transitive, the inheritance structure defined by a sequence of edge deletions must be a forest. (If the graph was strongly connected to begin with, this structure is a tree. Note also that the inheritance structure can be represented as a forest of SCSs prior to any edge being deleted. We shall exploit these ideas in the following paragraphs.)

- If there are no deletions, it is easy to maintain the decomposition into SCCs using the technique in [3]. For each vertex v, we maintain the largest integer v_{max} such that there is a directed path from v to the vertex with priority v_{max} and vice versa. v is then designated as belonging to $SCC\, v_{max}$.
- To deal with deletable edges, we introduce the concept of *enodes* (or "edge-nodes"). Associated with a deletable edge directed from node u to node v is an enode (β, say). Instead of adding an edge from node u to node v, we add two directed edges: one from u to β and the other from β to v. Each deletable edge has an associated DTS; the enode β is assigned a priority of (n + the DTS of the edge from u to v), where n is the number of nodes in the original graph.
- When an edge is deleted, we need to ensure that the connectivity provided by the deleted edge is not being used anymore. To ensure this, the following constraint is imposed on the BFS traversals from each vertex: Consider a BFS traversal rooted at v; a vertex u is visited by this traversal only if $pr(v) > pr(u)$. Since an edge with a higher DTS corresponds to an enode with a higher priority, any path from a vertex v to a vertex u, discovered by a BFS rooted at v, cannot pass through any enode β such that $pr(\beta) > pr(v)$. Consequently, this path cannot be disconnected due to the deletion of the edge associated with any enode β that has a priority greater than $pr(v)$. This constraint does create a new problem: given vertices u, v, and w with $pr(u) > pr(v) > pr(w)$ such that v lies on every path from w to u, the BFS traversal from w will not find a path to u. Nonetheless, u and w must be recognized as belonging to the same SCC. To overcome this problem, we carry out two BFS traversals from each vertex—one following the forward arcs and one following the backward arcs. Thus with each vertex x (a vertex could be a node or an enode) we have an associated SCS consisting of all vertices y such that there is a path p_f from x to y following the directed arcs in the forward direction, and a path p_b from x to y following the directed arcs in the reverse direction, such that neither path has a a vertex with priority greater than $pr(x)$. Such an arrangement correctly finds all the SCCs, since the node x which has the highest priority of all nodes in the SCC, will reach all nodes in the SCC both forwards and backwards, without passing through any node with priority greater than $pr(x)$. This arrangement also yields the SCS, x_{SCS}, for each vertex x; x_{SCS} contains all vertices y which can be reached from x both forwards and backwards without passing through any vertex with priority greater than $pr(x)$.

We formally define the concepts introduced above:

- *enode:* When a deletable edge is to be inserted from node u to node v, we introduce a special vertex called an enode (β, say). β has one incoming edge, (u, β), and one outgoing edge, (β, v). Each vertex of the graph can therefore

be a node or an enode. When an edge is deleted, the associated enode is removed from the graph.

- *Forward* BFS: A breadth first traversal of the graph starting at a specified root, v, that traverses the outgoing edges from each vertex that is visited.
- *Backward* BFS: A breadth first traversal of the graph starting at a specified root, v, that traverses the edges coming into each vertex that is visited, i.e., follows the directed arcs in the reverse direction.
- SCS *associated with x:* Associated with each vertex x is a SCS, x_{SCS}, consisting of all vertices y such that there exist two directed paths, p_f from x to y, and p_b from y to x, such that neither p_f nor p_b contains any vertex with a priority greater than $pr(x)$. As defined earlier, a SCS that is maximal is a SCC.
- *Forest of* SCSs: Consider an enode β such that β is the vertex with the highest priority in its SCC (Denote this SCC as β_{SCC}. Let β be associated with the deletable edge (x, y) and v be some vertex in β_{SCS}. Let w be the vertex in β_{SCS} with the highest priority such that w_{SCS} contains v. If the edge (x, y) were to be deleted at this point, then v would belong to the SCC in which w is the vertex with the highest priority (denoted w_{SCC}). Since w_{SCC} directly inherits v from β_{SCC}, we designate w_{SCS} as a child of β_{SCS}. This parent-child relationship defines a forest of SCSs, denoted by F_{SCS}. It follows that the size of F_{SCS} is $O(m + n)$. A SCS that is a root of some tree in F_{SCS} is maximal and is therefore a SCC.

Our data structure keeps track of the following information with each vertex x:

1. Two boolean arrays: F_x, which stores all the nodes visited by the forward BFS from x, and B_x, which stores all the nodes visited by the backward BFS from x. If x is an enode, these arrays are of size n^2; if x is a node, these are of size n.
2. The parent of x_{SCS}, if any, in F_{SCS}.
3. Two lists: a list of nodes in the corresponding *SCS* x_{SCS} and the list of children of x_{SCS} in F_{SCS}.
4. If x is a node, then we keep two lists—these are L_x^f, which stores all vertices whose forward BFS has visited x, and L_x^b, which stores all vertices whose backward BFS has visited x.
5. Two BFS queues: Q_x^f and Q_x^b
6. An integer x_{max}, which is the highest priority of all vertices y such that both forward and backward traversals from y have visited x.

The data structure supports the following operations.

- $Insert(u, v, t)$: Insert a deletable edge from node u to node v, with $DTS\ t$.
  1. Create an enode β and add directed edges from u to β and β to v.
  2. For each enode α such that $pr(\alpha) \geq n + t$, increment the priority of α; for each vertex x such that $x_{max} \geq n + t$, increment x_{max}.
  3. For each enode α in L_u^f, such that $pr(\alpha) > n + t$, insert β in Q_α^f and restart that traversal.
  4. For each enode α in L_v^b, such that $pr(\alpha) > n + t$, insert β in Q_α^b and restart that traversal.
  5. Do a forward BFS and a backward BFS from β and enumerate the items in β_{SCS} by taking the intersection of the sets of nodes visited by the two traversals.
  6. Find the enode α with the smallest priority such that $\beta \in \alpha_{SCS}$ and

make β_{SCS} the child of α_{SCS} in F_{SCS}.

- *Insert*$(u, v, 0)$: Insert a nondeletable edge from node u to node v.
  1. For each vertex x in L_u^f, such that $pr(x) > pr(v)$, insert v into Q_x^f and restart the forward BFS from x.
  2. For each vertex x in L_v^b, such that $pr(x) > pr(u)$, insert u into Q_x^b and restart the backward BFS from x.
- *SCC* (u, v): Check if node u and node v belong to the same strongly connected component.
  Return $u_{max} = v_{max}$.
- *delete*(): Delete the edge with the highest DTS.
  Let w be the enode with the highest priority.
  For each child v_{SCS} of w_{SCS}
  For each vertex x in v_{SCS}
  $x_{max} = pr(v)$.

The algorithms for forward and backward BFSs from a vertex x are as follows.

FORWARD BFS(x)

While Q_x^f is not empty
y = dequeue(Q_x^f); $F_x[y] = 1$; visit(y, x); add x to L_y^f.
For each vertex z such that there is a directed edge
from y to z, $pr(z) < pr(x)$ and $F_x[z] = 0$,
enqueue(Q_x^f, z);
end for;
end while;

BACKWARD BFS(x)

While Q_x^b is not empty
y = dequeue(Q_x^b); $B_x[y] = 1$; visit(y, x); add x to L_y^b.
For each vertex z such that there is a directed edge
from z to y, $pr(z) < pr(x)$ and $B_x[z] = 0$,
enqueue(Q_x^b, z);
end for;
end while;

In both BFS algorithms, the operation visit(y, x) does the following.

VISIT(y, x)

If $F_x[y] = B_x[y] = 1$ then
$y_{max} = \max(y_{max}, pr(x))$
Let p_{SCS} denote the parent of y_{SCS}
If $pr(p) > pr(x)$ then
Make x_{SCS} the parent of y_{SCS}.
end if;
Update x_{SCS} to contain y.
end if;

THEOREM 4.1. *The data structure described above has the following characteristics:*

1. *It incurs an expense of $O(m^\star + n)$ for each deletable edge, where $m^\star$ is the maximum number of edges present in the graph at any time in the lifespan of the deletable edge.*
2. *It inserts nondeletable edges in $O(n)$ amortized time.*
3. *It correctly answers SCC queries in $O(1)$ time.*

Proof.

1. Consider the operation $Insert(u, v, t)$. Step 1 takes $O(1)$ time. Since the number of enodes in the graph at the time of insertion is no more than $m^\star$, step 2 can be completed in $O(m^\star)$ time. Since we have two queues for each enode and no vertex is inserted into a queue more than once, the number of insert operations performed on the BFS queues of any enode is bounded by $O(m^\star + n)$. Step 5 consists of two BFS traversals and can therefore be completed in $O(m^\star + n)$ time. Step 6 and the operation *delete*() spend a constant amount of time on each vertex, and thus the total time spent dealing with a deletable edge is bounded by $O(m^\star + n)$.
2. Consider the operation $Insert(u, v, 0)$. When each forward BFS reaches u, the edge (u, v) would cause v to be inserted into a queue. There is one forward queue for each node and one for each enode. The cost of inserting into the enode queues is accounted for when we calculate the cost of inserting deletable edges. Since the graph has n nodes, the cost of inserting into the node queues is $O(n)$. Likewise, the vertex u is inserted into a queue when a backward BFS reaches v, giving us a total cost of $O(n)$.
3. The complexity follows immediately from the operation. To establish correctness, we make the following claim.
   Claim. For any two vertices x and y, such that $pr(y) > pr(x)$, $x \in y_{SCS}$ if and only if y_{SCS} is an ancestor of x_{SCS} in F_{SCS}. Furthermore, if there are two vertices y and w, such that $x \in y_{SCS}$ and $x \in w_{SCS}$ and $pr(y) > pr(w)$, then y_{SCS} is an ancestor of w_{SCS}.
   Proof. For the if part, note that if y_{SCS} is the parent of x_{SCS}, both the BFS traversals from y must visit x, i.e., $x \in y_{SCS}$. Since the relation $xSCSy$ (read "x belongs to y_{SCS}") is transitive, it follows that for any ancestor y_{SCS} of x_{SCS}, $x \in y_{SCS}$. The only if part can be argued by induction on the length of the path from x_{SCS} to y_{SCS}. From the way we carry out the visit operation during the traversals, it is obvious that if z is the vertex with the smallest priority such that both the forward and the backward BFS from z visit x, then z_{SCS} is the parent of x_{SCS}. By the induction hypothesis, y_{SCS} must be an ancestor of z_{SCS} and therefore an ancestor of x_{SCS} as well. This argument also tells us that if we have $x \in y_{SCS}$ and $x \in w_{SCS}$ and $pr(y) > pr(w)$, then y_{SCS} is an ancestor of w_{SCS}. □
   For any vertex x, let β be the enode with the highest priority in the SCC containing x. If there are no deletions, for every vertex z in β_{SCS}, z_{max} will be set to $pr(\beta)$ when the traversals from β visit z; i.e., the SCC queries will be answered correctly for any pair of vertices in β_{SCS}. When we delete the edge associated with the enode β, this SCC will be partitioned, and x_{max} will be assigned the value $pr(y)$, where y_{SCS} is a child of β_{SCS}. From the description of the delete operation, we know that the property $x \in y_{SCS}$ holds, and from the above claim we know that y is the vertex with the highest priority that satisfies this property. It follows that, after the deletion, y is the vertex with the highest priority in the SCC containing x, and therefore all SCC queries will continue to be answered correctly. □

4.2. A $O(mn)$ clustering algorithm. The approach here is to first add to the constraint graph all the Type0 and Type1 constraints for edges with length greater than D_{min}. We then add Type1 constraints for edges of length less than D_{min} in decreasing order of edge lengths. Whenever we reach an unsatisfiable 2-SAT instance,

Type0 constraints are deleted in increasing order of edge lengths until satisfiability is restored. By keeping track of the length of the edges whose constraints created unsatisfiability and the length of the edges whose constraints were removed to restore satisfiability, we can obtain all the (d_0, d_1) pairs needed in Step 2 of Algorithm Cluster.

Once again, let $l_1, l_2, \ldots, l_q$ be the q distinct edge lengths in the cluster graph sorted in nonincreasing order, and let S_i denote the set of all edges of length l_i in the cluster graph. To simplify the presentation, we define $l_0 = \infty$, $l_{q+1} = 0$, and S_0 and S_{q+1} as the associated (empty) sets. Let $D_{min} = l_p$, $1 \leq p \leq q$.

ALGORITHM CLUSTER2.

1. Insert all the Type1 constraints for edges with length greater than D_{min} into the constraint graph as undeletable edges.
2. Insert all the Type0 constraints for edges with length greater than D_{min} as follows:
   $DTS = 1$;
   For $i = 1$ to p-1
   For each edge e_{jk} in S_i that does not complete an even cycle in Kruskal's algorithm,
   $Insert(v_j, u_k, DTS++)$; $Insert(v_k, u_j, DTS++)$
   end for;
   end for;
3. $j = p$-1;
   For $i = p$-1 down to 1
   While (constraint graph is satisfiable)
   $j = j + 1$; Insert all Type1 constraints for edges in S_j;
   end while;
   Record (l_{i+1}, l_j) as a (d_0, d_1) pair.
   Remove Type0 constraints for all edges in S_i.
   end for;

THEOREM 4.2. *Algorithm Cluster2 correctly computes Step 2 of Algorithm Cluster in $O(mn)$ time.*

Proof. The argument for correctness is similar to the one for Theorem 3.3. If d_0 is l_{i+1}, the constraint graph contains Type0 constraints for all edges in $\bigcup_{k=1}^{i} S_k$. Since the constraint graph became unsatisfiable when Type1 constraints for edges in S_j were added, d_1 should be set to l_j to avoid these constraints.

There are $O(m)$ nondeletable edges inserted in step 1, each taking $O(n)$ time, and $O(n)$ deletable edges inserted in step 2, each requiring $O(m)$ time. Therefore both these steps can be done in $O(mn)$ time. In step 3, inside the while loop, at most $O(m)$ edges are inserted at an amortized cost of $O(n)$ per edge; the edges inserted in step 2 are deleted outside the while loop, but the cost of these deletions is already accounted for. Thus the entire process takes $O(mn)$ steps. □

5. Conclusion. We have discussed algorithms that solve the minimum sum-of-diameters clustering problem in $O(n^3)$ and $O(mn)$ time, respectively. In practice, the $O(n^3)$ algorithm may be better due to the fact that the only data structure used is an array. It would be interesting to determine experimentally at what value of m the second algorithm becomes more efficient. It is possible to use the ideas described in this paper to obtain an $O(qm)$ algorithm for minimum sum-of-diameters clustering, where q is the number of distinct edge lengths. This would actually give us better performance if q is $o(n)$. Whether there is a $o(mn)$ algorithm for the general

case remains an open question. Another interesting question would be to determine the relative complexities of maintaining the transitive closure vs. maintaining the decomposition into SCCs for general digraphs in the absence of lookahead.

Acknowledgments. The author would like to thank Pierre Hansen, Venkatesh Raman, and S. N. Maheshwari for several useful discussions and references, and the anonymous referees for several corrections.

REFERENCES

[1] B. Aspvall, M. F. Plass, and R. E. Tarjan, *A linear-time algorithm for testing the truth of certain quantified Boolean formulas*, Inform. Process. Lett., 8 (1979), pp. 121–123.

[2] P. Brucker, *On the complexity of clustering problems, optimization and operations research*, in Optimization and Operations Research, Lecture Notes in Econom. and Math. Systems 157, Springer-Verlag, Berlin, New York, 1978, pp. 45–54.

[3] S. Cicerone, D. Frigioni, U. Nanni, and F. Pugliese, *A uniform approach to semi-dynamic problems on digraphs*, Theoret. Comput. Sci., 203 (1998), pp. 69–90.

[4] R. M. Cormack, *A review of classification*, J. Roy. Statist. Soc. Ser. A, 134 (1971), pp. 321–367.

[5] M. Delattre and P. Hansen, *Bicriterion cluster analysis*, IEEE Trans. Pattern Analysis Machine Intelligence, PAMI-2(4), 1980, pp. 277–291.

[6] C. Demetrescu and G. F. Italiano, *Fully dynamic transitive closure: Breaking through the $O(n^2)$ barrier*, in Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science, Redondo Beach, CA, 2000, pp. 381–389.

[7] D. Dobkin and S. Suri, *Maintenance of geometric extrema*, J. ACM, 38 (1991), pp. 275–298.

[8] S. R. Doddi, M. V. Marathe, S. S. Ravi, D. S. Taylor, and P. Widmayer, *Approximation algorithms for clustering to minimize the sum of diameters*, Nordic J. Comput., 7 (2000), pp. 185–203.

[9] P. Hansen and B. Jaumard, *Minimum sum of diameters clustering*, J. Classification, 4 (1987), pp. 215–226.

[10] P. Hansen and B. Jaumard, *Cluster analysis and mathematical programming*, Math. Programming, 79 (1997), pp. 191–215.

[11] S. Khanna, R. Motwani, and R. H. Wilson, *On certificates and lookahead on dynamic graph problems*, in Proceedings of the Seventh ACM-SIAM Symposium on Discrete Algorithms, Atlanta, GA, 1996, pp. 222–231.

[12] V. King, *Fully dynamic algorithms for maintaining all-pairs shortest paths and transitive closure in digraphs*, in Proceedings of the 40th IEEE Symposium on Foundations of Computer Science, 1999, New York, pp. 81–91.

[13] V. King and G. Sagert, *Fully dynamic algorithm for maintaining the transitive closure*, in Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, Atlanta, GA, 1999, pp. 492–498.

[14] C. Monma and S. Suri, *Partitioning points and graphs to minimize the maximum or the sum of diameters*, in Proceedings of the 6th International Conference on Theory and Applications of Graphs, Kalamazoo, MI, 1989, pp. 899–912.

[15] M. R. Rao, *Cluster analysis and mathematical programming*, J. Amer. Statist. Assoc., 66 (1971), pp. 622–626.

SIAM J. DISCRETE MATH.
Vol. 18, No. 2, pp. 287–312

TWO EDGE-DISJOINT HOP-CONSTRAINED PATHS AND POLYHEDRA*

DAVID HUYGENS†, ALI RIDHA MAHJOUB‡, AND PIERRE PESNEAU‡

Abstract. Given a graph G with distinguished nodes s and t, a cost on each edge of G, and a fixed integer $L \geq 2$, the two edge-disjoint hop-constrained paths problem is to find a minimum cost subgraph such that between s and t there exist at least two edge-disjoint paths of length at most L. In this paper, we consider that problem from a polyhedral point of view. We give an integer programming formulation for the problem when $L = 2, 3$. An extension of this result to the more general case where the number of required paths is arbitrary and $L = 2, 3$ is also given. We discuss the associated polytope, $P(G, L)$, for $L = 2, 3$. In particular, we show in this case that the linear relaxation of $P(G, L)$, $Q(G, L)$, given by the trivial, the st-cut, and the so-called L-path-cut inequalities, is integral. As a consequence, we obtain a polynomial time cutting plane algorithm for the problem when $L = 2, 3$. We also give necessary and sufficient conditions for these inequalities to define facets of $P(G, L)$ for $L \geq 2$ when G is complete. We finally investigate the dominant of $P(G, L)$ and give a complete description of this polyhedron for $L \geq 2$ when $P(G, L) = Q(G, L)$.

Key words. survivable network, edge-disjoint paths, hop-constraints, polyhedron, facet

AMS subject classifications. 90B10, 90C27, 90C57

DOI. 10.1137/S0895480102419445

1. Introduction.

Given a graph $G = (N, E)$, with distinguished nodes s and t, and a fixed integer $L \geq 2$, an *L-st-path* in G is a path between s and t of length at most L, where the length of a path is the number of its edges. Given a function $c : E \to \mathbb{R}$ which associates a cost $c(e)$ with each edge $e \in E$, the *two edge-disjoint hop-constrained paths problem* (THPP) is to find a minimum cost subgraph such that between s and t there exist at least two edge-disjoint L-st-paths.

The THPP arises in the design of reliable communication networks. In fact, with the introduction of fiber optic technology in telecommunications, designing a minimum cost survivable network has become a major objective in the telecommunications industry. Survivable networks have to satisfy some connectivity requirements. As pointed out in [28], 2-edge connected networks have been shown to be cost effective and to provide an adequate level of survivability. In such networks, there are at least two edge-disjoint paths between each pair of nodes. So, if a link fails, it is always possible to reroute the traffic between two terminals along the second path.

However, this requirement is often insufficient regarding the reliability of a telecommunications network. In fact, the alternative paths could be too long to guarantee an effective routing. In data networks, such as the Internet, the elongation of the route of the information could cause a strong loss in the transfer speed. For other networks, the signal itself could be degraded by a longer routing. In such cases, the L-path requirement guarantees exactly the needed quality for the alternative routes.

*Received by the editors December 12, 2002; accepted for publication (in revised form) January 19, 2004; published electronically October 8, 2004. This research was partially supported by FRIA fellowship and a cooperation agreement CGRI–FNRS–CNRS, project 03/005.
http://www.siam.org/journals/sidma/18-2/41944.html

†Department of Computer Science, Université Libre de Bruxelles, Boulevard du Triomphe CP 210/01, B-1050 Ixelles, Belgium (dhuygens@smg.ulb.ac.be).

‡LIMOS, CNRS, UMR 6158, Université Blaise Pascal Clermont-Ferrand II, Complexe Scientifique des Cézeaux, F-63177 Aubière Cedex, France (Ridha.Mahjoub@math.univ-bpclermont.fr). The third author's current address: IAG/POMS, Université Catholique de Louvain, Place des Doyens, 1, B-1348 Louvain-la-Neuve, Belgium (pesneau@poms.ucl.ac.be).

The THPP can also be seen as a special case of the more general problem when more than one pair of terminals is considered. This is the case, for instance, when several commodities have to be routed in the network. Thus an efficient algorithm for solving the THPP would be useful to solve (or produce upper bounds for) this more general problem.

It is clear that an optimal solution of the THPP can be computed in polynomial time by enumerating all the L-st-paths. However, in a complete graph $G = (N, E)$ with $|N| = n$, there are $\mathcal{O}(n^{L-1})$ L-st-paths, which can also be enumeratively generated in $\mathcal{O}(n^{L-1})$ time. For every pair of such paths, one has to verify their edge-disjunction, which requires $\mathcal{O}(L^2)$ comparisons. Consequently, the whole enumerative algorithm for the THPP runs in $\mathcal{O}(L^2 n^{2(L-1)})$ time. Clearly, such a method is far from being applicable in practice. One of the principal aims of this paper is to devise a more efficient algorithm for the THPP. This algorithm, which will be a cutting plane method, will be based on a complete description of the associated polytope by a system of linear inequalities.

Given a graph $G = (N, E)$ and an edge subset $F \subseteq E$, the 0-1 vector $x^F \in \mathbb{R}^E$, such that $x^F(e) = 1$ if $e \in F$ and $x^F(e) = 0$ otherwise, is called the *incidence vector* of F. For $L \geq 2$, the *convex hull* of the incidence vectors of the solutions of the THPP on G, denoted by $P(G, L)$, will be called the THPP *polytope*. Given a vector $w \in \mathbb{R}^E$ and an edge subset $F \subseteq E$, we let $w(F) = \sum_{e \in F} w(e)$. If $W \subset N$ is a node subset of G, then the set of edges that have only one node in W is called a *cut* and is denoted by $\delta(W)$. We will write $\delta(v)$ for $\delta(\{v\})$. A cut $\delta(W)$ such that $s \in W$ and $t \in V \backslash W$ will be called an *st-cut*.

If x^F is the incidence vector of the edge set F of a solution of the THPP, then clearly x^F satisfies the inequalities

$$x(\delta(W)) \geq 2 \quad \text{for all } st\text{-cut } \delta(W), \tag{1.1}$$

$$1 \geq x(e) \geq 0 \quad \text{for all } e \in E. \tag{1.2}$$

Inequalities (1.1) will be called *st-cut inequalities* and inequalities (1.2) *trivial inequalities*.

In [12], Dahl considers the problem of finding a minimum cost path between two given terminal nodes s and t of length at most L. He describes a class of valid inequalities for the problem and gives a complete description of the associated L-path polyhedron when $L \leq 3$. In particular, he introduces a class of valid inequalities as follows.

Let $V_0, V_1, \ldots, V_{L+1}$ be a partition of N such that $s \in V_0$, $t \in V_{L+1}$, and $V_i \neq \emptyset$ for all $i = 1, \ldots, L$. Let T be the set of edges $e = uv$, where $u \in V_i$, $v \in V_j$, and $|i - j| > 1$. Then the inequality

$$x(T) \geq 1$$

is valid for the L-path polyhedron.

Using the same partition, this inequality can be generalized in a straightforward way to the THPP polytope as

$$x(T) \geq 2. \tag{1.3}$$

The set T is called an *L-path-cut* (or *L-star*), and a constraint of type (1.3) is called an *L-path-cut (or L-star) inequality*.

Let $Q(G, L)$ be the solution set of the system given by inequalities (1.1)–(1.3). In this paper, we show that inequalities (1.1)–(1.3), together with the integrality constraints, give an integer programming formulation of the THPP and of its generalization when more than two edge-disjoint L-st-paths are required for $L = 2, 3$. We then discuss the THPP polytope, $P(G, L)$, and show that $P(G, L) = Q(G, L)$ when $L = 2, 3$ for any graph. This yields a polynomial time cutting plane algorithm for the THPP in this case. We also give necessary and sufficient conditions for inequalities (1.1)–(1.3) to define facets for any $L \geq 2$ when the graph is complete. We finally investigate the dominant of $P(G, L)$, for which we give a complete description for any $L \geq 2$ when $P(G, L) = Q(G, L)$. As a consequence, we obtain the dominant of $P(G, L)$ when $L = 2, 3$.

Despite its interesting applications, the THPP has, to the best of our knowledge, never been studied before. There has been, however, a considerable amount of research on many related problems. In [14], Dahl and Johannessen consider the 2-path network design problem, which consists of finding a minimum cost subgraph connecting each pair of terminal nodes by at least one path of length at most 2. This problem is NP-hard. Dahl and Johannessen give an integer programming formulation for the problem and describe some classes of valid inequalities. Using these, they devise a cutting plane algorithm and present some computational results.

The closely related problem of finding a minimum cost spanning tree with hop-constraints is considered in [19], [20], [23]. Here, the hop-constraints limit the number of links between the root and any terminal in the network to a positive integer H. This problem is NP-complete even for $H = 2$. Gouveia [19] gives a multicommodity flow formulation for that problem and discusses a Lagrangian relaxation improving the LP bound. Gouveia [20] and Gouveia and Requejo [23] propose more efficient Lagrangian-based schemes for the problem and its Steiner version. Dahl [11] studies the problem for $H = 2$ from a polyhedral point of view and gives a complete description of the associated polytope when the graph is a wheel. Gouveia and Janssen [21] discuss a generalized problem where connectivity requirements are considered. They formulate the problem as a directed multicommodity flow model and use Lagrangian relaxation together with subgradient optimization to derive lower bounds. Gouveia and Magnanti [22] consider the problem that consists in finding a minimum spanning tree such that the number of edges in the tree between any pair of nodes is limited to a given bound (diameter). They present directed and undirected multicommodity formulations along with some computational experiments. Further hop-constrained survivable network design problems are studied in [1], [4], [5], [33], [34], [37].

In the framework of the minimum cost spanning tree problem with hop-constraints, Dahl and Gouveia [13] consider the hop-constrained path problem, that is, the problem of finding between two distinguished nodes s and t a minimum cost path with no more than K edges when K is fixed. They describe various classes of valid inequalities and show that some of these inequalities are sufficient to completely describe the associated polytope when $K \leq 3$. Then they discuss some applications to the hop-constrained minimum spanning tree problem. In [10], Coullard, Gamble, and Liu investigate the structure of the polyhedron associated with the st-walks of length K of a graph, where a walk is a path that may go through the same node more than once. They present an extended formulation of the problem, and, using projection, they give a linear description of the associated polyhedron. They also discuss classes of facets of that polyhedron.

Itai, Perl, and Shiloach [30] study the complexity of several variants of the maximum disjoint hop-constrained paths problem. This consists in finding the maximum

number of disjoint paths between two nodes s and t of length equal to (or bounded by) K, where K is a positive integer. They show that the problem is NP-complete for $K \geq 5$ and polynomially solvable for some of the variants for $K \leq 4$. In particular, they devise a polynomial time algorithm for the problem when the paths must be node- (resp., edge-) disjoint and of length bounded by K, with $K \leq 4$ (resp., $K \leq 3$). Bley [7] addresses approximation and computational issues for the edge- (node-) disjoint hop-constrained paths problem. In particular, he shows that the problem of computing the maximum number of edge-disjoint paths between two given nodes of length equal to 3 is polynomial. This answers an open question in [30]. In [35], Li, McCormick, and Simchi-Levi study the problem of finding K disjoint paths of minimum total cost between two distinguished nodes s and t, where each edge of the graph has K different costs and the jth edge-cost is associated with the jth path. They show that all the variants of the problem, when the graph is directed or undirected and the paths are edge- or node-disjoint, are NP-complete, even when $K = 2$.

Besides hop-constraints, another reliability condition, which is used in order to limit the length of the routing, requires that each link of the network belongs to a ring (cycle) of bounded length. In [16], Fortz, Labbé and Maffioli consider the 2-node connected subgraph problem with bounded rings. This problem consists in finding a minimum cost 2-node connected subgraph (N, F) such that each edge of F belongs to a cycle of length at most L. They describe several classes of facet defining inequalities for the associated polytope and devise a branch-and-cut algorithm for the problem. In [17], Fortz et al. study the edge version of that problem. They give an integer programming formulation of the problem in the space of the natural design variables and describe different classes of valid inequalities. They study the separation problem for these inequalities and discuss a branch-and-cut algorithm.

The related 2-edge connected subgraph problem and its associated polytope have also been the subject of extensive research in the past years. Grötschel and Monma [25] and Grötschel, Monma, and Stoer [26], [27] study the 2-edge connected subgraph problem within the framework of a general survivable model. They discuss the polyhedral aspects and devise cutting plane algorithms. In [36], Mahjoub shows that if G is series-parallel, then the 2-edge connected subgraph polytope is completely described by the trivial and the cut inequalities. This has been generalized by Baïou and Mahjoub [2] for the Steiner 2-edge connected subgraph polytope and by Didi Biha and Mahjoub [6] for the Steiner k-edge connected subgraph polytope for k even. In [3], Barahona and Mahjoub characterize this polytope for the class of Halin graphs. In [15], Fonlupt and Mahjoub study the fractional extreme points of the linear relaxation of the 2-edge connected subgraph polytope. They introduce an ordering on these extreme points and characterize the minimal extreme points with respect to that ordering. As a consequence, they obtain a characterization of the graph for which the linear relaxation of that polytope is integral. Kerivin, Mahjoub, and Nocq [32] describe a general class of valid inequalities for the 2-edge connected subgraph polytope, which generalizes the so-called F-partition inequalities [36], and introduce a branch-and-cut algorithm for the problem based on these inequalities, the trivial and the cut inequalities. Further work on the 2-edge and 2-node connected subgraph problems can be found in [9], [18], [28], [31].

The paper is organized as follows. In the next section, we give an integer programming formulation of the THPP and its generalization when the number of paths is arbitrary for $L \leq 3$. In section 3, we study the THPP polytope when $L = 2, 3$ and give our main result. In section 4, we study some structural properties of the facet defining inequalities of $P(G, L)$, which are used in section 5 for proving our main

result. In section 6, we describe necessary and sufficient conditions for the inequalities (1.1)–(1.3) to be facet defining. In section 7, we discuss the dominant of $P(G, L)$, and, in section 8, we give some concluding remarks.

The rest of this section is devoted to more definitions and notation. We assume the reader has familiarity with graphs and polyhedra. For specific details, the reader is referred to [8] and [38]. The graphs that we consider are finite, undirected, loopless, and may have multiple edges. A graph is denoted by $G = (N, E)$, where N is the *node set* and E is the *edge set*. Given W, W' two disjoint subsets of N, $[W, W']$ will denote the set of edges of G having one endnode in W and the other one in W'. If $W = \{v\}$, we will write $[v, W']$ instead of $[\{v\}, W']$. If G is a graph and e is an edge of E, then $G - e$ will denote the graph obtained from G by removing e. A *path* P of G is an alternate sequence of nodes and edges $(u_1, e_1, u_2, e_2, \ldots, u_{q-1}, e_{q-1}, u_q)$, where $e_i \in [u_i, u_{i+1}]$ for $i = 1, \ldots, q-1$. We will denote a path P by either its node sequence $(u_1, \ldots, u_q)$ or its edge sequence $(e_1, \ldots, e_{q-1})$.

2. Formulation for $L = 2, 3$. In this section, we show that the st-cut, L-path-cut, and trivial inequalities, together with integrality constraints, suffice to formulate the THPP as a 0-1 linear program when $L = 2, 3$. To this end, we first give a lemma.

LEMMA 2.1. *Let $G = (N, E)$ be a graph, s, t be two nodes of N, and $L \in \{2, 3\}$. Suppose that there do not exist k edge-disjoint L-st-paths in G, with $k \geq 2$. Then there exists a set of at most $k - 1$ edges that intersects every L-st-path.*

Proof. We first show the statement for $L = 3$. The proof uses ideas from [30] and [17]. Consider the capacitated directed graph $D = (N', A)$ obtained from G in the following way. The set N' consists of a copy s', t' of s, t and two copies N_1, N_2 of $N \backslash \{s, t\}$. For $u \in N \backslash \{s, t\}$, let u_1 and u_2 be the corresponding nodes in N_1 and N_2, respectively. To each edge $e \in [s, u]$, with $u \in N \backslash \{s, t\}$, we associate an arc e' from s' to u_1 of capacity 1. To each edge $e \in [v, t]$, with $v \in N \backslash \{s, t\}$, we associate an arc e' from v_2 to t' of capacity 1. For an edge $e \in [u, v]$, with $u, v \in N \backslash \{s, t\}$, we consider two arcs, one from u_1 to v_2 and the other from v_1 to u_2, both of capacity 1. Finally, we consider in D an arc from s' to t' of capacity 1 for every edge in $[s, t]$ and an arc from each node of N_1 to its peer in N_2 with infinite capacity (see Figure 1 for an illustration). Note that multiple edges in G yield multiple arcs in D. Observe that there is a one-to-one correspondence between the 3-st-paths in G and the directed $s't'$-paths in D.

Now consider a maximum flow $\phi \in \mathbb{R}_+^A$ from s' to t' in D. As the capacities of D are integer, ϕ can be supposed to be integer. Hence the flow value of each arc of capacity 1 is either 0 or 1. We claim that ϕ can be chosen so that no two arcs (u_1, v_2) and (v_1, u_2), corresponding to the same edge uv in G, have a positive value. Indeed, suppose that $\phi(u_1, v_2) = 1$ and $\phi(v_1, u_2) = 1$. Let $\phi' \in \mathbb{R}_+^A$ be the flow given by

$$\phi'(e) = \begin{cases} \phi(e) + 1 & \text{if } e \in \{(u_1, u_2), (v_1, v_2)\}, \\ 0 & \text{if } e \in \{(u_1, v_2), (v_1, u_2)\}, \\ \phi(e) & \text{otherwise.} \end{cases}$$

As (u_1, u_2) and (v_1, v_2) have infinite capacity and the flow going into u_2 and v_2 has not changed, ϕ' is still feasible. Moreover, ϕ' has the same value as ϕ.

As a consequence, an $s't'$-flow of value q in D corresponds to q edge-disjoint 3-st-paths in G. Since there do not exist, in G, k edge-disjoint 3-st-paths, the maximum flow in D is of value at most $k - 1$. Hence a minimum st-cut in D is of value at most $k - 1$ as well. Observe that such a cut does not contain arcs with infinite capacity. Hence, a minimum cut corresponds to a set of at most $k - 1$ edges that intersects all the 3-st-paths of G, and the proof for $L = 3$ is complete.

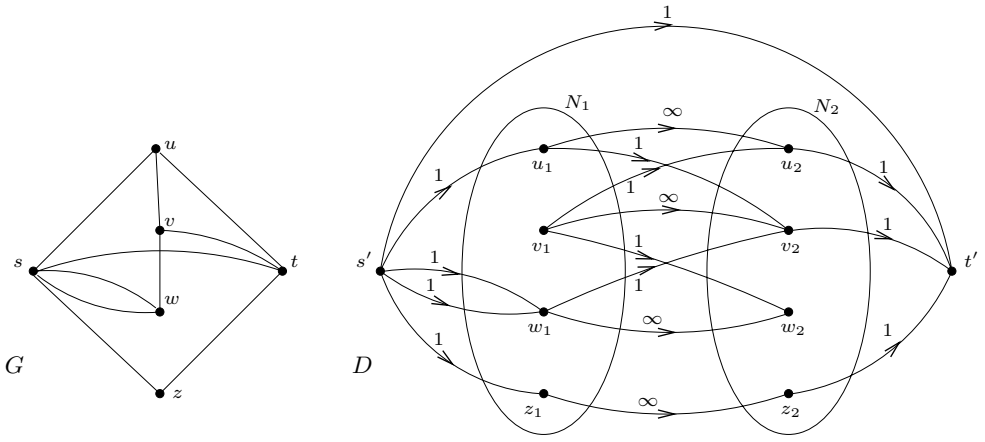


FIG. 1.

If $L = 2$, then we can similarly show the statement by considering the digraph $D = (N', A)$, where N' is a copy of N and to every edge $e \in [s, u]$ (resp., $[u, t]$), where $u \in N \backslash \{s, t\}$, corresponds an arc e' from s' to u' (resp., u' to t') of capacity 1 in D. Here u' is the copy of u in N' for every $u \in N$. □

THEOREM 2.2. *Let $G = (N, E)$ be a graph and $L \in \{2, 3\}$. Then the THPP is equivalent to the integer program*

$$\text{Min}\{cx;\, x \in Q(G, L),\, x \in \{0, 1\}^E\}.$$

Proof. To prove the theorem, it is sufficient to show that every 0-1 solution x of $Q(G, L)$ induces a solution of the THPP. Let us assume the contrary. Suppose that x does not induce a solution of the THPP but satisfies the st-cut and trivial constraints. We will show that x necessarily violates at least one of the L-path-cut constraints $x(T) \geq 2$. Let G_x be the subgraph induced by x. As x is not a solution of the problem, G_x does not contain two edge-disjoint L-st-paths. As $L \in \{2, 3\}$, it follows, by Lemma 2.1, that there exists at most one edge in G_x that intersects every L-st-path. Consider the graph $\tilde{G}_x$ obtained from G_x by deleting this edge. Obviously, $\tilde{G}_x$ does not contain any L-st-path.

We claim that $\tilde{G}_x$ contains at least one st-path of length at least $L + 1$. In fact, as x is a 0-1 solution and satisfies the st-cut inequalities, G_x contains at least two edge-disjoint st-paths. Since at most one edge was removed from G_x, at least one path remains between s and t in $\tilde{G}_x$. However, since $\tilde{G}_x$ does not contain an L-st-path, that path must be of length at least $L + 1$.

Now consider the partition $V_0, \ldots, V_{L+1}$ of N, with $V_0 = \{s\}$, V_i the set of nodes at distance i from s in $\tilde{G}_x$ for $i = 1, \ldots, L$, and $V_{L+1} = N \setminus (\bigcup_{i=0}^{L} V_i)$, where the distance between two nodes is the length of a shortest path between these nodes. Since there does not exist an L-st-path in $\tilde{G}_x$, it is clear that $t \in V_{L+1}$. Moreover, as, by the claim above, $\tilde{G}_x$ contains an st-path of length at least $L+1$, the sets $V_1, \ldots, V_L$ are nonempty. Furthermore, no edge of $\tilde{G}_x$ is a chord of the partition (that is, an edge between two sets V_i and V_j, where $|i - j| > 1$). In fact, suppose that there exists an edge $e = v_i v_j \in [V_i, V_j]$ with $|i - j| > 1$ and $i < j$. Therefore v_j is at distance $i + 1$ from s, a contradiction.

Thus, the edge deleted from G_x is the only edge that may be a chord of the partition in G_x. In consequence, if T is the set of chords of the partition in G, then

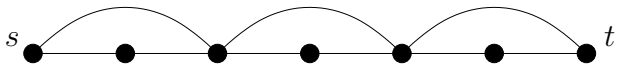


FIG. 2.

$x(T) \leq 1$. But this implies that the corresponding L-path-cut inequality is violated by x. □

If $L \geq 4$, inequalities (1.1)–(1.3), together with the integrality constraints $x(e) \in \{0,1\}$ for all $e \in E$, do not suffice to formulate the THPP as an integer program. Indeed, suppose that $L = 4$ and consider the graph shown in Figure 2. It is not hard to see that the solution induced by this graph satisfies inequalities (1.1)–(1.3), whereas the graph itself is not a feasible solution of the THPP.

However, using Lemma 2.1, Theorem 2.2 can be easily extended to the case where $L \in \{2,3\}$ and the number k of required L-st-paths is arbitrary. In other words, the problem in this case is equivalent to the integer program

$$\text{Min } \{cx;\, x \in Q_k(G,L),\, x \in \{0,1\}^E\}, \tag{2.1}$$

where $Q_k(G,L)$ is obtained from $Q(G,L)$ $(= Q_2(G,L))$ by replacing the right-hand side of inequalities (1.1) and (1.3) by k.

The separation problem for a system of inequalities consists in verifying whether a given solution $x^* \in \mathbb{R}^E$ satisfies the system and, if not, in finding an inequality of the system that is violated by x^*. The separation problem for inequalities (1.1) can be solved in polynomial time using any polynomial max-flow algorithm (see, e.g., [29]). Inequalities (1.3) can also be separated in polynomial time when $L \leq 3$. In fact, in this case, it is not hard to see that the separation problem reduces to finding a minimum weight edge subset that intersects all L-st-paths. Recently, Fortz et al. [17] have shown that this problem reduces to a max-flow problem (as described in the proof of Lemma 2.1) and hence can be solved in polynomial time.

Thus, by the ellipsoid method [24], problem (2.1) can be solved in polynomial time. It would then be interesting to characterize the graphs for which $Q_k(G,L)$ is integral. In what follows, we will show that for $k = 2$, that is, when (2.1) corresponds to the THPP, $Q_k(G,L)$ is integral for any graph for $L = 2,3$.

3. THPP polytope for $L = 2,3$. We first state our main result.

THEOREM 3.1. *$P(G,L) = Q(G,L)$ if $L = 2,3$.*

The proof of this theorem will be given in section 5. In what follows, we shall discuss the dimension of $P(G,L)$ and study some properties of its facial structure. Let $G = (N,E)$ be a graph. An edge $e \in E$ will be called *L-st-essential* if e belongs to an st-cut of cardinality 2 or an L-path-cut of cardinality 2. Let E^* denote the set of L-st-essential edges. Thus, $P(G-e,L) = \emptyset$ for all $e \in E^*$. The following theorem, which is easily seen to be true, characterizes the dimension of the polytope $P(G,L)$.

THEOREM 3.2. *If $L = 2,3$, $\dim(P(G,L)) = |E| - |E^*|$.*

COROLLARY 3.3. *If $G = (N,E)$ is complete with $|N| \geq 4$ and $L = 2,3$, then $P(G,L)$ is full dimensional.*

The following theorem gives a procedure for obtaining a linear description of the THPP polytope for a subgraph of G from that corresponding to G.

THEOREM 3.4. *Let $G = (N,E)$ be a graph, s,t be two nodes of N, and $L \geq 2$ be an integer. Let e be an edge of E. Let $G' = (N,E')$ be the graph obtained from G by deleting e. Then a linear system describing $P(G',L)$ can be obtained from a system describing $P(G,L)$ by removing the variables corresponding to e.*

Proof. The proof is easy. ∎

In the following, we will suppose that $G = (N, E)$ is complete with $|N| \geq 4$ and $L = 2, 3$. Hence, by Theorem 3.2, $P(G, L)$ is full dimensional. If $G = (N, E)$ is not complete, then a description of $P(G, L)$ can be obtained from that of $P(\overline{G}, L)$, by repeatedly using Theorem 3.4. Here $\overline{G}$ is the complete graph obtained from G by adding the missing edges. Moreover, it is clear that the problem can be reduced to that case by associating a big cost with the missing edges in the graph.

Let

$$T(G) = \{F \subseteq E \,|\, (N, F) \text{ is a solution of the THPP}\}.$$

Given an inequality $ax \geq \alpha$ that defines a facet of $P(G, L)$, we let

$$\tau_a = \{F \in T(G) \,|\, ax^F = \alpha\}.$$

In what follows, we will consider $a(e)$ as a weight on e. Hence, any solution S of τ_a will have a weight $a(S)$ equal to α and any solution of $T(G)$ a weight $\geq \alpha$.

LEMMA 3.5. (i) *Let $ax \geq \alpha$ be a facet defining inequality of $P(G, L)$ different from the trivial inequalities. Then for every edge $e \in E$, there exists an edge subset in τ_a that contains e and another one that does not.*

(ii) *Let $ax \geq \alpha$ be a facet defining inequality of $P(G, L)$ different from the st-cut inequalities. Then, for every st-cut $\delta(W)$, there exists an edge subset in τ_a containing at least three edges of $\delta(W)$.*

Proof. The proof is easy. ∎

Lemma 3.5 will be frequently used in what follows. At times we will use it without referring to it explicitly.

LEMMA 3.6. *Let $ax \geq \alpha$ be a facet defining inequality of $P(G, L)$ different from a trivial inequality. Then $a(e) \geq 0$ for all $e \in E$ and $\alpha > 0$.*

Proof. Assume, on the contrary, that there is an edge $e \in E$ such that $a(e) < 0$. Since $ax \geq \alpha$ is different from $-x(e) \geq -1$, by Lemma 3.5(i), there must exist a solution S of τ_a that does not contain e. As $S' = S \cup \{e\}$ still belongs to $T(G)$, this yields $\alpha \leq ax^{S'} = ax^S + a(e) < ax^S = \alpha$, a contradiction. Thus, $a(e) \geq 0$ for all $e \in E$. Since $ax \geq \alpha$ defines a facet of $P(G, L)$, there must exist at least one edge, say f, with $a(f) > 0$. Now, as $ax \geq \alpha$ is different from the inequality $x_f \geq 0$, there is an edge set of τ_a containing f. This implies that $\alpha > 0$. ∎

The following lemma shows that parallel edges in G have the same coefficient in every nontrivial facet defining inequality of $P(G, L)$ for $L = 2, 3$.

LEMMA 3.7. *Let $ax \geq \alpha$ be a facet defining inequality of $P(G, L)$ different from the trivial inequalities. Let $[u, v] = \{e_1, e_2, \ldots, e_p\}$ be the set of the parallel edges between two nodes u and v in G. Then $a(e_i) = a(e_j)$ for $i, j = 1, \ldots, p$.*

Proof. We will show the result for $L = 3$. The proof for $L = 2$ is similar. First we show that all edges in $[u, v]$ have the same coefficient, except possibly one, that may have a smaller coefficient. Indeed, if there are three edges $e_1, e_2, e_3 \in [u, v]$ such that $a(e_1) > a(e_2) \geq a(e_3)$, then there cannot exist an edge subset of τ_a containing e_1. Otherwise, one could replace e_1 by either e_2 or e_3 and get a solution which violates $ax \geq \alpha$, a contradiction. Now, suppose that there are two edges $e_1, e_2 \in [u, v]$ such that $a(e_1) > a(e_2)$. By the remark above, it follows that $a(e) = a(e_1)$ for all $e \in [u, v] \backslash \{e_1, e_2\}$.

Claim 1. Let S be a solution of τ_a.

(i) If S contains e_1, then it must contain e_2.

(ii) If S does not contain e_2, then it does not intersect $[u, v]$.

Proof. (i) If $e_1 \in S$ and $e_2 \notin S$, then $S' = (S\backslash\{e_1\}) \cup \{e_2\}$ is in $T(G)$. As $ax^{S'} < \alpha$, we have a contradiction.

(ii) Assume the contrary. Then we may suppose that S contains an edge e_i, $i \in \{1, \ldots, p\}\backslash\{2\}$, and $e_2 \notin S$. Since $a(e_i) > a(e_2)$, this is impossible by the argument given above. □

Now, since $ax \geq \alpha$ is different from a trivial inequality, by Lemma 3.5(i), there is an edge set of τ_a, say S_1, containing e_1. Let L_1 be a 3-st-path of S_1 that contains e_1. By Claim 1(i), it follows that e_2 belongs to the second 3-st-path of S_1, say L_2. Note that $L_1 \cap L_2 = \emptyset$. It is not hard to see that L_1 and L_2 go through e_1 and e_2, respectively, in the same direction starting from s. If not, one would have one path of the form (s, u, v, t) and the other one of the form (s, v, u, t). But then the edges e_1, e_2 might be deleted and one would obtain a feasible solution of weight smaller than α, a contradiction. So, let us assume, without loss of generality (w.l.o.g.), that u is the first node of e_1, e_2 used by L_1, L_2 going in this direction.

Let L_1^s, L_1^t (resp., L_2^s, L_2^t) be the subpaths of L_1 (resp., L_2) between s and u and between v and t. Obviously, $|L_i^s \cup L_i^t| \leq 2$ for $i = 1, 2$. Note that we have either $L_1^s = \emptyset = L_2^s$ or $L_1^s \neq \emptyset \neq L_2^s$. Moreover, if the latter case holds, we have that $|L_1^t| \leq 1$ and $|L_2^t| \leq 1$. Note also that, by symmetry, these properties remain true if we exchange s and t. Thus every st-path consisting of a combination of subpaths $L_i^s \cup \{e_j\} \cup L_k^t$ is of length at most 3 for $i, j, k = 1, 2$. In other words, we have that

$$|L_i^s \cup L_k^t| \leq 2 \qquad \text{for all } i, k \in \{1, 2\}.$$

By Lemma 3.5(i), there must also exist an edge set of τ_a, say S_2, that does not contain e_2. By Claim 1(ii), we have that $[u, v] \cap S_2 = \emptyset$. Let P_1 and P_2 be two edge-disjoint 3-st-paths in S_2. We have the following claim.

Claim 2. At least one of the sets $P_1 \cap L_1$ and $P_2 \cap L_2$ ($P_2 \cap L_1$ and $P_1 \cap L_2$) is nonempty.

Proof. Assume, on the contrary, that, for instance, $P_1 \cap L_1 = \emptyset = P_2 \cap L_2$. Then, since $P_2 \cup L_2 \in T(G)$, it follows that $a(P_2) \geq a(L_1)$. Now, let $L_1' = (L_1\backslash\{e_1\}) \cup \{e_2\}$. As $e_2 \notin S_2$ and hence $e_2 \notin P_1$, we have that $P_1 \cap L_1' = \emptyset$. Thus $P_1 \cup L_1' \in T(G)$, and therefore $a(L_1') \geq a(P_2)$. As a consequence, $a(L_1') \geq a(L_1)$, and hence $a(e_2) \geq a(e_1)$, a contradiction. □

By Claim 2, we may assume, w.l.o.g., that $P_1 \cap L_2 \neq \emptyset$. Also by the same claim, at least one of the sets $P_1 \cap L_1$ and $P_2 \cap L_2$ is nonempty. In what follows, we suppose that $P_2 \cap L_2 \neq \emptyset$. The case where $P_1 \cap L_1 \neq \emptyset$ can be treated along the same lines. As $e_2 \notin S_2$, it follows that $|L_2| = 3$. If $|L_2^s| = 2$, then $v = t$, and L_2 is of the form (s, w, u, t) with $w \neq s, t, u$. Let e_0 be the edge of $L_2 \cap [u, w]$. Note that one of the 3-st-paths of S_2, say P_1, uses e_0. Then P_1 is of the form (s, u, w, t). Let $\{f\} = P_1 \cap [w, t]$. As $(S_1\backslash\{e_0, e_1\}) \cup \{f\}$ and $(S_2\backslash\{e_0, f\}) \cup \{e_2\}$ are edge sets of $T(G)$, we obtain that $a(f) \geq a(e_0) + a(e_1)$ and $a(e_2) \geq a(e_0) + a(f)$, respectively. But this implies that $a(e_2) \geq a(e_1)$, a contradiction.

Consequently, $|L_2^s| \leq 1$, and, by symmetry, we also have that $|L_2^t| \leq 1$. Since $|L_2| = 3$, it follows that $|L_2^s| = |L_2^t| = 1$. So L_1 and L_2 are both of the form (s, u, v, t). As $P_1 \cap L_2 \neq \emptyset \neq P_2 \cap L_2$ and $S_2 \cap [u, v] = \emptyset$, we may assume, w.l.o.g., that $P_1 \cap [s, u] \neq \emptyset$ and $P_2 \cap [v, t] \neq \emptyset$. Moreover, this implies that $P_1 \cap L_1 = \emptyset = P_2 \cap L_1$. Now, by replacing e_1 and L_1^t by the subpath P_1^{ut} of P_1 between u and t, we get a solution, yielding $a(P_1^{ut}) \geq a(e_1) + a(L_1^t)$. Similarly, if we replace P_1^{ut} by e_2 and L_1^t in S_2, we obtain that $a(e_2) + a(L_1^t) \geq a(P_1^{ut})$. But this again yields $a(e_2) \geq a(e_1)$, which is impossible. □

By Lemma 3.7, the multiple edges have the same coefficient in any nontrivial facet of $P(G,L)$. For the rest of the paper, if $u,v \in N$, we will denote by uv a fixed edge of $[u,v]$. If P is a path of the form $(u_1,u_2,\ldots,u_q)$, then we will suppose that P uses the edges $u_1u_2,\ldots,u_{q-1}u_q$. If for a solution $S \in T(G)$ and two nodes $u,v \in N$ we have that S intersects $[u,v]$, then we will suppose that S uses edge uv and eventually further edges parallel to uv.

4. Structural properties. In this section we give some structural properties of the facet defining inequalities of $P(G,L)$ different from the trivial and the st-cut inequalities. These will be useful for the proof of our main result in section 5.

Let $L = 2,3$ and $ax \geq \alpha$ be a facet defining inequality of $P(G,L)$ different from the trivial and the st-cut inequalities. First, we give the following technical lemma, which will be frequently used in the subsequent proofs.

LEMMA 4.1. *Let S_1 and S_2 be two edge sets of τ_a. Let P_1 and P_1' be two edge-disjoint L-st-paths of S_1. Suppose that there is an L-st-path P_2 in S_2 such that $P_2 \cap P_1' = \emptyset$. Then, for every L-st-path P not intersecting S_2, we have $a(P) \geq a(P_1)$.*

Proof. Let S_1' (resp., S_2') be the edge set obtained from S_1 (resp., S_2) by replacing P_1 by P_2 (resp., P_2 by P). As $S_1', S_2' \in T(G)$, it follows that $a(P_2) \geq a(P_1)$ and $a(P) \geq a(P_2)$. Hence, $a(P) \geq a(P_1)$. □

LEMMA 4.2. *There cannot exist an L-st-path containing only edges with zero weight.*

Proof. We will show the result for $L = 3$. The proof for $L = 2$ can be done in a similar way.

Let us assume the contrary. Let P_0 be a shortest st-path such that $a(e) = 0$ for all $e \in P_0$. In what follows, we consider the case where $|P_0| = 3$. The cases where $|P_0| = 2$ or 1 can be treated similarly.

Let $P_0 = (s,u_1,u_2,t)$. Then $a(e) > 0$ for every chord of P_0. By Lemma 3.7, we have $a(e) = 0$ for all $e \in [s,u_1] \cup [u_1,u_2] \cup [u_2,t]$. As $ax \geq \alpha$ is different from a trivial inequality, by Lemma 3.5(i), there must exist an edge set S of τ_a not containing the edge u_2t of P_0. Let P_1, P_2 be two edge-disjoint 3-st-paths of S.

Claim 1. Let T be a solution of τ_a and T_1, T_2 be two edge-disjoint 3-st-paths of T. Then at least one of the paths T_1, T_2 has only edges with zero value if one of the following statements holds:

(i) $u_2t \notin T$,

(ii) $su_1 \notin T$,

(iii) $u_1u_2 \notin T$ and $|[u_2,t]| \geq 2$.

Proof. Suppose that both T_1 and T_2 use edges with positive weight. We first claim that both T_1 and T_2 intersect P_0. Otherwise, if, for instance, $T_1 \cap P_0 = \emptyset$, then $T_1 \cup P_0 \in T(G)$, yielding $a(P_0) \geq a(T_2)$. As $a(P_0) = 0$, we then have $a(T_2) = 0$, a contradiction.

Now suppose that $u_2t \notin T$. As $T_1 \cap T_2 = \emptyset$, one of the paths, say T_1, uses edge u_1u_2. Since T_1 uses at least one edge of positive weight and $a(e) = 0$ for all $e \in [s,u_1] \cup [u_2,t]$, T_1 must be of the form (s,u_2,u_1,t). By the remark above, we have indeed that $a(u_1t) > 0$. Now if we replace in T the edges u_1u_2 and u_1t by u_2t, we get a solution of $T(G)$. Moreover, as $a(u_2t) = a(u_1u_2) = 0$, it follows that $a(u_1t) = 0$, a contradiction.

If $su_1 \notin T$, then the statement follows by symmetry.

Suppose now that $u_1u_2 \notin T$ and $|[u_2,t]| \geq 2$. Denote by f an edge of $[u_2,t] \backslash \{u_2t\}$. Since $u_1u_2 \notin T$ and $T_1 \cap P_0 \neq \emptyset \neq T_2 \cap P_0$, we may suppose, w.l.o.g., that $su_1 \in T_1$ and $u_2t \in T_2$. Let $T_1^{u_1t}$ be the subpath of T_1 between u_1 and t. Observe that $a(T_1^{u_1t}) > 0$. Consider the solution obtained from T by replacing $T_1^{u_1t}$ by the edges u_1u_2 and f.

As $a(f) = a(u_1u_2) = 0$, this yields $a(T_1^{u_1t}) = 0$, a contradiction, which ends the proof of the claim. □

As $u_2t \notin S$, by Claim 1(i), it follows that at least one of the paths P_1 and P_2, say P_1, contains only edges with zero coefficient. Moreover, we have that $P_1 \cap P_0 \neq \emptyset$. Otherwise, there would exist a solution formed by P_1 and P_0 of weight zero, contradicting the fact that $\alpha > 0$.

Claim 2. (i) $|[u_2, t]| \geq 2$.

(ii) $|[s, u_1]| \geq 2$.

Proof. We will prove (i); the proof of (ii) follows by symmetry. Suppose that $|[u_2, t]| = 1$. We claim that the edge su_1 of P_0 belongs to P_1. In fact, if this is not the case, as $u_2t \notin S$ and $P_1 \cap P_0 \neq \emptyset$, P_1 must contain the edge u_1u_2. As $|[u_2, t]| = 1$ and $u_2t \notin S$, P_1 must use an edge of $[u_1, t]$ which is of positive weight, a contradiction. Thus P_1 is of the form (s, u_1, v, t) with $v \neq u_2$. We thus have $|[s, u_1]| = 1$. Otherwise, we would have two edge-disjoint 3-st-paths of zero weight, yielding $\alpha = 0$, a contradiction. By considering a solution of τ_a not containing su_1 and using Claim 1(ii) together with similar arguments as above, we can show that there exists a path P_1' of the form (s, w, u_2, t), with $w \neq u_1$, constituted of edges with zero coefficient. As P_1 and P_1' are edge-disjoint and hence form a solution of $T(G)$, this yields $\alpha = 0$, a contradiction. □

Since there are no two edge-disjoint 3-st-paths of weight zero, at least one of the sets $[s, u_1]$, $[u_1, u_2]$, $[u_2, t]$ must be reduced to a single edge. Consequently, by Claim 2, it follows that $|[u_1, u_2]| = 1$. Consider now a solution S' of τ_a not containing u_1u_2. Let P_1' and P_2' be two edge-disjoint 3-st-paths of S'. As, by Claim 2(ii), $|[u_2, t]| \geq 2$, we may, w.l.o.g., suppose by Claim 1(iii) that $a(P_1') = 0$. Also, since $\alpha > 0$, one should have $P_1' \cap P_0 \neq \emptyset$. Since $u_1u_2 \notin S'$, we may, w.l.o.g., suppose that $su_1 \in P_1'$. Therefore $P_1' = (s, u_1, v', t)$ with $v' \neq u_2$. As, by Claim 2, $|[s, u_1]| \geq 2$, the solution given by $P_0 \cup \tilde{P}_1$, where $\tilde{P}_1 = (f, u_1v', v't)$ with $f \in [s, u_1]\backslash\{su_1\}$, would be in $T(G)$ and of zero weight. But this is a contradiction, and the proof of the lemma is complete. □

Let us denote by U (resp., V) the subset of nodes u such that $a(e) = 0$ for all $e \in [s, u]$ (resp., $e \in [u, t]$). Note that, by Lemma 3.7, if for an edge $f \in [s, u]$ (resp., $f \in [u, t]$) for some $u \in N\backslash\{s, t\}$ we have $a(f) = 0$, then $u \in U$ (resp., $u \in V$). By Lemma 4.2, we have that $U \cap V = \emptyset$. Moreover, $a(e) > 0$ for all $e \in [s, t] \cup [s, V] \cup [U, t]$. If $L = 3$, we also have that $a(e) > 0$ for all $e \in [U, V]$. Let $W = N\backslash(\{s, t\} \cup U \cup V)$. Note that if $W \neq \emptyset$, $a(e) > 0$ for all $e \in [s, W] \cup [W, t]$.

LEMMA 4.3. *$U \neq \emptyset \neq V$.*

Proof. We will prove the lemma for U. The proof for V is similar. Since $ax \geq \alpha$ is different from the st-cut constraint corresponding to the node s, by Lemma 3.5(ii), there is an edge set F of τ_a that contains at least three edges of $\delta(s)$. As only two of these edges can be used by two edge-disjoint L-st-paths of F, there is an edge of $F \cap \delta(s)$, say $e_0 \in [s, u]$ with $u \in N\backslash\{s, t\}$, such that $F\backslash\{e_0\} \in T(G)$. This implies that $a(e_0) = 0$, and therefore $u \in U$. □

LEMMA 4.4. *Let $S \in \tau_a$ and P_1 be a 3-st-path of S going through a node u of $N\backslash\{s, t\}$. Let $\tilde{P}_1$ be the subpath of P_1 between s (resp., t) and u. Let P be a path between s (resp., t) and u such that $a(P) = 0$ and $|P| \leq |\tilde{P}_1|$. If $a(\tilde{P}_1) > 0$, then $P \cap P_2 \neq \emptyset$ for any 3-st-path P_2 of S, where $P_2 \cap P_1 = \emptyset$.*

Proof. If $P \cap P_2 = \emptyset$, as $|P| \leq |\tilde{P}_1|$, the edge set $(S\backslash\tilde{P}_1) \cup P$ belongs to $T(G)$, and hence $a(\tilde{P}_1) \leq a(P)$. As $a(P) = 0$ and $a(\tilde{P}_1) > 0$, this is impossible. □

The following lemma shows that the edges having both endnodes in U (V) all have zero coefficient. Moreover, if $L = 2$, the same holds for the edges between U and V.

LEMMA 4.5. (i) *If $L = 2$, then $a(e) = 0$ for all $e \in [U, V]$.*

(ii) *$a(e) = 0$ for all $e \in E(U) \cup E(V)$.*

Proof. (i) Let $e \in [U, V]$, and let S be a solution of τ_a containing e. As e cannot belong to a 2-st-path of S, $S\backslash\{e\}$ is also a solution of $T(G)$, and therefore $a(e) = 0$.

(ii) If $L = 2$ and $e \in E(U) \cup E(V)$, we can show as in (i) that $a(e) = 0$. Now let us consider the case where $L = 3$. Let us assume, on the contrary, that there exists an edge u_1u_2 with $u_1, u_2 \in U$ (the case where $u_1, u_2 \in V$ is similar) such that $a(u_1u_2) > 0$. Note that by Lemma 3.7 it follows that $a(e) > 0$ for all $e \in [u_1, u_2]$. Let us consider an edge set of τ_a, say S_1, that contains u_1u_2, and let P_1, P_1' be two edge-disjoint 3-st-paths in S_1. As $a(u_1u_2) > 0$, u_1u_2 must be in one of the 3-st-paths, say P_1. We can suppose, w.l.o.g., that P_1 is (s, u_1, u_2, t). Moreover, as $a(e) = 0$ for all $e \in [s, u_2]$, by Lemma 4.4, P_1' must contain every edge of $[s, u_2]$. However, this is possible only if $|[s, u_2]| = 1$. Consequently, we will assume in the rest of the proof that $[s, u_2] = \{su_2\}$ and $su_2 \in P_1'$. Let us assume that P_1' is of the form (s, u_2, z, t) with $z \neq s, t, u_2$. If P_1' consists of only two edges, then the proof is similar. Furthermore, $z \notin U$. Otherwise, one can consider the edge set $S_1' = (S_1\backslash\{su_1, u_1u_2, u_2z\}) \cup \{sz\}$, which is a solution of $T(G)$. As $a(sz) = 0$, we get $a(su_1) + a(u_1u_2) + a(u_2z) \leq 0$, and hence $a(u_1u_2) = 0$, a contradiction. Therefore $z \in V \cup W$.

Moreover, we have that $a(e) > 0$ for all $e \in [U\backslash\{u_1, u_2\}, u_2]$. Indeed, if $a(e) = 0$, then the edge set $(S_1\backslash\{su_1, u_1u_2\}) \cup \{su, e\}$, where u is the endnode of e different from u_2, would be a solution of $T(G)$ with a weight smaller than α, a contradiction.

Now, let us consider an edge set of τ_a, say S_2, that does not contain the edge su_2. Let P_2, P_2' be two edge-disjoint 3-st-paths in S_2. We claim that $[u_2, t] \cap S_2 = \emptyset$. In fact, if one of the 3-st-paths of S_2, say P_2, uses an edge of $[u_2, t]$, say u_2t, as $|[s, u_2]| = 1$ and $su_2 \notin S_2$, one should have $P_2 = (sw, wu_2, u_2t)$, where $w \in N\backslash\{s, u_2, t\}$. Moreover, we have $a(sw) + a(wu_2) > 0$. In fact, this is clear if $w \notin U$. If $w \in U$, then, as shown above, $a(wu_2) > 0$ and the statement follows. Now, by replacing in S_2 the subpath (sw, wu_2) by su_2, we get a solution of smaller weight, which is impossible.

Thus $[u_2, t] \cap S_2 = \emptyset$, and hence, as $su_2 \notin S_2$, no 3-st-path in S_2 goes through the node u_2. Let P be the path (su_2, u_2t). Thus, $P \cap S_2 = \emptyset$. Moreover, as neither su_2 nor u_2z belongs to S_2, at most one of the paths P_2, P_2' intersects P_1'. W.l.o.g., we may suppose that $P_2 \cap P_1' = \emptyset$. From Lemma 4.1, it then follows that $a(P) \geq a(P_1)$. But this implies that $a(u_1u_2) = 0$, a contradiction. ∎

LEMMA 4.6. (i) *If $L = 2$, then $W = \emptyset$.*

(ii) *If $L = 3$, then $W \neq \emptyset$.*

Proof. (i) Assume the contrary, and let $w \in W$. Then $a(e) > 0$ for all $e \in [s, w] \cup [w, t]$. We will show that $|[s, w] \cap F| = |[w, t] \cap F|$ for every $F \in \tau_a$. In fact, suppose, by contradiction, that there exists $F \in \tau_a$ such that, for instance, $|[s, w] \cap F| > |[w, t] \cap F|$. Since at most $|[w, t] \cap F|$ edge-disjoint 2-st-paths can go through w, there must exist an edge, say $\bar{e}$, of $[s, w] \cap F$ such that $F\backslash\{\bar{e}\} \in T(G)$. This implies that $a(\bar{e}) = 0$, a contradiction. Thus, the incidence vector of any solution of τ_a verifies the equation $x([s, w]) = x([w, t])$. As, by Lemma 3.6, this equation cannot be a positive multiple of $ax \geq \alpha$, we get a contradiction.

(ii) Assume that, on the contrary, $W = \emptyset$. Let $U' = U \cup \{s\}$. Since $ax \geq \alpha$ is different from the st-cut inequality associated with $\delta(U')$, there exists an edge set of τ_a, say F_1, that uses at least three edges of $\delta(U')$. Let P_1, P_1' be two edge-disjoint 3-st-paths of F_1. Since $W = \emptyset$, $a(e) > 0$ for all $e \in \delta(U')$, and hence every edge of $F_1 \cap \delta(U')$ must belong to one of the paths P_1 and P_1'. So, one of these paths, say P_1, must use at least two edges of $\delta(U')$. As any st-path intersects any st-cut an odd number of times, we have that P_1 contains exactly three edges of $\delta(U')$. Therefore, P_1 is of the form

(s, v, u, t), where $u \in U$ and $v \in V$. Let $F_2 = (F_1 \backslash (P_1' \cup \{vu\})) \cup \{su, vt\}$. Obviously, $F_2 \in T(G)$. As $a(su) = a(vt) = 0$, it follows that $a(vu) = 0$, a contradiction. □

For the rest of this section, we assume that $L = 3$.

LEMMA 4.7. (i) *If there are a node $w \in W$ and a node $u_1 \in U$ such that $a(u_1w) = 0$, then $a(e) = 0$ for all $e \in [U, w]$.*

(ii) *If there are a node $w \in W$ and a node $v_1 \in V$ such that $a(wv_1) = 0$, then $a(e) = 0$ for all $e \in [w, V]$.*

Proof. We show the result for U; the proof for V is similar. If $|U| = 1$, then the statement follows from Lemma 3.7. So, let us suppose that $|U| \geq 2$ and assume, on the contrary, that there is a node $u_2 \in U$ such that $a(u_2w) > 0$. Let S_1 be a solution of τ_a such that $u_2w \in S_1$. As $a(u_2w) > 0$, u_2w must belong to a 3-st-path P_1 in S_1. Let P_1' be a further 3-st-path of S_1 with $P_1 \cap P_1' = \emptyset$.

Claim 1. $P_1 = (s, u_2, w, t)$.

Proof. As $u_2w \in P_1$, P_1 is either of the form (s, w, u_2, t) or (s, u_2, w, t). Suppose that the first case holds. As $a(e) > 0$ for all $e \in [s, w]$ and $a(e) = 0$ for all $e \in [s, u_2]$, it follows from Lemma 4.4 that P_1' uses all the edges between s and u_2. Therefore $[s, u_2] \subseteq P_1'$. Moreover, since, by Lemma 4.5(ii), all the 2-su_2-paths going through u_1 have weight zero, again by Lemma 4.4, P_1' must also intersect all these paths. As P_1' cannot use more than one edge incident to s, one should have $[u_1, u_2] \subseteq P_1'$. As a consequence, $|[s, u_2]| = |[u_1, u_2]| = 1$, and P_1' is of the form (s, u_2, u_1, t). But, by adding edge su_1 and removing the edges sw, wu_2, we obtain a solution of lower weight, which is impossible. □

Consequently, $P_1 = (s, u_2, w, t)$. As $a(u_2w) > 0$ and therefore the weight of the subpath of P_1 between s and w is positive, it follows by Lemma 4.4 that P_1' must intersect every 2-sw-path of weight zero going through u_1. Since $a(u_1w) = 0$, by Lemma 3.7 $a(e) = 0$ for all $e \in [u_1, w]$. Thus, as $a(e) = 0$ for all $e \in [s, u_1]$, we obtain that at least one of the sets $[s, u_1]$ and $[u_1, w]$ is reduced to a single edge. If there is a node $u \in U \backslash \{u_1, u_2\}$ such that $a(e) = 0$ for some edge $e \in [u, w]$, then by Lemma 4.4, P_1' must also intersect the 2-sw-paths going through u. But as $|P_1'| \leq 3$, this is not possible. Therefore $a(e) > 0$ for all $e \in [U \backslash \{u_1\}, w]$.

Claim 2. $P_1' \cap [u_1, w] = \emptyset$.

Proof. Suppose, on the contrary, that P_1' uses, for instance, u_1w. If $P_1' = (s, w, u_1, t)$, then, as the weight of the subpath of P_1' between s and u_1 is positive and $a(e) = 0$ for all $e \in [s, u_1]$, by Lemma 4.4 it follows that P_1 uses all the edges between s and u_1. But this contradicts Claim 1. Hence P_1' is of the form (su_1, u_1w, h), where $h \in [w, t] \backslash \{wt\}$. We consider two cases.

Case 1. $|[s, u_1]| = 1$. Consider an edge set S_2 of τ_a such that $su_1 \notin S_2$. We may suppose that S_2 is minimal. Let P_2 and P_2' be the two edge-disjoint 3-st-paths of S_2. If S_2 uses an edge u_1z with $z \in V \cup W$, then u_1z belongs to one of the 3-st-paths of S_2, say P_2. As $su_1 \notin S_2$, $P_2 = (s, z, u_1, t)$. Observe that $a(e) > 0$ for all $e \in [s, z]$. Now by replacing the edges sz, zu_1 by su_1, we get a solution of $T(G)$ of weight less than α, a contradiction. As a consequence, we have $[u_1, V \cup W] \cap S_2 = \emptyset$, and therefore $[u_1, w] \cap S_2 = \emptyset$. Suppose now that $S_2 \cap [w, t] \neq \emptyset$ and, for instance, that $P_2 \cap [w, t] \neq \emptyset$. Since $a(e) > 0$ for all $e \in [U \backslash \{u_1\}, w]$, the subpath of P_2, say P_2^{sw}, between s and w has a positive weight. As $\{su_1, u_1w\}$ is a 2-sw-path of weight zero which does not intersect S_2, if we replace P_2^{sw} by su_1, u_1w, we get a solution of lower weight, which is impossible. Thus $S_2 \cap [w, t] = \emptyset$, and, in consequence, $P_1' \cap S_2 = \emptyset$. Let $P = P_1'$. By Lemma 4.1, it follows that $a(P) = a(P_1') \geq a(P_1)$. As $a(h) = a(wt)$ and $a(su_1) = a(u_1w) = 0$, this yields $a(u_2w) = 0$, a contradiction.

Case 2. $|[s,u_1]| \geq 2$. Since one of the sets $[s,u_1],[u_1w]$ contains exactly one edge, we have that $[u_1,w]=\{u_1w\}$. Let $\bar{S}_2$ be a solution of τ_a not containing u_1w. Suppose that $\bar{S}_2$ is minimal, and let $\bar{P}_2$ and $\bar{P}'_2$ be the two edge-disjoint 3-st-paths of $\bar{S}_2$. We can show, in a similar way as in Case 1, that $[w,t]\cap\bar{S}_2=\emptyset$. As $u_1w\notin\bar{S}_2$, it follows that $|\bar{S}_2\cap P'_1|\leq 1$. Hence, there is a 3-$st$-path of $\bar{S}_2$, say $\bar{P}_2$, that does not intersect P'_1. Therefore $\bar{P}_2\cup P'_1$ is a solution of $T(G)$, yielding $a(\bar{P}_2)\geq a(P_1)$. On the other hand, since $|[s,u_1]|\geq 2$, we may suppose that $\bar{P}'_2\cap P'_1=\emptyset$. So, if we replace, in $\bar{S}_2$, $\bar{P}_2$ by P'_1, we get a solution of $T(G)$, implying that $a(P'_1)\geq a(\bar{P}_2)$. Therefore $a(P'_1)\geq a(P_1)$, and hence $a(u_2w)=0$, a contradiction. □

By Claim 2, we then have $P'_1\cap[u_1,w]=\emptyset$. As P'_1 intersects all the 2-sw-paths going through u_1, it follows that $[s,u_1]=\{su_1\}$ and $su_1\in P'_1$.

If P'_1 uses an edge of $[u_1,t]$, then, by removing the edge u_2w and adding edges u_1w and u_1u_2, we get a solution of $T(G)$. But this implies that $a(u_2w)=0$, which is impossible. Along the same lines, we can also show that P'_1 does not go through any node of U. Hence P'_1 must use a node of $V\cup W$, say v.

Consider now a solution S_3 of τ_a not containing su_1. Let P_3 and P'_3 be two edge-disjoint 3-st-paths of S_3. Suppose that there is an edge, say u_1z, of $[u_1,V\cup W]$ that belongs to S_3. Since $su_1\notin S_3$, the 3-st-path containing u_1z, say P_3, must be of the form (s,z,u_1,t). Note that the subpath between s and u_1 has a positive weight. As $a(su_1)=0$, by Lemma 4.4, it follows that $su_1\in P'_3$, and hence $su_1\in S_3$, contradicting our hypothesis. Thus $[u_1,V\cup W]\cap S_3=\emptyset$, and hence $([u_1,v]\cup[u_1,w])\cap S_3=\emptyset$. Thus $|P'_1\cap S_3|\leq 1$. Consequently, there must exist a 3-st-path of S_3, say P_3, such that $P'_1\cap P_3=\emptyset$. Also we may show in a similar way that $[w,t]\cap S_3=\emptyset$. Consider now the path $P=(s,u_1,w,t)$. Observe that $P\cap S_3=\emptyset$. By Lemma 4.1, with respect to S_1 and S_3, it follows that $a(P)\geq a(P_1)$. But this implies that $a(u_2w)=0$, a contradiction, and the proof of the lemma is complete. □

LEMMA 4.8. *For all* $e,e'\in[U,t]$ *(resp.,* $e,e'\in[s,V]$*),* $a(e)=a(e')$.

Proof. We will prove the lemma for U; the proof for V is similar. If $|U|=1$, the statement follows from Lemma 3.7. So suppose $|U|\geq 2$. Let $u_1,u_2\in U$ such that $a(u_1t)=\min\{a(e),e\in[U,t]\}$ and $a(u_2t)=\max\{a(e),e\in[U,t]\}$. Assume that $a(u_2t)>a(u_1t)$.

Claim. (i) Let $S\in\tau_a$. If $S\cap[u_2,t]\neq\emptyset$, then $[u_1,t]\subseteq S$.

(ii) $|[u_1,t]|=1$.

Proof. (i) Suppose that $u_2t\in S$, and let T_1 and T_2 be two edge-disjoint 3-st-paths of S. As $a(u_2t)>0$, we may suppose, for instance, that $u_2t\in T_2$. Assume that there is an edge e_1 of $[u_1,t]$ that is not in S. If there is an edge $e\in[s,u_1]$ that is not in T_1, then we can replace u_2t by e and e_1 and get a solution of $T(G)$ of lower weight, a contradiction. Hence $[s,u_1]\subseteq T_1$, and therefore $[s,u_1]=\{su_1\}$, $su_1\in T_1$, and $[s,u_2]\cap T_1=\emptyset$. Furthermore, if T_1 contains an edge $e'\in[u_1,u_2]$, then, as $su_1\in T_1$, T_1 must use an edge f of $[u_2,t]\backslash\{u_2t\}$. Now it is easy to see that $(S\backslash\{f\})\cup\{e_1\}\in T(G)$. Since by Lemma 3.7, $a(e_1)=a(u_1t)$ and $a(f)=a(u_2t)$, it follows that $a(u_1t)\geq a(u_2t)$. But this contradicts our hypothesis. Therefore $[u_1,u_2]\cap T_1=\emptyset$. Consider now the solution $S'=(S\backslash\{u_2t\})\cup\{su_2,u_1u_2,e_1\}$. As $a(su_2)=a(u_1u_2)=0$, we have that $a(u_1t)=a(e_1)\geq a(u_2t)$, a contradiction.

(ii) Let $\bar{S}\in\tau_a$ such that $u_2t\in\bar{S}$. We may suppose that $\bar{S}$ is minimal. Let $\overline{T}_1,\overline{T}_2$ be the edge-disjoint 3-st-paths of $\bar{S}$, and suppose, w.l.o.g., that $u_2t\in\overline{T}_2$. From (i), it follows that $[u_1,t]\subseteq\bar{S}$. Moreover, as $u_2t\in\overline{T}_2$, $\overline{T}_2\cap[u_1,t]=\emptyset$, and hence $[u_1,t]\subseteq T_1$. This implies that $|[u_1,t]|=1$. □

Let S_1 be a solution of τ_a containing u_2t. By the claim above, S_1 also contains u_1t. As $a(su_1)=a(su_2)=0$ and $\{su_1,su_2,u_1t,u_2t\}$ is a solution of $T(G)$, we may assume that $S_1=\{su_1,su_2,u_1t,u_2t\}$.

Consider now a solution $S_2 \in \tau_a$ that does not contain u_1t, which may be supposed minimal. Since $u_1t \notin S_2$, by the claim it follows that $[u_2, t] \cap S_2 = \emptyset$; and, as a consequence, $[u_1, u_2] \cap S_2 = \emptyset$. Suppose that S_2 contains an edge su_1. Since S_2 is minimal, one of the two 3-st-paths of S_2, say T, contains su_1, and hence T is of the form (s, u_1, z, t), where $z \in N\backslash\{s, t, u_1, u_2\}$. Let T^{u_1t} be the subpath of T between u_1 and t. As the sets $(S_2\backslash T^{u_1t}) \cup \{u_1t\}$ and $(S_1\backslash\{u_2t\}) \cup (\{u_1u_2\} \cup T^{u_1t})$ are both solutions of $T(G)$, and, as by Lemma 4.5(ii) $a(u_1u_2) = 0$, we have that $a(u_1t) \geq a(T^{u_1t}) \geq a(u_2t)$, a contradiction. Consequently, $[s, u_1] \cap S_2 = \emptyset$.

Let $P_1 = (su_2, u_2t)$ and $P_1' = (su_1, u_1t)$ be the two 3-st-paths of S_1. Let $P = P_1'$ and P_2 be any 3-st-path of S_2. Note that $P \cap S_2 = P_1' \cap S_2 = \emptyset$, and hence $P_2 \cap P_1' = \emptyset$. By Lemma 4.1, it follows that $a(P) \geq a(P_1)$. However, as $a(su_1) = a(su_2) = 0$, this implies again that $a(u_1t) \geq a(u_2t)$, which is impossible. □

LEMMA 4.9. *Let S be a minimal solution of τ_a.*

(i) *If $U = \{u\}$ and $S \cap [s, u] = \emptyset$, then $\delta(u) \cap S = \emptyset$.*

(ii) *If $V = \{v\}$ and $S \cap [v, t] = \emptyset$, then $\delta(v) \cap S = \emptyset$.*

Proof. We will show (i); the proof of (ii) is similar. We first show that $[u, t] \cap S = \emptyset$. Assume, on the contrary, that $ut \in S$. Then, as $a(ut) > 0$, one of the 3-st-paths of S, say P, must contain ut. As $[s, u] \cap S = \emptyset$, P must be of the form (s, w, u, t), where $w \in N\backslash\{s, t, u\}$. Note that $w \notin U$, and hence $a(sw) > 0$. Thus, one can replace sw and wu by su in S and get a solution of $T(G)$ of weight less than α, a contradiction. Thus $[u, t] \cap S = \emptyset$. Now, by the minimality of S, no other edge of $\delta(u)$ may be used by S. □

LEMMA 4.10. *$a(e) = a(e')$ for all $e \in [U, t]$ and $e' \in [s, V]$.*

Proof. Assume the contrary. Thus, by Lemma 4.8, we may assume, w.l.o.g., that

$$a(e) > a(e') \qquad \text{for all } e \in [U, t] \text{ and } e' \in [s, V]. \tag{4.1}$$

Let $u_1 \in U$. Consider a solution S_1 of τ_a that contains u_1t, and suppose that S_1 is minimal. Let P_1 and P_1' be the two edge-disjoint 3-st-paths of S_1, and suppose that $u_1t \in P_1$.

Claim. $|V| = 1$.

Proof. Assume that $|V| \geq 2$. First observe that P_1 cannot go through a node $v \in V$. Otherwise, P_1 would be of the form (s, v, u_1, t). Since the subpaths of P_1 between s and u_1 and between v and t both have positive weight, by Lemma 4.4, P_1' must use edges su_1 and vt. Now, if we remove the edges of S_1 between u_1 and v, we still have a solution of $T(G)$. This implies that $a([u_1, v]) = 0$. But this contradicts the fact that $a(u_1v) > 0$. In consequence, since S_1 is minimal, S_1 may contain at most one edge from $[s, V]$. Suppose that S_1 contains edge sv_1, where $v_1 \in V$. Note that $sv_1 \in P_1'$. As $|V| \geq 2$, there is an edge sv_2, with $v_2 \in V$, that does not belong to S_1. If there is an edge $e \in [v_2, t]$ such that $e \notin S_1$, then, by replacing u_1t by sv_2 and e, we get a solution of $T(G)$. As $a(e) = 0$, this yields $a(sv_2) \geq a(u_1t)$, which contradicts (4.1). Thus $[v_2, t] \subseteq S_1$ and therefore $[v_2, t] \subseteq P_1'$. This implies that $[v_2, t] = \{v_2t\}$ and $P_1' = (s, v_1, v_2, t)$. By considering the solution obtained by replacing u_1t by sv_2 and v_1t, we obtain that $a(sv_2) \geq a(u_1t)$, which once again contradicts (4.1).

Consequently, $S_1 \cap [s, V] = \emptyset$. Now we remark that, since S_1 is minimal and $u_1t \in S_1$, S_1 cannot use two edges of $[V, t]$. Thus there is a node $z \in V$ such that $([s, z] \cup [z, t]) \cap S_1 = \emptyset$. By replacing u_1t by sz and zt in S_1, we get a solution of $T(G)$, yielding $a(sz) \geq a(u_1t)$. This contradicts (4.1), and the claim is proved. □

Let $V = \{v\}$. Let $P = (s, v, t)$ be an st-path of length 2 going through v. We claim that $P_1' \cap P \neq \emptyset$. In fact, if this is not the case, then, as the edge set obtained from S_1 by replacing P_1 by P is in $T(G)$, we would have that $a(sv) \geq a(u_1t)$. But

this contradicts (4.1). Therefore, P_1' must contain at least one of the sets $[s,v]$ and $[v,t]$. Thus at least one of the sets $[s,v]$ and $[v,t]$ is reduced to a single edge.

Case 1. $[v,t] = \{vt\}$. Consider a solution $S_2 \in \tau_a$ not containing vt, which is supposed minimal. Then, by Lemma 4.9, $S_2 \cap \delta(v) = \emptyset$, and hence $P \cap S_2 = \emptyset$. Moreover, as $P_1' \cap P \neq \emptyset$, P_1' does meet v, and therefore $|P_1' \cap S_2| \leq 1$. Thus there exists a 3-st-path of S_2, say P_2, that does not intersect P_1'. As $P \cap S_2 = \emptyset$, by Lemma 4.1, we have that $a(P) \geq a(P_1)$, and hence $a(sv) \geq a(u_1t)$. But this contradicts (4.1).

Case 2. $[s,v] = \{sv\}$. By Case 1, we may suppose that $|[v,t]| \geq 2$. As P_1' contains one of the sets $[s,v]$ and $[v,t]$, it follows that $sv \in P_1'$. Note that $\{su_1, u_1t, sv, vt\} \in T(G)$. As $a(su_1) = a(vt) = 0$ and S_1 is minimal, we may suppose, w.l.o.g., that $S_1 = \{su_1, u_1t, sv, vt\}$. Hence $P_1 = (su_1, u_1t)$ and $P_1' = (sv, vt)$. Consider now an edge set S_3 of τ_a not containing sv and suppose that S_3 is minimal. Since $|P_1' \cap S_3| \leq 1$, there must exist a 3-st-path in S_3, say P_3, such that $P_3 \cap P_1' = \emptyset$. If we replace, in S_1, P_1 by P_3, the resulting set is still a solution of $T(G)$, and therefore $a(P_3) \geq a(P_1)$. On the other hand, if there is an edge $h \in [v,t]$ such that $h \notin S_3$, then one can replace the path P_3 by the one formed by sv and h and get a solution of $T(G)$. But this implies that $a(P_3) \leq a(sv) + a(h)$. As $a(P_3) \geq a(P_1)$ and $a(h) = 0$, we obtain that $a(u_1t) \leq a(sv)$, contradicting (4.1). Thus $[v,t] \subseteq S_3$. As $|[v,t]| \geq 2$ and S_3 is minimal, it follows that $P_3 \cap [v,t] \neq \emptyset$. Let P_3^{sv} be the subpath of P_3 between s and v. By replacing, in S_3, P_3^{sv} by sv, we get a solution of $T(G)$, which yields $a(sv) \geq a(P_3^{sv})$. As $a(P_3) \geq a(P_1)$ and therefore $a(P_3^{sv}) \geq a(u_1t)$, we get $a(sv) \geq a(u_1t)$. But this again contradicts (4.1), which ends the proof of the lemma. □

Lemma 4.7 allows a partition of the set W into four subsets:

$W_1 = \{w \in W \mid a(e) = 0 \text{ for all } e \in [U,w], \text{ and } a(e') > 0 \text{ for all } e' \in [w,V]\}$,

$W_2 = \{w \in W \mid a(e) = 0 \text{ for all } e \in [U,w] \cup [w,V]\}$,

$W_3 = \{w \in W \mid a(e) > 0 \text{ for all } e \in [U,w], \text{ and } a(e') = 0 \text{ for all } e' \in [w,V]\}$,

$Z = W \backslash (W_1 \cup W_2 \cup W_3)$.

LEMMA 4.11. (i) *If* $U = \{u\}$, *then* $a(e) = a(e')$ *for all* $e \in [u,t]$ *and* $e' \in [W_1 \cup W_2, t]$.

(ii) *If* $V = \{v\}$, *then* $a(e) = a(e')$ *for all* $e \in [s,v]$ *and* $e' \in [s, W_2 \cup W_3]$.

Proof. We will prove only (i); the proof of (ii) is similar. Assume by contradiction that $a(ut) \neq a(wt)$ for some $w \in W_1 \cup W_2$. We will first give the following claim.

Claim. No solution of τ_a uses at the same time an edge of $[u,t]$ and an edge of $[w,t]$.

Proof. It suffices to show that there is no solution of τ_a containing at the same time ut and wt. Let us suppose, on the contrary, that there exists a solution $S \in \tau_a$ with $ut, wt \in S$. Let T_1 and T_2 be two edge-disjoint 3-st-paths of S. As $a(ut) > 0$ and $a(wt) > 0$, we may suppose that $ut \in T_1$ and $wt \in T_2$.

Suppose that $a(wt) < a(ut)$. The case where $a(wt) > a(ut)$ can be treated along the same lines. If $[s,u] \cap T_1 = \emptyset$, T_1 must go through a node $z \in N \backslash \{s,t,u\}$, and hence the subpath T_1^{su} of T_1 between s and u is of positive weight. By Lemma 4.4, it follows that $[s,u] \subseteq T_2$, and therefore $[s,u] = \{su\}$ and $T_2 = (s,u,w,t)$. If $z \in V$, then, by replacing wt by zt in S, we get a solution of $T(G)$. But, as $a(zt) = 0$, this implies that $a(wt) = 0$, a contradiction. Thus T_1 cannot go through V. As a consequence, as by Lemma 4.3, $V \neq \emptyset$, there is a node $v \in V$ such that sv and vt belong neither to T_1 nor to T_2. So, by replacing T_1 by (sv, vt), we get a solution of $T(G)$. However, since, from Lemma 4.10, we have $a(ut) = a(sv)$, we get $a(T_1^{su}) = 0$, a contradiction. Consequently, $[s,u] \cap T_1 \neq \emptyset$ and $T_1 = (s,u,t)$. By using similar arguments, we can also show that T_2 is of the form (f, uw, wt), where f is an edge parallel to su, and hence $|[s,u]| \geq 2$. Furthermore, at least one of the sets $[u,w]$ and $[w,t]$ is reduced to

a single edge. If not, one may replace ut by a 2-ut-path going through w and get a solution of $T(G)$. But this would imply that $a(wt) \geq a(ut)$, a contradiction.

Suppose that $|[w,t]| = 1$. The case where $|[u,w]| = 1$ is similar. Hence $[w,t] = \{wt\}$. Let $S' \in \tau_a$ such that $wt \notin S'$ and suppose that S' is minimal. If S' contains an edge $e \in [u,w]$, then, as S' is minimal, there must exist in S' a 3-st-path T containing e. Therefore T is of the form (s,w,u,t). Observe that in this case, the edge set obtained by deleting ut and adding wt is in $T(G)$, and then $a(ut) \leq a(wt)$, a contradiction. Consequently, $[u,w] \cap S' = \emptyset$. Hence, as $|T_2 \cap S'| \leq 1$, there is a 3-st-path, say T_1', in S' such that $T_1' \cap T_2 = \emptyset$. By replacing T_1 by T_1' in S, we get a solution of $T(G)$, and hence $a(T_1') \geq a(T_1)$. Note that only one edge of $[s,u]$ can be used by the second 3-st-path of S'. Thus one can replace T_1' by T_2 in S' and obtain a feasible solution, which yields $a(T_2) \geq a(T_1')$, and therefore $a(T_2) \geq a(T_1)$. But this implies that $a(wt) \geq a(ut)$, which is impossible. □

Suppose that $a(ut) > a(wt)$. The case where $a(ut) < a(wt)$ can be treated similarly. Let S_1 be a minimal solution of τ_a that contains ut, and let P_1 and P_1' be two edge-disjoint 3-st-paths of S_1. Suppose, w.l.o.g., that $ut \in P_1$. By the claim, we have $[w,t] \cap S_1 = \emptyset$. If S_1 contains an edge of $[u,w]$, then there is a 3-st-path of S_1 of the form (s,w,u,t). However, by removing ut and adding wt, we obtain a solution of $T(G)$, yielding $a(wt) \geq a(ut)$, a contradiction. Thus $[u,w] \cap S_1 = \emptyset$. Moreover, if there is an edge e of $[s,u]$ such that $e \notin P_1'$, one can replace ut by (e,uw,wt) and get a solution of $T(G)$. But this implies that $a(wt) \geq a(ut)$, a contradiction. Consequently, we have that $[s,u] \subseteq P_1'$. Hence $[s,u] = \{su\}$ and $P_1 = (s,z,u,t)$ with $z \in N \backslash \{s,t,u,w\}$. Observe that the subpath P_1^{su} of P_1 between s and u is of positive weight. If there are two edges $f \in [s,v]$ and $f' \in [v,t]$ such that $f, f' \notin P_1'$, where $v \in V$, then we can replace P_1 by the edges f and f' and still have a feasible solution. As by Lemma 4.10, $a(f) = a(ut)$, we obtain that $a(P_1^{su}) = 0$, a contradiction. Thus, for every node $v \in V$, the path P_1' must use all the edges of at least one of the sets $[s,v]$ and $[v,t]$. This implies that $V = \{v\}$. Moreover, as $su \in P_1'$, we have that $[s,v] \cap P_1' = \emptyset$, $[v,t] = \{vt\}$, and $P_1' = (s,u,v,t)$.

Let S_2 be a solution of τ_a that does not contain su. Recall that $[s,u] = \{su\}$. Suppose that S_2 is minimal. Thus S_2 consists of two edge-disjoint 3-st-paths, say P_2 and P_2'. As $|U| = 1$, by Lemma 4.9, we have that $\delta(u) \cap S_2 = \emptyset$. If S_2 contains an edge e of $[w,t]$, as $a(e) > 0$, e must belong to one of the 3-st-paths of S_2, say P_2. Since $(\{su\} \cup [u,w]) \cap S_2 = \emptyset$, P_2 must be of the form (s,z',w,t), where $z' \notin \{s,t,u\}$. We remark that the subpath of P_2 between s and w is of positive weight. Hence, by Lemma 4.4, P_2' must intersect every 2-sw-path going through u. But this contradicts the fact that $(\{su\} \cup [u,w]) \cap S_2 = \emptyset$. It then follows that $[w,t] \cap S_2 = \emptyset$. As $|P_1' \cap S_2| \leq 1$, there is a 3-st-path in S_2, say P_2, which does not intersect P_1'. Let P be a 3-st-path going through the nodes s,u,w,t. From Lemma 4.1, it follows that $a(P) \geq a(P_1)$. But then we have that $a(wt) \geq a(ut)$, a contradiction. □

5. Proof of Theorem 3.1. In this section, we prove Theorem 3.1; that is, $P(G,L) = Q(G,L)$ for $L = 2,3$. For this, we consider an inequality $ax \geq \alpha$ that defines a facet of $P(G,L)$ different from the trivial and the st-cut inequalities. We will show that $ax \geq \alpha$ is necessarily an L-path-cut inequality.

Case 1. $L = 2$. Let U, V, W be as defined in the previous section. By Lemma 4.6, it follows that $W = \emptyset$, and thus each 2-st-path uses exactly one edge with a nonzero coefficient. Thus, any solution of τ_a contains exactly two edges with a positive coefficient, which are exactly the edges of the 2-path-cut inequality induced by the

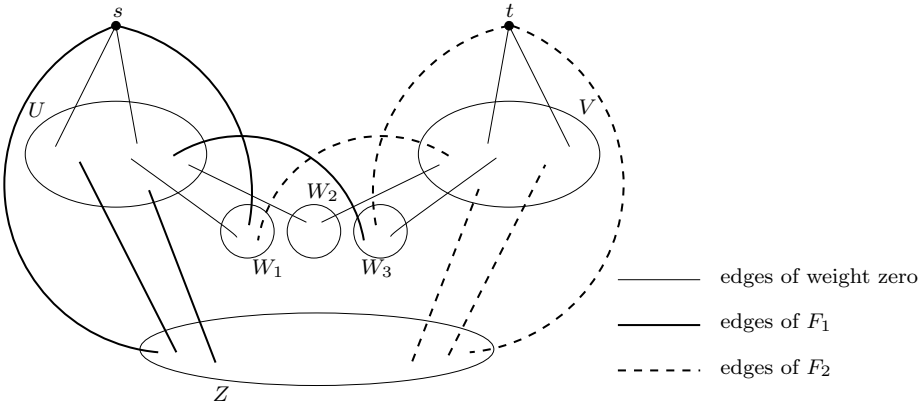


FIG. 3.

partition $\{s\}, U, V, \{t\}$. This implies that $ax \geq \alpha$ is the 2-path-cut inequality induced by this partition.

Case 2. $L = 3$. Let U, V, W_1, W_2, W_3, Z be as defined in the previous section. We consider two cases.

Case 2.1. $W_1 \cup W_3 \cup Z \neq \emptyset$. Let $F_1 = [\{s\} \cup U, Z] \cup [s, W_1] \cup [U, W_3]$ and $F_2 = [Z, V \cup \{t\}] \cup [W_3, t] \cup [W_1, V]$ (see Figure 3). We remark that $F_1 \cap F_2 = \emptyset$ and that there is no st-path of length 3 in G formed by edges only from F_1 and F_2. We have the following.

LEMMA 5.1. *For every solution S of τ_a, we have that $|S \cap F_1| = |S \cap F_2|$.*

Proof. Assume the contrary. Then there exists a solution, say S_1, such that, for one of its 3-st-path, say P_1, we have $|P_1 \cap F_1| \neq |P_1 \cap F_2|$. Let P_1' be the second 3-st-path in S_1. W.l.o.g., we may suppose that $P_1 \cap F_1 \neq \emptyset$.

Claim 1. $P_1 \cap F_2 = \emptyset$.

Proof. Since $P_1 \cap F_1 \neq \emptyset$ and $F_1 \cap F_2 = \emptyset$, we have that $|P_1 \cap F_2| \leq 2$. If $|P_1 \cap F_2| = 1$, as $|P_1 \cap F_1| \neq |P_1 \cap F_2|$ and $P_1 \cap F_1 \neq \emptyset$, $|P_1 \cap F_1| = 2$. Then, P_1 is of length 3 and contained in $F_1 \cup F_2$, which is impossible by the remark above. If $|P_1 \cap F_2| = 2$, then $|P_1 \cap F_1| = 1$, and again we have that P_1 is of length 3 and contained in $F_1 \cup F_2$, a contradiction. Thus, $|P_1 \cap F_2| = 0$ and the claim is proved. □

Claim 2. (i) $P_1 \cap [s, U] = \emptyset$.

(ii) $P_1 = (s, z, w, t)$ with $z \in Z \cup W_1$ and $w \in U \cup W_1 \cup W_2$ (z and w may be the same).

(iii) $[s, U] \subset P_1'$.

(iv) $|U| = 1$ and $|[s, U]| = 1$.

Proof. First note that (iv) is a consequence of (iii).

(i) If P_1 uses an edge of $[s, U]$, say su with $u \in U$, as $P_1 \cap F_1 \neq \emptyset$, P_1 would be of the form (s, u, z, t), where z belongs to either Z or W_3. But this implies that $P_1 \cap F_2 \neq \emptyset$, which contradicts Claim 1.

(ii) Suppose that P_1 contains an edge of $[U, W_3]$, say uw_3. Note that $a(uw_3) > 0$. As, by (i), $[s, U] \cap P_1 = \emptyset$, it follows that $P_1 = (s, w_3, u, t)$. By removing uw_3 and adding su and edges w_3v, vt for some $v \in V$, we get a solution of $T(G)$. As the added edges all have zero weight, this implies that $a(uw_3) = 0$, a contradiction. Consequently, we have that $P_1 \cap [U, W_3] = \emptyset$. Then, by (i) and the fact that $P_1 \cap F_1 \neq \emptyset$,

it follows that P_1 uses one of the edges of $[s, Z \cup W_1]$. As, by Claim 1, $P_1 \cap F_2 = \emptyset$, we obtain that $P_1 = (s, z, w, t)$, where $z \in Z \cup W_1$ and $w \in U \cup W_1 \cup W_2$.

(iii) Suppose that there is an edge of $[s, U]$, say su_0, that does not belong to P'_1. We have that $w \neq u_0$. Otherwise, P_1 would be (s, z, u_0, t). As by (ii) $z \in Z \cup W_1$ and hence $a(sz) > 0$, it follows that the subpath of P_1 between s and u_0 has a positive weight. But this implies by Lemma 4.4 that $su_0 \in P'_1$, a contradiction. We claim that $[u_0, w] \subseteq P'_1$. In fact, if, for instance, $u_0w \notin P'_1$, then consider the solution, say S'_1, obtained from S_1 by replacing sz and zw by su_0 and u_0w. Clearly, $S'_1 \in T(G)$, which implies that $a(su_0) + a(u_0w) \geq a(sz) + a(zw)$. As $a(u_0w) = a(su_0) = 0$, we obtain that $a(sz) = 0$, a contradiction. Thus $[u_0, w] \subseteq P'_1$, and hence $[u_0, w] = \{u_0w\}$. Suppose now that $P'_1 = (f, u_0w, g)$, where f (resp., g) is an edge of $[s, u_0]$ (resp., $[w, t]$) different from that used by P_1. By removing sz, zw, and g and adding the edges su_0 and u_0t, we get a solution of $T(G)$. As by Lemma 4.11 $a(u_0t) = a(g)$, it follows that $a(sz) = 0$, a contradiction. Consequently, $P'_1 = (s, w, u_0, t)$. Now, by considering the solution $\tilde{S}_1 = (S_1 \backslash \{sz, zw\}) \cup \{su_0\}$, one can get a contradiction along the same lines. This ends the proof of the claim. □

Now, by Claim 2(iv), we may suppose that $U = \{u\}$ and $[s, u] = \{su\}$. Let S_2 be a solution of τ_a that does not contain su. W.l.o.g., we may suppose that S_2 is minimal. Then, by Lemma 4.9, it follows that $S_2 \cap \delta(u) = \emptyset$. Let $P = \{s, u, t\}$. Clearly, $P \cap S_2 = \emptyset$. Moreover, as P'_1 goes through node u, $|P'_1 \cap S_2| \leq 1$. As a consequence, there must exist a 3-st-path of S_2, say P_2, such that $P_2 \cap P'_1 = \emptyset$. Now, by Lemma 4.1, we obtain that $a(P) \geq a(P_1)$. By Claim 2(ii), together with Lemma 4.11, it follows that $a(sz) \leq 0$. We then have a contradiction, and the lemma is proved. □

From Lemma 5.1, it follows that the facet defined by $ax \geq \alpha$ is contained in the face induced by the equation $x(F_1) - x(F_2) = 0$. As, by Lemma 3.6, this equation cannot be a positive multiple of $ax = \alpha$, we have a contradiction.

Case 2.2. $W_1 \cup W_3 \cup Z = \emptyset$. Since, by Lemma 4.6, $W \neq \emptyset$, we have necessarily that $W_2 \neq \emptyset$. Thus $\{s\}, U, W_2, V, \{t\}$ is a partition of N. Let T be the set of edges of the 3-path-cut induced by this partition (these edges are represented by solid lines in Figure 4). Note that $a(e) > 0$ for all $e \in T$. Moreover, $a(e) = 0$ for all $e \in E \backslash T$. This is clear for the edges of $E \backslash (T \cup E(W_2))$ from Lemma 4.5(ii) and the definition of U, V, W_2. If $a(z_1z_2) > 0$ for some $z_1, z_2 \in W_2$, then there must exist a solution $\tilde{S}$ of τ_a and a 3-st-path $\tilde{P}$ of $\tilde{S}$ containing z_1z_2. W.l.o.g., we may suppose that $\tilde{P} = (s, z_1, z_2, t)$. Let $\tilde{S}' = (\tilde{S} \backslash \{z_1z_2\}) \cup \{su, uz_2, z_1v, vt\}$ for some nodes $u \in U$ and $v \in V$. As $\tilde{S}' \in T(G)$ and all the added edges have zero weight, it follows that $a(z_1z_2) = 0$, a contradiction.

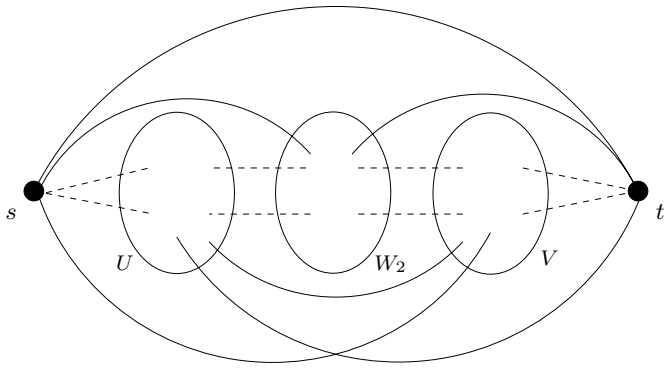


FIG. 4.

Now we claim that each solution of τ_a contains exactly two edges of T. First of all, note that, as the constraint (1.3) associated with T is valid for $P(G,3)$, every solution of τ_a must contain at least two edges of T. Assume that there is a solution S of τ_a with more than two edges of T. So, there must exist in S a 3-st-path P that contains at least two edges of T. We consider the case where $P = (s, w_2, w_2', t)$ with $w_2, w_2' \in W_2$. The other possible cases for P can be treated similarly ((s, w_2, t), (s, w_2, u, t) with $u \in U$, (s, v, w_2, t) with $v \in V$, (s, v, u, t)). Let P' be the second 3-st-path of S. By replacing P' by the edges su, uw_2', w_2v, vt in S, we get a solution of $T(G)$. As all these edges have zero weight, $a(P') = 0$, contradicting Lemma 4.2.

Thus, every solution of τ_a uses exactly two edges of T. This implies that $ax \geq \alpha$ is nothing but the 3-path-cut inequality induced by T, which ends the proof of Theorem 3.1.

6. Facets of $P(G, L)$. In this section, we give necessary and sufficient conditions for inequalities (1.1)–(1.3) to be facet defining for $P(G, L)$. This yields a minimal description of this polytope when $L \leq 3$. Throughout this section, $G = (N, E)$ is a complete graph with $|N| \geq 4$, which may contain multiple edges. Hence, by Corollary 3.3, $P(G, L)$ is full dimensional. The first two theorems, given without proof, describe when the trivial and the st-cut inequalities define facets of $P(G, L)$.

THEOREM 6.1. (i) *For $L \geq 2$, inequality $x(e) \leq 1$ defines a facet of $P(G, L)$.*

(ii) *For $L \geq 2$, inequality $x(e) \geq 0$ defines a facet of $P(G, L)$ if and only if $|N| \geq 5$, or $|N| = 4$ and e does not belong to either an st-cut or an L-path-cut, with exactly three edges.*

THEOREM 6.2. (i) *If $L = 2$, then the only st-cut inequalities that define facets of $P(G, 2)$ are those induced by $\{s\}$ and $N \setminus \{t\}$.*

(ii) *For $L \geq 3$, every st-cut inequality defines a facet of $P(G, L)$.*

We give now necessary and sufficient conditions for the L-path-cut inequalities to be facet defining for $P(G, L)$.

THEOREM 6.3. *For $L \geq 2$, inequality* (1.3) *defines a facet of $P(G, L)$ if and only if $|V_0| = |V_{L+1}| = 1$.*

Proof. Necessity. We will show that $x(T) \geq 2$ does not define a facet of $P(G, L)$ if $|V_0| \geq 2$. The case where $|V_{L+1}| \geq 2$ follows by symmetry.

Suppose that $|V_0| \geq 2$, and consider the partition given by

$$\overline{V}_0 = \{s\},$$
$$\overline{V}_1 = V_1 \cup (V_0 \setminus \{s\}),$$
$$\overline{V}_i = V_i,\ i = 2, \ldots, L+1.$$

This partition induces the L-path-cut inequality $x(\overline{T}) \geq 2$, where $\overline{T} = T \setminus [V_0 \setminus \{s\}, V_2]$. As G is complete, we have that $\overline{T}$ is strictly contained in T, and hence $x(T) \geq 2$ cannot be facet defining.

Sufficiency. Now, suppose that $|V_0| = |V_{L+1}| = 1$, that is, $V_0 = \{s\}$ and $V_{L+1} = \{t\}$. Let us denote inequality (1.3) by $ax \geq \alpha$, and let $bx \geq \beta$ be a facet defining inequality of $P(G, L)$ such that

$$\{x \in P(G, L) \,|\, ax = \alpha\} \subseteq \{x \in P(G, L) \,|\, bx = \beta\}.$$

We will show that $a = \rho b$ for some $\rho > 0$.

Let $V_0 = \{s\}, V_1, \ldots, V_L, V_{L+1} = \{t\}$ be the partition inducing $ax \geq \alpha$. Let $\overline{E} = E \setminus T = (\bigcup_{i=1}^{L} E(V_i)) \cup (\bigcup_{i=0}^{L} [V_i, V_{i+1}])$. Let $f \in [s, t]$ and $T_f = T \setminus \{f\}$. As the graph G is complete, it is easy to see that the sets given by

$$F_e = \overline{E} \cup \{f, e\} \qquad \text{for all } e \in T_f$$

induce solutions of the THPP whose incidence vectors satisfy $ax \geq \alpha$ with equality. Thus,

$$0 = bx^{F_e} - bx^{F_{e'}} = b(e) - b(e') \qquad \text{for all } e, e' \in T_f.$$

Hence,

$$b(e) = b(e') \qquad \text{for all } e, e' \in T_f. \tag{6.1}$$

Now let $g \in [V_0, V_L]$, $g' \in [V_1, V_{L+1}]$, and $F^* = \overline{E} \cup \{g, g'\}$. It is obvious that F^* induces a solution whose incidence vector satisfies $ax \geq \alpha$ with equality. Thus $bx^{F^*} - bx^{F_g} = b(g') - b(f) = 0$. This together with (6.1) yields

$$b(e) = \gamma \qquad \text{for all } e \in T \text{ for some } \gamma \in \mathbb{R}.$$

Now, we shall show that $b(e) = 0$ for all $e \in \overline{E}$. Suppose first that $e \in [V_0, V_1]$. Consider an edge $h \in [s, w]$ with $w \in V_2$ and the edge set $F_h \backslash \{e\}$, where F_h is as defined above. It is easy to see that $F_h \backslash \{e\}$ still induces a solution of the THPP whose incidence vector satisfies $ax \geq \alpha$ with equality. Thus,

$$0 = bx^{F_h} - bx^{F_h \backslash \{e\}} = b(e).$$

Similarly, we obtain that $b(e) = 0$ for all $e \in \bigcup_{i=0}^{L} [V_i, V_{i+1}]$. Consider now an edge $e \in E(V_i)$, $i \in \{1, \ldots, L\}$. Let $v \in V_L$ and $h' \in [s, v]$. Clearly, the set $F_{h'} \backslash \{e\}$ induces a solution of the problem. As $ax^{F_{h'}} = ax^{F_{h'} \backslash \{e\}} = \alpha$, we have that $bx^{F_{h'}} = bx^{F_{h'} \backslash \{e\}} = \alpha$, and hence $b(e) = 0$.

Consequently, we have that

$$\begin{aligned} b(e) &= 0 \qquad \text{for all } e \in \overline{E}, \\ b(e) &= \gamma \qquad \text{for all } e \in T. \end{aligned}$$

Since $\alpha > 0$, we have that $\gamma > 0$, and by setting $\rho = 1/\gamma$, we obtain that $a = \rho b$. □

Let E' be the set of edges that belong neither to an st-cut nor to an L-path-cut, consisting of exactly three edges. From the previous theorems, we have the following.

COROLLARY 6.4. *For $L = 2$, if $G = (N, E)$ is complete and $|N| \geq 4$, then a minimal complete linear description of $P(G, L)$ is given by*

$$\begin{aligned} &x(\delta(s)) \geq 2, \\ &x(\delta(t)) \geq 2, \\ &x(T) \geq 2 \qquad \textit{for all 2-path-cut } T \textit{ induced by } V_0 = \{s\}, V_1, V_2, V_3 = \{t\}, \\ &x(e) \leq 1 \qquad \textit{for all } e \in E, \\ &x(e) \geq 0 \qquad \textit{for all } e \in E'. \end{aligned}$$

COROLLARY 6.5. *For $L = 3$, if $G = (N, E)$ is complete and $|N| \geq 4$, then a minimal complete linear description of $P(G, L)$ is given by*

$$\begin{aligned} &x(\delta(W)) \geq 2 \quad \textit{for all } st\textit{-cut } \delta(W), \\ &x(T) \geq 2 \qquad \textit{for all 3-path-cut } T \textit{ induced by } V_0 = \{s\}, V_1, V_2, V_3, V_4 = \{t\}, \\ &x(e) \leq 1 \qquad \textit{for all } e \in E, \\ &x(e) \geq 0 \qquad \textit{for all } e \in E'. \end{aligned}$$

7. Dominant of $P(G, L)$. In this section, we consider the dominant of the polytope $P(G, L)$. We give a complete description of that polyhedron for any graph G and integer $L \geq 2$ such that $P(G, L) = Q(G, L)$.

Let $Dom(P(G, L))$ be the dominant of $P(G, L)$, that is,

$$Dom(P(G, L)) = \{y \in \mathbb{R}^E \mid \exists\, x \in P(G, L), x \leq y\}.$$

Let $D(G, L)$ be the polyhedron given by

$$\begin{aligned}
& y(\delta(W)) \geq 2 && \text{for all } st\text{-cut } \delta(W), \\
& y(\delta(W)\backslash\{e\}) \geq 1 && \text{for all } st\text{-cut } \delta(W),\ e \in \delta(W), \qquad (7.1) \\
& y(T) \geq 2 && \text{for all } L\text{-path-cut } T, \\
& y(T\backslash\{e\}) \geq 1 && \text{for all } L\text{-path-cut } T,\ e \in T, \qquad (7.2) \\
& y(e) \geq 0 && \text{for all } e \in E. \qquad (7.3)
\end{aligned}$$

THEOREM 7.1. *For every $L \geq 2$, if $P(G, L) = Q(G, L)$, then $Dom(P(G, L)) = D(G, L)$.*

Proof. We first prove that $Dom(P(G, L)) \subseteq D(G, L)$. Let $y \in Dom(P(G, L))$. Then there exists $\bar{x} \in P(G, L)$ such that $\bar{x} \leq y$. Hence, y satisfies (1.1), (1.3), and (7.3). We show that y also satisfies constraints (7.1) and (7.2).

Consider a constraint $y(\delta(W)\backslash\{e\}) \geq 1$ of type (7.1). As $\bar{x}(\delta(W)) \geq 2$ and $\bar{x}(e) \leq 1$, we have that

$$\begin{aligned}
y(\delta(W)\backslash\{e\}) &\geq \bar{x}(\delta(W)\backslash\{e\}) \\
&= \bar{x}(\delta(W)) - \bar{x}(e) \\
&\geq 2 - \bar{x}(e) \\
&\geq 1.
\end{aligned}$$

Now, in a similar way, we obtain that $y(T\backslash\{e\}) \geq 1$ for all L-path-cut T and $e \in T$. Therefore $Dom(P(G, L)) \subseteq D(G, L)$.

Next we prove that $D(G, L) \subseteq Dom(P(G, L))$. To this end, first let us note that the dominant of $D(G, L)$, $Dom(D(G, L))$, is $D(G, L)$ itself. Thus, to prove that $D(G, L) \subseteq Dom(P(G, L))$, it is sufficient to show that any extreme point $\bar{y}$ of $D(G, L)$ belongs to $P(G, L)$. Indeed, suppose that this is the case. Then any convex combination of extreme points of $D(G, L)$ is also in $P(G, L)$. On the other hand, since $Dom(D(G, L)) = D(G, L)$, any solution $y \in D(G, L)$ can be seen as $\tilde{y} + z$, where $\tilde{y}$ belongs to the convex hull of the extreme points of $D(G, L)$ and $z \geq 0$. As $\tilde{y} \in P(G, L)$, we have therefore that $y \in Dom(P(G, L))$.

So let $\bar{y}$ be an extreme point of $D(G, L)$. As $P(G, L) = Q(G, L)$ and all inequalities in $Q(G, L)$ are in $D(G, L)$ except $x(e) \leq 1, e \in E$, in order to show that $\bar{y} \in P(G, L)$, it suffices to show that $\bar{y}(e) \leq 1$ for all $e \in E$.

Suppose that $\bar{y}(e_0) > 1$ for some $e_0 \in E$. Since $\bar{y}$ is an extreme point of $D(G, L)$, there exists at least one constraint among (1.1), (7.1), (1.3) (7.2) involving the variable $x(e_0)$ and that is tight for $\bar{y}$.

If $\bar{y}(\delta(W)\backslash\{f\}) = 1$ with $e_0 \in \delta(W)\backslash\{f\}$, then, clearly, $\bar{y}(e_0) \leq \bar{y}(\delta(W)\backslash\{f\}) = 1$, a contradiction.

If $\bar{y}(\delta(W)) = 2$ with $e_0 \in \delta(W)$, then $\bar{y}(e_0) + \bar{y}(\delta(W)\backslash\{e_0\}) = 2$, and hence $\bar{y}(e_0) = 2 - \bar{y}(\delta(W)\backslash\{e_0\})$. As $\bar{y}$ satisfies (7.1), it follows that $\bar{y}(e_0) \leq 1$, which is impossible.

We obtain a similar contradiction if one of the constraints (1.3), (7.2) is tight for $\bar{y}$. □

It would be interesting to investigate the dominant of the THPP polytope when $P(G, L) \neq Q(G, L)$.

An immediate consequence of Theorems 3.1 and 7.1 is the following.

COROLLARY 7.2. *If $L = 2, 3$, then $Dom(P(G, L)) = D(G, L)$.*

8. Concluding remarks. In this paper, we have considered the problem of finding a minimum cost edge set containing at least two edge-disjoint paths between two terminals s and t of length no more than L, where $L \geq 2$ is a given integer. We have given a formulation for this problem and extended this formulation to the case where more than two paths are required between s and t. We have also investigated its polyhedral structure when $L = 2, 3$. In particular, we have shown in that case that the associated polytope $P(G, L)$ is described by the trivial, st-cut, and L-path-cut inequalities. Moreover, we have given necessary and sufficient conditions for these inequalities to be facet defining for any $L \geq 2$. This yielded a minimal linear description for $P(G, L)$ when $L = 2, 3$. We have finally considered the dominant of $P(G, L)$, for which we have given a complete description for any $L \geq 2$ when $P(G, L)$ is given by those inequalities.

Since the separation problems for inequalities (1.1) and (1.3) can be solved in polynomial time when $L \leq 3$, from Theorem 3.1 it follows that, for $L \leq 3$, the THPP can be solved in polynomial time using a cutting plane algorithm. To the best of our knowledge, this is the first (nonenumerative) polynomial algorithm devised for this problem.

Let $P_k(G, L)$ be the polytope associated with the problem where the number of edge-disjoint paths k is arbitrary. A natural question that may be posed is whether the linear relaxation of this problem is integral. We have made some investigations in this direction. These motivate us to give the following conjecture.

CONJECTURE 8.1. *$P_k(G, L) = Q_k(G, L)$ if $L = 2, 3$, where $Q_k(G, L)$ is as defined in section 2.*

As already mentioned, if $L \geq 4$, the formulation given in section 2 is no longer valid for the THPP. Unfortunately, so far we do not know a formulation for the problem in that case. However, for $L \leq 3$, it is not hard to see that the formulation given in section 2 for the THPP (and also for its generalization when the number k of required edge-disjoint L-st-paths is more than two) can be easily extended to the case where more than one pair of terminals is considered. Here the formulation is given by the st-cut and L-path-cut inequalities for every pair $\{s, t\}$ of terminals, together with the trivial inequalities. However, these inequalities do not suffice to completely describe the associated polytope for this general case even for $L \leq 3$. In fact, consider the graph shown in Figure 5 with two pairs of terminals $\{s, t\}$ and $\{s', t'\}$. Suppose that $L = 3$. Here, a feasible solution must contain at least two edge-disjoint 3-st-paths and at least two edge-disjoint 3-$s't'$-paths. It is not hard to see that the fractional point $\bar{x} = (1, 1, 1, 1, 0, 0, 0, 1/2, 1/2, 1/2)$ satisfies all trivial, st-cut, and L-path-cut inequalities (with respect to the two pairs of terminals). Moreover, $\bar{x}$ is an extreme point of the polyhedron given by these inequalities. Actually, one can easily see that the inequality

$$x(e_5) + x(e_6) + x(e_7) + x(e_8) + x(e_9) + x(e_{10}) \geq 2$$

is valid for the problem but violated by $\bar{x}$. Furthermore, this inequality is facet

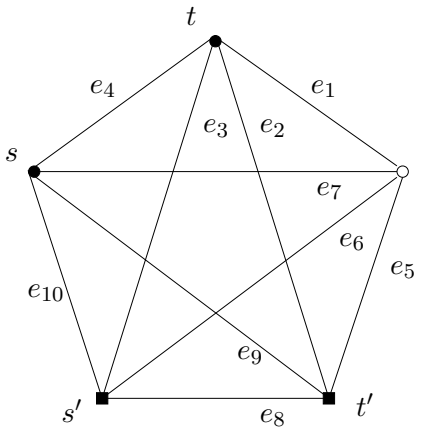


FIG. 5.

defining for the associated polytope.

Finally, let us note that the results given in this paper can be exploited to devise a branch-and-cut algorithm for that general problem when $L = 2, 3$. For this, we should identify further families of facet defining inequalities. These should take into account the interaction between the different pairs of terminals. Our results can also be used to obtain upper bounds for that problem. If $L \leq 3$, one can solve the THPP in the underlying graph G for every pair of terminals using the cutting plane algorithm developed in this paper. Then, by considering the union of the different solutions obtained this way, one get a feasible solution for the problem. This approach can be used to provide upper bounds even when $L \geq 4$. On the other hand, it would be interesting to investigate the extension of the results, related to the formulation of the THPP when $L \leq 3$ as well as the facial structure of its associated polytope, to the more general case when k and L are both arbitrary. This is our aim for future work.

Acknowledgment. We thank the anonymous referees for their valuable comments.

REFERENCES

[1] A. Balakrishnan and K. Altinkemer, *Using a hop-constrained model to generate alternative communication network design*, ORSA J. Comput., 4 (1992), pp. 192–205.

[2] M. Baïou and A. R. Mahjoub, *Steiner 2-edge connected subgraph polytopes on series-parallel graphs*, SIAM J. Discrete Math., 10 (1997), pp. 505–514.

[3] F. Barahona and A. R. Mahjoub, *On two-connected subgraphs polytopes*, Discrete Math., 147 (1995), pp. 19–34.

[4] W. Ben-Ameur, *Constrained length connectivity and survivable networks*, Networks, 36 (2000), pp. 17–33.

[5] W. Ben-Ameur and E. Gourdin, *Internet routing and related topology issues*, SIAM J. Discrete Math., 17 (2003), pp. 18–49.

[6] M. D. Biha and A. R. Mahjoub, *Steiner k-edge connected subgraph polytope*, J. Comb. Optim., 4 (2000), pp. 131–144.

[7] A. Bley, *On the complexity of vertex-disjoint length-restricted path problems*, Comput. Complexity, to appear.

[8] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, Universities Press, Belfast, 1976.

[9] S. C. Boyd and T. Hao, *An integer polytope related to the design of survivable communication networks*, SIAM J. Discrete Math., 6 (1993), pp. 612–630.

[10] C. R. Coullard, A. B. Gamble, and J. Liu, *The k-walk polyhedron*, in Advances in Optimiza-

tion and Approximation, Nonconvex Optim. Appl. 1, D.-Z. Du and J. Sun, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994, pp. 9–29.
[11] G. Dahl, *The* 2*-hop spanning tree problem*, Oper. Res. Lett., 23 (1998), pp. 21–26.
[12] G. Dahl, *Notes on polyhedra associated with hop-constrained paths*, Oper. Res. Lett., 25 (1999), pp. 97–100.
[13] G. Dahl and L. Gouveia, *On Formulations of the Hop-Constrained Minimum Spanning Tree Problem*, preprint, 2001.
[14] G. Dahl and B. Johannessen, *The* 2*-path network problem*, Networks, 43 (2004), pp. 190–199.
[15] J. Fonlupt and A. R. Mahjoub, *Critical extreme points of the* 2*-edge connected spanning subgaph polytope*, in Integer Programming and Combinatorial Optimization, Lecture Notes in Comput. Sci. 1610, G. Cornuéjols, R. E. Burkard, and G. J. Woeginger, eds., Springer–Verlag, Berlin, 1999, pp. 166–182.
[16] B. Fortz, M. Labb , and F. Maffioli, *Solving the two-connected network with bounded meshes problem*, Oper. Res., 48 (2000), pp. 866–877.
[17] B. Fortz, A. R. Mahjoub, S. T. McCormick, and P. Pesneau, *The* 2*-Edge Connected Subgraph Problem with Bounded Rings*, Working paper 98/03, IAG, Université Catholique de Louvain, Louvain-la-Neuve, Belgium, 2003.
[18] H. N. Gabow, M. X. Goemans, and D. P. Williamson, *An efficient approximation algorithm for the survivable network design problem*, *Networks and matroids; Sequencing and scheduling.* Math. Programming, 82 (1998), pp. 13–40.
[19] L. Gouveia, *Multicommodity flow models for spanning trees with hop constraints*, European J. Oper. Res., 95 (1996), pp. 178–190.
[20] L. Gouveia, *Using variable redefinition for computing lower bounds for minimum spanning and Steiner trees with hop constraints*, INFORMS J. Comput., 10 (1998), pp. 180–188.
[21] L. Gouveia and E. Janssen, *Designing reliable tree networks with two cable technologies*, European J. Oper. Res., 105 (1998), pp. 552–568.
[22] L. Gouveia and T. L. Magnanti, *Modelling and Solving the Diameter-Constrained Minimum Spanning Tree Problem*, Technical report, DEIO-CIO, Faculdade de Ciências da Universidade de Lisboa, 2000.
[23] L. Gouveia and C. Requejo, *A new Lagrangean relaxation approach for the hop-constrained minimum spanning tree problem*, European J. Oper. Res., 132 (2001), pp. 539–552.
[24] M. Grötschel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica, 1 (1981), pp. 169–197.
[25] M. Grötschel and C. L. Monma, *Integer polyhedra arising from certain networks design problems with connectivity constraints*, SIAM J. Discrete Math., 3 (1990), pp. 502–523.
[26] M. Grötschel, C. L. Monma, and M. Stoer, *Polyhedral approaches to network survivability*, in Reliability of Computer and Communication Networks, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 5, F. Roberts, F. Hwang, and C. L. Monma eds., AMS, Providence, RI, 1991, pp. 121–141.
[27] M. Grötschel, C. L. 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.
[28] M. Grötschel, C. L. Monma, and M. Stoer, *Design of survivable networks*, in Network Models, Handbooks Oper. Res. Management Sci. 7, M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, eds., North–Holland, Amsterdam, 1995, pp. 617–671.
[29] J. Hao and J. B. Orlin, *A faster algorithm for finding the minimum cut in a directed graph*, J. Algorithms, 17 (1994), pp. 424–446.
[30] A. Itai, Y. Perl, and Y. Shiloach, *The complexity of finding maximum disjoint paths with length constraints*, Networks, 12 (1982), pp. 277–286.
[31] K. Jain, *A factor* 2 *approximation algorithm for the generalized Steiner network problem*, Combinatorica, 21 (2001), pp. 39–60.
[32] H. Kerivin, A. R. Mahjoub, and C. Nocq, $(1, 2)$*-survivable networks: Facets and branch-and-cut*, in The Sharpest Cut: The Impact of Manfred Padberg and His Work, MPS/SIAM Ser. Optim., M. Grötschel, ed., SIAM, Philadelphia, 2004, pp. 121–152.
[33] L. J. Leblanc and R. Reddoch, *Reliable Link Topology/Capacity and Routing in Backbone Telecommunication Networks*, Working paper 90-08, Owen Graduate School of Management, Vanderbilt University, Nashville, TN, 1990.
[34] L. J. Leblanc, R. Reddoch, J. Chifflet, and P. Mahey, *Packet routing in telecommunications networks with path and flow restrictions*, INFORMS J. Comput., 11 (1999), pp. 188–197.
[35] C.-L. Li, S. T. McCormick, and D. Simchi-Levi, *Finding disjoint paths with different path-costs: Complexity and algorithms*, Networks, 22 (1992), pp. 653–667.

[36] A. R. Mahjoub, *Two-edge connected spanning subgraphs and polyhedra*, Math. Programming, 64 (1994), pp. 199–208.
[37] H. Pirkul and S. Soni, *New formulations and solution procedures for the hop constrained network design problem*, European J. Oper. Res., 148 (2003), pp. 126–140.
[38] W. R. Pulleyblank, *Polyhedral combinatorics*, in Optimization, Handbooks Oper. Res. Management Sci. 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. 18, No. 2, pp. 313–319

A NOTE ON THE PROOF OF NIHO'S CONJECTURE*

XIANG-DONG HOU†

Abstract. A longstanding conjecture by Niho on the maximally nonlinearity of certain power functions was proved recently by Hollmann and Xiang using a result of Dobbertin on the almost perfect nonlinearity of the Niho power functions. A key ingredient of the proof, a bound for certain binary weights, was obtained using a computer. In this note, we provide a noncomputer proof for the bound of the binary weights.

Key words. almost perfect nonlinear function, cross-correlation, maximally nonlinear function, Niho's conjecture

AMS subject classifications. 11A63, 11T23, 94A55

DOI. 10.1137/S0895480103432817

1. Introduction.

Let $n = 2m + 1$ be an odd integer. A function $f : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ is called maximally nonlinear if

$$\sum_{x \in \mathbb{F}_{2^n}} (-1)^{\mathrm{Tr}_{\mathbb{F}_{2^n}/\mathbb{F}_2}(af(x)+bx)} = 0, \ \pm 2^{m+1}$$

for all $a, b \in \mathbb{F}_{2^n}$ with $a \neq 0$. If d is an integer with $\gcd(d, 2^n - 1) = 1$, then $f(x) = x^d$ is maximally nonlinear if and only if

$$\sum_{x \in \mathbb{F}_{2^n}} (-1)^{\mathrm{Tr}_{\mathbb{F}_{2^n}/\mathbb{F}_2}(x^d+bx)} = 0, \ \pm 2^{m+1} \quad \text{for all } b \in \mathbb{F}_{2^n}. \tag{1.1}$$

Let α be a primitive element of $\mathbb{F}_{2^n}$. Equation (1.1) is equivalent to the claim that the sequence $\mathrm{Tr}_{\mathbb{F}_{2^n}/\mathbb{F}_2}(\alpha^i)$ and its decimation by d, i.e., $\mathrm{Tr}_{\mathbb{F}_{2^n}/\mathbb{F}_2}(\alpha^{di})$, have exactly three cross-correlation values -1, $-1 \pm 2^{m+1}$. For more details about sequences with preferred cross-correlation values and their applications in communication, see [2], [7], [8], [11], [12], [13].

The well-known Welch and Niho conjectures made in 1972 claim that x^d is maximally nonlinear on $\mathbb{F}_{2^n}$ in the following two cases:

Welch's conjecture (see [12]): $n = 2m + 1$, $d = 2^m + 3$.

Niho's conjecture (see [12]): *n odd,* $d = 2^{2r} + 2^r - 1$, *where* $4r + 1 \equiv 0 \pmod{n}$.

Both conjectures have been proved recently: Welch's conjecture was proved by Canteaut, Charpin, and Dobbertin [3] and by Hollmann and Xiang [9]; Niho's conjecture was proved by Hollmann and Xiang [9].

We briefly describe the idea in the proofs of [3] and [9]. A function $f : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ is called almost perfect nonlinear (APN) if for each $0 \neq a \in \mathbb{F}_{2^n}$ and $b \in \mathbb{F}_{2^n}$, the equation

$$f(x + a) + f(x) = b$$

*Received by the editors August 7, 2003; accepted for publication (in revised form) January 28, 2004; published electronically November 9, 2004. This research was supported by NSA grant MDA 904-02-1-0080.

http://www.siam.org/journals/sidma/18-2/43281.html

†Department of Mathematics, University of South Florida, Tampa, FL 33620 (xhou@mail.cas.usf.edu).

has exactly two solutions $x \in \mathbb{F}_{2^n}$. Chabaud and Vaudenay [4] showed that maximally nonlinear functions are necessarily APN. On the other hand, Canteaut, Charpin, and Dobbertin [3] proved that an APN power function x^d on $\mathbb{F}_{2^n}$ is maximally nonlinear if and only if

$$\sum_{x\in\mathbb{F}_{2^n}} (-1)^{\mathrm{Tr}_{\mathbb{F}_{2^n}/\mathbb{F}_2}(x^d+bx)} \equiv 0 \pmod{2^{m+1}} \tag{1.2}$$

for all $b \in \mathbb{F}_{2^n}$. A major step in the proofs of the Welch and Niho conjectures is the following result due to Dobbertin.

THEOREM 1.1 (see [5], [6]). *The Welch power functions and the Niho power functions are APN.*

Therefore, to prove the Welch and Niho conjectures, it suffices to prove (1.2) for the Welch and Niho exponents d.

For each $a \in \mathbb{Z}$ with $a \not\equiv 0 \pmod{2^n-1}$, write

$$a \equiv \sum_{i=0}^{n-1} a_i 2^i \pmod{2^n-1}, \quad a_i \in \{0,1\},$$

and define

$$w(a) = \sum_{i=0}^{n-1} a_i.$$

A theorem of McEliece [10] on the weights of cyclic codes shows that (1.2) holds for all $b \in \mathbb{F}_{2^n}$ if and only if

$$w(da) - w(a) \le m \tag{1.3}$$

for all $0 < a < 2^n - 1$. (For readers familiar with Ax's work [1], the equivalence between (1.2) and (1.3) can be seen more directly without involving cyclic codes.) So the final step in the proofs of the Welch and Niho conjectures is the establishment of the bound (1.3).

In [9], the bound (1.3) in the Niho case was obtained through a computer analysis of a weighted digraph on 1296 vertices. The purpose of this note is to provide a noncomputer proof of the bound (1.3) in the Niho case.

2. A noncomputer proof of $w(da) - w(a) \le m$ in the Niho case. Let $n = 2m+1 > 0$ be an odd integer and $d = 2^{2r} + 2^r - 1$, where $4r + 1 \equiv 0 \pmod{n}$. The goal is to prove that for each integer $0 < a < 2^n - 1$,

$$w(da) - w(a) \le m. \tag{2.1}$$

We shall follow the idea of [9]. A sequence $\{u_i\}_{i\in\mathbb{Z}}$ is called periodic with period n if $u_i = u_j$ whenever $i \equiv j \pmod{n}$. All sequences in this section are periodic with period n. Write

$$a = \sum_{i=0}^{n-1} a_i 2^i, \quad a_i \in \{0,1\},$$

and extend $a_0, \ldots, a_{n-1}$ to a periodic sequence with period n. We have

$$da = \sum_{i=0}^{n-1} (-a_i + a_{i-r} + a_{i-2r}) 2^i \equiv \sum_{i=0}^{n-1} b_i 2^i \pmod{2^n-1},$$

where

$$b_i = 1 - a_i + a_{i-r} + a_{i-2r} \in \{0, 1, 2, 3\}.$$

Write

$$\sum_{i=0}^{n-1} b_i 2^i \equiv \sum_{i=0}^{n-1} s_i 2^i \pmod{2^n - 1}$$

with $s_i \in \{0, 1\}$. By Theorem 13 of [9], there exists a sequence $\{c_i\}$ such that

$$s_i = b_i - 2c_i + c_{i-1} \tag{2.2}$$

and $c_i \in \{0, 1, 2\}$. c_i is the carryover from the ith binary digit of $\sum_{i=0}^{n-1} b_i 2^i$ to its next binary digit. Note that

$$\begin{aligned} & w(da) - w(a) \\ &= \sum_{i=0}^{n-1} s_i - \sum_{i=0}^{n-1} a_i \\ &= \sum_{i=0}^{n-1} (1 - a_i + a_{i-r} + a_{i-2r} - 2c_i + c_{i-1}) - \sum_{i=0}^{n-1} a_i \\ &= n - \sum_{i=0}^{n-1} c_i. \end{aligned}$$

Thus (2.1) is equivalent to

$$\sum_{i=0}^{n-1} c_i \geq m + 1. \tag{2.3}$$

Since $\gcd(r, n) = 1$, the sequence $c_0, c_r, c_{2r}, \ldots, c_{(n-1)r}$ is a rearrangement of $c_0, c_1, \ldots, c_{n-1}$. Therefore, it suffices to show that for each $i \in \mathbb{Z}$, there exists an integer $k \geq 0$ such that

$$c_i + c_{i+r} + \cdots + c_{i+kr} \geq \frac{k+1}{2}.$$

(In fact, we will see that $k \leq 3$.) In the proof, we will frequently use the table and facts listed below.

FACT 1. *If $c_i \geq 1$ and $b_{i-4r}, b_{i-8r}, \ldots, b_{i-4kr}$ are all ≥ 1, then $c_{i-4kr} \geq 1$.*

FACT 2. *If $c_i \geq 2$ and $b_{i-4r}, b_{i-8r}, \ldots, b_{i-4(k-1)r}$ are all ≥ 2, then $c_{i-4kr} \geq 1$.*

To see Fact 1, observe that $b_{i-4r} \geq 1$ and $c_{i-4r-1} = c_i \geq 1$ imply $c_{i-4r} \geq 1$. Proceed inductively to arrive at $c_{i-4kr} \geq 1$. Fact 2 follows from the same observation. Put

$$A = \begin{bmatrix} a_0 & a_r & \cdots & a_{(n-1)r} \\ b_0 & b_r & \cdots & b_{(n-1)r} \end{bmatrix}.$$

The proof in this section is based on the analysis of the matrix A.

LEMMA 2.1. *There does not exist a string*

$$a_i a_{i+r} \cdots a_{i+kr} = 1\ 1\ 0\ *\ \cdots\ *\ 0\ 0\ 1 \tag{2.4}$$

TABLE 1
Values of $a_{i-2r}, a_{i-r}, a_i,\ b_i$ *and* c_i.

a_{i-2r}	a_{i-r}	a_i	b_i	c_i
0	0	0	1	≤ 1
1	0	0	2	≥ 1
0	1	0	2	≥ 1
0	0	1	0	≤ 1
0	1	1	1	≤ 1
1	0	1	1	≤ 1
1	1	0	3	≥ 1
1	1	1	2	≥ 1

such that $b_i \geq 1,\ c_i = c_{i+r} = 0,\ c_{i+2r} = 1$, *and* $k \equiv 0 \pmod 4$.

Proof. Suppose to the contrary that such a string exists. Let the string in (2.4) be of the shortest length. For convenience, assume $i = 0$. For Table 1, we have $b_r \geq 1$ and $b_{2r} = 3$. Thus

$$A = \begin{bmatrix} 1 & 1 & 0 & a_{3r} & a_{4r} & a_{5r} & a_{6r} & \cdots \\ \geq 1 & \geq 1 & 3 & b_{3r} & b_{4r} & b_{5r} & b_{6r} & \cdots \end{bmatrix}.$$

Since $c_0 = c_r = 0$ and $b_0 \geq 1$, $b_r \geq 1$, by Fact 1 we have $c_{4r} = c_{5r} = 0$; hence $b_{4r} \leq 1$, $b_{5r} \leq 1$. We claim that $b_{6r} \leq 1$. Otherwise, $c_{6r} \geq 1$, i.e., $c_{2r-1} \geq 1$. Since $b_{2r} = 3$, by (2.2) we must have $c_{2r} \geq 2$, which is a contradiction. Thus we have

$$A = \begin{bmatrix} 1 & 1 & 0 & a_{3r} & a_{4r} & a_{5r} & a_{6r} & \cdots \\ \geq 1 & \geq 1 & 3 & & \leq 1 & \leq 1 & \leq 1 & \cdots \end{bmatrix}.$$

Since b_{4r}, b_{5r}, b_{6r} are all ≤ 1, a quick inspection of Table 1 reveals that $a_{3r} = 0$ and $(a_{4r}, a_{5r}, a_{6r}) = (0,1,1)$ or $(0,0,1)$ or $(0,0,0)$.

Case (i) $(a_{4r}, a_{5r}, a_{6r}) = (0,1,1)$. By Fact 1,

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 & a_{7r} & a_{8r} & \cdots \\ \geq 1 & \geq 1 & 3 & 2 & 1 & 0 & 1 & & \leq 1 & \end{bmatrix}.$$

Since $b_{8r} \leq 1$, it is clear that $(a_{7r}, a_{8r}) = (0,1)$. Thus by Fact 1,

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & a_{9r} & a_{10r} & \cdots \\ \geq 1 & \geq 1 & 3 & 2 & 1 & 0 & 1 & 3 & 1 & & \leq 1 & \end{bmatrix}.$$

The same argument shows that $a_{9r}a_{10r}a_{11r}a_{12r}\cdots = 0\,1\,0\,1\cdots$, which contradicts (2.4).

Case (ii) $(a_{4r}, a_{5r}, a_{6r}) = (0,0,1)$. By Fact 1,

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 & a_{7r} & a_{8r} & a_{9r} & a_{10r} & \cdots \\ \geq 1 & \geq 1 & 3 & 2 & 1 & 1 & 0 & & \leq 1 & \leq 1 & & \end{bmatrix}.$$

Clearly, $(a_{7r}, a_{8r}, a_{9r}) = (0,1,1)$. By Table 1, $b_{8r} = 1$ and $c_{10r} \geq 1$. We claim that $a_{10r} = 1$. In fact, since $c_0 = c_r = 0$, by Fact 1 we have $c_{8r} = c_{9r} = 0$. Since $b_{2r} = 3$ and $c_{2r} = 1$, by (2.2) we must have $c_{6r} = c_{2r-1} = 0$. It follows from Fact 2 that $c_{10r} \leq 1$. Thus $c_{10r} = 1$. If, to the contrary, $a_{10r} = 0$, then $a_{8r}a_{9r}\cdots a_{kr}$ is a shorter string having the same properties as (2.4). Since string (2.4) is the shortest of its kind, we have a contradiction. Thus we have proved that $a_{10r} = 1$. By Fact 1,

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & a_{11r} & a_{12r} & a_{13r} & a_{14r}\cdots \\ \geq 1 & \geq 1 & 3 & 2 & 1 & 1 & 0 & 2 & 1 & 1 & 2 & & \leq 1 & \leq 1 & \end{bmatrix}.$$

The same argument shows that

$$a_{11r}\cdots a_{14r}\,a_{15r}\cdots a_{18r}\cdots = 0\,1\,1\,1\;\;0\,1\,1\,1\;\cdots,$$

which contradicts (2.4).

Case (iii) $(a_{4r}, a_{5r}, a_{6r}) = (0,0,0)$. Let $l > 1$ be the smallest integer such that $(a_{4lr}, a_{(4l+1)r}, a_{(4l+2)r}) \neq (0,0,0)$. By Fact 1,

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & & a_{4lr} & a_{(4l+1)r} & a_{(4l+2)r} & \cdots \\ \geq 1 & \geq 1 & 3 & 2 & B & \leq 1 & \leq 1 & \leq 1 & \end{bmatrix},$$

where B is made of 2×4 blocks of the form

$$\begin{bmatrix} 0 & 0 & 0 & * \\ \geq 1 & \geq 1 & \geq 1 & * \end{bmatrix}.$$

Since $b_{4lr}, b_{(4l+1)r}$ and $b_{(4l+2)r}$ are all ≤ 1, we must have $(a_{4lr}, a_{(4l+1)r}, a_{(4l+2)r}) = (0,1,1)$ or $(0,0,1)$. If $(a_{4lr}, a_{(4l+1)r}, a_{(4l+2)r}) = (0,1,1)$, the argument in Case (i) applies. If $(a_{4lr}, a_{(4l+1)r}, a_{(4l+2)r}) = (0,0,1)$, the argument in Case (ii) applies. Therefore the proof of the lemma is complete. □

COROLLARY 2.2. *If there exists $i \in \mathbb{Z}$ such that $a_i a_{i+r} a_{i+2r} = 1\,1\,0$, $b_i \geq 1$, $c_i = c_{i+r} = 0$, and $c_{i+2r} = 1$, then the sequence $a_0 a_r a_{2r} \cdots$ does not contain a string $0\,0$.*

Proof. Otherwise, there exists $l > 0$ such that $a_{i+lr} a_{i+(l+1)r} a_{i+(l+2)r} = 0\,0\,1$. Let $t > 0$ be an integer such that $l + 2 + tn \equiv 0 \pmod 4$, and put $k = l + 2 + tn$. Then $a_i a_{i+r} \cdots a_{i+kr} = 1\,1\,0 * \cdots * 0\,0\,1$, which is impossible by Lemma 2.1. □

Now we are ready to prove the main result.

THEOREM 2.3. *For each $i \in \mathbb{Z}$, there exists an integer $0 \leq k \leq 3$ such that*

$$c_i + c_{i+r} + \cdots + c_{i+kr} \geq \frac{k+1}{2}. \tag{2.5}$$

Proof. For convenience, let $i = 0$. Assume to the contrary that such a k does not exist. We must have $c_0 = 0$ since otherwise (2.5) is satisfied with $k = 0$. We have from Table 1 that $(a_{-2r}, a_{-r}, a_0) = (0,0,0)$ or $(0,0,1)$ or $(1,0,1)$ or $(0,1,1)$. The possibility of $(0,1,1)$ is immediately dismissed since otherwise $b_r \geq 2$ and $c_r \geq 1$.

Case 1. $(a_{-2r}, a_{-r}, a_0) = (0,0,0)$. We can write

$$A = \begin{bmatrix} 0 & \cdots & 0 & 1 & a_{lr} & a_{(l+1)r} & a_{(l+2)r} & \cdots & 0 & 0 \\ 1 & \cdots & 1 & 0 & & & & & & \end{bmatrix}$$

for some $l \geq 2$. We claim that at most one of $b_{lr}, b_{(l+1)r}, b_{(l+2)r}$ is ≥ 2. (Otherwise, by Fact 1, either one of c_0, c_r is ≥ 1 or two of c_0, c_r, c_{2r}, c_{3r} are ≥ 1.) An inspection of Table 1 shows that $(a_{lr}, a_{(l+1)r}, a_{(l+2)r}) = (1,0,1)$ or $(0,1,1)$.

Case 1.1. $(a_{lr}, a_{(l+1)r}, a_{(l+2)r}) = (1,0,1)$, i.e.,

$$A = \begin{bmatrix} 0 & \cdots & 0 & 1 & 1 & 0 & 1 & a_{(l+3)r} & \cdots & 0 & 0 \\ 1 & \cdots & 1 & 0 & 1 & 3 & 1 & & & & \end{bmatrix}.$$

We must have $c_{lr} = c_{(l+2)r} = 0$. (Otherwise, two of $c_{lr}, c_{(l+1)r}, c_{(l+2)r}$ are ≥ 1. By Fact 1, either one of c_0, c_r is ≥ 1 or two of c_0, c_r, c_{2r}, c_{3r} are ≥ 1.) By the argument in Case (i) of the proof of Lemma 2.1, $a_{(l+3)r} a_{(l+4)r} \cdots = 0\,1\,0\,1\cdots$, which is impossible.

Case 1.2. $(a_{lr}, a_{(l+1)r}, a_{(l+2)r}) = (0,1,1)$, i.e.,

$$A = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & 1 & 1 & a_{(l+3)r} & \cdots & 0 & 0 \\ 1 & \cdots & 1 & 0 & 2 & 1 & 1 & & & & \end{bmatrix}.$$

We must have $c_{(l+1)r} = c_{(l+2)r} = 0$. (Otherwise, two of $c_{lr}, c_{(l+1)r}, c_{(l+2)r}$ are ≥ 1, which is impossible.) We must also have $c_{(l+3)r} \leq 1$. (Otherwise, by Fact 2, $c_{(l-1)r} \geq 1$. It is impossible to have both $c_{(l-1)r}$ and $c_{lr} \geq 1$.) By Corollary 2.2, $a_{(l+3)r} \neq 0$. By Fact 1, we have

$$A = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & 1 & 1 & 1 & a_{(l+4)r} & a_{(l+5)r} & a_{(l+6)r} & a_{(l+7)r} & \cdots & 0 & 0 \\ 1 & \cdots & 1 & 0 & 2 & 1 & 1 & 2 & & \leq 1 & \leq 1 & & & & \end{bmatrix}.$$

Since $b_{(l+5)r} \leq 1$ and $b_{(l+6)r} \leq 1$, we see that $(a_{(l+4)r}, a_{(l+5)r}, a_{(l+6)r}) = (0,1,1)$. By Fact 2, $c_{(l+7)r} \leq 1$ and by Corollary 2.2, $a_{(l+7)r} = 1$. Thus $a_{(l+4)r} \cdots a_{(l+7)r} = 0\,1\,1\,1$. In the same way, we have

$$a_{(l+4)r} a_{(l+5)r} \cdots = 0\,1\,1\,1\ \ 0\,1\,1\,1\ \cdots,$$

which is impossible.

Case 2. $(a_{-2r}, a_{-r}, a_0) = (0,0,1)$ or $(1,0,1)$. We have

$$A = \begin{bmatrix} 1 & a_r & a_{2r} & a_{3r} & \cdots & 0 \\ & \leq 1 & & & & \end{bmatrix}.$$

Since $b_r \leq 1$, we have $a_r = 1$. Since at most one of b_{2r}, b_{3r} is ≥ 2, we see that $(a_{2r}, a_{3r}) = (0,1)$. Thus

$$A = \begin{bmatrix} 1 & 1 & 0 & 1 & a_{4r} & \cdots & 0 \\ & 1 & 3 & 1 & & & \end{bmatrix}. \tag{2.6}$$

We claim that $a_{4r} a_{5r} \cdots a_{(n-1)r}$ contains a string $0\,0$. (Otherwise, it is clear from (2.6) that $b_i \geq 1$ for all i. Since $c_{2r} \geq 1$, Fact 1 implies that $c_i \geq 1$ for all i, which is impossible.) Thus we can write

$$A = \begin{bmatrix} 1 & 1 & 0 & 1 & * & \cdots & * & 1 & 0 & 0 & a_{tr} & \cdots \\ & 1 & 3 & 1 & \geq 1 & \cdots & \geq 1 & \geq 1 & \geq 2 & \geq 2 & & \end{bmatrix},$$

where $* \cdots *$ does not contain a string $0\,0$. Since both $c_{(t-2)r}$ and $c_{(t-1)r}$ are ≥ 1, by Fact 1 at least one of c_r and c_{3r} is ≥ 1. Meanwhile, since $b_{2r} = 3$, we have $c_{2r} \geq 1$. We have a contradiction since (2.5) is satisfied with $k = 3$. □

REFERENCES

[1] J. Ax, *Zeros of polynomials over finite fields*, Amer. J. Math., 86 (1964), pp. 255–261.

[2] A. R. Calderbank, G. McGuire, B. Poonen, and M. Rubinstein, *On a conjecture of Helleseth regarding pairs of binary m-sequences*, IEEE Trans. Inform. Theory, 42 (1996), pp. 988–990.

[3] A. Canteaut, P. Charpin, and H. Dobbertin, *Binary m-sequences with three valued cross-correlation: A proof of Welch's conjecture*, IEEE Trans. Inform. Theory, 46 (2000), pp. 4–8.

[4] F. Chabaud and S. Vaudenay, *Links between differential and linear cryptanalysis*, in Advances in Cryptology—EUROCRYPT '94, Lecture Notes in Comput. Sci. 950, A. De Santis, ed., Springer-Verlag, New York, Berlin, 1995, pp. 356–365.

[5] H. Dobbertin, *Almost perfect nonlinear power functions on* GF(2^n)*: The Welch case*, IEEE Trans. Inform. Theory, 45 (1999), pp. 1271–1275.

[6] H. Dobbertin, *Almost perfect nonlinear power functions on* GF(2^n)*: The Niho case*, Inform. and Comput., 151 (1999), pp. 57–72.
[7] S. W. Golomb, *Theory of transformation groups of polynomials over* GF(2) *with applications to linear shift register sequences*, Inform. Sci., 1 (1968), pp. 87–109.
[8] T. Helleseth, *Some results about the cross-correlation function between two maximal linear sequences*, Discrete Math., 16 (1976), pp. 209–232.
[9] H. Hollmann and Q. Xiang, *A proof of the Welch and Niho conjectures on cross-correlations of binary m-sequences*, Finite Fields Appl., 7 (2001), pp. 253–286.
[10] R. J. McEliece, *Weight congruence for p-ary cyclic codes*, Discrete Math., 3 (1972), pp. 177–192.
[11] G. McGuire and A. R. Calderbank, *Proof of a conjecture of Sarwate and Pursley regarding pairs of binary m-sequences*, IEEE Trans. Inform. Theory, 41 (1995), pp. 1153–1155.
[12] Y. Niho, *Multi-valued Cross-Correlation Functions Between Two Maximal Linear Recursive Sequences*, Ph. D. thesis, University of Southern California, Los Angeles, CA, 1972.
[13] D. V. Sarwate and M. B. Pursley, *Cross-correlation properties of pseudorandom and related sequences*, Proc. IEEE, 68 (1980), pp. 593–619.

SIAM J. DISCRETE MATH.
Vol. 18, No. 2, pp. 320–331

DETERMINISTIC HYPERGRAPH COLORING AND ITS APPLICATIONS*

CHI-JEN LU†

Abstract. Given a hypergraph and a set of colors, we want to find a vertex coloring to minimize the size of any monochromatic set in an edge. We give a deterministic polynomial time approximation algorithm with performance close to the best bound guaranteed by an existential argument. This can be applied to support divide and conquer approaches to various problems. We give two examples. For deterministic DNF approximate counting, this helps us explore the importance of a previously ignored parameter, the maximum number of appearances of any variable, and we construct algorithms that are particularly good when this parameter is small. For partially ordered sets, we are able to constructivize the dimension bound given by Füredi and Kahn [*Order*, 3 (1986), pp. 15–20].

Key words. hypergraph, approximation algorithm, Lovász local lemma, derandomization, approximate counting, partially ordered set

AMS subject classifications. 05C65, 68R05, 68W25

DOI. 10.1137/S0895480100367664

1. Introduction.

A hypergraph $H(V, E)$ consists of a set V of vertices and a set E of edges, where each edge is a subset of vertices. An undirected graph is just a hypergraph where each edge contains exactly two vertices. There are four natural parameters associated with a hypergraph:

- $n = |V|$, the number of vertices.
- $m = |E|$, the number of edges.
- $t = \max_{e \in E} |e|$, the size of the largest edge. We will write $E \subseteq V^{\leq t}$.
- $d = \max_{v \in V} deg(v)$, where $deg(v) = |\{e \in E : v \in e\}|$. We call d the degree of H.

Given k colors, we want to color vertices so that no color appears more than c times in any edge. Call such a coloring a (k, c)-coloring. The objective here is to minimize c. Clearly, for a (k, c)-coloring, $c \geq \mu \equiv \frac{t}{k}$, and we want to have c as small and close to μ as possible. This problem was studied before by Srinivasan [13], who gave the following (nonconstructive) existential bound for c:

$$c = \begin{cases} O(\mu) & \text{if } \mu = \Omega(\log d), \\ O\left(\frac{\log d}{\log((\log d)/\mu)}\right) & \text{otherwise.} \end{cases}$$

In this paper, we give a simple deterministic polynomial time algorithm for finding a coloring with

$$c = \begin{cases} O(\mu) & \text{if } \mu = \Omega(\log(td)), \\ O\left(\frac{\log(dt)}{\log((\log(dt))/\mu)}\right) & \text{otherwise.} \end{cases} \tag{1}$$

When $t = d^{O(1)}$ or $\mu = \Omega(\log(td))$, our bound for c is within a constant factor of the bound given by Srinivasan. For applications we consider here, our bound

*Received by the editors February 13, 2000; accepted for publication (in revised form) February 9, 2004; published electronically November 9, 2004. A preliminary version of this paper appeared in *Proceedings of the 2nd International Workshop on Randomization and Approximation Techniques in Computer Science*, 1998, pp. 35–46.
http://www.siam.org/journals/sidma/18-2/36766.html
†Institute of Information Science, Academia Sinica, Taipei, Taiwan (cjlu@iis.sinica.edu.tw).

suffices. After the appearance of the conference version of our paper, Leighton, et al. [7] succeeded in constructivizing Srinivasan's bound for c but using a much more involved method.

This hypergraph coloring problem is NP-complete, as the standard NP-complete graph coloring problem can be reduced to it. Our algorithm can be seen as an approximation algorithm to this problem, which guarantees an approximation ratio $O(1)$ when $\mu = \Omega(\log(td))$ and an approximation ratio $O(\log(td))$ in general. We do not know of any nonapproximability result for this problem. Ahuja and Srivastav [1] studied a similar hypergraph coloring problem, which differs from ours in that c is given and the objective is to minimize k, the number of colors. For the case of $c = \Omega(\log d)$, they gave an approximation algorithm with an approximation ratio $(1 + \delta)$ for any constant $\delta \in (0, 1)$. For $c = O(1)$, they showed that no polynomial time algorithm can achieve an approximation ratio $m^{\frac{1}{2}-\delta}$, for any constant $\delta > 0$, unless NP $\subseteq$ ZPP.

Notice that we can use a k-coloring to partition the original hypergraph into k subhypergraphs, one for each color, in a natural way. Then with a (k, c)-coloring, we can guarantee that any edge in each subhypergraph has at most c vertices. This turns out to be useful for supporting some divide and conquer approaches. We will give two examples.

Our first application is to the deterministic DNF approximate counting problem. Given a DNF formula F of n variables and m terms, we want to estimate its *volume*, defined as

$$vol(F) = \Pr_{x \in \{0,1\}^n}[F(x) = 1],$$

within an additive error ε. Luby, Velicković, and Wigderson [9], following the work of Nisan [10], and Nisan and Wigderson [11], gave a deterministic $2^{O(\log^4 \frac{nm}{\varepsilon})}$ time algorithm. Luby and Velicković [8] gave a deterministic $2^{(\log \frac{m \log n}{\varepsilon})(\frac{1}{\varepsilon})(2^{O(\sqrt{\log\log \frac{m}{\varepsilon}})})}$ time algorithm, which is good when a large error ε is allowed. Note that a DNF formula F can be naturally modelled by a hypergraph with vertices corresponding to variables and edges corresponding to terms. Now the degree d of the hypergraph indicates the maximum number of times a variable appears in F. This parameter has not received attention before for this problem, and it is our focus here. We construct two deterministic algorithms, one with running time

$$2^{O((\log \frac{mn}{\varepsilon})(\log^3 \frac{d \log m}{\varepsilon}))}$$

and the other with running time

$$2^{(\log \frac{m \log n}{\varepsilon})(\log \frac{1}{\varepsilon})\left(2^{O\left(\sqrt{\log(d \log \frac{m}{\varepsilon})}\right)}\right)}.$$

Note that d is at most m, so our first algorithm is never worse than that of Luby, Velicković, and Wigderson [9] and is particularly good when d is small and ε is large. Our second algorithm is better than that of Luby and Velicković [8] when $d \leq 2^{O(\log^2 \frac{1}{\varepsilon})}$ and is better than our first algorithm when $d \leq 2^{O((\log\log \frac{m}{\varepsilon})^2)}$.

Our second application is to constructivize some dimension bound of partially ordered sets (posets). Let $(P, <)$ be a poset. Its dimension, denoted as $\dim(P)$, is defined to be the minimum number of linear extensions $L_1, \ldots, L_d$ such that $P = L_1 \cap \cdots \cap L_d$ (i.e., $x < y$ if and only if $x <_{L_i} y$ for all i). For $x \in P$, let $U(x) = \{y \in P : y \geq x\}$ be the set of upper bounds for x, and let $C(x) = \{y \in P : y \geq x \text{ or } y \leq x\}$

be the set of elements comparable to x. Füredi and Kahn [6] gave the following existential bound: for some constants c_1 and c_2,

$$\dim(P) \le r \equiv \min\{c_1 t \log^2 t, c_2 u \log |P|\},$$

where $t \equiv \max_{x \in P} |C(x)|$ and $u \equiv \max_{x \in P} |U(x)|$. One key ingredient in their proof is the hypergraph coloring problem, where a poset $(P, <)$ is modelled by a hypergraph $H(V, E)$ with $V = P$ and $E = \{U(x) : x \in P\}$. Using our hypergraph coloring algorithm, together with other ideas, we are able to constructivize their existential bound. That is, we give a deterministic polynomial time algorithm for finding $O(r)$ linear extensions with intersection equal to the given poset.

We believe that there should be more applications of our hypergraph coloring algorithm.

2. Hypergraph coloring. Consider a hypergraph $H(V, E)$ with $n = |V|$, $m = |E|$, $E \subseteq V^{\le t}$, and degree d. We want to color its vertices with k colors such that no edge contains c vertices of the same color. When $k \ge (t-1)d + 1$, the task is easy according to the following lemma, so we assume $k < (t-1)d + 1$ from now on.

LEMMA 2.1 (see [6]). *Given a hypergraph $H(V, E)$ with $E \subseteq V^{\le t}$ and degree d, there exists a coloring with $(t-1)d+1$ colors such that no edge has two vertices of the same color. Furthermore, such a coloring can be found by a simple greedy algorithm in deterministic polynomial time.*

Given a coloring on H, call an edge *bad* if it contains c vertices of the same color, and call a set of edges bad if all edges in it are bad. Our goal is to find a *good* k-coloring γ such that no edge is bad. If we choose γ randomly, then for any edge e,

$$\Pr_{\gamma}[e \text{ is bad}] \le \binom{t}{c} \left(\frac{1}{k}\right)^c k \le \left(\frac{3t}{ck}\right)^c k.$$

From the Lovász local lemma [4], there exists a good k-coloring, provided $(\frac{3t}{ck})^c kdt \le 1/4$. However, a random k-coloring could be good with exponentially small probability, and it is not obvious how to find such a good coloring, even probabilistically. Beck [3] had the first success in derandomizing the local lemma, and Alon [2] later adapted Beck's idea to derandomize more applications of the local lemma. We will follow their approach closely.

Our main result in this section is a deterministic polynomial time algorithm to find a good k-coloring satisfying

$$\left(\frac{9t}{ck}\right)^c \left(\frac{k}{3}\right) (dt)^4 \le \frac{1}{4},$$

which leads to the bound for c given in (1). For this value of c, a random $\frac{k}{3}$-coloring turns an edge e bad with probability less than $p \equiv (\frac{1}{dt})^4$. We first give a randomized algorithm and then derandomize it.

2.1. A randomized algorithm. There will be at most three phases, each using a distinct set of $\frac{k}{3}$ colors. The intuition is that a random $\frac{k}{3}$-coloring is unlikely to have a large "cluster" of bad edges and that each bad cluster can be recolored separately for a proper definition of cluster. For a hypergraph H, its line graph L_H is the graph where nodes of L_H correspond to edges of H and two nodes of L_H are adjacent if

and only if the two corresponding edges in H intersect.[1] Let $L_H^{(a,b)}$ be the graph derived from L_H, which has the same node set as L_H, but now two nodes of $L_H^{(a,b)}$ are adjacent if and only if their distance is exactly a or b in L_H. Call a set of edges in H a $(1,2)$-tree if the corresponding nodes in $L_H^{(1,2)}$ are connected. Call a set of edges in H a $(2,3)$-tree if the corresponding nodes in $L_H^{(2,3)}$ are connected but no two corresponding nodes are adjacent in L_H.

Our algorithm consists of phases. In the first phase, we find a $\frac{k}{3}$-coloring such that all bad $(1,2)$-trees have size $O(dt \log m/\log(dt))$. In the second phase, we use a new set of $\frac{k}{3}$ colors and try to recolor each bad $(1,2)$-tree separately. If m is small, the recoloring can be done successfully. Otherwise, we need another phase, using another set of $\frac{k}{3}$ colors.

Phase 1. In this phase, we will find a $\frac{k}{3}$-coloring such that all bad $(1,2)$-trees have size $O(dt \log m/\log(dt))$. First we need the following lemma.

LEMMA 2.2. *For some $u = O(\log m/\log(dt))$, the probability that a random $\frac{k}{3}$-coloring has a bad $(2,3)$-tree of size u is $(1/m)^{\Omega(1)}$.*

Proof. Any two edges of a $(2,3)$-tree have no vertex in common, so the events of each being bad are independent. Since there are at most $\frac{m}{((dt)^3-1)u+1}\binom{(dt)^3u}{u}$ $(2,3)$-trees of size u and each one is bad with probability at most p^u, the probability that a random $\frac{k}{3}$-coloring has a bad $(2,3)$-tree of size u is less than

$$m(3(dt)^3p)^u \le m\left(\frac{3}{dt}\right)^u = \left(\frac{1}{m}\right)^{\Omega(1)}. \qquad \square$$

As a $(1,2)$-tree of size dtu must contain a $(2,3)$-tree of size u, a random $\frac{k}{3}$-coloring with high probability will have no bad $(1,2)$-tree of size $dtu = O(dt \log m/\log(dt))$. In the next section, we will show how to find such a $\frac{k}{3}$-coloring deterministically, using the standard technique of conditional probability with a pessimistic estimator.

Phase 2. Suppose we have found a $\frac{k}{3}$-coloring with no bad $(1,2)$-tree of size dtu. Next, we try to recolor each *maximal* bad (1,2)-tree. Let $T = (V_T, E_T)$ be a maximal bad $(1,2)$-tree. When we recolor vertices in T, those good edges intersecting T are also affected, and we want to make sure that they will not turn bad after the recoloring. So together with T, we also take into account those good edges but with vertices not in T removed. That is, we consider the coloring problem for the hypergraph $S = (V_S, E_S)$ with $V_S = V_T$ and $E_S = \{e \cap V_S : e \in E_H, e \cap V_S \neq \emptyset\}$. It is easy to see that the condition of the local lemma still holds and a good $\frac{k}{3}$-coloring exists for S. We will use a different set of $\frac{k}{3}$ colors in this phase. If we can find a good $\frac{k}{3}$-coloring for S, then after this recoloring, no edge of H intersecting T is bad. Now as each edge of H intersects at most one maximal bad $(1,2)$-tree (from the definition of maximal $(1,2)$-tree), we can repeat this recoloring process for each maximal bad $(1,2)$-tree, using the same new set of $\frac{k}{3}$ colors.

Note that $|E_S| \le (dt)^2u$. Suppose $\sqrt{\log m/\log\log m} \le dt$. Then the probability that a random $\frac{k}{3}$-coloring has a bad edge in S is at most

$$|E_S|p < (dt)^4\left(\frac{1}{dt}\right)^4 = 1.$$

[1] Note that, to avoid confusion, we use the term *node* instead of *vertex* for graphs and call two nodes of a graph *adjacent* instead of using the term *edge*. The terms *vertex* and *edge* are reserved for hypergraphs.

In this case, we can find a good $\frac{k}{3}$-coloring in deterministic polynomial time, using again the technique of conditional probability.

Otherwise, we have $dt < \sqrt{\log m/\log\log m}$. As in Phase 1, we now can find a $\frac{k}{3}$-coloring such that all bad $(1,2)$-trees have size at most

$$O(dt\log((dt)^2\log m)/\log(dt)) = O\left(\sqrt{\log m\log\log m}/\log(dt)\right).$$

Then we enter Phase 3.

Phase 3. Now as $t \le \sqrt{\log m/\log\log m}$, each bad $(1,2)$-tree contains

$$t\cdot O\left(\sqrt{\log m\log\log m}/\log(dt)\right) = O(\log m/\log(dt))$$

vertices of the hypergraph H. So we can use an exhaustive search to find a good $\frac{k}{3}$-coloring in deterministic time $(\frac{k}{3})^{O(\log m/\log(dt))}$, which is $m^{O(1)}$, as we can assume $k = O(dt)$ due to Lemma 2.1.

2.2. Derandomization of Phase 1. Let R denote the set of all $(2,3)$-trees of size $u = O(\frac{\log m}{\log(dt)})$, and let B denote the event that some tree in R is bad. From Lemma 2.2, we know that the bad event B is unlikely to happen under a random $\frac{k}{3}$-coloring. But how do we find a good coloring deterministically? The idea is to use the standard technique of conditional probability with a pessimistic estimator, introduced by Raghavan [12]. We want to color vertices one by one. The color of each vertex is chosen to minimize the probability of having the bad event B if we randomly $\frac{k}{3}$-color the remaining vertices. The hope is that the final coloring is a good one because the final conditional probability, which is either 0 or 1, is at most the original unconditional one, which is less than 1. However, it is not easy to compute the exact conditional probability at each step here. So we use a pessimistic estimator instead.

Suppose that we have already assigned colors $\gamma_1,\dots,\gamma_i$ to vertices $v_1,\dots,v_i$. We will overestimate the conditional probability

$$P_i(\gamma_1,\dots,\gamma_i) \equiv \Pr_{\gamma_{i+1},\dots,\gamma_n}[B \mid \gamma_1,\dots,\gamma_i]$$

by the following pessimistic estimator:

$$A_i(\gamma_1,\dots,\gamma_i) = \sum_{T\in R}\prod_{e\in T}\sum_{I\subseteq e,|I|=c}\Pr_{\gamma_{i+1},\dots,\gamma_n}[I \text{ is monochromatic} \mid \gamma_1,\dots,\gamma_i].$$

Clearly, $P_i(\gamma_1,\dots,\gamma_i) \le A_i(\gamma_1,\dots,\gamma_i)$ for all i and all $\gamma_1,\dots,\gamma_i$, and $A_0 = (1/m)^{\Omega(1)}$ from Lemma 2.2. Now,

$$\begin{aligned}
A_i(\gamma_1,\dots,\gamma_i) &= \sum_{T\in R}\prod_{e\in T}\mathrm{E}_{\gamma_{i+1}}\sum_{I\subseteq e,|I|=c}\Pr_{\gamma_{i+2},\dots,\gamma_n}[I \text{ is monochromatic} \mid \gamma_1,\dots,\gamma_{i+1}]\\
&= \sum_{T\in R}\mathrm{E}_{\gamma_{i+1}}\prod_{e\in T}\sum_{I\subseteq e,|I|=c}\Pr_{\gamma_{i+2},\dots,\gamma_n}[I \text{ is monochromatic} \mid \gamma_1,\dots,\gamma_{i+1}]\\
&= \mathrm{E}_{\gamma_{i+1}} A_i(\gamma_1,\dots,\gamma_{i+1}),
\end{aligned}$$

where the second equality is because each edge e in T intersects no other edges in T. We pick γ_{i+1} to minimize $A_i(\gamma_1,\dots,\gamma_i,\gamma_{i+1})$. Then we have

$$1 > A_0 \ge A_1(\gamma_1) \ge A_2(\gamma_1,\gamma_2) \ge \cdots \ge A_n(\gamma_1,\dots,\gamma_n) \ge P_n(\gamma_1,\dots,\gamma_n). \tag{2}$$

$P_n(\gamma_1, \ldots, \gamma_n)$ is either 1 or 0 depending on whether the coloring $\gamma_1, \ldots, \gamma_n$ results in a bad $(2,3)$-tree of size u. As $P_n(\gamma_1, \ldots, \gamma_n) < 1$ from (2), we know $P_n(\gamma_1, \ldots, \gamma_n) = 0$, so there is no bad $(2,3)$-tree of size u, and we have found a good $\frac{k}{3}$-coloring.

It remains to show that for any i and any $\gamma_1, \ldots, \gamma_i$, $A_i(\gamma_1, \ldots, \gamma_i)$ can be computed efficiently. There are $m^{O(1)}$ $(2,3)$-trees of size u in R. It can be shown that enumerating all of them takes deterministic polynomial time (see, for example, [7]). For each $(2,3)$-tree T and any edge e in T, the value

$$\sum_{I \subseteq e, |I|=c} \Pr_{\gamma_{i+1}, \ldots, \gamma_n} [I \text{ is monochromatic} \mid \gamma_1, \ldots, \gamma_i]$$

can also be easily computed. So $A_i(\gamma_1, \ldots, \gamma_i)$ can be computed in deterministic polynomial time.

3. DNF approximate counting. Each finite set is associated with a natural distribution, the uniform distribution over its elements, and we will not make the distinction between a set and its natural distribution when it is clear from the context. Given a DNF formula F on n variables, we would like to know its volume, $vol(F) = \Pr_{x \in \{0,1\}^n}[F(x) = 1]$. Valiant [14] has shown that it is #P-complete to compute the exact value, so we settle for an approximation. The standard approach is to find a pseudorandom distribution using many fewer random bits that can still fool F.

DEFINITION 3.1. *A function $g : \{0,1\}^r \to \{0,1\}^n$ is called an ε-generator for a Boolean function $F : \{0,1\}^n \to \{0,1\}$ if*

$$\left| \Pr_{x \in \{0,1\}^n}[F(x) = 1] - \Pr_{y \in \{0,1\}^r}[F(g(y)) = 1] \right| \leq \varepsilon.$$

A function g is called an ε-generator for a class of Boolean functions if it is an ε-generator for each function in this class.

So an algorithm for approximating $vol(F)$ is to find an ε-generator g for F and then compute $\frac{1}{2^r} \sum_{y \in \{0,1\}^r} F(g(y))$, the expected value of F over the pseudorandom distribution generated by g. The running time is proportional to 2^r, and the goal is to reduce r. Notice that we can have a different generator g for a different input function F.

Clearly there are three important parameters in this problem: the number n of variables, the number m of terms, and the error ε allowed. We discover the importance of another parameter d, the maximum number of terms in which a variable can appear. Let DNF_d denote the set of DNF formulas with each variable appearing in at most d terms. Such formulas are usually called *read-d-times* DNF formulas. In the following, we will also assume that each term in a formula F contains at most $t = \log \frac{m}{\varepsilon}$ literals. This is because we can always remove those terms containing more than t literals to get another formula F' such that $|\Pr_x[F(x) = 1] - \Pr_x[F'(x) = 1]| \leq \varepsilon$ and then consider the formula F' instead. Let $t\text{DNF}_d$ denote the set of DNF_d formulas with no term containing more than t literals. This is the class of formulas we consider in this section. For convenience, we also assume that $n = m^{\Theta(1)}$.

A $t\text{DNF}_d$ formula F of n variables and m terms can be modelled by a hypergraph $H(V, E)$ with $|V| = n$, $|E| = m$, $E \subseteq V^{\leq t}$, and degree d. We can use the algorithm in the previous section to find a (k, c)-coloring for some k and c to be chosen later. Let V_i, $1 \leq i \leq k$, be the set of variables with color i. If we fix values for all variables not in V_i, we get a $c\text{DNF}_d$ formula on V_i. So a $t\text{DNF}_d$ formula gives rise to k classes of $c\text{DNF}_d$ formulas, and we show next that it suffices to be able to fool each class

separately. Suppose that for $1 \le i \le k$, $g_i : \{0,1\}^{r_i} \to \{0,1\}^{|V_i|}$ is an ε-generator for all $c\mathrm{DNF}_d$ formulas on the variable set V_i. Define $g : \{0,1\}^r \to \{0,1\}^n$, with $r = r_1 + \cdots + r_k$ and $n = |V_1| + \cdots + |V_k|$, such that those bits corresponding to V_i are generated by g_i. The following is a standard result.

LEMMA 3.2. *The function g defined above is a $k\varepsilon$-generator for F.*

Proof. For $1 \le i \le k$, let U_i denote the uniform distribution over $\{0,1\}^{|V_i|}$ for the variables in V_i, and let S_i be the corresponding pseudorandom distribution generated by g_i. Let D_i denote the distribution $S_1 \times \cdots \times S_i \times U_{i+1} \times \cdots \times U_k$, and let D'_i denote the distribution $S_1 \times \cdots \times S_{i-1} \times U_{i+1} \times \cdots \times U_k$. For $y \in D'_i$, let F_y denote the resulting formula from F by assigning the value y to the corresponding variables, and note that F_y is a $c\mathrm{DNF}_d$ formula on variable set V_i. Then

$$\begin{aligned}
\left| vol(F) - \Pr_{x \in D_k}[F(x) = 1] \right| &= \left| \Pr_{x \in D_0}[F(x) = 1] - \Pr_{x \in D_k}[F(x) = 1] \right| \\
&\le \sum_{i=0}^{k-1} \left| \Pr_{x \in D_i}[F(x) = 1] - \Pr_{x \in D_{i+1}}[F(x) = 1] \right| \\
&\le \sum_{i=1}^{k} \mathop{\mathrm{E}}_{y \in D'_i} \left| \Pr_{z \in S_i}[F_y(z) = 1] - \Pr_{z \in U_i}[F_y(z) = 1] \right| \\
&\le k\varepsilon.
\end{aligned}$$

D_k is the pseudorandom distribution generated by g. So g is a $k\varepsilon$-generator for F. ∎

So we have reduced the problem of finding a generator for a $t\mathrm{DNF}_d$ formula to the problem of finding a generator fooling all $c\mathrm{DNF}_d$ formulas. For small c, it becomes an easier task. This is one example where our hypergraph coloring algorithm supports some kind of divide and conquer approach. It remains to find such ε-generators for $c\mathrm{DNF}_d$. We will give two constructions according to two different values of k and c. Before proceeding to that, let us summarize what our algorithm does with a DNF formula F as input:

1. Remove those terms with more than $t = \log \frac{m}{\varepsilon}$ variables.
2. Determine the parameters d, k, c.
3. Run the hypergraph coloring algorithm to find a (k, c)-coloring for F.
4. Construct generators $g_1, \ldots, g_k$ and the generator g.
5. Compute the average of F under the pseudorandom distribution generated by g.

3.1. Construction I: $c = O(\log \frac{dt}{\varepsilon})$ and $k = O(\frac{t}{c})$. A given formula might contain constants in some terms, which would allow it to be simplified further. When we talk about the number of terms in a formula later, we mean the number of terms left after this simplification. The following lemma says that when both c and d are small, it suffices to be able to fool $c\mathrm{DNF}_d$ formulas with very few terms.

LEMMA 3.3. *Let $l = 2^c \ln \frac{1}{\varepsilon} = (\frac{dt}{\varepsilon})^{O(1)}$ and $m' = d(l-1)c = (\frac{dt}{\varepsilon})^{O(1)}$. If h is an ε-generator for all $c\mathrm{DNF}_d$ formulas with at most m' terms, then h is a 2ε-generator for all $c\mathrm{DNF}_d$ formulas.*

Proof. Consider a $c\mathrm{DNF}_d$ formula G. If G has at most m' terms, h is certainly an ε-generator for G. If G has more than m' terms, it has l disjoint terms because otherwise some variable would appear more than $m'/((l-1)c) = d$ times. Let T

denote the OR of those l disjoint terms. Then

$$1 \geq vol(G) \geq vol(T) > 1 - (1 - 2^{-c})^l \geq 1 - e^{-2^{-c}2^c \ln 1/\varepsilon} = 1 - \varepsilon.$$

As h is an ε-generator for T, we have

$$1 \geq \Pr_y[G(h(y)) = 1] \geq \Pr_y[T(h(y)) = 1] \geq vol(T) - \varepsilon \geq 1 - 2\varepsilon.$$

Then $|vol(G) - \Pr_y[G(h(y)) = 1]| \leq 2\varepsilon$. □

It remains to show how to find such an ε-generator for all $c\text{DNF}_d$ formulas of at most m' terms. Any $c\text{DNF}_d$ formula of at most m' terms contains at most $n' = cm'$ variables, out of the n variables, and is much easier to fool. The generator of Luby, Velicković, and Wigderson [9] can be slightly modified to suit this purpose. The keys in their result are the construction of a set system and its use in the generator construction of Nisan and Wigderson [11]. The formulas we consider here are more restricted, so set systems with better parameters can be constructed.

Let us first fix the following parameters: $b = \log \frac{4n'm'}{\varepsilon}$, $s = b^2$, and $r = 24cb^3$. Consider a $c\text{DNF}_d$ formula T and a family S of n subsets $S_1, \ldots, S_n$, where $S_i \subseteq \{1, \ldots, r\}$ for each i. Call S *good* for T if the following two conditions hold:

- For any variable x_i of T, $|S_i| \leq s$.
- For any variable x_i of T and any term of T with variables $\{x_j : j \in B\}$, $|S_i \cap (\cup_{j \in B \setminus \{i\}} S_j)| < b$.

With a good set S, one can construct a good generator for T, as stated in the following.

LEMMA 3.4 (see [9]). *If S is good for T, then the function $h_S : \{0,1\}^r \to \{0,1\}^n$, defined by*

$$h_S(y) \equiv \left(\bigoplus_{j \in S_1} y_j, \ldots, \bigoplus_{j \in S_n} y_j \right),$$

is an ε-generator for T.

As we want to construct one generator for all $c\text{DNF}_d$ formulas, it may not be possible to find one set system that is good for all such formulas. The strategy is to generate the set system randomly and prove that for any $c\text{DNF}_d$ formula T, the generated set system is good for T with high probability. Following [9], we will show that S can be generated using an approximate $2b$-wise independent space.

DEFINITION 3.5. *An (n, k, p, δ) space consists of n binary random variables, $X_1, \ldots, X_n$, such that, for any k indices $i_1, \ldots, i_k$ and any k bits $x_1, \ldots, x_k$,*

$$\left| P[X_{i_1} = x_1 \wedge \cdots \wedge X_{i_k} = x_k] - p^{|\{i : x_i = 1\}|}(1-p)^{|\{i : x_i = 0\}|} \right| \leq \delta.$$

Let $\delta = \frac{\varepsilon}{4m'n'(3cr)^{2b}}$. Let X_{ij}, for $1 \leq i \leq n$ and $1 \leq j \leq r$, be sampled from an $(nr, 2b, \frac{2s}{r}, \delta)$ space. Each sample point in this space can be efficiently indexed by a string of $v = O(\log\log n + b\log\frac{r}{s} + \log\frac{1}{\delta}) = O(\log^2 \frac{dt}{\varepsilon})$ bits [5]. Let $S(w) = \{S_1(w), \ldots, S_n(w)\}$, where $S_i(w) = \{j : X_{ij}(w) = 1\}$. Then using Lemma 1 of [9] in our setting, we have the following.

LEMMA 3.6. *For any $T \in cDNF_d$, $\Pr_w[S(w)$ not good for $T] \leq \varepsilon$.*

Proof.

$$\Pr_w[S(w) \text{ not good for } T] \leq n' \left(\frac{1}{s^b} + r^{2b}\delta \right) + m'n' \binom{r}{b} c^b \left(\left(\frac{2s}{r} \right)^{2b} + \delta \right)$$

$$\begin{aligned}
&\le \frac{n'}{s^b} + m'n'\left(\frac{12cs^2}{rb}\right)^b + n'r^{2b}\delta + m'n'\left(\frac{3cr}{b}\right)^b\delta \\
&\le \frac{\varepsilon}{4} + \frac{\varepsilon}{4} + \frac{\varepsilon}{4} + \frac{\varepsilon}{4} \\
&\le \varepsilon. \qquad \square
\end{aligned}$$

Consider the generator $h : \{0,1\}^{v+r} \to \{0,1\}^n$, defined as

$$h(w,y) \equiv h_{S(w)}(y) = \left(\bigoplus_{j\in S_1(w)} y_j, \ldots, \bigoplus_{j\in S_n(w)} y_j\right).$$

Combining Lemmas 3.4 and 3.6, we know that h is a 2ε-generator for all cDNF$_d$ formulas with at most m' terms. From Lemma 3.3, h is a 4ε-generator for all cDNF$_d$ formulas. From Lemma 3.2, given $F \in$ DNF$_d$, we can get a $4k\varepsilon$-generator for F. The number of random bits used is

$$k(r+v) = O\left(\frac{t}{c}\log^4\frac{dt}{\varepsilon}\right) = O\left(\log\frac{m}{\varepsilon}\log^3\frac{d\log m}{\varepsilon}\right).$$

Finally, with ε replaced by $\frac{\varepsilon}{4k}$, we have the following theorem.

THEOREM 3.7. *Given a DNF formula, we can approximate its volume within an additive error ε in deterministic $2^{O(\log\frac{m}{\varepsilon}\log^3\frac{d\log m}{\varepsilon})}$ time.*

3.2. Construction II: $c = O(\sqrt{\log(dt)})$ and $k = t2^{O(c)}$. Our second construction follows the framework of Luby and Velicković [8]. Let us recall their approach. First, they construct a small sample set of "almost" (k_0, c_0)-coloring[2] with $c_0 = O(\log t)$ and $k_0 = O(t2^{c_0+1}/\varepsilon)$. Next, they show that an $(n, l_0, 1/2, \delta_0)$ space gives rise to a good generator for c_0DNF for some l_0 and δ_0. The algorithm of [8] then does the following. Each almost (k_0, c_0)-coloring in the sample set is used for the purpose of divide and conquer to divide the given DNF formula F into k_0 classes of c_0DNF formulas, and then k_0 independent copies of $(n, l_0, 1/2, \delta_0)$ spaces are used to construct a generator for F with which an estimate of $vol(F)$ is computed. Each almost (k_0, c_0)-coloring gives an estimate of $vol(F)$, and the algorithm of [8] outputs the largest among these estimates as the final answer.

Here, we use our (k, c)-coloring algorithm to replace the almost (k_0, c_0)-coloring of [8] with $c = O(\sqrt{\log(dt)})$ and $k = t2^{O(c)}$. That is, we use our hypergraph coloring algorithm to produce one (k, c)-coloring, instead of a set of colorings. As a result, we need to calculate only one estimate, which is used as the final answer. The (k, c)-coloring divides the tDNF$_d$ formula into k classes of cDNF formulas.

Let $l = \lceil\log\frac{8k}{\varepsilon}\rceil c2^c$ and $\delta = \frac{\varepsilon}{8k2^l}$. Let $\hat{h} : \{0,1\}^s \to \{0,1\}^n$ be the mapping that generates the $(n, l, 1/2, \delta)$ space. From [8], we know that $\hat{h}$ is a good generator for cDNF.

LEMMA 3.8 (see [8]). *$\hat{h}$ is an $\frac{\varepsilon}{k}$-generator for cDNF.*

Using k independent copies of $\hat{h}$, each with an independent random seed, we get an ε-generator for the tDNF$_d$ formula F, according to Lemma 3.2. The total number of random bits used by the generator is

$$ks = O(k(\log\log n + l + \log 1/\delta)) = O(kl) = O\left(\log\frac{m}{\varepsilon}\log\frac{1}{\varepsilon}2^{O(\sqrt{\log(d\log\frac{m}{\varepsilon})})}\right).$$

[2]More precisely, for any term, only a small fraction of k_0-coloring in this set can result in more than c_0 variables in that term having the same color.

So we have the following theorem.

THEOREM 3.9. *Given a DNF formula, we can approximate its volume within an additive error ε in deterministic $2^{\log \frac{m}{\varepsilon} \log \frac{1}{\varepsilon} 2^{O(\sqrt{\log(d \log \frac{m}{\varepsilon})})}}$ time.*

4. Dimensions of posets. In this section, we will constructivize the existential bounds, given by Füredi and Kahn [6], on the dimensions of posets.

DEFINITION 4.1 (see [6]). *Let $(P, <)$ be a poset. Its dimension, denoted as $\dim(P)$, is defined to be the minimum number of linear extensions $L_1, \ldots, L_d$ such that $P = L_1 \cap \cdots \cap L_d$, i.e.,*

$$\forall x, y \in P, \;\; x < y \Leftrightarrow x <_{L_i} y \quad \forall i. \tag{3}$$

DEFINITION 4.2 (see [6]). *For $x \in P$, define $U(x) \equiv \{y \in P : y \geq x\}$, $L(x) \equiv \{y \in P : y \leq x\}$, and $C(x) \equiv U(x) \cup L(x)$. Define $u \equiv \max_{x \in P} |U(x)|$, $\ell \equiv \max_{x \in P} |L(x)|$, and $t \equiv \max_{x \in P} |C(x)|$.*

For $x \in P$, $S \subseteq P$, and π a permutation over P, we will write $x <_\pi S$ for the condition for all $y \in S, x <_\pi y$.

LEMMA 4.3 (see [6]). *The dimension of $(P, <)$ is equal to the minimum number of permutations, $\pi_1, \ldots, \pi_d$, such that*

$$\forall x, y \in P, \;\; y \not\leq x \Rightarrow \exists i, \;\; x <_{\pi_i} U(y). \tag{4}$$

In addition, there is a deterministic polynomial time algorithm for converting a set of d permutations satisfying condition (4) *to a set of d permutations satisfying condition* (3).

This lemma provides an alternative way to compute the dimension of a poset. It is especially useful for obtaining the upper bound of the dimension: just find a set Π of permutations such that any pair (x, y) in the set

$$K_P = \{(x, y) \in P^2 : y \not\leq x\}$$

is *killed* by some permutation $\pi \in \Pi$ in the sense that $x <_\pi U(y)$. Based on this, Füredi and Kahn [6] gave a simple upper bound: $\dim(P) = O(u \log |P|)$. Their argument can be turned into a deterministic algorithm.

THEOREM 4.4. *Given any poset $(P, <)$, we can find a set of $O(u \log |P|)$ permutations satisfying condition* (3) *in deterministic polynomial time.*

Proof. We want to find $d = O(u \log |P|)$ permutations $\pi_1, \ldots, \pi_d$, one by one, to kill every pair in K_P. Note that for any $(x, y) \in K_P$, $x \notin U(y)$, so a random permutation kills (x, y) with probability at least $\frac{1}{u+1}$. Then for any $K \subseteq K_P$, the expected fraction of K killed by a random permutation is at least $\frac{1}{u+1}$. Thus, given $\pi_1, \ldots, \pi_{i-1}$, there exists some π_i that can kill at least $\frac{1}{u+1}$ fraction of those remaining pairs in K_P not killed by $\pi_1, \ldots, \pi_{i-1}$. We can find such a π_i by fixing its components one by one in $|P|$ steps, again using the technique of conditional probability. After choosing $\pi_1, \ldots, \pi_d$ in this way, the number of pairs not yet killed is less than $|P|^2(1 - \frac{1}{u+1})^d < 2^{2 \log_2 |P| - d/(u+1)} \leq 1$ for some $d = O(u \log |P|)$. These d permutations satisfy condition (4) and can be converted to d permutations satisfying condition (3) by Lemma 4.3. It is easy to see that the whole process can be done in deterministic polynomial time. ∎

Füredi and Kahn [6] gave another upper bound: $\dim(P) = O(t \log^2 t)$. Again, their argument can be turned into a deterministic algorithm. Assume without loss of generality that $t^{\log t} \leq |P|$ (otherwise, we can just use the previous theorem).

THEOREM 4.5. *Given any poset $(P,<)$, we can find a set of $O(u\log^2 t) = O(t\log^2 t)$ permutations satisfying condition (3) in deterministic polynomial time.*

Proof. Consider the hypergraph $H(V,E)$ with $V=P$ and $E=\{U(y): y\in P\}$. The goal is to find a set of permutations over V_i such that for any $x\in V$ and any $e\in E$, some permutation puts x before every element in e. Note that H has degree ℓ and $E\subseteq V^{\le u}$. Using our hypergraph coloring algorithm, we can find a (k,c)-coloring with $c=\Theta(\log t)$ and $k=O(u/c)$. H can now be partitioned into k hypergraphs $H_1(V_1,E_1),\ldots,H_k(V_k,E_k)$, where V_i is the set of vertices with color i and $E_i=\{e\cap V_i : e\in E\}=\{U(y)\cap V_i : y\in V\}$.

We will have k groups, $G_1,\ldots,G_k$, of permutations with G_i designed to kill those pairs $(x,y)\in K_P$ with $x\in V_i$. Permutations in G_i place V_i ahead of $\bar{V}_i\equiv V\setminus V_i$ and use an arbitrary permutation T_i for $\bar{V}_i$. This guarantees that for any $(x,y)\in K_P$ with $x\in V_i$, we have $x<_\tau U(y)\cap\bar{V}_i$ for any permutation $\tau\in G_i$. It remains to guarantee that for any $(x,y)\in K_P$ with $x\in V_i$, there exists a permutation $\pi\in G_i$ such that $x<_\pi U(y)\cap V_i$. Notice that each $U(y)\cap V_i$ corresponds to an edge in E_i, so it suffices to find a set of permutations over V_i such that for any $x\in V_i$ and any $e\in E_i$, some permutation puts x before every element in e for $1\le i\le k$. That is, we have reduced the original problem to k smaller subproblems.

For $H_i(V_i,E_i)$, use Lemma 2.1 to color vertices in V_i with $r=O(c\ell)=O(\ell\log t)$ colors such that all vertices in any edge of E_i have different colors. Let $V_{i,j}$, $1\le j\le r$, denote those vertices in V_i having color j. For the order among $V_{i,j}$, we use an arbitrary permutation $R_{i,j}$ together with its converse $R'_{i,j}$ (reversing the order in $R_{i,j}$). This guarantees that for any $(x,y)\in K_P$ with $x\in V_{i,j}$, either $R_{i,j}$ or $R'_{i,j}$ puts x before $U(y)\cap V_{i,j}$ (which has at most one element) for $1\le j\le r$. It remains to guarantee that for any $(x,y)\in K_P$ with $x\in V_{i,j}$, some permutation puts x before $U(y)\cap V_i\setminus V_{i,j}$ (which has at most $c-1$ elements) for $1\le j\le r$. That is, it suffices to find a set of permutations on r colors such that, for any color j and any set J of $c-1$ colors not containing j, some permutation puts j ahead of J. A random collection of $q=O(c^2\log r)$ permutations will fail with probability at most

$$r\binom{r-1}{c-1}\left(1-\frac{1}{c}\right)^q<1.$$

Using an idea similar to that in Theorem 4.4, a good set of permutations $\gamma_1,\ldots,\gamma_q$ can be found one by one, each in deterministic $r^{O(c)}=|P|^{O(1)}$ time. This makes sure that for any $(x,y)\in K_P$ with $x\in V_{i,j}$, there exists some $s\in[q]$ such that the permutation (over V_i)

$$\big(R_{i,\gamma_s(1)},R_{i,\gamma_s(2)},\ldots,R_{i,\gamma_s(r)}\big)$$

puts x before $U(y)\cap V_i\setminus V_{i,j}$ for $1\le j\le r$.

In summary, for each i and s with $1\le i\le k$ and $1\le s\le q$, we have two permutations:

- $\pi_{i,s}=\big(R_{i,\gamma_s(1)},R_{i,\gamma_s(2)},\ldots,R_{i,\gamma_s(r)},T_i\big)$.
- $\pi'_{i,s}=(R'_{i,\gamma_s(1)},R'_{i,\gamma_s(2)},\ldots,R'_{i,\gamma_s(r)},T_i)$.

The total number of permutations is thus

$$2kq=O((u/c)(c^2\log r))=O(u\log^2 t)=O(t\log^2 t),$$

and they can be found in deterministic polynomial time. As a result, we have the theorem. ∎

Acknowledgment. We would like to thank David A. Mix Barrington for some helpful comments on an earlier version of this paper.

REFERENCES

[1] N. Ahuja and A. Srivastav, *On constrained hypergraph coloring and schedule*, in Proceedings of the 5th International Workshop on Approximation Algorithms for Combinatorial Optimization, Rome, Italy, 2002, pp. 14–25.

[2] N. Alon, *A parallel algorithmic version of the local lemma*, Random Structures Algorithms, 2 (1991), pp. 367–378.

[3] J. Beck, *An algorithmic approach to the Lovász local lemma*, Random Structures Algorithms, 2 (1991), pp. 343–365.

[4] P. Erdös and L. Lovász, *Problems and results on 3-chromatic hypergraphs and some related questions*, in Infinite and Finite Sets, A. Hajnal, L. Lovasz, and V. T. Sos, eds., North–Holland, Amsterdam, 1975, pp. 609–628.

[5] G. Even, O. Goldreich, M. Luby, N. Nisan, and B. Velicković, *Approximations of general independent distributions*, in Proceedings of the 24th Annual ACM Symposium on Theory of Computing, Victoria, BC, Canada, 1992, pp. 10–16.

[6] Z. Füredi and J. Kahn, *On the dimensions of ordered sets of bounded degree*, Order, 3, 1986, pp. 15–20.

[7] T. Leighton, C.-J. Lu, S. Rao, and A. Srinivasan, *New algorithmic aspects of the Local lemma with applications to routing and partitioning*, SIAM J. Comput., 31 (2001), pp. 626–641.

[8] M. Luby and B. Velicković, *On deterministic approximate counting of DNF*, Algorithmica, 16 (1996), pp. 415–433.

[9] M. Luby, B. Velicković, and A. Wigderson, *Deterministic approximate counting of depth-2 circuits*, in Proceedings of the 2nd Israeli Symposium on Theory of Computing and Systems, Natanya, Israel, 1993.

[10] N. Nisan, *Pseudo-random bits for constant depth circuits*, Combinatorica, 11 (1991), pp. 63–70.

[11] N. Nisan and A. Wigderson, *Hardness vs. randomness*, J. Comput. System Sci., 49 (1994), pp. 149–167.

[12] P. Raghavan, *Probabilistic construction of deterministic algorithm: Approximating packing integer programs*, J. Comput. System Sci., 38 (1994), pp. 683–707.

[13] A. Srinivasan, *An extension of the Lovász local lemma, and its applications to integer programming*, in Proceedings of the 7th ACM-SIAM Symposium on Discrete Algorithms, Atlanta, GA, 1996, pp. 6–15.

[14] L. G. Valiant, *The complexity of enumeration and reliability problems*, SIAM J. Comput., 8 (1979), pp. 410–421.

SIAM J. DISCRETE MATH.
Vol. 18, No. 2, pp. 332–346

FASTER DETERMINISTIC BROADCASTING IN AD HOC RADIO NETWORKS*

DARIUSZ R. KOWALSKI† AND ANDRZEJ PELC‡

Abstract. We consider radio networks modeled as directed graphs. In ad hoc radio networks, every node knows only its own label and a linear bound on the size of the network but is unaware of the topology of the network or even of its own neighborhood. The fastest currently known deterministic broadcasting algorithm working for arbitrary n-node ad hoc radio networks has running time $\mathcal{O}(n \log^2 n)$. Our main result is a broadcasting algorithm working in time $\mathcal{O}(n \log n \log D)$ for arbitrary n-node ad hoc radio networks of radius D. The best currently known lower bound on broadcasting time in ad hoc radio networks is $\Omega(n \log D)$; hence our algorithm is the first to shrink the gap between bounds on broadcasting time in radio networks of arbitrary radius to a logarithmic factor. We also show a broadcasting algorithm working in time $\mathcal{O}(n \log D)$ for *complete layered* n-node ad hoc radio networks of radius D. The latter complexity is optimal.

Key words. distributed algorithms, radio networks, broadcasting

AMS subject classifications. 68W15, 68W40, 68R10

DOI. 10.1137/S089548010342464X

1. Introduction.

1.1. The model. A radio network is a collection of transmitter-receiver stations. It is modeled as a directed graph on the set of these stations, referred to as *nodes*. A directed edge $e = (u, v)$ means that the transmitter of u can reach v. Nodes send messages in synchronous *steps* (time slots). In every step, every node acts either as a *transmitter* or as a *receiver*. A node acting as a transmitter sends a message which can potentially reach all of its out-neighbors. A node acting as a receiver in a given step gets a message if and only if exactly one of its in-neighbors transmits in this step. The message received in this case is the one that was transmitted. If at least two in-neighbors v and v' of u transmit simultaneously in a given step, none of the messages is received by u in this step. In this case we say that a *collision* occurred at u. It is assumed that the effect at node u of more than one of its in-neighbors transmitting is the same as that of no in-neighbor transmitting; i.e., a node cannot distinguish a collision from silence.

The goal of *broadcasting* is to transmit a message from one node of the network, called the *source*, to all other nodes. Remote nodes get the source message via intermediate nodes along paths in the network. In order to make broadcasting feasible, we

*Received by the editors March 25, 2003; accepted for publication (in revised form) February 3, 2004; published electronically November 9, 2004. A preliminary version of this paper appeared in *Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science* (STACS 2003), Berlin, Germany, 2003, Lecture Notes in Comput. Sci. 2607, H. Alt and M. Habib, eds., Springer-Verlag, Berlin, 2003, pp. 109–120.

http://www.siam.org/journals/sidma/18-2/42464.html

†Instytut Informatyki, Uniwersytet Warszawski, Banacha 2, 02-097 Warszawa, Poland (darek@mimuw.edu.pl), and Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, Saarbrücken, 66123 Germany. This work was done in part during this author's stay at the Research Chair in Distributed Computing of the Université du Québec en Outaouais as a postdoctoral fellow. This author's research was supported in part by KBN grant 4T11C04425.

‡Département d'informatique, Université du Québec en Outaouais, Hull, Québec J8X 3X7, Canada (Andrzej.Pelc@uqo.ca). This author's research was supported in part by NSERC grant OGP 0008136 and by the Research Chair in Distributed Computing of the Université du Québec en Outaouais.

assume that there is a directed path from the source to any node of the network. We study one of the most important and widely investigated performance parameters of a broadcasting algorithm, which is the total time, i.e., the number of steps it uses to inform all the nodes of the network.

We consider deterministic distributed broadcasting in ad hoc radio networks. In such networks, a node does not have any a priori knowledge of the topology of the network, its maximum degree, its radius, nor even of its immediate neighborhood: the only a priori knowledge of a node is its own label and a linear upper bound r on the number of nodes. Labels of all nodes are distinct integers from the interval $[0, \ldots, r]$. Broadcasting in ad hoc radio networks was investigated, e.g., in [3, 5, 6, 7, 9, 10, 12, 13, 15]. We use the same definition of running time of a broadcasting algorithm working for ad hoc radio networks as, e.g., in [12]. We say that the algorithm works in time t for networks of a given class if t is the smallest integer such that the algorithm informs all nodes of any network of this class in at most t steps. We do not suppose the possibility of spontaneous transmissions; i.e., only nodes which have already gotten the source message are allowed to send messages. Of course, since we are only concerned with upper bounds on broadcasting time, all our results remain valid if spontaneous transmissions are allowed. The format of all messages is the same: a node transmits the source message and the current step number.

We denote by n the number of nodes in the network, by r an upper bound on label values, by D the radius of the network (the maximum length of a shortest directed path from the source to any other node), and by Δ the maximum in-degree of a node in the network. We assume that r is linear in n ($r = cn$ for some constant c). Among these parameters, only r is known to nodes of the network.

1.2. Related work. In many papers on broadcasting in radio networks (e.g., [1, 2, 16, 19, 17]), the network is modeled as an undirected graph, which is equivalent to the assumption that the directed graph, which models the network in our scenario, is symmetric. A lot of effort has been devoted to finding good upper and lower bounds on deterministic broadcast time in such radio networks under the assumption that nodes have full knowledge of the network. In [1] the authors proved the existence of a family of n-node networks of radius 2 for which any broadcast requires time $\Omega(\log^2 n)$, while in [16] it was proved that broadcasting can be done in time $O(D + \log^5 n)$ for any n-node network of diameter D. (Note that for symmetric networks, diameter is of the order of the radius.)

As for broadcasting in ad hoc symmetric radio networks, an $\mathcal{O}(n)$ algorithm assuming spontaneous transmissions was constructed in [9]. If spontaneous transmissions are precluded, the best currently known results on broadcasting in such networks are those from [18]: an algorithm with running time $\mathcal{O}(n \log n)$ and a lower bound $\Omega(n \frac{\log n}{\log(n/D)})$.

Deterministic broadcasting in arbitrary directed radio networks was studied, e.g., in [6, 7, 8, 9, 10, 12, 13, 15]. In [8], a $\mathcal{O}(D \log^2 n)$-time broadcasting algorithm was given for all n-node networks of radius D, assuming that nodes know the topology of the network. (The later algorithm from [16], giving upper bound $O(D + \log^5 n)$ on broadcasting time, works only for undirected radio networks.) Other above-cited papers studied broadcasting time in ad hoc directed radio networks. The best known lower bound on this time is $\Omega(n \log D)$, proved in [13]. As for the upper bounds, a series of papers presented increasingly faster algorithms, starting with time $O(n^{11/6})$, in [9], then $O(n^{5/3} \log^{1/3} n)$ in [15], then $O(n^{3/2})$ in [10], and finally $O(n \log^2 n)$ in [12], which corresponds to the fastest algorithm, working

for ad hoc networks of arbitrary maximum degree, known before the present paper. In another approach, broadcasting time is studied for ad hoc radio networks of maximum degree Δ. This work was initiated in [6], where the authors constructed a broadcasting scheme working in time $O(D\frac{\Delta^2}{\log^2\Delta}\log^2 n)$ for arbitrary n-node networks with radius D and maximum degree Δ. (While the result was stated only for undirected graphs, it is clear that it holds for arbitrary directed graphs, not just symmetric ones.) This result was further investigated, both theoretically and using simulations, in [7, 4]. On the other hand, a protocol working in time $O(D\Delta\log^{\log\Delta} n)$ was constructed in [3]. Finally, an $O(D\Delta\log n\log(n/\Delta))$ protocol was described in [13] (for the case when nodes know n but not Δ). If n is also unknown, the algorithm from [13] works in time $O(D\Delta\log^a n\log(n/\Delta))$ for any $a>1$.

Finally, randomized broadcasting in ad hoc radio networks was studied, e.g., in [2, 19, 18, 14]. In [2], the authors give a simple randomized protocol running in expected time $O(D\log n+\log^2 n)$. We later improved this upper bound to $O(D\log(n/D)+\log^2 n)$ in [18] (see also [14]). In [19] it was shown that, for any randomized broadcast protocol and parameters D and n, there exists an n-node network of radius D, requiring expected time $\Omega(D\log(n/D))$ to execute this protocol.

1.3. Our results. Our main result is a deterministic broadcasting algorithm working in time $\mathcal{O}(n\log n\log D)$ for arbitrary n-node ad hoc radio networks of radius D. This improves the best currently known broadcasting time $\mathcal{O}(n\log^2 n)$ from [12], e.g., for networks of radius polylogarithmic in size. Also, for $D\Delta\in\omega(n)$, this improves the upper bound $O(D\Delta\log n\log(n/\Delta))$ from [13]. The best currently known lower bound on broadcasting time in ad hoc radio networks is $\Omega(n\log D)$ [13]; hence our algorithm is the first to shrink the gap between bounds on deterministic broadcasting time for radio networks of arbitrary radius to a logarithmic factor. Our algorithm is nonconstructive in the same sense as the one from [12]. Using the probabilistic method, we prove the existence of a combinatorial object, which all nodes use in the execution of the deterministic broadcasting algorithm. (Since we do not count local computations in our time measure, such an object—the same for all nodes—could be found by an exhaustive search performed locally by all nodes without changing our result.)

We also show a broadcasting algorithm working in time $\mathcal{O}(n\log D)$ for *complete layered* n-node ad hoc radio networks of radius D. The latter complexity is optimal, due to the matching lower bound $\Omega(n\log D)$, which was proved in [13] for this class of networks, even assuming that nodes know parameters n and D. The best previous upper bound on broadcasting time in complete layered n-node ad hoc radio networks of radius D was $\mathcal{O}(n\log n)$ [12]. Hence we obtain a gain for the same range of values of D as before.

If nodes do not know any upper bound on the size of the network, the upper bound $\mathcal{O}(n\log^2 n)$ from [12] remains valid, using a simple doubling technique, which probes possible values of n. In our case, the doubling technique cannot be used directly, since we deal with two unknown parameters, n and D. However, we can modify our algorithm in this case, obtaining running time $\mathcal{O}(n\log n\log\log n\log D)$, which still beats the time from [12], e.g., for networks of radius polylogarithmic in size. For $D\Delta\in\Omega(n)$, this also improves the upper bound $O(D\Delta\log^a n\log(n/\Delta))$, for any $a>1$, proved in [13] for the case of unknown n and Δ.

Note added in proof. Recently, our upper bound $\mathcal{O}(n\log n\log D)$ on deterministic broadcasting time in arbitrary n-node ad hoc radio networks of radius D

has been improved in [14] to $\mathcal{O}(n \log^2 D)$. As in our case, the algorithm in [14] is nonconstructive.

2. The broadcasting algorithm. In this section we show a deterministic broadcasting algorithm working in time $\mathcal{O}(n \log n \log D)$ for arbitrary n-node ad hoc radio networks of radius D. Recall that r is a linear upper bound on the number of nodes and that labels of all nodes are distinct integers from the interval $[0, \ldots, r]$. Taking $2^{\lceil \log r \rceil}$ instead of r, we can assume that $\log r$ is a positive integer. The parameter r is known to all nodes. We first show our upper bound under the additional assumption that D is known to all nodes. At the end of this section, we show how this assumption can be removed without changing the result.

We will use the Procedure Fast-Broadcasting, formally defined below, which intuitively works as follows. Communication is divided into stages of length $1 + \log r$. Current time is appended to all messages. Upon receiving a message, a node updates time and waits until the end of the current stage. This is the only adaptive part of the procedure. Then a sequence of transmissions is performed by any node v, based on a predefined vector $T(v)$ of length $\mathcal{O}(r \log r \log D)$. This sequence of steps is interleaved with round-robin transmissions according to global time computed by every node which already got a message. (The role of round-robin transmissions is purely technical: the procedure could be modified by incorporating them in the description of vectors $T(v)$, but separating these substeps improves clarity of analysis for bounded D and does not change asymptotic complexity of the procedure.)

Given a 0-1 matrix $T = [T_i(v)]_{i \le t; v \le r}$, where $t = 3600 \cdot (1 + \log r) \cdot r \log D$, we formally define the above-described procedure in the following way.

PROCEDURE FAST-BROADCASTING(T).

After receiving the source message and the current step number $a(1 + \log r) + b$ for some parameters a, b such that $0 \le b \le \log r$ and $0 < a(1 + \log r) + b \le t$, node v waits until step $t_v = (a + 1)(1 + \log r) - 1$.

for $i = t_v + 1, \ldots, t$ **do**

Substep A. if $T_i(v) = 1$ **then** v transmits in step i.

Substep B. if $i \equiv v \mod r$ **then** v transmits in step i.

We now define the following random 0-1 matrix $\hat{T} = [\hat{T}_i(v)]_{i \le t; v \le r}$. For all parameters a, b such that $0 \le b \le \log r$ and $0 < a(1 + \log r) + b \le t$, we have $\mathsf{Pr}[T_{a(1+\log r)+b}(v) = 1] = 1/2^b$, and all events $\hat{T}_i(v) = 1$ are independent. The period consisting of steps $a(1 + \log r), \ldots, (a + 1)(1 + \log r) - 1$ is called *stage* a.

Algorithm Fast-Broadcasting consists of executing Procedure Fast-Broadcasting($\hat{T}$) for the above-defined random matrix $\hat{T}$.

Path-graphs and simple-path-graphs. In what follows, v_0 denotes the source. In order to analyze Algorithm Fast-Broadcasting, we define the following classes of directed graphs. A *path-graph* consists of a directed path $v_0, \ldots, v_k$, where $k \le D$, possibly with some additional edges $(v_l, v_{l'})$, for $l > l'$, and with some additional nodes v, whose only out-neighbors are among $v_1, \ldots, v_k$. The path $v_0, \ldots, v_k$ is called the *main path*. A *simple-path-graph* consists of a directed path $v_0, \ldots, v_k$, where $k \le D$, with some additional nodes v, each of which has exactly one out-neighbor, and this out-neighbor is among $v_1, \ldots, v_k$. For any path-graph $G = (V, E)$, the graph $\bar{G}$ is the subgraph of G on the same set of nodes containing the main path and satisfying the condition that for every node v outside the main path, v has exactly one neighbor v_l in $\bar{G}$, where $l = \max\{l' : (v, v_{l'}) \in E\}$. By definition, for every path-graph G, the graph $\bar{G}$ is a simple-path-graph. See Figure 1.

Intuitively, a path-graph corresponds to a directed path in a graph, together with all neighboring nodes that can influence transmissions from the source to the target. There can be as many as $\Theta(2^{nD})$ such graphs. A simple-path-graph can be obtained from a path-graph by deleting all "backward edges" leading to the main path. Since every node in such a graph has only one out-neighbor, the number of simple-path-graphs is much smaller, at most $\Theta(D^n)$. We show that deleting all "backward edges" does not influence the speed of transmission from the source to the target.

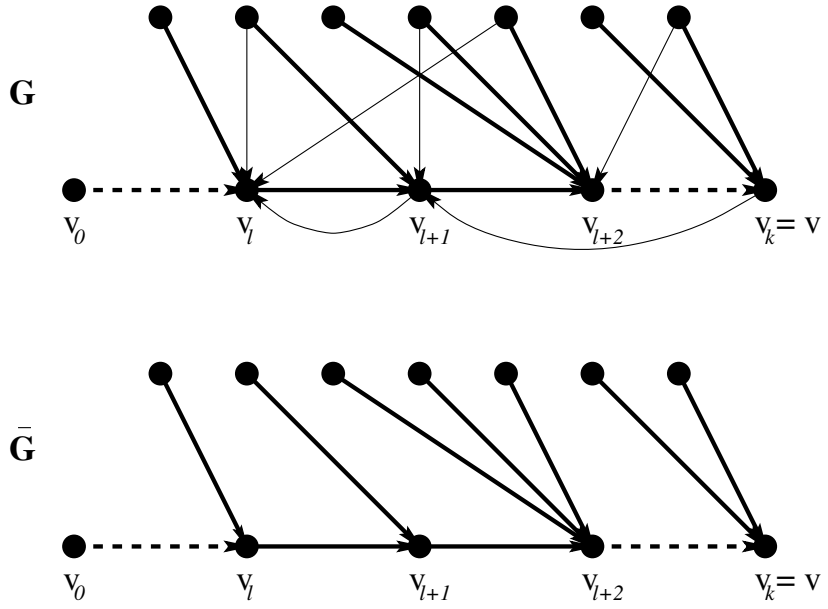


FIG. 1. *Path-graph G and corresponding simple-path-graph $\bar{G}$.*

Adversarial models. In general, path-graphs do not satisfy our assumption that there is a directed path from the source v_0 to any other node. Hence, for path-graphs, we modify our model of broadcasting as follows and call it the *adversarial wake-up model*. In this model the goal is not to wake up all processors, since the network may not be strongly connected, but to wake up the last node on the main path. We assume that nodes outside the main path also get the source message but are woken up by an adaptive adversary in various time steps not exceeding t. More precisely, the adversary can wake up any node v in the main path in any step $t_v \leq t$, providing v with the source message and step number t_v. The adversary acts according to some *wake-up pattern* $\mathcal{W}_m$ from the family $\{\mathcal{W}_m\}_{m \leq M}$ of all possible wake-up patterns. A wake-up pattern is a function assigning to every vertex v a step number $t_v \leq t$. The analysis of the progress of the algorithm working on path-graphs is relatively easy to extend to arbitrary graphs (see the proof of Theorem 5). This is not the case with simple-path-graphs, which may be too "trimmed" compared to an arbitrary graph. To overcome this obstacle we introduce a slightly stronger notion of the adversarial model, called the *simple-adversarial model*, which is the adversarial model with the following rule:

> If node v_l in the main path is not the rightmost node in the main path having woken-up in-neighbors, then v_l may be woken up only according to the adversarial wake-up pattern (*not* by some of its transmitting neighbors!).

Broadcasting under the simple-adversarial model in the simple-path-graph introduces similar "effects" to the presence of "backward edges" (we prove it in Lemma 4). For other applications of adversarial models and path-graphs, see also [11].

For every wake-up pattern $\mathcal{W}_m$ and stage a of Procedure Fast-Broadcasting(T), we define an integer $f_a(m)$ as follows (independently of the presence of an adversary or simple adversary). If node v_k has the source message after stage a, we fix $f_a(m) = k$. Otherwise, $f_a(m) = l$, where v_{l+1} is the last node on the main path which does not have the source message after stage a but has an in-neighbor having the source message after stage a, assuming that the pattern $\mathcal{W}_m$ is used. See Figure 2.

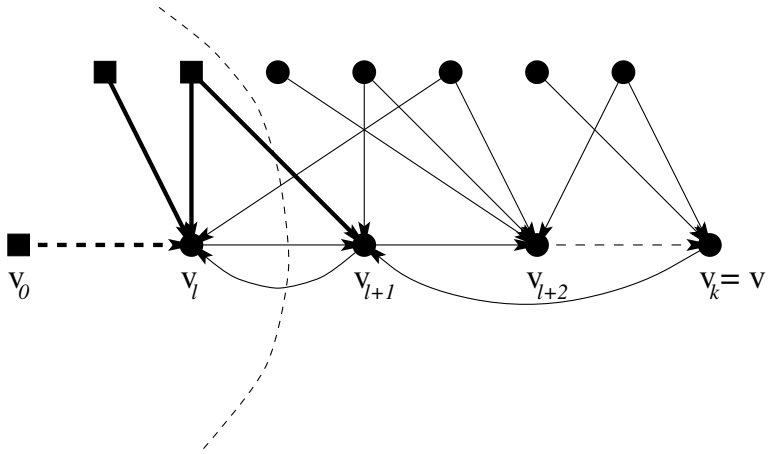


FIG. 2. *Definition of $f_a(m)$. Illustration for $f_a(m) = l$.*

We now describe a high-level outline of the proof of our upper bound. First, using the probabilistic method, we show the existence of a matrix T such that Procedure Fast-Broadcasting(T), applied to any simple-path-graph and to any wake-up pattern under the simple-adversarial model, delivers the source message to the last node of the main path in time $\mathcal{O}(n \log n \log D)$. Next, we show that the same is true for any path-graph under the adversarial model. Finally, we show that Procedure Fast-Broadcasting(T) completes broadcasting in all graphs in time $\mathcal{O}(n \log n \log D)$.

Fix an integer a and consider the simple-adversarial model. Suppose that $\hat{T}_i(v)$ are fixed for all $i < (a+1)(1+\log r)$. Our first goal is to show that, for a fixed simple-path-graph G under the simple-adversarial model, the set $\{m : f_a(m) < f_{a+1}(m) \leq k\}$ contains a constant fraction of values $\{m : f_a(m) < k\}$, for any stage a, with high probability. More precisely, we call stage $a+1$ *successful* if either $\{m : f_a(m) < k\} = \emptyset$ or $|\{m : f_a(m) < f_{a+1}(m) \leq k\}| \geq \frac{1}{36} \cdot |\{m : f_a(m) < k\}|$ in the execution of Algorithm Fast-Broadcasting.

LEMMA 1. *With probability at least* 0.1*, stage $a+1$ is successful.*

Proof. We consider only substeps of type A. Let $X = \{m : f_a(m) < k\}$. Let $X_l = \{m : f_a(m) = l\}$ for $l = 1, \ldots, k-1$. Observe that $\{X_l\}_{l=1}^{k-1}$ is a partition

of X. Consider the set X_l for a fixed $l < k$. For each $m \in X_l$, let H_m be the set of all in-neighbors of v_l having the source message at the end of stage a. By the definition of $f_a(m)$, which is equal to l, we have $H_m \neq \emptyset$ for all $m \in X_l$. Let X_l^j, for $j = 0, \ldots, \log r$, be the set of all $m \in X_l$ such that $2^{j-1} < |H_m| \leq 2^j$. Consider step $(a+1)(1+\log r)+j$ in stage $a+1$. In this step, every in-neighbor $v \in H_m$ of v_l transmits with probability $1/2^j$. If $j = 0$, then $|H_m| = 1$ for all X_l^0. Hence, with probability 1, node v_l receives the source message in step $(a+1)(1+\log r)$, and consequently $f_{a+1}(m) \geq l > f_a(m)$. If $j > 0$, then the probability that exactly one node in H_m transmits in step $(a+1)(1+\log r)+j$, for a fixed $m \in X_l^j$, is at least

$$|H_m| \cdot \frac{1}{2^j} \cdot \left(1 - \frac{1}{2^j}\right)^{|H_m|-1} > \frac{1}{2}\left(1 - \frac{1}{2^j}\right)^{2^j} \geq \frac{1}{2} \cdot \frac{1}{4} = \frac{1}{8} .$$

Using the Markov inequality applied to the number of sets H_m (for $m \in X_l^j$) for which exactly one node transmits in step $(a+1)(1+\log r)+j$, we obtain that, with probability at least 0.1, $|X_l^j \cap \{m : f_a(m) < f_{a+1}(m) \leq k\}| \geq \frac{1}{36}|X_l^j|$.

Indeed, for any $m \in X_l^j$, define the random binary variable χ_m as follows: $\chi_m = 0$ if and only if v_l receives the source message by step $(a+1)(1+\log r)+j$. Define $\chi = \sum_{m \in X_l^j} \chi_m$. Since $\mathsf{Pr}(\chi_m = 1) < 7/8$, the expected value of χ is smaller than $\frac{7}{8}|X_l^j|$, and consequently

$$\mathsf{Pr}\left(\chi \geq \frac{35}{36}|X_l^j|\right) = \mathsf{Pr}\left(\chi \geq \frac{10}{9} \cdot \frac{7}{8}|X_l^j|\right) \leq \mathsf{Pr}\left(\chi \geq \frac{10}{9} \cdot E(\chi)\right) \leq 0.9 .$$

Hence $\mathsf{Pr}(\chi < \frac{35}{36}|X_l^j|) \geq 0.1$.

Observe that, for different values j, the inequalities $|X_l^j \cap \{m : f_a(m) < f_{a+1}(m) \leq k\}| \geq \frac{1}{36}|X_l^j|$ can be proved using disjoint steps and therefore disjoint sets of random independent trials. Consequently, $|X_l \cap \{m : f_a(m) < f_{a+1}(m) \leq k\}| \geq \frac{1}{36}|X_l|$ with probability at least 0.1, and, for the same reason, we have $|X \cap \{m : f_a(m) < f_{a+1}(m) \leq k\}| \geq \frac{1}{36}|X|$ with probability at least 0.1, which completes the proof. □

Note that in Lemma 1 we used the rule of radio broadcasting only to transmit to node v_{l+1}, which is the rightmost node in the main path having some active in-neighbor, so the assumptions of the simple-adversarial wake-up model are satisfied. The next lemma implies that in Algorithm Fast-Broadcasting under the simple-adversarial model the last node of the main path of a simple-path-graph gets the source message by step $\mathcal{O}(n \log n \log D)$ with probability at least $1 - (0.45)^{n \log D}$.

LEMMA 2. *Fix a simple-path-graph G and consider all wake-up patterns under the simple-adversarial wake-up model. By stage $3600 \cdot n \log D$, the number of successful stages is at least $72D \log n$ with probability at least $1 - (0.45)^{n \log D}$. This number of successful stages is sufficient to deliver the source message to node v_k.*

Proof. We consider only substeps of type A. Consider $3600 \cdot n \log D$ consecutive stages since the beginning of Algorithm Fast-Broadcasting. By Lemma 1, stage a is successful with probability at least 0.1; moreover, this probability is at least 0.1 independently of successes in other stages. Note, however, that such events are not independent; only conditional probabilities of success are at least 0.1. Hence the commonly used probabilistic inequalities, such as Chernoff-type bounds, cannot be used in our case. We present short and simple calculations to avoid using more advanced probabilistic bounds for conditional probabilities.

The probability that among $3600n\log D$ consecutive stages at most $72D\log n$ are successful is at most

$$\sum_{k=0}^{72D\log n}\binom{3600n\log D}{k}\cdot\left(\frac{9}{10}\right)^{3600n\log D-k}$$

$$\le\binom{3600n\log D}{0}\cdot\left(\frac{9}{10}\right)^{3600n\log D}+\binom{3600n\log D}{1}\cdot\left(\frac{9}{10}\right)^{3600n\log D-1}$$

$$+\sum_{k=2}^{72D\log n}\frac{(3600n\log D)^{3600n\log D+1}}{k^k(3600n\log D-k)^{3600n\log D-k}}\cdot\left(\frac{9}{10}\right)^{3600n\log D-k}$$

$$\le 4001n\log D\cdot\left(\frac{9}{10}\right)^{3600n\log D}$$

$$+\sum_{k=2}^{72D\log n}3600n\log D\cdot\left(\frac{3600n\log D}{k}\right)^{k}\left[\frac{9}{10}\cdot\left(1+\frac{k}{3600n\log D-k}\right)\right]^{3600n\log D-k}$$

$$\le 4001n\log D\cdot\left(\frac{9}{10}\right)^{3600n\log D}$$

$$+72D\log n\cdot 3600n\log D\cdot\left(\frac{3600n\log D}{72n\log D}\right)^{72n\log D}\left[\frac{9}{10}\cdot\frac{50}{49}\right]^{49\cdot 72n\log D}$$

$$\le 4001n\log D\cdot\left(\frac{9}{10}\right)^{3600n\log D}+72D\log n\cdot 3600n\log D\cdot\left(50^{72}\cdot(0.92)^{49\cdot 72}\right)^{n\log D},$$

which is at most $(0.45)^{n\log D}$, for sufficiently large n. We used the inequalities $\frac{b^b}{e^b}\le b!\le\frac{b^{b+1}}{e^b}$ for $b\ge 2$ and the fact that function $\left(\frac{C}{x}\right)^x$ is increasing for $0<x\le\frac{C}{e}$ for positive constant C. We also used the inequality $\frac{D}{\log D}\le\frac{n}{\log n}$.

In order to complete the proof, we show that if there are at least $72D\log n$ successful stages by stage a, then $\{m : f_a(m)<k\}=\emptyset$. Let a_x, for $x=1,\ldots,\log n$, be the first stage after $72D\cdot x$ successful stages. It is enough to prove, by induction on x, that $|\{m : f_{a_x}(m)<k\}|<n/2^x$. For $x=1$ this is straightforward. Assume that this inequality is true for x. We show it for $x+1$. Suppose the contrary: $|\{m : f_{a_{x+1}}(m)<k\}|\ge n/2^{x+1}$. It follows that during each successful stage b, where $a_x\le b<a_{x+1}$, at least $\frac{n/2^{x+1}}{36}$ values $f_b(m)$ are smaller than $f_{b+1}(m)$. Since in stage a_x there were fewer than $n/2^x$ such values and each of them may increase at most D times, we obtain that in stage a_{x+1} the set $\{m : f_{a_{x+1}}(m)<k\}$ would have fewer than

$$\frac{\frac{n}{2^x}\cdot D}{\frac{n}{36\cdot 2^{x+1}}\cdot 72D}=1$$

elements, which contradicts the assumption that $|\{m : f_{a_{x+1}}(m)<k\}|>n/2^{x+1}$. □

LEMMA 3. *There exists a matrix T of format $r\times 3600r(1+\log r)\log D$ such that, for any n-node simple-path-graph G of radius at most D and any wake-up pattern, Procedure Fast-Broadcasting(T) delivers the source message to the last node of the main path of G in $3600n\log D$ stages for sufficiently large n.*

Proof. Since r is linear in n, we have $r=cn$ for some constant c. Knowledge of c is not necessary in the analysis, but, since we do not know this constant, the proof has to be split into two cases.

First, consider D such that $\log D > 20(1+c)$. In this case we consider only substeps of type A. There are at most

$$\sum_{k=1}^{D} \binom{r}{n} \cdot \binom{n}{k} \cdot k! \cdot k^{n-k} \leq D^{\alpha n}$$

different simple-path-graphs G with n nodes and radius at most D for some positive constant $\alpha < 1.1$. This is because

$$\binom{r}{n} \cdot \binom{n}{k} \cdot k! \cdot k^{n-k} \leq 2^r \cdot 2^n \cdot k^n = 2^{n(1+c)+n\log k} \leq 2^{n(1+c)+n\log D} < D^{1.05 \cdot n}$$

for any $k \leq D$, and consequently

$$\sum_{k=1}^{D} \binom{r}{n} \cdot \binom{n}{k} \cdot k! \cdot k^{n-k} \leq D \cdot D^{1.05 \cdot n} < D^{1.1 \cdot n}$$

for sufficiently large n ($n > 20$).

Let $\hat{T}$ be the random matrix defined previously. By Lemma 2, the probability that Algorithm Fast-Broadcasting working on $\hat{T}$ delivers the source message to the last node of the main path of G by stage $3600n \log D$, for all simple-path-graphs G and all wake-up patterns $\mathcal{W}_m$, is at least

$$1 - \sum_{G} (0.45)^{n \log D} \geq 1 - D^{\alpha n} \cdot (0.45)^{n \log D} > 1 - D^{1.1n} \cdot (0.45)^{n \log D} > 0 \ ,$$

where the sum is taken over all simple-path-graphs G with n nodes chosen from r labels, and of radius at most D. Using the probabilistic method we obtain that there is a matrix T satisfying the lemma.

Next, consider D such that $\log D \leq 20(1+c)$. In this case we consider only substeps of type B. Since D is constant by step $D \cdot r \in \mathcal{O}(n)$, every node of the main path will receive the source message by the round-robin argument. This concludes the proof. □

For every wake-up pattern $\mathcal{W}_m$ and every step i of Procedure Fast-Broadcasting(T), we define an integer $f'_i(m)$ as follows by analogy to $f_a(m)$. If node v_k has the source message after step i, we fix $f'_i(m) = k$. Otherwise, $f'_i(m) = l$, where v_{l+1} is the last node on the main path which does not have the source message after step i but has a neighbor having the source message after step i, assuming that pattern $\mathcal{W}_m$ is used. Obviously $f_a(m) = f'_{(a+1)(1+\log r)-1}(m)$. Note that the above definition, similar to the definition of f_i, may be applied to the case without adversary or under simple adversary or normal adversary.

LEMMA 4. *Fix a path-graph G and a matrix T. For any wake-up pattern $\mathcal{W}_m$ and any step i, values $f'_i(m)$ are the same for network G under the adversarial model and for network $\bar{G}$ under the simple-adversarial model. Consequently, for any stage a, values $f_a(m)$ are the same for network G under the adversarial model and for network $\bar{G}$ under the simple-adversarial model.*

Proof. We consider two executions of Procedure Fast-Broadcasting(T): the first on graph G under the adversarial model and the second on graph $\bar{G}$ under the simple-adversarial model. The idea of the proof is to overcome two major differences between the two executions: additional ("backward") edges in graph G and the additional communication rule in the second execution under the simple adversary. We have to

show that these two differences do not influence the "front" (rightmost active layer) of computation in these two executions. For H equal to G or to $\bar{G}$, we adopt the following notation:

- $N_l[H]$ is the set of in-neighbors of v_l in H;
- $M_j[H]$ is the set of nodes in H which have the source message after step j;
- $f'_i(m)[H]$ has the meaning defined before for Procedure Fast-Broadcasting(T) working on H.

In order to prove the lemma, it is enough to prove the following invariant after step i of Procedure Fast-Broadcasting(T):

A_i: $f'_i(m)[G] = f'_i(m)[\bar{G}] = l$,
B_i: $N_l[G] \cap M_i[G] = N_l[\bar{G}] \cap M_i[\bar{G}]$.

We prove the invariant by induction on i. For $i = 1$ it is straightforward. Assume A_i and B_i hold. We prove A_{i+1} and B_{i+1}.

Proof of A_{i+1}. Let $f'_i(m)[G] = l$. By assumption A_i, we get also $f'_i(m)[\bar{G}] = l$. In view of the adversarial and simple-adversarial models of the algorithm description and of assumption B_i, node v_l gets the source message after step i in G under the adversarial model if and only if it gets the source message after step i in $\bar{G}$ under the simple-adversarial model. Since, for all $l' > l$, $v_{l'}$ does not have an in-neighbor which holds the source message after step i, for both executions, in view of A_i, we conclude that $v_{l'}$ does not have the source message after step $i+1$ in either G or in $\bar{G}$.

Suppose that $f'_{i+1}(m)[G] > f'_{i+1}(m)[\bar{G}]$. Hence there exists a node v outside the main path, which is woken up after step $i+1$, that is an in-neighbor of $v_{f'_{i+1}(m)[G]}$ in G but is not an in-neighbor of $v_{f'_{i+1}(m)[G]}$ in $\bar{G}$. This is a contradiction because, by definition of $\bar{G}$, v has an out-neighbor $v_{l'}$ in $\bar{G}$ for some $l' > f'_{i+1}(m)[G]$; hence $f'_{i+1}(m)[\bar{G}] \geq l' > f'_{i+1}(m)[G]$.

Suppose that $f'_{i+1}(m)[G] < f'_{i+1}(m)[\bar{G}]$. Hence there exists a node v outside the main path, which is woken up after step $i+1$, that is an in-neighbor of $v_{f'_{i+1}(m)[\bar{G}]}$ in $\bar{G}$ but is not an in-neighbor of $v_{f'_{i+1}(m)[\bar{G}]}$ in G. This contradicts the fact that $\bar{G}$ is a subgraph of G.

Proof of B_{i+1}. We have proved that $f'_{i+1}(m)[G] = f'_{i+1}(m)[\bar{G}] = l'$ for some $l' \geq l$. We show that $N_{l'}[G] \cap M_{i+1}[G] = N_{l'}[\bar{G}] \cap M_{i+1}[\bar{G}]$. Notice that for $l'' > l'$, $v_{l''} \notin M_{i+1}[G]$ and $v_{l''} \notin M_{i+1}[\bar{G}]$ by definitions of $f'_{i+1}(m)[G]$ and $f'_{i+1}(m)[\bar{G}]$.

If $v_{l'-1} \in N_{l'}[G] \cap M_{i+1}[G]$, then from the proof of invariant A_{i+1} we get $l'-1 = l$ but also $v_{l'-1} = v_l \in N_{l'}[\bar{G}] \cap M_{i+1}[\bar{G}]$ because this node got the source message in step $i+1$ (in view of the adversarial and simple-adversarial models of the algorithm description and of assumption B_i).

If v is outside the main path and $v \in N_{l'}[G] \cap M_{i+1}[G]$, then $v \in M_{i+1}[\bar{G}]$ because the same wake-up pattern is used. Also $v \in N_{l'}[\bar{G}]$ because otherwise there would exist an index $l'' > l'$ such that $v \in N_{l''}[\bar{G}]$, which contradicts the inequality $f'_{i+1}(m)[\bar{G}] = l' < l''$.

If $v_{l'-1} \in N_{l'}[\bar{G}] \cap M_{i+1}[\bar{G}]$, then from the proof of invariant A_{i+1} we get $l'-1 = l$ and $v_{l'-1} = v_l \in N_{l'}[G] \cap M_{i+1}[G]$, similar to the above argument for the dual case (where G is interchanged with $\bar{G}$).

If v is outside the main path and $v \in N_{l'}[\bar{G}] \cap M_{i+1}[\bar{G}]$, then, since $\bar{G}$ is a subgraph of G, we have $v \in N_{l'}[G]$. Since the same wake-up pattern is used in G, we have $v \in M_{i+1}[G]$. □

THEOREM 5. *There is a matrix T of format $r \times 3600r(1+\log r)\log D$ such that, for every n-node graph G of radius D, Procedure Fast-Broadcasting(T) performs broadcasting on G in time $\mathcal{O}(n \log n \log D)$.*

Proof. Take the matrix T from Lemma 3. Suppose the contrary: after step $3600n(1+\log r)\log D$, there is a node v without the source message. Consider the subgraph H of G, which contains a shortest directed path $v_0,\ldots,v_k=v$ from the source to node v, with all induced edges between nodes of this path, and all those in-neighbors v' of nodes $v_1,\ldots,v_k$ which received the source message by step $3600n(1+\log r)\log D$, together with the corresponding arcs (v',v_l). By definition, H is a path-graph, and hence $\bar{H}$ is a simple-path-graph. By Lemma 3 we obtain that v received the source message in $\bar{H}$ by step $3600n(1+\log r)\log D$. (We need to apply Lemma 3, under the simple-adversarial model, to the wake-up pattern "generated" by Procedure Fast-Broadcasting(T) working on G: every node in $\bar{H}$ is woken up, under the simple-adversarial wake-up model, in the time step in which it gets the source message for the first time when Procedure Fast-Broadcasting(T) is executed on G.) From Lemma 4, we obtain that the same is true in H under the adversarial model. Since the considered wake-up pattern is generated by Procedure Fast-Broadcasting(T) working on G, we conclude that v received the source message by step $3600n(1+\log r)\log D$ when Procedure Fast-Broadcasting(T) is executed on G. This is a contradiction, which concludes the proof. □

We conclude this section by observing that the assumption that radius D is known to all nodes can be removed without changing our result. It is enough to apply Algorithm Fast-Broadcasting for parameter r and for eccentricities 2^{2^i} for $i=1,\ldots,\lceil\log\log r\rceil$. Broadcasting will be completed after the execution of Algorithm Fast-Broadcasting for $i=\lceil\log\log D\rceil$. The total time will be at most four times larger than the running time of Algorithm Fast-Broadcasting when D is known.

Observe that using only substeps of type B in Procedure Fast-Broadcasting(T) we can trivially get the estimate $\mathcal{O}(nD)$ on broadcasting time. Hence the upper bound from Theorem 5 can be refined to $\mathcal{O}(n\cdot\min\{\log n\log D, D\})$.

3. Broadcasting with unknown bound on network size. In this section we show how our estimate of broadcasting time changes if the nodes do not know any parameters of the network: neither its radius D nor any upper bound r on the number of nodes. Denote by $AFB(x,y)$ the execution of Algorithm Fast-Broadcasting for the upper bound x on the size of the network and for radius y, running in time $3600x(1+\log x)\log y$ (it exists by Theorem 5). We construct the following.

ALGORITHM MODIFIED-FAST-BROADCASTING.

$i:=1$
repeat forever
  $i:=i+1$, $l:=1$
  while $2^{2^l}<2^{i-l}$ **do**
    $AFB(2^{i-l},2^{2^l})$ (1)
    $l:=l+1$
  $AFB(2^{i-l},2^{i-l})$ (2)

The above algorithm uses a doubling technique to estimate unknown parameters D and r. Since the running time of $AFB(x,y)$ depends superlinearly on the upper bound x on the size of the network and logarithmically on radius y, we increase the estimate of the size exponentially and the estimate of the radius doubly exponentially. Of course, since we know neither r nor D, the loop repeat is executed without termination. As defined in the introduction, time of broadcasting for a given network is the smallest integer t such that all nodes of the network are informed after step t.

The following observations hold.

1. For every n-node graph G with parameters r and D, broadcasting on G is completed by the time when algorithm $AFB(2^{\lceil \log r \rceil}, \min\{2^{\lceil \log r \rceil}, 2^{2^{\lceil \log \log D \rceil}}\})$ is executed. This happens for $i = \lceil \log r \rceil + \lceil \log \log D \rceil$ and $l = \lceil \log \log D \rceil$, either in (1) if $\lceil \log r \rceil > 2^{\lceil \log \log D \rceil}$ or in (2) otherwise.

2. $ABF(2^{i-l}, 2^{2^l})$ performs broadcasting in $3600 \cdot 2^{i-l} \cdot (i-l+1) \cdot 2^l$ steps. $ABF(2^{i-l}, 2^{i-l})$ performs broadcasting in $3600 \cdot 2^{i-l} \cdot (i-l+1) \cdot (i-l)$ steps.

3. Fix i. Let l_0 be the largest index l for which $AFB(2^{i-l}, 2^{2^l})$ is executed in (1). The execution of loop "while" lasts at most

$$\sum_{l=1}^{l_0} 3600 \cdot 2^{i-l} \cdot (i-l+1) \cdot 2^l \leq \sum_{l=1}^{\lceil \log i \rceil} 3600 \cdot 2^{i-l} \cdot (i-l+1) \cdot 2^l \leq 3600 \cdot \lceil \log i \rceil \cdot 2^i \cdot i$$

steps. The execution of (2) lasts

$$3600 \cdot 2^{i-l_0-1} \cdot (i-l_0-1+1) \cdot (i-l_0-1) \leq 3600 \cdot 2^{i-l_0-1} \cdot (i-l_0+1) \cdot 2^{l_0+1}$$

steps. The latter inequality follows from the condition $2^{2^{l_0}} \geq 2^{i-l_0-1}$. We further have

$$3600 \cdot 2^{i-l_0-1} \cdot (i-l_0+1) \cdot 2^{l_0+1} = 3600 \cdot 2^{i-l_0}(i-l_0+1) \cdot 2^{l_0} \leq 3600 \lceil \log i \rceil \cdot 2^j \cdot i \ .$$

4. The total number of steps until the execution of "repeat" for $i = i_0$ is at most

$$\sum_{i=2}^{i_0} 2 \cdot 3600 \cdot \lceil \log i \rceil \cdot 2^i \cdot i \leq 7200 \cdot 2^{i_0+1} \cdot i_0 \cdot \lceil \log i_0 \rceil \ ,$$

since (2) lasts at most the same time as the last preceding execution of (1).

5. For any r and D, the total running time of Algorithm Modified-Fast-Broadcasting is at most

$$7200 \cdot 2^{\lceil \log r \rceil + \lceil \log \log D \rceil + 1} \cdot (\lceil \log r \rceil + \lceil \log \log D \rceil) \cdot \lceil \log(\lceil \log r \rceil + \lceil \log \log D \rceil) \rceil \ ,$$

which is of order $\mathcal{O}(r \log r \log \log r \log D) = \mathcal{O}(n \log n \log \log n \log D)$. This proves the following theorem.

THEOREM 6. *Algorithm Modified-Fast-Broadcasting completes broadcasting on any n-node network of radius D in time $\mathcal{O}(n \log n \log \log n \log D)$, even when nodes do not know any parameters of the network or any bound on its size.*

Similar to section 2, the above upper bound on broadcasting time can be refined to $\mathcal{O}(n \cdot \min\{\log n \log \log n \log D, D\})$.

4. Optimal broadcasting in complete layered networks. In [13] the authors prove a lower bound $\Omega(n \log D)$ on deterministic broadcasting time on any n-node network of radius D. This is done using complete layered networks. All nodes of such networks can be partitioned into layers $L_0, L_1, \ldots, L_D$ where L_0 consists of the source and the set of directed edges is $\{(v,w) : v \in L_i, w \in L_{i+1}, i = 0, 1, \ldots, D-1\}$. More precisely, it is shown in [13] that for every deterministic broadcasting algorithm there is a complete layered n-node network of radius D, such that this algorithm requires time $\Omega(n \log D)$ to perform broadcast on this network. This result holds even when n and D are known to all nodes.

In this section we present a deterministic broadcasting algorithm which works on every complete layered n-node network of radius D in time $\mathcal{O}(n \log D)$, and thus it is optimal. Hence, any lower bound sharper than $\Omega(n \log D)$, on broadcasting time in arbitrary radio networks, would have to be established for graphs more complicated than complete layered networks. Our result is also an improvement of the upper bound $\mathcal{O}(n \log n)$, proved in [12] for n-node complete layered networks.

We use the following definition of an (r,k)-selective family. A family $\mathcal{F}$ of subsets of R is called (r,k)-*selective*, for $k \leq r$, if, for every subset Z of $\{1,\ldots,r\}$, such that $|Z| \leq k$, there is a set $F \in \mathcal{F}$ and element $z \in Z$ such that $Z \cap F = \{z\}$.

LEMMA 7 (see [13]). *For every $r \geq 2$ and $k \leq r$, there exists an (r,k)-selective family $\mathcal{F}$ of size $\mathcal{O}(k \log((r+1)/k))$.*

Let $\mathcal{F}_i$ denote an $(r, 2^i)$-selective family for $i = 1, \ldots, \log r$. By Lemma 7, we can assume that $\phi_i = |\mathcal{F}_i| \leq \alpha 2^i \log((r+1)/2^i)$ for some constant $\alpha > 0$ and for all $i = 1, \ldots, \log r$. Let $\mathcal{F}_i = \{F_i(1), \ldots, F_i(\phi_i)\}$.

ALGORITHM COMPLETE-LAYERED.
for $i = 1, \ldots, \log r$ **do**
  for $j = 1, \ldots, \phi_i$ **do**
    if $v \in F_i(j)$ and v got the source message **then** v transmits
for $j = 1, \ldots, r$ **do**
  if $v = j$ and v got the source message **then** v transmits.

THEOREM 8. *Algorithm Complete-Layered completes broadcasting in $\mathcal{O}(n \log D)$ time for any n-node complete layered network of radius D.*

Proof. For $D = 1$ the proof is obvious. Assume $D \geq 2$. Fix an n-node complete layered network G of radius D. Let L_l denote the lth layer of G, and let $d_l = |L_l|$, for $l = 0, \ldots, D$. Let t_l denote the step in which all nodes in L_l received the source message for the first time.

Claim. $t_{l+1} - t_l \leq 4\alpha d_l \log\big(2(r+1)/d_l\big)$ for every $l = 0, \ldots, D-1$.

In step $t_l + 1$ all nodes in L_l start transmitting. After at most

$$\begin{aligned}
\sum_{i=1}^{\lceil \log d_l \rceil} f_i &\leq \alpha \sum_{i=1}^{\lceil \log d_l \rceil} 2^i \log((r+1)/2^i) \\
&\leq \alpha \left[\sum_{i=1}^{\lceil \log d_l \rceil} 2^i \log(r+1) - \sum_{i=1}^{\lceil \log d_l \rceil} i \cdot 2^i \right] \\
&\leq \alpha \Big[2^{\lceil \log d_l \rceil + 1} \log(r+1) - (2^{\lceil \log d_l \rceil + 1} \lceil \log d_l \rceil - 2^{\lceil \log d_l \rceil + 1} + 1) \Big] \\
&\leq \alpha 2^{\lceil \log d_l \rceil + 1} \log \frac{2(r+1)}{d_l} \\
&\leq 4\alpha d_l \log \frac{2(r+1)}{d_l}
\end{aligned}$$

steps, all nodes in L_l complete transmissions according to the selective family $\mathcal{F}_{\lceil \log d_l \rceil}$. By definition of $\mathcal{F}_{\lceil \log d_l \rceil}$, there is a step among $t_l + 1, \ldots, t_l + \lfloor 4\alpha d_l \log \frac{2(r+1)}{d_l} \rfloor$ such that exactly one node in L_l transmits in this step. Consequently all nodes in L_{l+1} get the source message by step $t_l + \lfloor 4\alpha d_l \log \frac{2(r+1)}{d_l} \rfloor$. This completes the proof of the claim.

Since $\sum_{l=0}^{D} d_l = n$ and $t_0 = 0$, we have

$$\begin{aligned} t_D = \sum_{l=0}^{D-1}(t_{l+1} - t_l) &\leq 4\alpha \sum_{l=0}^{D-1} d_l \log \frac{2(r+1)}{d_l} \\ = 4\alpha \sum_{l=0}^{D-1} \left(\log \frac{(r+1)^{d_l}}{d_l^{d_l}} + d_l \right) &\leq 4\alpha \log \frac{(r+1)^{n-d_D}}{\Pi_{l=0}^{D-1} d_l^{d_l}} + 4\alpha n \\ \leq 4\alpha(n-d_D) \log \frac{r+1}{(n-d_D)/D} + 4\alpha n \ . \end{aligned}$$

We used the fact that

$$\prod_{l=0}^{D-1} d_l^{d_l} \geq \left(\frac{n-d_D}{\sum_{l=0}^{D-1} d_l \cdot \frac{1}{d_l}} \right)^{n-d_D} = \left(\frac{n-d_D}{D} \right)^{n-d_D} ,$$

which follows from the inequality between geometric and harmonic averages.

Since, for $D \geq 2$, the function $x \cdot \log \frac{D(r+1)}{x}$ is increasing for $x \leq r+1$, we have

$$4\alpha(n-d_D) \log \frac{r+1}{(n-d_D)/D} + 4\alpha n \leq 4\alpha n \log \frac{r+1}{n/D} + 4\alpha n \in \mathcal{O}(n \log D) \ . \qquad \square$$

5. Conclusion. We presented a deterministic broadcasting algorithm working in time $\mathcal{O}(n \log n \log D)$ for arbitrary n-node ad hoc radio networks of radius D, thus shrinking the gap between the upper bound and the best currently known lower bound $\Omega(n \log D)$ [13] on broadcasting time in ad hoc radio networks to a logarithmic factor. While our upper bound has been recently improved to $\mathcal{O}(n \log^2 D)$ in [14], a logarithmic gap between the bounds still remains. Closing this gap is a challenging open problem.

REFERENCES

[1] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg, *A lower bound for radio broadcast,* J. Comput. System Sci., 43 (1991), pp. 290–298.

[2] R. Bar-Yehuda, O. Goldreich, and A. Itai, *On the time complexity of broadcast in radio networks: An exponential gap between determinism and randomization,* J. Comput. System Sci., 45 (1992), pp. 104–126.

[3] S. Basagni, D. Bruschi, and I. Chlamtac, *A mobility-transparent deterministic broadcast mechanism for ad hoc networks,* IEEE/ACM Trans. Networking, 7 (1999), pp. 799–807.

[4] S. Basagni, A. D. Myers, and V.R. Syrotiuk, *Mobility-independent flooding for real-time multimedia applications in ad hoc networks,* in Proceedings of the IEEE Emerging Technologies Symposium on Wireless Communications & Systems, Richardson, TX, 1999.

[5] D. Bruschi and M. Del Pinto, *Lower bounds for the broadcast problem in mobile radio networks,* Distrib. Comput., 10 (1997), pp. 129–135.

[6] I. Chlamtac and A. Faragó, *Making transmission schedules immune to topology changes in multi-hop packet radio networks,* IEEE/ACM Trans. Networking, 2 (1994), pp. 23–29.

[7] I. Chlamtac, A. Faragó, and H. Zhang, *Time-spread multiple access (TSMA) protocols for multihop mobile radio networks,* IEEE/ACM Trans. Networking, 5 (1997), pp. 804–812.

[8] I. Chlamtac and O. Weinstein, *The wave expansion approach to broadcasting in multihop radio networks,* IEEE Trans. Communications, 39 (1991), pp. 426–433.

[9] B.S. Chlebus, L. Gasieniec, A. Gibbons, A. Pelc, and W. Rytter, *Deterministic broadcasting in unknown radio networks,* Distrib. Comput., 15 (2002), pp. 27–38.

[10] B.S. Chlebus, L. Gąsieniec, A. Östlin, and J.M. Robson, *Deterministic radio broadcasting,* in Proceedings of the 27th International Colloquium on Automata, Languages and Programming (ICALP'2000), Geneva, Switzerland, 2000, Lecture Notes in Comput. Sci. 1853, U. Montanari, J.D.P. Rolim, and E. Welzl, eds., Springer-Verlag, Berlin, 2000, pp. 717–728.

[11] M. Chrobak, L. Gąsieniec, and D. Kowalski, *The wake-up problem in multi-hop radio networks,* in Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, 2004, pp. 985–993.

[12] M. Chrobak, L. Gąsieniec, and W. Rytter, *Fast broadcasting and gossiping in radio networks,* in Proceedings of the 41st Symposium on Foundations of Computer Science (FOCS'2000), Redondo Beach, CA, pp. 575–581.

[13] A.E.F. Clementi, A. Monti, and R. Silvestri, *Selective families, superimposed codes, and broadcasting on unknown radio networks,* in Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms, Washington, DC, 2001, pp. 709–718.

[14] A. Czumaj and W. Rytter, *Broadcasting algorithms in radio networks with unknown topology,* in Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS'2003), Cambridge, MA, pp. 492–501.

[15] G. De Marco and A. Pelc, *Faster broadcasting in unknown radio networks,* Inform. Process. Lett., 79 (2001), pp. 53–56.

[16] I. Gaber and Y. Mansour, *Centralized broadcast in multihop radio networks,* J. Algorithms, 46 (2003), pp. 1–20.

[17] D. Kowalski and A. Pelc, *Deterministic broadcasting time in radio networks of unknown topology,* in Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'2002), Vancouver, BC, Canada, pp. 63–72.

[18] D. Kowalski and A. Pelc, *Broadcasting in undirected ad hoc radio networks,* in Proceedings of the 22nd Annual ACM Symposium on Principles of Distributed Computing (PODC'2003), Boston, MA, pp. 73–82.

[19] E. Kushilevitz and Y. Mansour, *An $\Omega(D \log(N/D))$ lower bound for broadcast in radio networks,* SIAM J. Comput., 27 (1998), pp. 702–712.

SIAM J. DISCRETE MATH.
Vol. 18, No. 2, pp. 347–352

ON-LINE LOAD BALANCING OF TEMPORARY TASKS ON IDENTICAL MACHINES*

YOSSI AZAR† AND LEAH EPSTEIN‡

Abstract. We prove an exact lower bound of $2-\frac{1}{m}$ on the competitive ratio of any deterministic algorithm for load balancing of temporary tasks on m identical machines. We also show a lower bound of $2-\frac{2}{m+1}$ for randomized algorithms. For small values of m we give an improved randomized lower bound of $2-\frac{1}{m}$.

Key words. on-line, competitive ratio, temporary tasks, load balancing, scheduling

AMS subject classifications. 68Q20, 68Q25

DOI. 10.1137/S0895480197329296

1. Introduction.

We consider the problem of nonpreemptive on-line load balancing of temporary tasks on m identical machines. Tasks (jobs) arrive at arbitrary times. Each task has a weight and a duration. A task has to be assigned upon its arrival to exactly one of the machines, thereby increasing the *load* on this machine by the weight of the task. The increase of the load is only for a time period equal to the duration of the task. The duration of each task becomes known only upon its termination. This model is called temporary tasks of unknown duration. Once a task has been assigned to a machine it cannot be reassigned to another machine. We define the cost of an algorithm to be the maximum load over machines and time. The goal is to minimize the cost. We say that the competitive ratio of a deterministic on-line algorithm is r if for all sequences the cost of the on-line algorithm is at most r times the cost of an optimal off-line algorithm which knows all events in advance. Note that this definition corresponds to the ratio of the maximum load over machines and time of the on-line algorithm to the maximum load over machines and time of the optimal off-line algorithm. One may also consider a momentary competitive ratio, that is, the maximum over time of the ratio between the maximum load over the machines of the on-line algorithm and the maximum load over the machines of the optimal off-line algorithm. This is a much stronger definition, and it is folklore that a reasonable (i.e., constant) competitive ratio cannot be achieved for this definition.

The problem of scheduling tasks on identical machines was first introduced by Graham [12, 13]. He gave a greedy algorithm, "List Scheduling," which is $2-\frac{1}{m}$ competitive, where m is the number of machines. The upper bound was proved for permanent tasks, i.e., tasks that start at arbitrary times but continue forever. Nevertheless, his $2-\frac{1}{m}$ analysis of the upper bound holds for temporary tasks as well.

In this paper we show that the algorithm of Graham is optimal by proving a matching lower bound. We show a lower bound of $2-\frac{1}{m}$ on the competitive ratio

*Received by the editors October 21, 1997; accepted for publication (in revised form) March 9, 2004; published electronically November 9, 2004. A preliminary version of this paper appeared in the *Proceedings of the 5th Israeli Symposium on Theory of Computing and Systems*, 1997, pp. 119–125.
http://www.siam.org/journals/sidma/18-2/32929.html

†School of Computer Science, Tel-Aviv University, Tel-Aviv, 69978, Israel (azar@tau.ac.il). This author's research was supported in part by the Israel Science Foundation.

‡School of Computer Science, The Interdisciplinary Center, Herzliya, 46150, Israel (lea@idc.ac.il). This work carried out when the author was at Tel-Aviv University. This author's research was supported in part by the Israel Science Foundation.

of any deterministic on-line algorithm for load balancing of temporary tasks on identical machines. One could hope that with randomized algorithms, an upper bound strictly less than 2 can be achieved. The regular definition of the competitive ratio for randomized algorithms corresponds to an oblivious adversary (see [7]). Specifically, the competitive ratio of a randomized on-line algorithm is r if for all sequences, the expected cost of the on-line algorithm over its coin flips is at most r times the cost of the optimal off-line algorithm. Note that this definition corresponds to an adversary that knows the specification of the on-line algorithm but does not know the coin flips and the assignments. Moreover, a lower bound for an oblivious adversary is also a lower bound for an adaptive one (see [7] for the definition of adaptive adversaries).

We also show a lower bound of $2-\frac{2}{m+1}$ on the competitive ratio of any randomized on-line algorithm for the problem. In fact, for $m = 2, 3, 4$, we can improve the lower bound to $2 - \frac{1}{m}$, which implies that "List Scheduling" is also optimal in these cases. The randomized lower bound for general m requires a sequence of tasks of super-polynomial length in m. If we restrict the sequence to have a polynomial length, we prove a lower bound of $2 - O(\frac{\log\log m}{\log m})$ on the competitive ratio for any randomized algorithm.

Recall that Graham [12, 13] considered only permanent tasks. He showed that the greedy algorithm "List Scheduling" has a competitive ratio of exactly $2 - \frac{1}{m}$ for m machines. In other words, he showed a family of instances for which the algorithm "List Scheduling" delivers a schedule where the maximum load is $2-\frac{1}{m}$ the maximum load in an optimal schedule. For $m = 2, 3$, the algorithm is optimal [10]. However, the algorithm of Graham is not optimal (for all $m \geq 4$) [11, 9]. Bartal et al. [5] were the first to show an algorithm whose competitive ratio is strictly below $c < 2$ (for all m). More precisely, their algorithm achieves a competitive ratio of $2 - \frac{1}{70}$. Later, the algorithm was generalized by Karger, Phillips, and Torng [14] to yield an upper bound of 1.945. Very recently, Albers [1] designed a 1.923 competitive algorithm and improved the lower bound to 1.852 (the previous lower bound for permanent tasks was 1.8370 [6]). The best lower bound known for randomized algorithms is 1.582 (for large m) [8, 15]. For $m = 2$, the randomized competitive ratio is precisely 4/3 [5]. We show that in contrast to permanent tasks, the simple algorithm of Graham turns out to be optimal for temporary tasks. Moreover, even randomization cannot reduce the competitive ratio below $2 - o(1)$. Note that for $m = 2$, our tight randomized lower bound is 3/2. We also prove the same lower bound for the known duration case. This is in contrast to the competitive ratio for permanent jobs that is 4/3.

To prove our randomized lower bound we introduce a new technique that converts a lower bound for deterministic algorithms to a lower bound for randomized algorithms. More precisely, we show that a lower bound for deterministic algorithms that maintains a fixed value for the cost of the optimal assignment is a lower bound for randomized algorithms.

The problem of on-line load balancing of temporary tasks was introduced by Azar, Broder, and Karlin [2]. They studied the restricted assignment case, i.e., each task can be assigned only to a machine in a subset which may depend on the task. They showed an $\Omega(\sqrt{m})$ lower bound in contrast to the $\Theta(\log m)$ competitive ratio for permanent tasks [4]. A matching upper bound was given in [3]. Load balancing of temporary tasks was also studied for the related machines model. In this model the increase of the load on a machine is the ratio of the weight of the task and the speed of that machine. An algorithm which is 20 competitive and a general lower bound of $3 - o(1)$ were given by [3].

2. Notation. We denote the input sequence by $\sigma = \sigma_1, \ldots, \sigma_r$. Each event σ_i is an arrival or a departure of a job (task). We view σ as a sequence of times; the time σ_i is the moment after the ith event happens. We denote the weight of job j by w_j, its arrival time by a_j, and its departure time (which is unknown until it departs) by d_j. An on-line algorithm has to assign a job upon its arrival without knowing the future jobs and the durations of jobs that have not departed yet. We compare the performance of on-line algorithms and the optimal off-line algorithm that knows the sequence of jobs and their durations in advance.

Let $J_i = \{j | a_j \leq \sigma_i < d_j\}$ be the active jobs at time σ_i. For a given algorithm A (on-line or off-line) let A_j be the machine to which job j is assigned. Let

$$l_k^A(i) = \sum_{\{j | A_j = k, j \in J_i\}} w_j$$

be the load on machine k at time σ_i, which is the sum of weights of all jobs assigned to k that are active at this time. The cost of an algorithm A is the maximum load ever achieved by any of the machines, i.e., $C_A = max_{i,k}\ l_k^A(i)$. The competitive ratio of A is r if for any sequence $\mathrm{E}(C_A) \leq r \cdot C_{opt}$, where the expectation is taken over the coin flips of the on-line algorithm and C_{opt} is the cost of the optimal off-line algorithm.

3. A lower bound for deterministic algorithms. In this section we prove the deterministic lower bound.

THEOREM 3.1. *Any deterministic on-line algorithm for load balancing of temporary tasks has the competitive ratio of at least* $2 - \frac{1}{m}$.

Proof. We consider the following sequence. First $m(m-1)$ unit jobs arrive ($w_j = 1$). Since there are m machines, there is at least one machine (call it x) with load at least $m-1$. Then all jobs depart, except $m-1$ jobs on this machine. Next $m-1$ jobs of load $m-1$ arrive. There are two possible cases:

1. The on-line algorithm keeps at least one machine empty (with no jobs assigned to it). In this case, the load of the on-line algorithm on some machine is at least $2(m-1)$ since at least two jobs of load $m-1$ were assigned to one machine, or a job of load $m-1$ was assigned to machine x.
   In the first phase, the off-line algorithm assigns the $m-1$ unit jobs which will not depart to one machine and distributes the $(m-1)^2$ unit jobs that will depart evenly on the other $m-1$ machines. Then it assigns one job of weight $m-1$ to each of the other machines to have the maximum load of $m-1$. In this case, the ratio between the costs of the on-line and the off-line algorithms is at least $2(m-1)/(m-1) = 2$.
2. The on-line algorithm does not keep an empty machine; i.e., there is now one job of weight $m-1$ on each machine except x, on which there are $m-1$ unit jobs. Next one additional (and final) job of weight m arrives. The on-line algorithm must assign it to one of the machines, which results in a load of $2m-1$.
   We show that the off-line algorithm can assign the jobs, having a maximum load of m. In the first phase, the unit jobs that do not depart are each assigned to a different machine. The other jobs are distributed so that the load on each machine is exactly $m-1$. In the second phase, only $m-1$ unit jobs, each scheduled on a different machine, are left. The jobs of load $m-1$ are added to those machines, yielding a load of m and keeping one machine empty. Finally, the job of load m is assigned to the empty machine. In this case, the competitive ratio is at least $(2m-1)/m = 2 - 1/m$.

In both possible cases, the competitive ratio is at least $2-\frac{1}{m}$, and thus any on-line algorithm has at least this competitive ratio. □

4. Lower bounds for randomized algorithms. In this section we prove lower bounds for randomized on-line algorithms. We first introduce a general theorem that converts lower bounds for deterministic algorithms to lower bounds for randomized algorithms. In order to make this conversion possible, the value of the optimal off-line cost should be fixed and known in advance. We represent a lower bound for a deterministic algorithm by a tree. Each path in the tree is one possible lower bound sequence. Each node in the tree is a subsequence, and a child node of a node is one possible way to continue the sequence. The size $|T|$ of a tree T is defined to be the number of leaves in T (the number of possible sequences). We consider both the unknown duration case and the known duration case. In the first case the duration of a job is known only when it departs, whereas in the second case the duration of a job is known upon its arrival.

THEOREM 4.1. *Let r_1 (r_2, respectively) be a deterministic lower bound on the competitive ratio for load balancing of temporary tasks with unknown (known, respectively) duration, where the optimal value of the load is known in advance. Then, r_1 (r_2, respectively) is also a lower bound for randomized algorithms for the same problem, i.e., load balancing of temporary tasks with unknown (known, respectively) duration.*

Proof. Consider a lower bound tree T' for deterministic algorithms with a fixed value of optimal load which is known in advance. We show how to convert it into a lower bound for randomized algorithms. A lower bound for temporary tasks with unknown duration is converted into a lower bound for randomized unknown duration, and a lower bound for known duration is converted into a lower bound for randomized known duration. We first slightly modify the lower bound tree as follows: each possible sequence $\sigma \in T'$ is followed by the departures of all existing jobs. This can be done for unknown duration and also for known duration. Note that the new tree T satisfies $|T| = |T'|$. Next we recall the adaptation of Yao's theorem for on-line algorithms. It states that a lower bound for the competitive ratio of deterministic algorithms on any distribution on the input is also a lower bound for randomized algorithms and is given by $E(C_{on}/C_{opt})$. The main idea of the proof is to construct sequences in which, on one hand, the new value C'_{opt} is the same as the known optimal value C_{opt} in the lower bound of the original tree T and, on the other hand, with high probability the new value of C'_{on} is also the same as the value C_{on} of T. To construct the lower bound we choose (uniformly at random) a leaf of the tree T that corresponds to a sequence. Define this short sequence as a segment. Repeat the choice of segments $|T|k$ times and concatenate the sequences into one long sequence. This defines a distribution on the set of possible long sequences. Since the optimal off-line costs of all possible segments are the same, the optimal off-line cost of every resulting sequence is C_{opt} as well. For any deterministic algorithm, there exists a leaf in T that has a cost C_{on} for the on-line algorithm. With probability at least $\frac{1}{|T|}$, the cost of the on-line algorithm on a specific segment (and thus for the whole sequence) is C_{on}. The probability that the cost C_{on} would not be achieved in one segment is at most $(1-\frac{1}{|T|})^{|T|k} \leq e^{-k}$, and thus with probability at least $1-e^{-k}$ the competitive ratio is C_{on}/C_{opt}; otherwise it is at least 1. We calculate $E(\frac{C'_{on}}{C'_{opt}})$, where C'_{opt}, and C'_{on} are, respectively, the off-line and on-line costs of the long sequence

$$E\left(\frac{C'_{on}}{C'_{opt}}\right) \geq (1-e^{-k})\frac{C_{on}}{C_{opt}} + e^{-k} .$$

Since this is true for every k,

$$E\left(\frac{C'_{on}}{C'_{opt}}\right) \geq \frac{C_{on}}{C_{opt}} ,$$

which is exactly the competitive ratio of the lower bound of the tree T. □

Now, we would like to apply Theorem 4.1 to Theorem 3.1. However, in Theorem 3.1 the optimal value of the load is not fixed and hence cannot be known in advance. Therefore, we prove a slightly smaller deterministic lower bound with a fixed known optimal value.

THEOREM 4.2. *Any deterministic on-line algorithm for load balancing of temporary tasks has a competitive ratio of at least* $2 - \frac{2}{m+1}$ *even if the optimal value is known in advance.*

Proof. We consider the following sequence. First m^2 unit jobs ($w_j = 1$) arrive. Since there are m machines, there is at least one machine (call it x) with a load of at least m. Now all jobs depart, except m jobs on this machine. Now m jobs of weight m arrive. Since machine x has load m, the on-line algorithm must assign two jobs of weight m to one machine, or assign one job of weight m to the machine x. In both cases, the maximum load of the on-line algorithm is at least $2m$. The off-line algorithm distributes the m large jobs and the m unit jobs that remained from the first phase evenly on the m machines, and thus has a load of $m + 1$. The other $m(m-1)$ unit jobs of the first phase are also distributed evenly on the m machines. The ratio between the costs of the two algorithms is at least $\frac{2m}{m+1} = 2 - \frac{2}{m+1}$. □

COROLLARY 4.3. *Any randomized on-line algorithm for load balancing of temporary tasks has a competitive ratio of at least* $2 - \frac{2}{m+1}$.

Proof. The proof follows from Theorems 4.2 and 4.1. In this case, there are $(m^2)^m = m^{2m}$ different possibilities for the m^2 unit jobs to be placed, and there are m^{2m} leaves in the lower bound tree. □

It is possible to improve the lower bound for small numbers of machines. The following remark can be proved using similar techniques to the lower bound for general m, where case analysis is applied (for complete details see the preliminary version of this paper).

Remark 1. Any randomized on-line algorithm for load balancing of temporary tasks on $m \leq 4$ machines has a competitive ratio of at least $2 - \frac{1}{m}$. Moreover, for $m = 2$ the competitive ratio is at least $\frac{3}{2}$ even in the known duration model.

The general randomized lower bound is proved using sequences of exponential length. We can achieve a lower bound of $2 - o(1)$ using sequences of polynomial length as well.

Remark 2. Any randomized on-line algorithm for load balancing of temporary tasks has a competitive ratio of at least $2 - O(\frac{\log\log m}{\log m})$ even when the input sequence is of polynomial length in m.

To prove this remark we use the following sequence repeated a polynomial number of times. Take an integer h such that $2h^h \leq m$. The sequence starts with the arrival of mh unit jobs. Then $mh - m$ of them are chosen uniformly at random and depart. Next m jobs of weight h arrive and afterward all jobs depart. The calculations are omitted (for complete details see the preliminary version of this paper).

5. Concluding remarks. We proved lower bounds of $2-o(1)$ on the competitive ratio of the load balancing problem of temporary tasks. Note that there is a small gap between the randomized lower bound and the optimal deterministic one. One would like to know the exact bound for randomized algorithms or at least if randomization

helps at all to reduce the competitive ratio in these problems. It follows from our results that randomization may help slightly only for $m \geq 5$. Knowing whether the durations of the tasks can help in reducing the competitive ratio strictly below 2 for both deterministic and randomized algorithms remains an open question.

6. Acknowledgment. We would like to thank Allan Borodin for many helpful discussions.

REFERENCES

[1] S. Albers, *Better bounds for online scheduling*, SIAM J. Comput., 29 (1999), pp. 459–473.
[2] Y. Azar, A. Broder, and A. Karlin, *On-line load balancing*, Theoret. Comput. Sci., 130 (1994), pp. 73–84.
[3] Y. Azar, B. Kalyanasundaram, S. Plotkin, K. Pruhs, and O. Waarts, *On-line load balancing of temporary tasks*, J. Algorithms, 22 (1997), pp. 93–110.
[4] Y. Azar, J. Naor, and R. Rom, *The competitiveness of on-line assignments*, J. Algorithms, 18 (1995), pp. 221–237.
[5] Y. Bartal, A. Fiat, H. Karloff, and R. Vohra, *New algorithms for an ancient scheduling problem*, J. Comput. System Sci., 1995, pp. 359–366.
[6] Y. Bartal, H. Karloff, and Y. Rabani, *A better lower bound for on-line scheduling*, Inform. Process. Lett., 50 (1994), pp. 113–116.
[7] A. Borodin and R. El-Yaniv, *Online Computation and Competitive Analysis*, Cambridge University Press, Cambridge, UK, 1998.
[8] B. Chen, A. van Vliet, and G. Woeginger, *A lower bound for randomized on-line scheduling algorithms*, Inform. Process. Lett., 51 (1994), pp. 219–222.
[9] B. Chen, A. van Vliet, and G. Woeginger, *New lower and upper bounds for on-line scheduling*, Oper. Res. Lett., 16 (1994), pp. 221–230.
[10] U. Faigle, W. Kern, and G. Turan, *On the performance of online algorithms for partition problems*, Acta Cybernet., 9 (1989), pp. 107–119.
[11] G. Galambos and G. Woeginger, *An on-line scheduling heuristic with better worst case ratio than Graham's list scheduling*, SIAM J. Comput., 22 (1993), pp. 349–355.
[12] R. Graham, *Bounds for certain multiprocessor anomalies*, Bell System Tech. J., 45 (1966), pp. 1563–1581.
[13] R. Graham, *Bounds on multiprocessing timing anomalies*, SIAM J. Appl. Math., 17 (1969), pp. 416–429.
[14] D. Karger, S. Phillips, and E. Torng, *A better algorithm for an ancient scheduling problem*, J. Algorithms, 20 (1996), pp. 400–430.
[15] J. Sgall, *On-line scheduling on parallel machines*, Technical report CMU-CS-94-144, Carnegie Mellon University, Pittsburgh, PA, 1994.

SIAM J. DISCRETE MATH.
Vol. 18, No. 2, pp. 353–361

AN INTERLACING RESULT ON NORMALIZED LAPLACIANS*

GUANTAO CHEN†, GEORGE DAVIS†, FRANK HALL†, ZHONGSHAN LI†,
KINNARI PATEL†, AND MICHAEL STEWART†

Abstract. Given a graph G, the normalized Laplacian associated with the graph G, denoted $\mathcal{L}(G)$, was introduced by F. R. K. Chung and has been intensively studied in the last 10 years. For a k-regular graph G, the normalized Laplacian $\mathcal{L}(\mathcal{G})$ and the standard Laplacian matrix $L(G)$ satisfy $L(G) = k\mathcal{L}(\mathcal{G})$, and hence they have the same eigenvectors and their eigenvalues are directly related. However, for an irregular graph G, $\mathcal{L}(\mathcal{G})$ and $L(G)$ behave quite differently, and the normalized Laplacian seems to be more natural. In this paper, Cauchy interlacing-type properties of the normalized Laplacian are investigated, and the following result is established. Let G be a graph, and let $H = G - e$, where e is an edge of G. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n = 0$ be the eigenvalues of $\mathcal{L}(G)$, and let $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_n$ be the eigenvalues of $\mathcal{L}(H)$. Then, $\lambda_{k-1} \geq \theta_k \geq \lambda_{k+1}$ for each $k = 1, 2, 3, \ldots, n$, where $\lambda_0 = 2$ and $\lambda_{n+1} = 0$. Applications are given for eigenvalues of graphs obtained from special graphs by adding or deleting a few edges. A short proof is given of the result that G is a graph with each component a nontrivial bipartite graph if and only if $2 - \lambda$ is an eigenvalue of $\mathcal{L}(G)$ for each eigenvalue λ of $\mathcal{L}(G)$.

Key words. normalized Laplacian, eigenvalues, graphs, subgraphs, Cauchy interlacing property

AMS subject classifications. 05C50, 15A18

DOI. 10.1137/S0895480103438589

1. Introduction. All graphs in this paper are simple graphs, namely, finite graphs without loops or parallel edges. Let G be a graph, and let $V(G)$ and $E(G)$ denote the vertex set and the edge set of G, respectively. Two vertices are adjacent if they are two end vertices of an edge, and two edges are adjacent if they share a common end vertex. A vertex and an edge are incident if the vertex is one end vertex of the edge. For any vertex $v \in V(G)$, let d_v denote the degree of v.

Suppose that $V(G) = \{v_1, v_2, \ldots, v_n\}$. An $n \times n$ (0,1)-matrix $A := A(G) = (a_{ij})$ is called the *adjacency matrix* of G if

$$a_{ij} = \begin{cases} 1 & \text{if } v_i v_j \in E(G), \\ 0 & \text{otherwise.} \end{cases}$$

The eigenvalues of $A(G)$ have been studied extensively. We refer the reader to Biggs [1] and Schwenk and Wilson [11] for literature in this area.

The *standard Laplacian* $L := L(G) = (L_{ij})$ of a graph G of order n is the $n \times n$ matrix L defined as follows:

$$L_{ij} = \begin{cases} d_{v_i} & \text{if } v_i = v_j, \\ -1 & \text{if } v_i v_j \in E(G), \\ 0 & \text{otherwise.} \end{cases}$$

Let T denote the diagonal matrix with the (i,i)th entry having value d_{v_i}. The *normalized Laplacian* of G is the $n \times n$ matrix $\mathcal{L} := \mathcal{L}(G) = (\mathcal{L}_{ij})$ given by

*Received by the editors December 11, 2003; accepted for publication (in revised form) February 5, 2004; published electronically November 9, 2004.
http://www.siam.org/journals/sidma/18-2/43858.html
†Department of Mathematics and Statistics, Georgia State University, Atlanta, GA 30303 (gchen@mathstat.gsu.edu, gdavis@mathstat.gsu.edu, fhall@mathstat.gsu.edu, zli@mathstat.gsu.edu, kpatel@mathstat.gsu.edu, mstewart@mathstat.gsu.edu). The research of the first author was partially supported by NSF grant DMS-0070059 and NSA grant H98230-04-01-0030.

$$\mathcal{L}_{ij} = \begin{cases} 1 & \text{if } v_i = v_j \text{ and } d(v_i) \neq 0, \\ -\frac{1}{\sqrt{d_{v_i} d_{v_j}}} & \text{if } v_i v_j \in E(G), \\ 0 & \text{otherwise.} \end{cases}$$

We can write $\mathcal{L} = T^{-1/2} L T^{-1/2}$ with the convention that $T^{-1}(i,i) = 0$ if $d_{v_i} = 0$. For convenience, we simply call $\mathcal{L}$ the (normalized) *Laplacian*. The study of Laplacians began with a book by Cvetković, Doob, and Sachs [7]. Laplacians have been intensively studied by Chung and her collaborators in a series of papers [3, 4, 5, 6] and by Chung in [2]. Clearly, for k-regular graphs, we have $L = kI - A$ and $\mathcal{L} = \frac{1}{k} L = I - \frac{1}{k} A$, so we have an easy one-to-one correspondence between the eigenvalues of $\mathcal{L}$, L, and A. For nonregular graphs, there is different behavior among these three, and the standard Laplacian would seem to be the most natural one. However, as pointed out in [2, p. 2], the eigenvalues of normalized Laplacians are in a "normalized" form, and the spectra of normalized Laplacians relate well to other graph invariants for general graphs in a way that the other two definitions fail to do. The advantages of this definition are perhaps due to the fact that it is consistent with the eigenvalues in spectral geometry and in stochastic processes. Many results known for the Laplacians of regular graphs can be generalized to all graphs.

Chung [2] notes that for any graph G, its Laplacian can be written as

$$\mathcal{L} = SS^T,$$

where S is the matrix whose rows are indexed by the vertices and whose columns are indexed by the edges of G such that each column corresponding to an edge $e = v_i v_j$ has an entry $1/\sqrt{d_{v_i}}$ in the row corresponding to v_i, has an entry $-1/\sqrt{d_{v_j}}$ in the row corresponding to v_j, and has zero entries elsewhere. Hence, all eigenvalues of $\mathcal{L}$ are real and nonnegative. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ be the eigenvalues of $\mathcal{L}$. The following results can be found in [2].

THEOREM 1.1. *For a graph G on n vertices, we have the following.*

1. $\lambda_n = 0$.
2. $\sum_i \lambda_i \leq n$ *with equality holding if and only if G has no isolated vertices.*
3. *For $n \geq 2$, $\lambda_{n-1} \leq \frac{n}{n-1}$ with equality holding if and only if G is a complete graph on n vertices. Also, for a graph G without isolated vertices, we have $\lambda_1 \geq \frac{n}{n-1}$.*
4. *For a graph which is not a complete graph, we have $\lambda_{n-1} \leq 1$.*
5. *If G is connected, then $\lambda_{n-1} > 0$. If $\lambda_{n-i+1} = 0$ and $\lambda_{n-i} \neq 0$, then G has exactly i connected components.*
6. *For all $i \leq n$, we have $\lambda_i \leq 2$ with $\lambda_1 = 2$ if and only if a connected component of G is a nontrivial bipartite graph.*
7. *The spectrum of a graph is the union of the spectra of its connected components.*

THEOREM 1.2. *The eigenvalues of the Laplacians of some special graphs are given below.*

1. *For the complete graph K_n on n vertices, the eigenvalues are 0 and $n/(n-1)$ (with multiplicity $n-1$).*
2. *For the complete bipartite graph $K_{m,n}$ on $m+n$ vertices, the eigenvalues are 0, 1 (with multiplicity $m+n-2$), and 2.*
3. *For the star S_n on n vertices, the eigenvalues are 0, 1 (with multiplicity $n-2$), and 2.*

THEOREM 1.3. *A connected graph is a nontrivial bipartite graph if and only if, for each eigenvalue λ_i of the Laplacian, $2-\lambda_i$ is also an eigenvalue of the Laplacian.*

Chung [2] develops many of her results through the use of harmonic eigenfunctions. An alternative approach is more purely matrix theoretic. We now give another proof of Theorem 1.3. The reader is referred to [10] for terminology.

Proof. Suppose that $2-\lambda_i$ is an eigenvalue of $\mathcal{L}(G)$ for each eigenvalue λ_i. Since 0 is an eigenvalue of $\mathcal{L}(G)$, 2 is also an eigenvalue. Thus, G is a nontrivial bipartite graph by point 6 of Theorem 1.1.

Conversely, suppose that G is a connected nontrivial bipartite graph. By point 6 of Theorem 1.1, both 0 and 2 are eigenvalues of $\mathcal{L}$. We first observe that $I-\mathcal{L}$ is a nonnegative matrix. Since G is connected, $I-\mathcal{L}$ is irreducible. Since the eigenvalues of $\mathcal{L}$ are in the closed interval $[0,2]$, the eigenvalues of $I-\mathcal{L}$ are in the closed interval $[-1,1]$. In particular, 1 and -1 are eigenvalues of $I-\mathcal{L}$. By the Perron–Frobenius theorem [10, p. 508], the eigenvalue 1 of $I-\mathcal{L}$ (and hence the eigenvalue 0 of $\mathcal{L}$) has multiplicity 1 (this gives another proof of point 5 of Theorem 1.1).

We now use the "equal spacing" property [10, p. 511] of the nonzero eigenvalues of the nonnegative irreducible matrix $I-\mathcal{L}$. This implies that -1 has multiplicity 1 (so there are $k=2$ eigenvalues of maximum modulus); so does eigenvalue 2 of $\mathcal{L}$. Further, from the "equal spacing" property, all the nonzero eigenvalues of $I-\mathcal{L}$ occur in pairs centered at the origin. Hence, the eigenvalues of $\mathcal{L}$ not equal to 1 occur in pairs centered at 1. Thus, λ_i is an eigenvalue of $\mathcal{L}$ if and only if $2-\lambda_i$ is an eigenvalue of $\mathcal{L}$. □

2. Interlacing eigenvalues. Eigenvalue interlacing provides a useful tool for obtaining inequalities and regularity results concerning the structure of graphs in terms of eigenvalues of adjacency matrices and Laplacians. Much research has been done in this area. For a survey of literature, we refer the reader to Haemers [8]. The following result is known as Cauchy's interlacing theorem.

THEOREM 2.1. *Let A be a real $n\times n$ symmetric matrix and B be an $(n-1)\times(n-1)$ principal submatrix of A. If*

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \quad \textit{and} \quad \theta_1 \geq \theta_2 \geq \cdots \geq \theta_{n-1}$$

are the eigenvalues of A and B, respectively, then

$$\lambda_1 \geq \theta_1 \geq \lambda_2 \geq \theta_2 \geq \cdots \geq \theta_{n-1} \geq \lambda_n.$$

Let G be a graph of order n, and let $H = G - v$, where v is a vertex of G. Theorem 2.1 gives an interlacing property of the eigenvalues of G and the eigenvalues of H, which we refer to as the vertex version of the interlacing property. Theorem 2.1 does not directly apply to the standard Laplacian (or the Laplacian) of G and H since the principal submatrices of a standard Laplacian (or Laplacian) may no longer be the standard Laplacian (or Laplacian) of a subgraph. However, the following result due to van den Heuvel [9] reflects an edge version of the interlacing property.

THEOREM 2.2. *Let G be a graph, and let $H = G - e$, where e is an edge of G. If*

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n = 0 \quad \textit{and} \quad \theta_1 \geq \theta_2 \geq \cdots \geq \theta_n = 0$$

are the eigenvalues of $L(G)$ and $L(H)$, respectively, then

$$\lambda_1 \geq \theta_1 \geq \lambda_2 \geq \cdots \geq \theta_{n-1} \geq \lambda_n.$$

Since the trace of $\mathcal{L}$ is n, when there are no isolated vertices, it is impossible to have an exactly parallel result to Theorem 2.2. The purpose of this article is to establish the following interlacing result on normalized Laplacians.

THEOREM 2.3. *Let G be a graph, and let $H = G - e$, where e is an edge of G. If*

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \quad \textit{and} \quad \theta_1 \geq \theta_2 \geq \cdots \geq \theta_n$$

are the eigenvalues of $\mathcal{L}(G)$ and $\mathcal{L}(H)$, respectively, then

$$\lambda_{i-1} \geq \theta_i \geq \lambda_{i+1} \quad \textit{for each } i = 1, 2, 3, 4, \ldots, n,$$

where $\lambda_0 = 2$ and $\lambda_{n+1} = 0$.

The following are direct consequences of Theorem 2.3.

COROLLARY 2.4. *Let G be a graph, and let H be a spanning subgraph of G such that $|E(G - H)| \leq t$ for some positive integer t. If*

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \quad \textit{and} \quad \theta_1 \geq \theta_2 \geq \cdots \geq \theta_n$$

are the eigenvalues of $\mathcal{L}(G)$ and $\mathcal{L}(H)$, respectively, then

$$\lambda_{i-t} \geq \theta_i \geq \lambda_{i+t} \quad \textit{for each } i = 1, 2, \ldots, n$$

with the convention of

$$\lambda_{-t+1} = \lambda_{-t+2} = \cdots = \lambda_0 = 2,$$
$$\lambda_{n+1} = \lambda_{n+2} = \cdots = \lambda_{n+t} = 0.$$

COROLLARY 2.5. *If G is a graph on n vertices and only t edges away from complete, then $n/(n-1)$ is an eigenvalue of $\mathcal{L}(G)$ with multiplicity at least $n-2t-1$.*

COROLLARY 2.6. *If G is a graph on $m+n$ vertices and the edge set $E(G)$ can be obtained from $K_{m,n}$ by deleting at most t edges, then 1 is an eigenvalue of $\mathcal{L}(G)$ with multiplicity at least $m + n - 2(t + 1)$.*

Let G be a graph, and let $x \in V(G)$. The neighborhood of x is

$$N(x) = \{y \ : \ xy \in E(G)\}.$$

For any two vertices u and v of G, we use $G/\{u, v\}$ to denote the graph obtained from G by contracting u and v to one vertex; i.e., $G/\{u, v\}$ is the graph obtained from G by deleting the vertices u and v and adding a new vertex (uv) such that the neighborhood of (uv) is the union of the neighborhoods of u and v. When u and v are adjacent, $G/\{u, v\}$ is the graph obtained from G by contracting the edge uv. Contraction of edges and vertices has many applications in graph theory. By contracting two special vertices of a graph, we obtain the following interlacing result.

THEOREM 2.7. *Let G be a graph, and let u and v be two vertices of G. Let*

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \quad \textit{and} \quad \theta_1 \geq \theta_2 \geq \cdots \geq \theta_{n-1}$$

be the eigenvalues of $\mathcal{L}(G)$ and $\mathcal{L}(G/\{u, v\})$, respectively. If $N(u) \cap (N(v) \cup \{v\}) = \emptyset$, then

$$\lambda_{i-1} \geq \theta_i \geq \lambda_{i+1} \quad \textit{for each } i = 1, 2, 3, 4, \ldots, n,$$

where $\lambda_0 = 2$ and $\lambda_{n+1} = 0$.

The proof of Theorem 2.3 will heavily depend on the Courant–Fischer theorem. Here and subsequently, the notation $g \perp g^{(k+1)}, \ldots, g^{(n)}$ means that g is orthogonal to $\operatorname{span}(g^{(k+1)}, \ldots, g^{(n)})$.

THEOREM 2.8 (Courant–Fischer). *For a real, symmetric $n \times n$ matrix A with eigenvalues*

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_n,$$

we have

$$\lambda_k = \max_{g^{(k+1)}, g^{(k+2)}, \ldots, g^{(n)} \in \mathbb{R}^n} \; \min_{\substack{g \perp g^{(k+1)}, g^{(k+2)}, \ldots, g^{(n)} \\ g \neq 0}} \frac{g^T A g}{g^T g}$$

and

$$\lambda_k = \min_{g^{(1)}, g^{(2)}, \ldots, g^{(k-1)} \in \mathbb{R}^n} \; \max_{\substack{g \perp g^{(1)}, g^{(2)}, \ldots, g^{(k-1)} \\ g \neq 0}} \frac{g^T A g}{g^T g}.$$

The following two lemmas will also be useful in what follows.

LEMMA 2.9. *Suppose that for real a, b, and γ,*

$$a^2 - 2\gamma^2 \geq 0, \qquad b^2 - \gamma^2 > 0, \qquad \text{and} \qquad \frac{a^2}{b^2} \leq 2.$$

Then

$$\frac{a^2 - 2\gamma^2}{b^2 - \gamma^2} \leq \frac{a^2}{b^2}.$$

Proof. The result follows from

$$\frac{a^2 - 2\gamma^2}{b^2 - \gamma^2} = \frac{a^2}{b^2} \frac{1 - 2\gamma^2/a^2}{1 - \gamma^2/b^2} \leq \frac{a^2}{b^2}.$$

The final inequality is clearly true when

$$\frac{\gamma^2}{b^2} \leq \frac{2\gamma^2}{a^2},$$

which is equivalent to $a^2/b^2 \leq 2$. ∎

LEMMA 2.10. *Let G be a graph on n vertices, let $L = L(G)$ be the standard Laplacian of G, and let $f = (f_1, \ldots, f_n)^T$ be a column vector in $\mathbb{R}^n$. Then,*

$$f^T L f = \sum_{i \sim j} (f_i - f_j)^2,$$

where $\sum_{i \sim j}$ runs over all unordered pairs $\{i, j\}$ for which v_i and v_j are adjacent.

Proof. Lemma 2.10 directly follows from the definition of L. ∎

3. Proof of Theorem 2.3. We adapt the Courant–Fischer theorem to the Laplacian using harmonic eigenfunctions. Recall that

$$\mathcal{L} = T^{-1/2} L T^{-1/2}.$$

We assume that $T^{1/2}$ is invertible; that is, there are no vertices of degree zero.

For vectors g and $g^{(j)}$, define the vectors

$$f = T^{-1/2} g \qquad \text{and} \qquad f^{(j)} = T^{1/2} g^{(j)}.$$

Note that

$$g \perp g^{(k+1)}, g^{(k+2)}, \ldots, g^{(n)}$$

if and only if

$$f \perp f^{(k+1)}, f^{(k+2)}, \ldots, f^{(n)}.$$

Applying the Courant–Fischer theorem to get the eigenvalues λ_k of $\mathcal{L}$ gives

$$\begin{aligned}
\lambda_k &= \max_{g^{(k+1)}, g^{(k+2)}, \ldots, g^{(n)} \in \mathbb{R}^n} \; \min_{\substack{g \perp g^{(k+1)}, g^{(k+2)}, \ldots, g^{(n)} \\ g \neq 0}} \frac{g^T T^{-1/2} L T^{-1/2} g}{g^T g} \\
&= \max_{g^{(k+1)}, g^{(k+2)}, \ldots, g^{(n)} \in \mathbb{R}^n} \; \min_{\substack{f \perp f^{(k+1)}, f^{(k+2)}, \ldots, f^{(n)} \\ f \neq 0}} \frac{f^T L f}{f^T T f} \\
&= \max_{f^{(k+1)}, f^{(k+2)}, \ldots, f^{(n)} \in \mathbb{R}^n} \; \min_{\substack{f \perp f^{(k+1)}, f^{(k+2)}, \ldots, f^{(n)} \\ f \neq 0}} \frac{f^T L f}{f^T T f} \\
&= \max_{f^{(k+1)}, f^{(k+2)}, \ldots, f^{(n)} \in \mathbb{R}^n} \; \min_{\substack{f \perp f^{(k+1)}, f^{(k+2)}, \ldots, f^{(n)} \\ f \neq 0}} \frac{\sum_{i \sim j} (f_i - f_j)^2}{\sum_j f_j^2 d_j},
\end{aligned}$$

where f_j is the jth component of f, $d_j = d(v_j)$ is the degree of v_j, and $\sum_{i \sim j}$ runs over all unordered pairs $\{i, j\}$ for which v_i and v_j are adjacent. The third line depends on the invertibility of T so that maximizing over vectors $f^{(k)}$ is equivalent to maximizing over vectors $g^{(k)}$. The final line depends on Lemma 2.10. The vector f can be viewed as a function $f(v)$ on the set of vertices that maps v_j to f_j. The function $f(v)$ is a harmonic eigenfunction.

The other half of the Courant–Fischer theorem gives

$$\lambda_k = \min_{f^{(1)}, f^{(2)}, \ldots, f^{(k-1)} \in \mathbb{R}^n} \; \max_{\substack{f \perp f^{(1)}, f^{(2)}, \ldots, f^{(k-1)} \\ f \neq 0}} \frac{\sum_{i \sim j} (f_i - f_j)^2}{\sum_j f_j^2 d_j}.$$

Without loss of generality, we assume that an edge between the particular vertices v_1 and v_2 is removed, and we consider the eigenvalues θ_k of the Laplacian of the modified graph. Two changes occur in the Courant–Fischer theorem when an edge is removed. The degrees of v_1 and v_2 are decreased from $d(v_1)$ and $d(v_2)$ to $d(v_1) - 1$ and $d(v_2) - 1$ so that

$$\sum_j f_j^2 d_j \to \sum_j f_j^2 d_j - f_1^2 - f_2^2.$$

Also, since v_1 and v_2 are no longer adjacent, the sum no longer includes the pair $\{1,2\}$ so that

$$\sum_{i\sim j}(f_i-f_j)^2 \to \sum_{i\sim j}(f_i-f_j)^2-(f_1-f_2)^2.$$

Note that the sum $\sum_{i\sim j}$ still runs over vertices that are adjacent in the *original graph*; in applying the theorem to the modified graph we explicitly subtract out $(f_1-f_2)^2$ instead of modifying the index set of the sum.

Thus,

$$\begin{aligned}
\theta_k &= \max_{f^{(k+1)},f^{(k+2)},\dots,f^{(n)}\in\mathbb{R}^n} \ \min_{\substack{f\perp f^{(k+1)},f^{(k+2)},\dots,f^{(n)}\\ f\neq 0}} \frac{\sum_{i\sim j}(f_i-f_j)^2-(f_1-f_2)^2}{\sum_j f_j^2 d_j - f_1^2 - f_2^2}\\
&\le \max_{f^{(k+1)},f^{(k+2)},\dots,f^{(n)}\in\mathbb{R}^n} \ \min_{\substack{f\perp f^{(k+1)},f^{(k+2)},\dots,f^{(n)}\\ f\neq 0\\ f_1=-f_2}} \frac{\sum_{i\sim j}(f_i-f_j)^2-(f_1-f_2)^2}{\sum_j f_j^2 d_j - f_1^2 - f_2^2}\\
&= \max_{f^{(k+1)},f^{(k+2)},\dots,f^{(n)}\in\mathbb{R}^n} \ \min_{\substack{f\perp f^{(k+1)},f^{(k+2)},\dots,f^{(n)},e_1+e_2\\ f\neq 0}} \frac{\sum_{i\sim j}(f_i-f_j)^2-4f_1^2}{\sum_j f_j^2 d_j - 2f_1^2}\\
&\le \max_{f^{(k+1)},f^{(k+2)},\dots,f^{(n)}\in\mathbb{R}^n} \ \min_{\substack{f\perp f^{(k+1)},f^{(k+2)},\dots,f^{(n)},e_1+e_2\\ f\neq 0}} \frac{\sum_{i\sim j}(f_i-f_j)^2}{\sum_j f_j^2 d_j}\\
&\le \max_{f^{(k)},f^{(k+1)},\dots,f^{(n)}\in\mathbb{R}^n} \ \min_{\substack{f\perp f^{(k)},f^{(k+1)},\dots,f^{(n)}\\ f\neq 0}} \frac{\sum_{i\sim j}(f_i-f_j)^2}{\sum_j f_j^2 d_j} = \lambda_{k-1}.
\end{aligned}$$

The vectors e_1 and e_2 are the standard basis vectors. In line four we have used Lemma 2.9, which is applicable with $\gamma^2 = 2f_1^2$ because of the inequality

$$\frac{\sum_{i\sim j}(f_i-f_j)^2}{\sum_j f_j^2 d_j} \le \frac{\sum_{i\sim j} 2(f_i^2+f_j^2)}{\sum_j f_j^2 d_j} = 2.$$

In a similar manner the second half of the Courant–Fischer theorem gives a lower bound on θ_k:

$$\begin{aligned}
\theta_k &= \min_{f^{(1)},f^{(2)},\dots,f^{(k-1)}\in\mathbb{R}^n} \ \max_{\substack{f\perp f^{(1)},f^{(2)},\dots,f^{(k-1)}\\ f\neq 0}} \frac{\sum_{i\sim j}(f_i-f_j)^2-(f_1-f_2)^2}{\sum_j f_j^2 d_j - f_1^2 - f_2^2}\\
&\ge \min_{f^{(1)},f^{(2)},\dots,f^{(k-1)}\in\mathbb{R}^n} \ \max_{\substack{f\perp f^{(1)},f^{(2)},\dots,f^{(k-1)}\\ f\neq 0\\ f_1=f_2}} \frac{\sum_{i\sim j}(f_i-f_j)^2}{\sum_j f_j^2 d_j - 2f_1^2}\\
&\ge \min_{f^{(1)},f^{(2)},\dots,f^{(k-1)}\in\mathbb{R}^n} \ \max_{\substack{f\perp f^{(1)},f^{(2)},\dots,f^{(k-1)},(e_1-e_2)\\ f\neq 0}} \frac{\sum_{i\sim j}(f_i-f_j)^2}{\sum_j f_j^2 d_j}\\
&\ge \min_{f^{(1)},f^{(2)},\dots,f^{(k)}\in\mathbb{R}^n} \ \max_{\substack{f\perp f^{(1)},f^{(2)},\dots,f^{(k)}\\ f\neq 0}} \frac{\sum_{i\sim j}(f_i-f_j)^2}{\sum_j f_j^2 d_j} = \lambda_{k+1}.
\end{aligned}$$

Thus,

$$\lambda_{k-1} \geq \theta_k \geq \lambda_{k+1}$$

with the convention $\lambda_0 = 2$ and $\lambda_{n+1} = 0$. The proof of the upper bound does not apply to θ_1, and the proof of the lower does not apply to θ_n. The values of λ_0 and λ_{n+1} have been chosen to make the upper and lower bounds true for θ_1 and θ_n. This follows immediately from bounds that hold for all eigenvalues of a normalized Laplacian: $0 \leq \theta_k \leq 2$.

We have assumed throughout that T is invertible (i.e., there is no vertex of degree zero). However, this is not restrictive; the inequality holds in general. If $d(v) = 0$ for m vertices, then the normalized Laplacian can be permuted so that

$$P\mathcal{L}P^T = \begin{pmatrix} \mathcal{L}_1 & 0 \\ 0 & 0_{m\times m} \end{pmatrix}$$

for some permutation matrix P. Thus, for $\mathcal{L}$,

$$\lambda_{n-m} = \lambda_{n-m+1} = \cdots = \lambda_n = 0.$$

The removal of an edge then affects only $\mathcal{L}_1$ so that the theorem can be applied to the submatrix. The additional zero eigenvalues, $\theta_{n-m+1} = \theta_{n-m+2} = \cdots = \theta_n = 0$, satisfy

$$\theta_k \geq \lambda_{k+1} = 0 \qquad \text{and} \qquad \theta_k \leq \lambda_{k-1} = 0$$

for $k = n - m + 1, \ldots, n$. Also $\theta_{n-m} \geq \lambda_{n-m+1} = 0$. Interlacing bounds for all other θ_k follow from the interlacing theorem applied to $\mathcal{L}_1$.

4. Concluding remarks. Finally, we give a few remarks on the proof of Theorem 2.7. We assume that the vertices v_i are indexed by integers with $v_1 = u$ and $v_2 = v$, where u and v are as in the statement of the theorem. Let J be an index set such that $j \in J$ if and only if $v_j \in N(v_1)$. Since $N(v_1) \cap (N(v_2) \cup \{v_2\}) = \emptyset$, we can view the contraction as the removal of all edges v_1v_j and the simultaneous addition of all edges v_2v_j for $j \in J$. Thus, the Courant–Fischer theorem applied to $\mathcal{L}(G/\{v_1, v_2\})$ is

$$\theta_k = \max_{f^{(k+1)},\ldots,f^{(n)}\in\mathbb{R}^n} \min_{\substack{f\perp f^{(k+1)},\ldots,f^{(n)} \\ f\neq 0}} \frac{\sum_{i\sim j}(f_i - f_j)^2 + \sum_{j\in J}(f_2 - f_j)^2 - (f_1 - f_j)^2}{\sum_j f_j^2 d_j - d_1 f_1^2 + d_1 f_2^2}.$$

If we impose the constraint $f_1 = f_2$, then these modifications disappear, and a nearly identical argument to that used to prove Theorem 2.3 gives the upper bound on θ_i. The lower bound follows similarly from the min max part of the Courant–Fischer theorem. Strictly speaking, the above expression for θ_k applies not to the graph $G/\{v_1, v_2\}$ but to $G/\{v_1, v_2\}$ together with the newly isolated vertex v_1. However, as before, the addition of an extra zero eigenvalue does not affect the interlacing.

REFERENCES

[1] N. L. Biggs, *Algebraic Graph Theory*, 2nd ed., Cambridge University Press, Cambridge, UK, 1993.

[2] F. R. K. Chung, *Spectral Graph Theory*, CBMS. Reg. Conf. Ser. Math. 92, AMS, Providence, RI, 1997.

[3] F. R. K. Chung, *Diameters and eigenvalues*, J. Amer. Math. Soc., 2 (1989), pp. 187–196.
[4] 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.
[5] F. R. K. Chung, A. Grigor'yan, and S.-T. Yau, *Upper bounds for eigenvalues of the discrete and continuous standard Laplacian operators*, Adv. Math., 117 (1996), pp. 165–178.
[6] F. R. K. Chung and S.-T. Yau, *A Harnack inequality for homogeneous graphs and subgraphs*, Comm. Anal. Geom., 2 (1994), pp. 627–640.
[7] D. M. Cvetković, M. Doob, and H. Sachs, *Spectra of Graphs, Theory and Application*, Academic Press, New York, London, 1980.
[8] W. H. Haemers, *Interlacing eigenvalues and graphs*, Linear Algebra Appl., 226/228 (1995), pp. 593–616.
[9] J. van den Heuvel, *Hamilton cycles and eigenvalues of graphs*, Linear Algebra Appl., 226/228 (1995), pp. 723–730.
[10] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
[11] A. J. Schwenk and R. J. Wilson, *Eigenvalues of graphs*, in Selected Topics in Graph Theory, L. W. Beineke and R. J. Wilson, eds., Academic Press, New York, London, 1978, pp. 307–336.

SIAM J. DISCRETE MATH.
Vol. 18, No. 2, pp. 362–381

THREE-DIMENSIONAL PERIODICITY AND ITS APPLICATION TO PATTERN MATCHING*

ZVI GALIL†, JONG GEUN PARK‡, AND KUNSOO PARK§

Abstract. We study three-dimensional periodicity in finite arrays and introduce four categories of three-dimensional periodicity: edge-periodic, side-aperiodic, side-periodic, and completely periodic. We analyze three-dimensional periodicity thoroughly, and we characterize periods of a finite three-dimensional array by a small number of vectors. This periodicity analysis leads to an efficient algorithm for three-dimensional pattern matching.

Key words. three-dimensional periodicity, pattern matching

AMS subject classifications. 68R15, 68R05, 68Q25

DOI. 10.1137/S0895480101390308

1. Introduction.

Let A be a d-dimensional array. A vector v is called a *period* of A if $A[w] = A[w+v]$ for every pair of points w and $w+v$ in A. If v is not a period of A, then there exists at least one pair of points $w, w+v$ such that $A[w] \neq A[w+v]$. Such a pair of points $(w, w+v)$ is called a *witness* of A against v.

Periodicity in an *infinite* array A is easily characterized by the number of independent periods of A. That is, an infinite d-dimensional array can have at most d independent periods. However, periodicity in a *finite* array is more complicated [AB, GP], and it has been an important subject of research, especially in connection with pattern matching [KMP, ABF, KR].

Periodicity in a finite one-dimensional array A is not much different from that in an infinite array because when p is the smallest period of A, A can be produced by many copies of the prefix of A of length p. This concept can be generalized to two and higher dimensions, where a finite d-dimensional array can be produced by many copies of a d-dimensional polyhedron. In two and higher dimensions, however, there are other kinds of periodicities due to finiteness of arrays. Amir and Benson [AB] were the first to study two-dimensional periodicity in finite arrays. They defined three categories of two-dimensional periodicity: line-periodic, radiant-periodic, and lattice-periodic. This study on two-dimensional periodicity has led to significant progress in two-dimensional pattern matching [ABF, GP, CCG, CPP].

In this paper we study three-dimensional periodicity in finite arrays and introduce the following four categories of three-dimensional periodicity: edge-periodic, side-aperiodic, side-periodic, and completely periodic. Karpinski and Rytter [KR] gave a classification based on two-dimensional periodicity of the faces of a three-dimensional array. We analyze three-dimensional periodicity thoroughly, and we characterize periods of a finite three-dimensional array by a small number of vectors. This periodicity analysis has an application to three-dimensional pattern matching.

*Received by the editors June 2, 2001; accepted for publication (in revised form) January 7, 2004; published electronically November 9, 2004. A preliminary version of this paper appeared in *Proceedings of the 9th ACM Symposium on Parallel Algorithms and Architectures*, 1997, pp. 53–62.
http://www.siam.org/journals/sidma/18-2/39030.html

†Department of Computer Science, Columbia University, New York, NY 10027 (galil@cs.columbia.edu). This author's research was partially supported by NSF grant CCR-93-16209.

‡Department of Mathematics Education, Chonbuk National University, Chonju, Korea.

§School of Computer Science and Engineering, Seoul National University, Seoul 151-742, Korea (kpark@theory.snu.ac.kr). This author's research was supported by MOST grant M1-0309-06-0003.

Three-dimensional pattern matching is defined as follows: Given a pattern P of size $m_1 \times m_2 \times m_3$ and a text T of size $n_1 \times n_2 \times n_3$, find all occurrences of the pattern in the text. For a simple description of time complexities, we assume $m = m_1 = m_2 = m_3$ and $n = n_1 = n_2 = n_3$. Let Σ be the alphabet from which the symbols of P and T are drawn. In contrast to one and two dimensions, there are only a few results on three-dimensional pattern matching. The two-dimensional matching algorithm due to Baker [Ba] and Bird [Bi] also solves three-dimensional pattern matching in $O((n^3 + m^3) \log m)$ time. Breslauer [Br] obtained an $O(n^3 + m^3 \log m)$-time algorithm using preprocessing, which needs a large space or randomization. Karpinski and Rytter [KR] gave a parallel algorithm whose text search takes optimal $O(\log m)$ time on the CREW PRAM and whose preprocessing takes $O(\log m)$ time using m^3 processors on the arbitrary CRCW PRAM. See [Ja] for various models of PRAM. A sequential version of Karpinski and Rytter's algorithm runs in $O(n^3 + m^3 \log m)$ time.

Our periodicity analysis has led to a parallel algorithm for three-dimensional pattern matching whose text search is alphabet-independent (i.e., with no assumptions on the alphabet Σ [ABF]) and which runs in optimal constant time on the common CRCW PRAM. Its time and processor complexities for preprocessing are the same as those of Karpinski and Rytter's algorithm. This algorithm was described in the preliminary version of this paper [GPP]. To avoid PRAM details, here we present a sequential version of the algorithm whose text search is alphabet-independent and runs in $O(n^3)$ time. Its preprocessing computes a witness table for the pattern, and it runs in $O(m^3)$ time if Σ is a constant-size alphabet or an integer alphabet in the range $[0, m^c]$ for some constant c [Fa], and in $O(m^3 \log m)$ time if Σ is an unbounded alphabet and only symbol comparisons are allowed on Σ. Hence, the sequential algorithm takes $O(n^3 + m^3)$ time in most cases of the alphabet Σ.

Basic definitions are given in the next section. In section 3 we revisit two-dimensional periodicity and find some of its properties. In section 4 we study three-dimensional periodicity in finite arrays. In section 5 we describe our text search algorithm based on three-dimensional periodicity. In section 6 we describe the computation of a witness table. Finally, we conclude in section 7.

2. Preliminaries. Let A be a three-dimensional array of size $m_1 \times m_2 \times m_3$. The positions of array A start from 0. The point $(0, 0, 0)$ is called the *origin* of array A. For any vector v, let A_v be the subarray of A consisting of all points $w \in A$ such that $w - v \in A$; see Figure 1. Note that a vector is defined by its start point and end point, and thus its start point does not need to be the origin as in $-v$ in Figure 1.

A *period* of A is a vector such that two copies of A, one shifted by the vector over the other, overlap without a mismatch. Formally, a vector v is a period of A (or we say that A is periodic with v) if $A[w] = A[w + v]$ for every pair of points $w, w + v$ in A (i.e., $A_{-v} = A_v$). If v is not a period of A, then there exists at least one pair of points $w, w + v$ such that $A[w] \neq A[w + v]$. Such a pair of points $(w, w + v)$ is called a *witness* of A against v.

For a vector $v = (a, b, c)$, let $\ell_x(v)$, $\ell_y(v)$, and $\ell_z(v)$ denote the absolute values of a, b, and c, respectively. The *length* of a vector v is the maximum of the absolute values of its coordinates, i.e., $|v| = \max(\ell_x(v), \ell_y(v), \ell_z(v))$. We say that v is a *valid* vector of array A if $\max\left(\frac{\ell_x(v)}{m_1}, \frac{\ell_y(v)}{m_2}, \frac{\ell_z(v)}{m_3}\right) < \frac{1}{8}$.

We define precedence relations on points as follows. If an array A has a period v in one direction, then $-v$ is also a period in the opposite direction. Hence we need consider only one of two opposite directions. Let $u = (a, b, c)$ and $v = (i, j, k)$.

1. If $a \leq i$, $b \leq j$, and $c \leq k$, then $u \prec_{+++} v$.

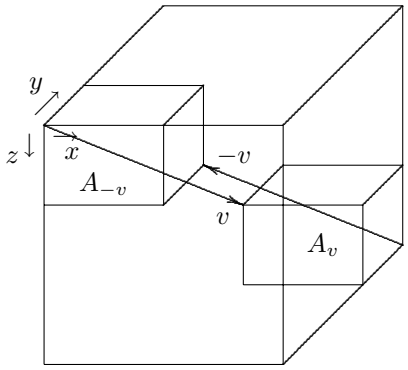


FIG. 1. A_v *and* A_{-v}.

2. If $a \le i$, $b \le j$, and $c \ge k$, then $u \prec_{++-} v$.
3. If $a \le i$, $b \ge j$, and $c \le k$, then $u \prec_{+-+} v$.
4. If $a \ge i$, $b \le j$, and $c \le k$, then $u \prec_{-++} v$.

We will also use 0 in place of + or − to mean equality; e.g., if $a \le i$, $b \le j$, and $c = k$, then $u \prec_{++0} v$. A sequence of points $u_1, \ldots, u_t$ is called a *monotone line* if $u_1 \prec_i \cdots \prec_i u_t$ for a fixed precedence relation $\prec_i$ such as $\prec_{++0}$.

EXAMPLE 1. *Four points* $(5,5,0)$, $(7,8,0)$, $(9,8,0)$, *and* $(9,9,0)$ *form a monotone line because* $(5,5,0) \prec_{++0} (7,8,0) \prec_{++0} (9,8,0) \prec_{++0} (9,9,0)$.

Let p, q, and r be three independent vectors. Two points (or vectors) u and v are *congruent modulo* p, q, r if $u - v = ap + bq + cr$ for some integers a, b, c. A point congruent to $(0,0,0)$ modulo p, q, r is called a *lattice point* on p, q, r. Similarly we can define congruence and lattice points in two dimensions.

We now define edge vectors, side vectors, and general vectors in three dimensions.

DEFINITION 1. *A vector* (a,b,c) *is an edge vector if exactly two coordinates of* a,b,c *are* 0*; a side vector if exactly one coordinate is* 0*; and a general vector if no coordinates are* 0. *There are three directions* $(+,0,0)$, $(0,+,0)$, $(0,0,+)$ *for edge vectors (called edge directions); six directions* $(+,+,0)$, $(+,-,0)$, $(+,0,+)$, $(+,0,-)$, $(0,+,+)$, $(0,+,-)$ *for side vectors (called side directions); and four directions* $(+,+,+)$, $(+,+,-)$, $(+,-,+)$, $(-,+,+)$ *for general vectors (called general directions). Side directions can be divided into three types* $(x,y,0)$, $(x,0,z)$, $(0,y,z)$, *each of which has two directions.*

DEFINITION 2. *A side direction is adjacent to a general direction if the two nonzero coordinates of the side direction have the same signs as the general direction. For example,* $(-,+,0)$, $(-,0,+)$, *and* $(0,+,+)$ *are adjacent to* $(-,+,+)$.

First we need a lemma that holds for any dimensions.

LEMMA 1. *Let* p *and* v *be vectors of the same direction or adjacent directions. Suppose that* p *is a period of an array* A. *Then* v *is a period of* A_p *if and only if* $p+v$ *is a period of* A.

Proof. Assume that v is a period of A_p. Since p and v are periods of A and A_p, respectively, we have $A_{p+v} = (A_p)_v = (A_p)_{-v} = (A_{-p})_{-v} = A_{-p-v}$, and thus $p+v$ is a period of A. We can prove the *if* case similarly. □

3. Two-dimensional periodicity revisited. Before we go on to three-dimensional periodicity, we need to study two-dimensional periodicity, because a three-

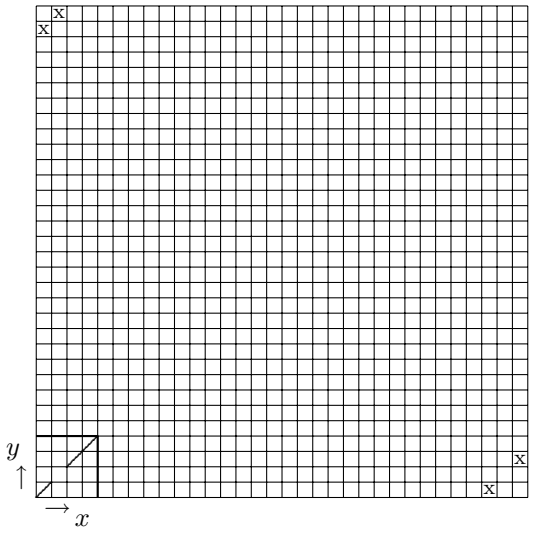


FIG. 2. *A* 32×32 *array* A *whose elements are either* x *or blank. The points marked with slashes in the lower-left corner (except the origin) are valid periods of* A*, which are* $(2,2)$ *and* $(3,3)$.

dimensional array A of size $m_1 \times m_2 \times m_3$ can be regarded as a two-dimensional array in the xy-plane, where each symbol is a string of length m_3 parallel to the z-axis.

DEFINITION 3. *A two-dimensional vector* (a,b) *is of quad-*I *direction if it is of direction* $(+,+)$ *or* $(+,0)$*, and of quad-*II *direction if* $(-,+)$ *or* $(0,+)$.

According to [AB, GP], there are the following three categories of two-dimensional periodicity:

1. line-periodic: One of the quad-I and quad-II directions has no valid periods, and the other has valid periods that are on a line going through $(0,0)$. See Figure 2.

2. radiant-periodic: One of the quad-I and quad-II directions has no valid periods, and the other has at least two independent valid periods. See Figure 3.

3. lattice-periodic: Both quad-I and quad-II directions have valid periods. The shortest quad-I period and the shortest quad-II period are called the *basis vectors*.

It was shown in [AB, GP] that all valid periods in the lattice-periodic case are represented by the basis vectors. Here we will show that all valid periods in the line-periodic and radiant-periodic cases can be represented in special forms by a small number of vectors. This result will be extended to three dimensions in the next section.

3.1. Line-periodicity. Suppose that a two-dimensional array A is line-periodic. Let $p_1, p_2, \ldots, p_s$ be all valid periods of A from shortest to longest. Let $u_i = p_i - p_{i-1}$ for $1 \le i \le s$, where $p_0 = (0,0)$. Let V be the subset of $\{u_1, \ldots, u_s\}$ such that u_1 is in V and u_i, $i \ge 2$, belongs to V if and only if $u_i \ne u_{i-1}$. We rename the vectors in V as $v_1, v_2, \ldots, v_t$ and they will be called *step vectors*.

EXAMPLE 2. *If the valid periods* p_i *from shortest to longest are* $(11,22)$*,* $(22,44)$*,* $(27,54)$*,* $(30,60)$*,* $(31,62)$*,* $(32,64)$*, and* $(33,66)$*, then the* u_i*'s and* v_i*'s are as follows:*

i	1	2	3	4	5	6	7
p_i	$(11,22)$	$(22,44)$	$(27,54)$	$(30,60)$	$(31,62)$	$(32,64)$	$(33,66)$
u_i	$(11,22)$	$(11,22)$	$(5,10)$	$(3,6)$	$(1,2)$	$(1,2)$	$(1,2)$
	v_1		v_2	v_3	v_4		

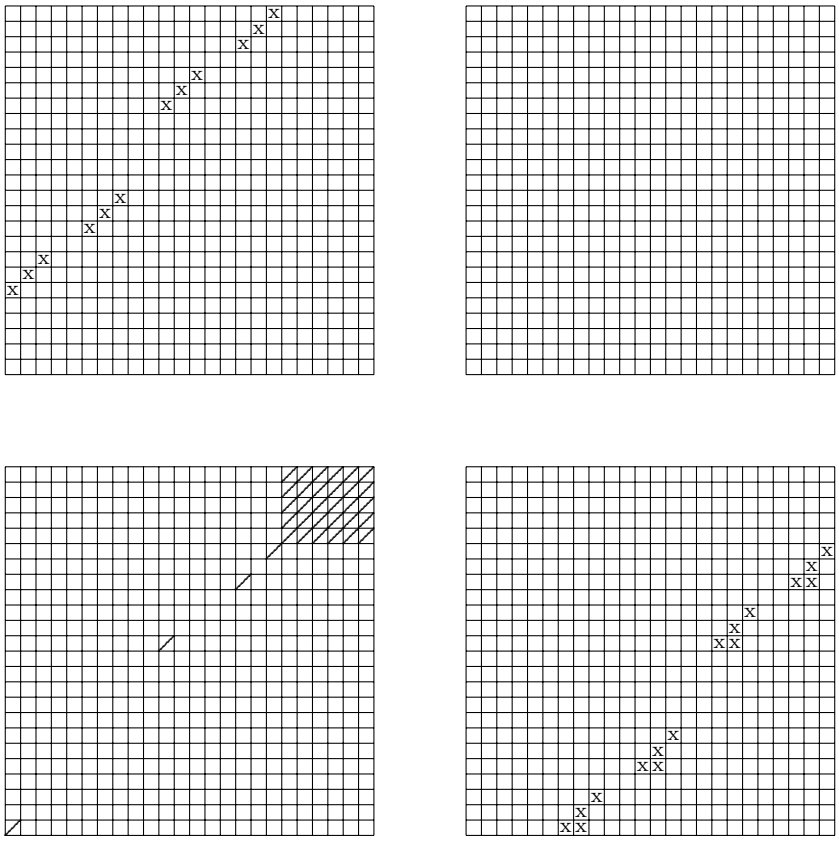


FIG. 3. *The four corner* 24×24 *subarrays of a* 192×192 *array* A *are shown. All the other elements of* A *are blank. The points marked with slashes (except the origin) are valid periods of* A*, where the shortest and the second shortest periods are* $(10, 12)$ *and* $(15, 16)$.

LEMMA 2. *Let* $h = \min(\ell_x(p_1), \ell_y(p_1))$. *The size of* $V = \{v_1, \ldots, v_t\}$ *is* $O(h)$*, and all valid periods of* A *are uniquely represented by the form* $c_i + kv_i$, $0 \le k < d_i$*, for some vectors* c_i *and some integers* d_i, $1 \le i \le t$.

Proof. For each $1 \le i < t$, let v_i and v_{i+1} be the renamings of u_j and u_l $(j < l)$, respectively. Since $u_j = p_j - p_{j-1}$ and $u_l = p_l - p_{l-1}$, periods $p_j, p_{j+1}, \ldots, p_{l-1}$ are p_j, $p_j + v_i$, $p_j + 2v_i, \ldots$, by definition of the v_i's. If we let $c_i = p_j$ and $d_i = \frac{|p_l - p_j|}{|v_i|}$, then periods from p_j to p_{l-1} are represented by the form $c_i + kv_i$, $0 \le k < d_i$.

We now show that $|v_{i+1}| < |v_i|$. Since p_{j-1} and p_j are periods of A, v_i $(= p_j - p_{j-1})$ is a period of $A_{p_{j-1}}$ (and thus a period of $A_{p_{l-1}}$) by Lemma 1. Since p_{l-1} and v_i are periods of A and $A_{p_{l-1}}$, respectively, $p_{l-1} + v_i$ is a period of A by Lemma 1. Since p_l is the period of A just after p_{l-1} in length, we have $|p_l| \le |p_{l-1} + v_i|$. Since $v_{i+1} = p_l - p_{l-1}$, $|v_{i+1}| \le |v_i|$. By definition of the v_i's, $v_{i+1} \ne v_i$, and thus $|v_{i+1}| < |v_i|$. Therefore, the size of V is $O(h)$.

Finally, let v_t be the renaming of u_ℓ $(= p_\ell - p_{\ell-1})$. If we let $c_t = p_\ell$ and $d_t = \frac{|p_s - p_\ell|}{|v_t|} + 1$, then periods from p_ℓ to p_s are represented by the form $c_t + kv_t$, $0 \le k < d_t$. □

EXAMPLE 3. *For the periods in Example* 2, $c_1 = (11, 22)$, $c_2 = (27, 54)$, $c_3 = (30, 60)$, $c_4 = (31, 62)$ *and* $d_1 = 2$, $d_2 = 1$, $d_3 = 1$, $d_4 = 3$. *Hence,* $p_7 = (33, 66)$ *is*

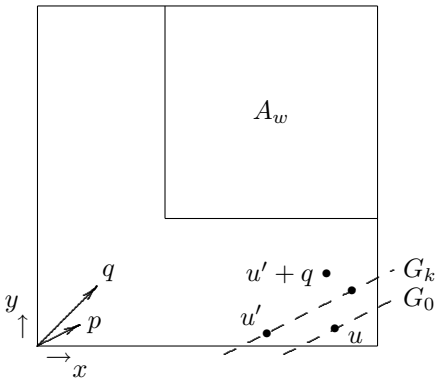


FIG. 4. *Proof of Lemma* 3.

represented by $c_4 + 2v_4$.

3.2. Radiant-periodicity. Suppose that A is radiant-periodic. It was shown in [GP] that there exists a vector w such that A_w and A_{-w} are lattice-periodic with basis vectors b_1, b_2, and that a valid vector v in A_w is a period of A if and only if v is a lattice point on b_1, b_2 (Theorem 3 in [GP]). The vectors w, b_1, b_2 are called the *characteristic vectors* of A. Furthermore, if there are more periods outside A_w, these periods are lattice points on b_1, b_2 and form a monotone line [GP].

EXAMPLE 4. *In Figure* 3 *the characteristic vectors are* $w = (18, 19)$, $b_1 = (1, 0)$, *and* $b_2 = (0, 1)$. *The valid periods outside* A_w *are* $(10, 12)$, $(15, 16)$, *and* $(17, 18)$.

Here we will show that the periods outside A_w are divided into two parts, an initial line of periods parallel to the shortest period of A and the rest, and they are represented by a small number of vectors. Assume without loss of generality that the periods of A are of direction $(+, +)$.

LEMMA 3. *Let* p *and* q *be the shortest and the second shortest periods of* A *outside* A_w, *respectively. If* q *is independent of* p, *then* $3p$ *is in* A_w.

Proof. We first define *defects*. If $u \in A$ and $v \in A_w$ are lattice-congruent modulo b_1, b_2 and $A[u] \neq A[v]$, then u is called a *defect*. Since A_w and A_{-w} are lattice-periodic with b_1, b_2, they contain no defects. Assume that q is independent of p. (See Figure 4.)

Suppose that $3p$ is not in A_w. Since $3p$ is not in A_w, there exists at least one defect u such that $u + 3p$ or $u - 3p$ is in A. Assume without loss of generality that u is in the lower-right quadrant of A as in Figure 4.

We will show that defects would be spread all over A starting from u by periods p and q. Assume that q is counterclockwise with respect to p (i.e., q becomes parallel to p when q is rotated clockwise by less than 90°). (The other case is similar.) Let G_i, $i \geq 0$, be the line passing through $u + iq$ and parallel to p. Let k be the largest number such that G_k contains a defect, and let u' be the first defect (from the left) on G_k. Since $u' - p$ is not in A, $u' + 2p$ is in A because $G_k \cap A$ is long enough to contain two points whose distance is $2p$. Since $u' + 2p$ is in A and $|q| \leq |2p|$ (because q is second shortest), $u' + q$ is in A. Hence, $u' + q$ is a defect by period q. Since $u' + q \in G_{k+1}$, we have a contradiction to the maximality of k. □

LEMMA 4. *Let* p *and* q, $p \prec_{++} q$, *be periods of* A *outside* A_w. *Then* $\ell_x(p) < \ell_x(q)$ *and* $\ell_y(p) < \ell_y(q)$.

Proof. Suppose that $\ell_x(p) = \ell_x(q)$ and $\ell_y(p) < \ell_y(q)$ (or $\ell_x(p) < \ell_x(q)$ and

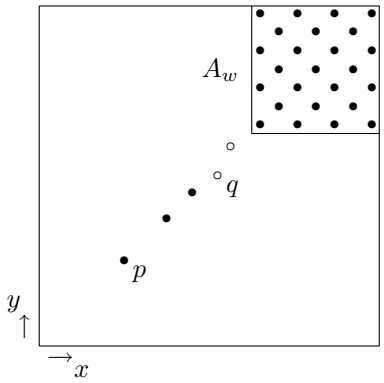


FIG. 5. *Class* 1 *is black dots inside* A_w. *Classes* 2 *and* 3 *are black and white dots, respectively, outside of* A_w.

$\ell_y(p) = \ell_y(q)$). As in the proof of Lemma 3, defects would be spread all over A by periods p and q. □

Lemma 4 means that the lengths of the periods of A outside A_w are simply increasing, and Lemma 3 means that there are not many periods of A outside A_w that are independent of the shortest period.

DEFINITION 4. *When a two-dimensional array* A *is radiant-periodic with characteristic vectors* w, b_1, b_2, *we divide the valid periods of* A *into the following three classes, where each class may be empty (see Figure* 5*):*

1. *Class* 1 *is the periods in* A_w. *They are lattice points on* b_1, b_2, *by Theorem* 3 *in* [GP].

2. *If there are periods outside* A_w, *let* p *be the shortest period of* A, *and* q *the shortest period independent of* p. *Let* $h = \min(\ell_x(p), \ell_y(p))$. *Class* 2 *is the periods* v *such that* v *is parallel to* p *and* $v \prec_{++} q$. *(Class* 2 *will be called the initial line of periods.) These periods are represented by* $O(h)$ *step vectors* v_i*'s and corresponding* c_i*'s and* d_i*'s, by Lemma* 2.

3. *Class* 3 *is the rest of the periods outside* A_w. *(These periods will be called broken periods.)*

LEMMA 5. *In class* 3 *there are* $O(h)$ *broken periods.*

Proof. Let p_i and p_{i+1} be the two consecutive periods just before q (i.e., p_i, p_{i+1}, and q are three consecutive periods of A outside A_w). Note that if q is the second shortest period of A, then p_i is 0 and p_{i+1} is the shortest period p. By applying Lemmas 3 and 4 to A_{p_i}, the number of broken periods is $O(\min(\ell_x(p_{i+1} - p_i), \ell_y(p_{i+1} - p_i)))$, which is $O(h)$. □

EXAMPLE 5. *In Figure* 3, *period* $(10, 12)$ *belongs to class* 2, *and the broken periods of class* 3 *are* $(15, 16)$ *and* $(17, 18)$. *Class* 1 *consists of* 30 *periods in* A_w.

Since the line-periodic case is a special case of the radiant-periodic case, we will consider only the radiant-periodic case in what follows.

4. Three-dimensional periodicity. We classify three-dimensional arrays A of size $m_1 \times m_2 \times m_3$ into the following four cases by the existence of valid periods:

1. edge-periodic: A has at least one valid edge period. (In the following cases, A has no valid edge periods.)

2. side-aperiodic: A has no valid side periods in at least one type. If A is edge-periodic, then A has a simple repetitive structure, i.e., the whole array A is produced

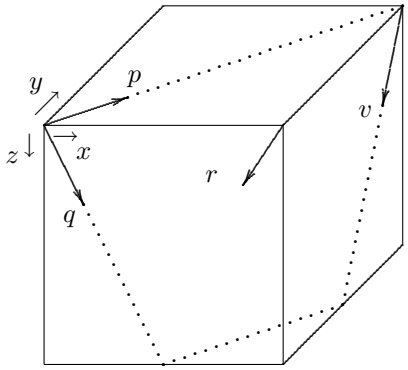


FIG. 6. *The missing general direction r: The periods produced by p and q are on the plane marked with dotted lines.*

by many copies of a rectilinear subarray. The side-aperiodic case is also simple for pattern matching purposes as we will see later. The interesting case is when A has valid side periods in all three types. Even with these side periods in all three types, there may exist one general direction (and its adjacent side directions) where A has no periods. These directions will be called the *missing directions* of the side periods. We have two more cases by the existence of periods in the missing directions.

3. side-periodic: A has valid side periods in all three types. However, A has no valid periods in the missing directions of the side periods.

4. completely periodic: In addition to valid side periods in all three types, A has at least one valid period in the missing directions of the side periods.

EXAMPLE 6. *If a three-dimensional array A has valid side periods p, q, and v in Figure* 6*, but not r, then it is side-periodic. If A has period r in addition to p, q, and v, then it is completely periodic.*

Now we will characterize the periods of three-dimensional array A in the side-periodic and completely periodic cases. To introduce missing directions and their implications, we will describe the completely periodic case first.

4.1. Complete periodicity. We will first define missing directions, and then show that two side periods and a period in the missing directions determine complete periodicity.

EXAMPLE 7. *Let p and q be side vectors in two different types. Consider what kinds of vectors can be produced by a linear combination of p and q. For example, if* $p = (1, 1, 0)$ *and* $q = (1, 0, 1)$*, then* $p - q = (0, 1, -1)$ *is a side vector of the third type. Notice that p and q produce general vectors in three directions:* $(+, +, +)$*,* $(+, +, -)$*, and* $(+, -, +)$*, i.e.,* $p + q = (2, 1, 1)$*,* $2p - q = (1, 2, -1)$*, and* $2q - p = (1, -1, 2)$*. However, p and q cannot produce vectors in direction* $(-, +, +)$*. The complete periodicity of three dimensions has to do with this direction.*

LEMMA 6. *Let p and q be side vectors in two different types. Then p and q produce general vectors in three directions, but no general vectors in the remaining one direction, which is denoted by* α*. Furthermore, they produce no side vectors in the three directions adjacent to* α*, but produce side vectors in the remaining three directions.*

Proof. Let $p = (i, j, 0)$ and $q = (k, 0, l)$, where $i, j, k, l > 0$. (The other cases

are similar.) Then $p+q$ is a vector in direction $(+,+,+)$. For $a,b>0$ such that $ai-bk\neq 0$, $ap-bq=(ai-bk,aj,-bl)$ is a vector in $(+,+,-)$ or $(-,+,-)$. Hence p and q produce general vectors in three directions $(+,+,+)$, $(+,+,-)$, and $(+,-,+)$.

The vector p itself is a side vector in $(+,+,0)$, and q in $(+,0,+)$. Also for a,b such that $ai-bk=0$, $ap-bq$ is a side vector in $(0,+,-)$. That is, p and q produce side vectors in three directions $(+,+,0)$, $(+,0,+)$, and $(0,+,-)$.

Finally, $ap+bq=(ai+bk,aj,bl)$ cannot be a vector in the general direction $(-,+,+)$ or its adjacent side directions for any a,b because

1. if $aj>0$ and $bl>0$, then $ai+bk>0$;
2. if $ai+bk<0$ and $aj>0$, then $bl<0$;
3. if $ai+bk<0$ and $bl>0$, then $aj<0$. □

DEFINITION 5. *In Lemma* 6 *the directions (one general direction and three side directions adjacent to it) in which* p *and* q *cannot produce vectors will be called the* missing directions *of* p,q. *See Figure* 6. *(Let* v *be a side period in the third type as in Figure* 6. *Notice that the missing directions remain the same even if any two of* p, q, *and* v *are chosen in Lemma* 6.*)*

We now show that two side periods and a period in the missing directions (e.g., p,q,r in Figure 6) determine complete periodicity.

LEMMA 7. *Let* p *be a valid side vector of* A, *and* r *be a valid vector that has the same sign as* p *in one and a different sign (or* 0*) in the other of the two coordinates where* p *has nonzero values (e.g.,* p *in* $(+,+,0)$ *and* r *in* $(-,+,+)$*). If* w *and* $w'=w+ap+br+u$ *are in* A, *where* $a,b\geq 1$ *are integers and* u *is a vector that has the same signs as* r *(or* 0*) in the coordinate where* p *has* 0 *and in the coordinate where* p *has the same sign as* r *(e.g.,* u *in* $(+,+,+)$*), then at least one of* $w+p$ *and* $w+r$ *is in* A.

Proof. Assume that p is of direction $(+,+,0)$, r of $(-,+,+)$, and u of $(+,+,+)$. (The other cases are similar.) The y-coordinate (z-coordinate) of $w+p$ is between the y-coordinates (z-coordinates) of w and w' because the y-coordinates (z-coordinates) of p, r, and u are ≥ 0. The same holds for $w+r$. If the x-coordinate of w is $<\frac{m_1}{2}$, then $w+p$ is in A because p is a valid vector; if it is $\geq\frac{m_1}{2}$, then $w+r$ is in A. □

LEMMA 8. *Let* p *and* q *be valid side vectors of* A *in two different types, and let* r *be a valid vector in the missing directions of* p,q. *If* w *and* w' *are two points in* A *that are congruent modulo* p,q,r, *then there exists a sequence of points* $w=w_0,w_1,\ldots,w_k=w'$ *such that every* w_i, $0\leq i\leq k$, *is in* A, *and every* w_i-w_{i-1}, $1\leq i\leq k$, *is one of* p, q, *and* r.

Proof. Assume without loss of generality that p, q, and r are of direction $(+,+,0)$, $(+,0,+)$, and $(-,+,+)$, respectively. Let $w'=w+ap+bq+cr$. We will prove the lemma for the case $a,b,c\geq 0$. (The other cases are similar.) First we show that at least one of $w+p$, $w+q$, and $w+r$ (which will be w_1) is in A by the following cases:

1. All of a, b, and c are positive: One of $w+p$ and $w+r$ is in A by Lemma 7.
2. One of a, b, and c is 0: If $b=0$ (resp., $a=0$, $c=0$), one of $w+p$ and $w+r$ (resp., one of $w+q$ and $w+r$, one of $w+p$ and $w+q$) is in A by Lemma 7.
3. Two of a, b, and c are 0: If $a>0$ (resp., $b>0$, $c>0$), $w+p$ (resp., $w+q$, $w+r$) is in A.

By repeating the same procedure, one can show that the lemma holds. □

THEOREM 1. *Suppose that* A *has valid side periods in all three types, two of which are* p *and* q *in two different types. If additionally* A *has a valid period* r *in the missing directions of* p,q, *then every lattice point on* p,q,r *is a period of* A.

Proof. Consider a vector $ap+bq+cr$. Let w and w' be two points in A such that $w'=w+ap+bq+cr$. By Lemma 8 there exists a sequence of points $w=$

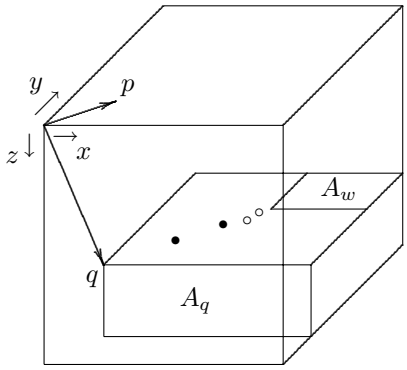


FIG. 7. *The side-periodic case: Black and white dots represent periods in the initial line and broken periods, respectively.*

$w_0, w_1, \ldots, w_k = w'$ such that every w_i is in A and every $w_i - w_{i-1}$ is one of p, q, and r. Since p, q, and r are periods of A, we have $A[w] = A[w_1] = \cdots = A[w']$. Hence $ap + bq + cr$ is a period of A. □

4.2. Side-periodicity. Array A has valid side periods in all three types, but it has no valid periods in the missing directions. That is, A has no side periods in the three directions adjacent to the missing general direction. These three side directions consist of one direction in each type. Hence, for each type of side direction, there are periods in one direction but no periods in the other direction.

EXAMPLE 8. *If the array A in Figure 6 has valid side periods p, q, and v, but not r, then it is side-periodic. Since period p is in direction $(+,+,0)$ of type $(x,y,0)$, there are no periods in $(+,-,0)$ because $(+,-,0)$ is a missing direction of p, q.*

Array A may have another period u that is not a linear combination of p and q, but such a period u must not be in one of the missing directions by definition of this case. In each type of side direction, therefore, A may be line-periodic or radiant-periodic, but not lattice-periodic.

We now show that the periods of A can be represented by a small number of vectors. Suppose that A is side-periodic with side periods in $(+,+,0)$, $(+,0,+)$, and $(0,+,-)$, as in Figure 6. Consider the periods of A with z-coordinate j for some fixed $j > 0$. These periods can be of direction $(+,+,+)$, $(-,-,+)$, and $(+,-,+)$ since the fourth direction $(-,+,+)$ is the missing general direction. We will characterize the periods of direction $(+,+,+)$. The periods of direction $(-,-,+)$ are similar. Direction $(+,-,+)$ will be dealt with separately.

Let U be the set of all valid periods of A of direction $(+,+,+)$ with z-coordinate j. Let p be a shortest period of A in direction $(+,+,0)$, and let $h = \min(\ell_x(p), \ell_y(p))$; see Figure 7. A period q in U is called an *anchor period* if $q \prec_{++0} v$ for all periods $v \in U$. There are two cases depending on whether there exists an anchor period in U or not.

Consider first the case that there exists an anchor period in U. If there exists an anchor period q in U, we can work with A_q (rather than A) by Lemma 1. Let U' be the set of vectors v such that $v \neq (0,0,0)$ and $q + v \in U$. Assume without loss of generality that A_q is radiant-periodic in the xy-plane with characteristic vectors w, b_1, b_2.

LEMMA 9. *If a shortest period of A_q in direction $(+,+,0)$ is not parallel to p, there are $O(h)$ periods in U' outside A_w.*

Proof. Let r be a shortest period of A_q in direction $(+,+,0)$. If r is independent of p, then $|r| \leq |p|$ because p is also a period of A_q. If $q+p$ is inside A_w, there are $O(h)$ periods in U' outside A_w because these periods form a monotone line. If $q+p$ is outside A_w, let k be the largest integer such that $kr \prec_{++0} p$. By applying Lemmas 3 and 4 to A_{kr}, the number of periods in $A_{kr} - A_w$ is $O(h)$. The number of periods outside A_{kr} is also $O(h)$ because of $kr \prec_{++0} p$. Therefore, there are $O(h)$ periods in U' outside A_w. □

DEFINITION 6. *If there exists an anchor period q in U, assume that A_q is radiant-periodic in the xy-plane with characteristic vectors w, b_1, b_2. We divide all periods of U' into the following three classes, where each class may be empty (see Figure 7):*

1. *Class 1 is the periods in A_w. They are lattice points on b_1, b_2.*

2. *If there are periods outside A_w, let r be the shortest one. If r is parallel to p, class 2 is the initial line of periods that is parallel to p. Since $|r| \leq |p|$, these periods are represented by $O(h)$ step vectors v_i and corresponding c_i's and d_i's. (Here we assume that v_i's and c_i's are vectors of A_q, not vectors of A.)*

3. *Class 3 is the rest of the periods outside A_w (called broken periods). By Lemmas 5 and 9, there are $O(h)$ broken periods.*

Note that Definition 6 is essentially the same as Definition 4 except that we require in Definition 6 that the initial line of periods in class 2 be parallel to p.

EXAMPLE 9. *Consider the array A in Figure 7. Suppose that p is $(10,12,0)$ and that the xy-plane of A_q is the two-dimensional array in Figure 3. Then, class 2 has one period $(10,12,0)$, and class 3 has $(15,16,0)$ and $(17,18,0)$. The 30 periods in A_w of Figure 3 (with z-coordinates 0) belong to class 1.*

LEMMA 10. *If there is no anchor period in U, then there exists a vector w with z-coordinate j such that A_w is lattice-periodic in the xy-plane and A_w contains all periods in U.*

Proof. Let $q \in U$ be a shortest period in the x- and y-coordinates (i.e., $\max(\ell_x(q), \ell_y(q))$ is smallest). Let $r \in U$ be a shortest period in the x- and y-coordinates such that $q \prec_{-+0} r$. (Since there is no anchor period in U, there exists such a period r.)

We first show that $q-r$ is a period of A_q of direction $(+,-,0)$. Let u be a point in $(A_q)_{-(q-r)}$ (i.e., $u+q-r$ is in A_q). Then $u-r$ is in A because $u+q-r$ is in A_q. Since $A[u] = A[u-r] = A[u+q-r]$ by periods q and r, $q-r$ is a period of A_q. Similarly, $q-r$ is a period of A_r. Since p is a period of A_q and A_r of direction $(+,+,0)$, both A_q and A_r have quad-I and quad-II periods in the xy-plane, and thus they are lattice-periodic in the xy-plane. Hence there exists a vector w with z-coordinate j such that A_w is lattice-periodic in the xy-plane and A_w includes both A_q and A_r (Lemma 7 in [GP]). (This vector w is one of the characteristic vectors of a radiant-periodic array.) Since q and r are shortest in the x- and y-coordinates among the periods in U, A_w contains all periods in U. □

In the case that there is no anchor period in U, all periods in U can be regarded as class 1, and classes 2 and 3 are empty by Lemma 10. Hence the case that there exists an anchor period is more general.

To deal with directions $(+,-,+)$ and $(-,+,+)$ in the pattern matching algorithm, we need a couple of lemmas.

LEMMA 11. *Let p and q be valid periods of direction $(+,+,0)$ and $(+,-,+)$, respectively. Then $q+p$ and $q-p$ are periods of A.*

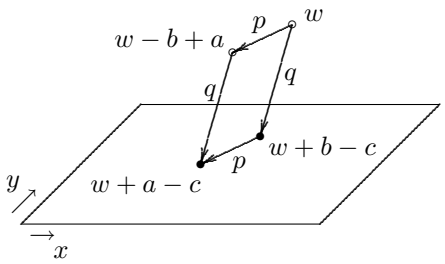


FIG. 8. *Proof of Lemma* 12.

Proof. We prove the lemma for $p+q$. Let w be a point in $A_{-(p+q)}$. Since $w+p+q$ is in A, at least one of $w+p$ and $w+q$ (say, $w+p$) is in A by Lemma 7. Since $A[w]=A[w+p]=A[w+p+q]$ by periods p and q, $p+q$ is a period of A. □

LEMMA 12. *Let a and b be two points such that $b-a$ is a valid side period of A of direction $(+,+,0)$, and let c be a point such that both $a-c$ and $b-c$ are valid vectors of direction $(+,-,+)$ (or $(-,+,+)$). If there exists a witness of A against $a-c$, then there also exists a witness of A against $b-c$, and vice versa.*

Proof. We prove the lemma for the case that $a-c$, $b-c$ are of direction $(+,-,+)$ and there exists a witness $(w, w+a-c)$ of A against $a-c$. (The other cases are similar.) See Figure 8. Let $p=-(b-a)$ and $q=b-c$. Consider the parallelogram whose four corners are w, $w+b-c$ $(=w+q)$, $w-b+a$ $(=w+p)$, and $w+a-c$ $(=w+p+q)$. Since $(w, w+p+q)$ is a witness, both w and $w+p+q$ are in A and $A[w]\neq A[w+p+q]$. Since p is of direction $(-,-,0)$ and q is of $(+,-,+)$, at least one of $w+p$ and $w+q$ is in A by Lemma 7. If $w+q$ is in A, $(w, w+q)$ is a witness of A against q because $A[w+q]=A[w+p+q]\neq A[w]$. If $w+p$ is in A, $(w+p, w+p+q)$ is a witness against q because $A[w+p]=A[w]\neq A[w+p+q]$. □

5. Three-dimensional pattern matching. Consider three-dimensional pattern matching for a pattern P of size $m_1\times m_2\times m_3$ and a text T of size $n_1\times n_2\times n_3$. For a point u in the text, the $m_1\times m_2\times m_3$ text block whose origin is placed on u is called the *area* of u. If the area of u is a possible occurrence of the pattern, u is called a *candidate*. Candidates are *consistent* if, for every pair u and v of the candidates, the vector $u-v$ is a period of the pattern.

We will use various techniques in our three-dimensional pattern matching algorithm. First we introduce a new technique based on the following two notions.

DEFINITION 7. *When the pattern P has a set R of periods, a generator of the pattern with respect to R is a subpattern $P'\subseteq P$ such that every point w in P has a sequence of points $w=w_0, w_1, \ldots, w_k$ such that every w_i is in P, every w_i-w_{i-1} is a period in R, and $w_k\in P'$.*

DEFINITION 8. *When the pattern has a period v, a pair of text points $(w, w+v)$ such that $T[w]\neq T[w+v]$ is called a periodic mismatch.*

When the pattern P has a generator P' with respect to a set R of periods, the *technique of generators and periodic mismatches* finds pattern occurrences in two stages.

1. For each period $v\in R$, find periodic mismatches for every point w of the text. A periodic mismatch $(w, w+v)$ implies that w and $w+v$ cannot be inside an occurrence of the pattern; i.e., it eliminates all candidates whose areas contain w and $w+v$.

2. If a candidate u survives the periodic mismatches for all periods in R, the area of u is periodic with every period in R. In addition, if the area of u contains an occurrence of generator P', then it is an occurrence of the pattern by definition of generators.

Other techniques that we use are as follows.

• Duels [Vi]: Consider two candidates u and v such that u and v are inconsistent. If we have a witness against $u - v$, at least one of u and v can be eliminated (or killed) by a constant number of symbol comparisons, which is called a *duel*. A table that contains witnesses against nonperiods among all valid vectors is called a *witness table*.

• Duels on a line of candidates [ABF]: Suppose we have m' candidates which are on one line. By applying duels from one end of the line to the other end, we can obtain consistent candidates from the given m' candidates in $O(m')$ time.

• The wave method [ABF]: Given a set of consistent candidates, the wave method finds the occurrences of the pattern in linear time.

As in most pattern matching algorithms, our three-dimensional matching algorithm consists of two parts:

1. preprocessing: Find periods of the pattern P and witnesses against nonperiods among all valid vectors.

2. text search: Initially every point of the text is a candidate. We divide all candidates into disjoint blocks of size $\frac{m_1}{8} \times \frac{m_2}{8} \times \frac{m_3}{8}$ (each block is called a *candidate block*). We process each candidate block as follows.

The text search is based on the three-dimensional periodicity of the pattern P. We will describe below our text search algorithm in each of the four cases of three-dimensional periodicity.

5.1. Edge-periodicity. The pattern has at least one valid edge period. Since the pattern is generated by a rectilinear subpattern, the text search problem in this case is reduced to one with a smaller pattern.

Let P' be the rectilinear subpattern such that its length in a coordinate with valid edge periods is $2|p|$, where p is the shortest edge period in that coordinate, and the length in a coordinate with no valid edge periods is the same as that of P. Let R be the set of the shortest valid edge period in each coordinate. Then P' is a generator of P with respect to R.

EXAMPLE 10. *If the pattern P has the shortest valid edge periods $p = (a, 0, 0)$ and $q = (0, b, 0)$ for $a, b > 0$ in the x- and y-coordinates, respectively, and no valid edge periods in the z-coordinate, then $P' = P[0..2a-1, 0..2b-1, 0..m_3-1]$ and $R = \{p, q\}$.*

Since P' has no edge periods shorter than those of P by the periodicity lemma of one-dimensional strings [KMP], P' has no valid edge periods. We find occurrences of P' in text T using the other cases, since P' is not edge-periodic. Then we can find occurrences of the whole pattern P by the technique of generators and periodic mismatches. Since the size of R is $O(1)$, checking periodic mismatches takes $O(1)$ time for each point in the area of a candidate, and thus it takes linear $O(m_1m_2m_3)$ time for a candidate block.

5.2. Side-aperiodicity. The pattern has a plane (say, the xy-plane) where there are no valid side periods. By using duels, we will make the number of candidates small enough to apply the wave method. There are two cases by the shape of the pattern.

Assume first that $m_1m_2 \geq m_3$. Assume without loss of generality that $m_1 \geq m_2$. Divide a candidate block into lines parallel to the x-axis, and apply duels on the candidates in each line. Since the pattern P has no valid edge periods, we obtain at

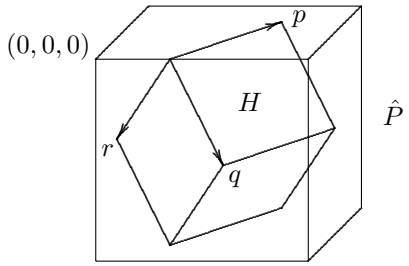


FIG. 9. *Polyhedron H and rectilinear array $\hat{P}$ that encompasses H.*

most one surviving candidate per line.

We now divide the candidate block into planes parallel to the xy-plane. Since there is at most one candidate in each line parallel to the x-axis, there are $O(m_2)$ candidates in each plane. Perform $O(m_2^2)$ duels for all pairs of $O(m_2)$ candidates in each plane. Since there are no side periods of type $(x, y, 0)$, there will be at most one survivor in each plane, i.e., $O(m_3)$ survivors in total. Check consistency of the survivors by doing $O(m_3^2)$ duels, since $m_3^2 \leq m_1 m_2 m_3$. We end up with a set of consistent candidates.

Assume next that $m_1 m_2 < m_3$. In this case we divide the candidate block into lines parallel to the z-axis. By applying duels in each line, we obtain at most one survivor per line, i.e., $O(m_1 m_2)$ survivors in total. In consistency checking, $O((m_1 m_2)^2)$ duels can be done in linear time since $(m_1 m_2)^2 \leq m_1 m_2 m_3$.

Given a set of consistent candidates, we can find the occurrences of the pattern by the wave method in linear $O(m_1 m_2 m_3)$ time.

5.3. Complete periodicity. The pattern has valid side periods in all three types and at least one valid period in the missing directions. We will find a rectilinear generator of the pattern that is neither edge-periodic nor completely periodic. Then the text search problem is reduced to one with a smaller pattern.

Let p be a shortest side period of the pattern, and q a shortest side period among those whose type is different from p. Let r be a shortest period in the missing directions of p, q. Assume without loss of generality that $p = (a_x, a_y, 0)$, $q = (b_x, 0, b_z)$, and $r = (-c_x, c_y, c_z)$, where $a_x, a_y, b_x, b_z, c_x, c_y, c_z > 0$.

The rectilinear generator of the pattern is $P' = P[0..2(a_x + b_x + c_x) - 1, 0..2(a_y + c_y) - 1, 0..2(b_z + c_z) - 1]$. Notice that $\hat{P} = P[0..a_x + b_x + c_x - 1, 0..a_y + c_y - 1, 0..b_z + c_z - 1]$ is a rectilinear array that encompasses a polyhedron (denoted by H) whose edges are vectors p, q, r; see Figure 9. We now show that P' is a generator of P, and P' is neither edge-periodic nor completely periodic.

LEMMA 13. *P' is a generator of pattern P with respect to $R = \{p, q, r\}$.*

Proof. For every point $w \in P$, there is a point $w' \in H \subseteq P'$ which is congruent to w modulo p, q, r. By Lemma 8 there exists a sequence of points $w = w_0, w_1, \ldots, w_k = w'$ such that every w_i is in P and every $w_i - w_{i-1}$ is one of p, q, r. Hence P' is a generator of P with respect to p, q, r. □

LEMMA 14. *P' has no valid periods that are not periods of P.*

Proof. Suppose that we divide the three-dimensional space into polyhedra that have the same shape as H in Figure 9 by planes parallel to p and q, planes parallel to

q and r, and planes parallel to p and r such that one of the polyhedra is H. We say that two polyhedra A and B are *adjacent* if A and B overlap completely when one of them is shifted by a vector $ap + bq + cr$, where each of a, b, and c is 0, 1, or -1.

We show that if P' has a valid period u, then u is also a period of P. Consider any pair of points $w, w + u \in P$. Let H' be the polyhedron (by the division above) which contains w. Since u is a valid vector of P', $w + u$ is either in H' or in a polyhedron H'' adjacent to H'. Since P' is large enough to contain two adjacent polyhedra, there exist $w', w' + u \in P' \subseteq P$ which are congruent to $w, w + u$ modulo p, q, r, respectively. Hence we have $P[w] = P'[w'] = P'[w' + u] = P[w + u]$, and therefore u is a period of P. □

COROLLARY 1. *P' is not edge-periodic.*

LEMMA 15. *P' is not completely periodic.*

Proof. By the way p and q were chosen, we have $|p| \leq |q|$. There are two cases. If $|r| \geq |q|$, then r is not a valid vector of P' because a largest coordinate of r is larger than one eighth of that coordinate of P'. Since r is shortest in the missing directions of p, q, P' has no valid periods in the missing directions of p, q, by Lemma 14. Hence P' is not completely periodic.

If $|r| < |q|$, then q is not a valid vector of P'. By the way q was chosen, any side period v of P in types except $(x, y, 0)$ (the type of p) satisfies $|v| \geq |q|$, i.e., v is not a valid vector of P'. Since P' has no valid side periods in two types, P' is not completely periodic. □

When the pattern is completely periodic, we find all occurrences of P'. Since P' is neither edge-periodic nor completely periodic, it is finished by the side-aperiodic or side-periodic case. Since P' is a generator of the pattern, we can find the occurrences of the pattern by the technique of generators and periodic mismatches. Again the technique takes linear time because the size of R is 3.

5.4. Side-periodicity. The pattern has valid side periods in all three types, but it has no valid periods in the missing directions. In each type of side directions the pattern is line-periodic or radiant-periodic, but not lattice-periodic. We will divide a candidate block into planes and perform duels in each plane to get a set of consistent candidates. Between different planes we will perform duels using a small number of processors by the characterization of periods in section 4.2.

Assume without loss of generality that P has side periods in $(+, +, 0)$, $(+, 0, +)$, $(0, +, -)$ (i.e., $(-, +, +)$ is the missing general direction) and that $m_1 \geq m_2 \geq m_3$. Since the radiant-periodic case is more general than the line-periodic case, we consider the radiant-periodic case only in the xy-plane. Let p be a shortest period of P in direction $(+, +, 0)$, and let $h = \min(\ell_x(p), \ell_y(p))$. Let $\hat{p}$ be the shortest integer-valued vector such that $p = g\hat{p}$ for integer g. Note that $1 \leq g \leq h$.

EXAMPLE 11. *Suppose that $p = (11, 22, 0)$ is a shortest period of the pattern P. Then $h = 11$, and $\hat{p} = (1, 2, 0)$ since $p = 11 \cdot \hat{p}$. Note that $\hat{p}$ is the shortest (possible) step vector for the periods parallel to p, as shown in Example 2.*

Divide a candidate block into lines parallel to the x-axis, and get at most one survivor per line by applying duels in each line. Divide the candidate block into planes parallel to the xy-plane, and perform duels in each plane to get consistent survivors per plane, which will be one *group* of survivors. The consistent survivors in a group form a monotone line, and any two neighboring survivors are at least p apart from each other. Hence we have $O(m_3)$ groups, each of which contains $O(\frac{m_2}{h})$ consistent survivors. A difficulty in the side-periodic case is that performing all possible duels of $O(\frac{m_2 m_3}{h})$ candidates requires more than linear time. We will reduce the number of

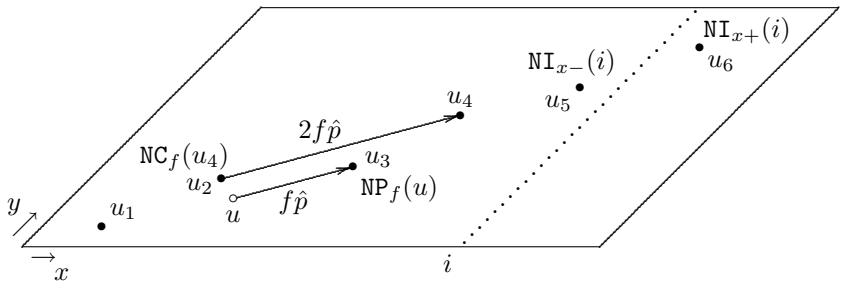


FIG. 10. *Nearest candidates.*

duels by the characterization of periods in section 4.2.

In the following, we will perform duels between different groups and eliminate some candidates by duels so that all the remaining candidates are consistent. Consider a pair of candidates a, b with $\ell_z(a) < \ell_z(b)$. Then the relation between a and b is one of $a \prec_{+++} b$, $a \prec_{+-+} b$, $a \prec_{--+} b$, and $a \prec_{-++} b$, among which $(-,+,+)$ is the missing direction. The relations $\prec_{+++}$ and $\prec_{--+}$ are harder to deal with than $\prec_{+-+}$ and $\prec_{-++}$.

We first consider relations $\prec_{+++}$ and $\prec_{--+}$. We will eliminate candidates so that every pair of candidates a, b with $a \prec_{+++} b$ are consistent. The relation $\prec_{--+}$ can be handled similarly. Recall that there are $O(m_3)$ groups, each of which contains $O(\frac{m_2}{h})$ consistent candidates. First we need some precomputations on each group of candidates in which we will find nearest candidates in various forms. Let $u_1, u_2, \ldots$ be the candidates in a group C.

Definitions of nearest candidates:

1. Nearest candidate from a candidate: For every candidate $u_i \in C$ and every $1 \le f \le g$, let $\mathtt{NC}_f(u_i)$ denote the candidate $u' \in C$ nearest to u_i such that $u' \prec_{++0} u_i$ and $u_i - u'$ is a multiple of $f\hat{p}$.

2. Nearest candidate from a point: Consider the plane D (parallel to the xy-plane) of a candidate block that contains u_i's. A line on D parallel to p which contains at least one candidate in C will be called a *check line*. Since there are $O(\frac{m_2}{h})$ candidates in C, there are $O(\frac{m_2}{h})$ check lines on D. For every point u in check lines and every $1 \le f \le g$, let $\mathtt{NP}_f(u)$ denote the candidate $u' \in C$ nearest to u such that $u \prec_{++0} u'$ and $u' - u$ is a multiple of $f\hat{p}$. That is, $\mathtt{NP}_f(u)$ is the nearest candidate to u whose distance to u is a multiple of (possible) step vector $f\hat{p}$.

3. Nearest candidate from a coordinate: For each x-coordinate i, let $\mathtt{NI}_{x+}(i)$ denote the candidate in C whose x-coordinate is the smallest integer $\ge i$, and $\mathtt{NI}_{x-}(i)$ the candidate in C whose x-coordinate is the largest integer $\le i$. Since the candidates in group C are monotone and no two candidates have the same x-coordinates (because P is not edge-periodic), $\mathtt{NI}_{x+}$ and $\mathtt{NI}_{x-}$ are well defined. For each y-coordinate j, $\mathtt{NI}_{y+}(j)$ and $\mathtt{NI}_{y-}(j)$ are similarly defined.

EXAMPLE 12. *Let* $C = \{u_1, \ldots, u_6\}$ *as in Figure* 10. *Then* $\mathtt{NC}_f(u_4) = u_2$ *because* $u_2 \prec_{++0} u_4$ *and* $u_4 - u_2 = 2f\hat{p}$. $\mathtt{NP}_f(u) = u_3$ *because* $u \prec_{++0} u_3$ *and* $u_3 - u = f\hat{p}$. *Finally,* $\mathtt{NI}_{x+}(i) = u_6$ *and* $\mathtt{NI}_{x-}(i) = u_5$.

We can compute the nearest candidates defined above in linear $O(m_1 m_2 m_3)$ time as follows. Again let $u_1, u_2, \ldots$ be the candidates in a group C.

Computation of nearest candidates:

1. $\texttt{NC}_f(u_i)$: Since there are $O(\frac{m_2}{h})$ candidates in C and $g = O(h)$, we need $O(\frac{m_2^2}{h})$ time to compute NC for a group C. For all groups, it takes $O(\frac{m_2^2 m_3}{h})$ time. $\texttt{NC}_f$ is used to compute $\texttt{NP}_f$.

2. $\texttt{NP}_f(u)$: Let D be the plane of a candidate block that contains u_i's, and let u be a point in a check line of D. We compute NP for plane D as follows. Initialize $\texttt{NP}_f(u) = nil$ for all u and f. For each pair of $u_i \in C$ and f, we set $\texttt{NP}_f(u) = u_i$ for all points u in the check line containing u_i such that $\texttt{NC}_f(u_i) \prec_{++0} u \prec_{++0} u_i$ ($u \prec_{++0} u_i$ if $\texttt{NC}_f(u_i)$ is undefined) and $u_i - u$ is a multiple of $f\hat{p}$. Since the length of a check line is $O(m_1)$, we need $O(m_1)$ time for each pair of $u_i \in C$ and f. Since there are $O(\frac{m_2}{h})$ candidates in C and $g = O(h)$, $O(m_1 m_2)$ time is needed for each plane D; overall it takes $O(m_1 m_2 m_3)$ time.

3. NI: Since there are $O(m_1)$ values of i and $O(\frac{m_2}{h})$ candidates in a group C, we can compute $\texttt{NI}_{x+}(i)$ and $\texttt{NI}_{x-}(i)$ for C in $O(\frac{m_1 m_2}{h})$ time. For all groups, $O(\frac{m_1 m_2 m_3}{h})$ time is needed.

Consider every pair of a group C and a candidate v in another group. We will find consistent candidates from v and C as follows. Let $C' = \{u \in C \mid v \prec_{+++} u\}$, and let $u_1, u_2, \ldots$ be the candidates in C' such that $u_1 \prec_{++0} u_2 \prec_{++0} \cdots$, where C' can be found using $\texttt{NI}_{x+}$ and $\texttt{NI}_{y+}$ with v. Let j be the difference between the z-coordinates of v and the u_i's. Let U be the set of all periods of the pattern of direction $(+,+,+)$ whose z-coordinates are j. We assume the most general case for U, i.e., that U consists of the periods in P_w (which are lattice points on b_1, b_2), an initial line of periods (which are represented by q, $O(h)$ step vectors v_i, and corresponding c_i, d_i), and $O(h)$ broken periods, by Definition 6. The main task in performing duels between different groups is to find the candidate in C' that is consistent with v and nearest to v. We do the task in $O(h)$ time as follows. Since there are $O(m_3)$ groups for C and $O(\frac{m_2 m_3}{h})$ candidates for v, this task can be done in $O(m_2 m_3^2)$ time.

Finding the nearest consistent candidate:

1. Any two candidates u_i and u_k in C' are congruent modulo b_1, b_2 because $u_k - u_i$ is a period of P, which is in turn a period of P_q. Hence, if $v + q$ is not congruent to $u_1 \in C'$ modulo b_1, b_2, none of C' are consistent with v. If it is congruent, $v + q$ is congruent to all candidates in C' modulo b_1, b_2, and in this case we find nearest consistent candidates to v in each class of Definition 6 as follows.

2. For periods in P_w (class 1), find the candidate $\hat{u} \in C'$ nearest to v such that $v + w \prec_{++0} \hat{u}$ using $\texttt{NI}_{x+}$ and $\texttt{NI}_{y+}$. Since $\hat{u}$ is congruent to $v + q$ modulo b_1, b_2, candidate $\hat{u}$ is consistent with v.

3. For each step vector v_i (class 2), find the candidate $u \in C'$ nearest to v such that $u - (v + q)$ is a period of P_q uniquely represented by the form $c_i + kv_i$ for $1 \le k \le d_i$ (i.e., u is consistent with v) as follows. Since the initial line of P_q is parallel to p by Definition 6, step vector v_i is parallel to p. Let f be the integer such that $v_i = f\hat{p}$. The nearest candidate u is $\texttt{NP}_f(v + q + c_i)$ if $\texttt{NP}_f(v + q + c_i)$ is defined and $|\texttt{NP}_f(v + q + c_i) - (v + q + c_i)|/|v_i| < d_i$; otherwise, it does not exist.

4. For each broken period p_i (class 3), check whether $v + p_i$ is a candidate in C' or not.

5. Among the $O(h)$ candidates found in steps 2–4, find the candidate nearest to v, which we denote by u_l.

Since the u_i's in C' are consistent, v is consistent with all u_i, $i \ge l$. Perform a duel between v and u_{l-1} (the farthest inconsistent candidate). If v is killed by the duel, the candidates in C' remain and they are consistent. If u_{l-1} is killed, then all

u_i, $i \leq l-1$, are killed by their consistency. Since the killed candidates in C' are consecutive ones in the monotone line of candidates, we mark the first and the last of the killed candidates.

We now consider the relation $\prec_{+-+}$. Given a group C and a candidate $v \notin C$, find $C' = \{u \in C \mid v \prec_{+-+} u\}$ using $\mathtt{NI}_{x+}$ and $\mathtt{NI}_{y-}$. If C' is empty, we are done. Otherwise, perform a duel between v and an arbitrary candidate u in C'. When v and u are consistent, v is consistent with all candidates in C', by Lemma 11. When v and u are inconsistent, at least one of them is killed by the duel. If v is killed, we are done. If u is killed, then all candidates in C' are killed, by Lemma 12. We again mark the first and the last of the killed candidates.

Finally, consider the relation $\prec_{-++}$. Given a group C and a candidate $v \notin C$, find $C' = \{u \in C \mid v \prec_{-++} u\}$ using $\mathtt{NI}_{x-}$ and $\mathtt{NI}_{y+}$. The rest of this case is the same as the case of $\prec_{+-+}$ except that v and $u \in C'$ are always inconsistent because $(-,+,+)$ is the missing direction.

Now we remove all killed candidates (the ones between marked candidates) in linear time. Then all the remaining candidates are consistent. Find pattern occurrences by the wave method in linear time. In summary, we have the following theorem.

THEOREM 2. *Given a witness table of the pattern, three-dimensional pattern matching can be done in linear $O(n_1 n_2 n_3)$ time.*

6. Witness computation. In this section we assume for simplicity that the pattern P is of size m^3. We will compute a witness table for three-dimensional pattern P by combining the suffix tree construction in [Fa] and two-dimensional witness computation [GP]. The suffix tree construction is the bottleneck in our algorithm, and it takes $O(m^3)$ time for a constant-size alphabet or an integer alphabet since the time complexity of suffix tree construction has been shown to be equivalent to that of sorting [Fa, FFM]. It takes $O(m^3 \log m)$ time only in the case that Σ is an unbounded alphabet and only symbol comparisons are allowed on Σ.

We first construct a suffix tree with the pattern P and do the following preprocessings on the suffix tree, which were also used in the two-dimensional case [ABF]:

1. Divide the pattern into lines parallel to the z-axis and construct a suffix tree ST for the concatenation of all the lines by Farach's algorithm [Fa]. Its time complexity is equivalent to that of sorting.

2. Process suffix tree ST for LCA (lowest common ancestor) queries [HT, ScV] in $O(m^3)$ time such that the LCA of two nodes can be found in $O(1)$ time.

3. Process suffix tree ST in $O(m^3)$ time such that each node v has the length of the substring labeled from the root to v [LV].

With these preprocessings, a query for the length of the longest common prefix of two substrings (parallel to the z-axis) of P can be answered in $O(1)$ time.

We will find periods of the pattern P and witnesses against nonperiods among all valid vectors of P as follows. For a fixed i, consider the vectors of the form (x, y, i). Let $u = (0, 0, i)$. Then P_u is an array of size $m \times m \times (m-i)$ (bottom $m-i$ planes of P), but it can be considered as a two-dimensional array of size $m \times m$, where each symbol is a line segment parallel to the z-axis. Let A_i be the two-dimensional array of size $2m \times 2m$ whose lower-left and lower-right quadrants are P_{-u} and upper-left and upper-right quadrants are P_u. Then a vector $v = (a, b, i)$ is a period of the pattern if and only if $(m, m) + (a, b)$ is a period of A_i; see Figure 11. Hence all periods of the pattern can be found by solving $O(m)$ copies of the two-dimensional problem. To solve the two-dimensional problems, we use the alphabet-independent linear-time algorithm in [GP]. A comparison of two symbols α, β in A_i can be done in $O(1)$ time

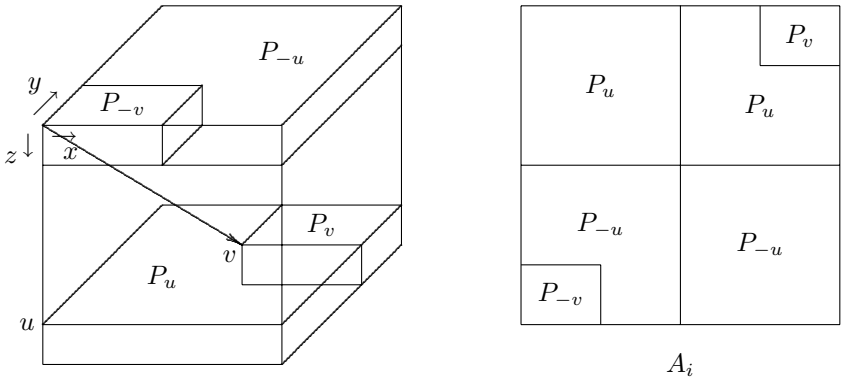


FIG. 11. P_u *and* P_v.

by a query on ST, which tells us whether α and β are the same or not. And if α and β are not the same, it gives the position of a mismatch, which can be used as a witness. Therefore, given the suffix tree ST preprocessed as above, computing a witness table takes $O(m^3)$ time.

7. Conclusion. We studied three-dimensional periodicity and presented a three-dimensional pattern matching algorithm based on it. The parallel algorithm in [GPP] has essentially the same structure as the one in section 5, except that it uses some parallel techniques such as deterministic samples [Vis, CPR] and the parallel version of the wave method [CMR]. As in two dimensions, three-dimensional periodicity plays an important role in developing pattern matching algorithms in three dimensions.

The witness computation algorithm in section 6, though it takes linear time, is not alphabet-independent. An open problem in three dimensions is to develop an alphabet-independent linear-time algorithm that computes a witness table.

Acknowledgments. We are grateful to the referees for their valuable comments, which helped greatly improve the presentation of this paper.

REFERENCES

[AB] A. AMIR AND G. BENSON, *Two-dimensional periodicity and its applications*, in Proceedings of the 3rd ACM-SIAM Symposium on Discrete Algorithms, Orlando, FL, 1992, pp. 440–452.

[ABF] A. AMIR, G. BENSON, AND M. FARACH, *An alphabet independent approach to two-dimensional pattern matching*, SIAM J. Comput., 23 (1994), pp. 313–323.

[Ba] T.P. BAKER, *A technique for extending rapid exact-match string matching to arrays of more than one dimension*, SIAM J. Comput., 7 (1978), pp. 533–541.

[Bi] R.S. BIRD, *Two dimensional pattern matching*, Inform. Process. Lett., 6 (1977), pp. 168–170.

[Br] D. BRESLAUER, *Dictionary-matching on unbounded alphabet: Uniform-length dictionaries*, in Proceedings of the 5th Symposium on Combinatorial Pattern Matching, Lecture Notes in Comput. Sci. 807, Springer, New York, 1994, pp. 184–197.

[CCG] R. COLE, M. CROCHEMORE, Z. GALIL, L. GĄSIENIEC, R. HARIHARAN, S. MUTHUKRISHNAN, K. PARK, AND W. RYTTER, *Optimally fast parallel algorithms for preprocessing and pattern matching in one and two dimensions*, in Proceedings of the 34th IEEE Symposium on the Foundations of Computer Science, Palo Alto, CA, 1993, pp. 248–258.

[CPR] M. CROCHEMORE, Z. GALIL, L. GĄSIENIEC, K. PARK, AND W. RYTTER, *Constant-time randomized parallel string matching*, SIAM J. Comput., 26 (1997), pp. 950–960.

[CMR] M. Crochemore, L. Gąsieniec, R. Hariharan, S. Muthukrishnan, and W. Rytter, *A constant time optimal parallel algorithm for two-dimensional pattern matching*, SIAM J. Comput., 27 (1998), pp. 668–681.

[CPP] A. Czumaj, Z. Galil, L. Gąsieniec, K. Park, and W. Plandowski, *Work-time optimal parallel algorithms for string problems*, in Proceedings of the 27th ACM Symposium on the Theory of Computing, Las Vegas, NM, 1995, pp. 713–722.

[Fa] M. Farach, *Optimal suffix tree construction with large alphabets*, in Proceedings of the 38th IEEE Symposium on the Foundations of Computer Science, Miami Beach, FL, 1997, pp. 137–143.

[FFM] M. Farach-Colton, P. Ferragina, and S. Muthukrishnan, *On the sorting-complexity of suffix tree construction*, J. ACM, 47 (2000), pp. 987–1011.

[GP] Z. Galil and K. Park, *Alphabet-independent two-dimensional witness computation*, SIAM J. Comput., 25 (1996), pp. 907–935.

[GPP] Z. Galil, J.G. Park, and K. Park, *Three-dimensional pattern matching*, in Proceedings of the 9th ACM Symposium on Parallel Algorithms and Architectures, Newport, RI, 1997, pp. 53–62.

[HT] D. Harel and R.E. Tarjan, *Fast algorithms for finding nearest common ancestors*, SIAM J. Comput., 13 (1984), pp. 338–355.

[Ja] J. JaJa, *An Introduction to Parallel Algorithms*, Addison–Wesley, Reading, MA, 1992.

[KR] M. Karpinski and W. Rytter, *Alphabet-independent optimal parallel search for three-dimensional patterns*, Theoret. Comput. Sci., 205 (1998), pp. 243–260.

[KMP] D.E. Knuth, J.H. Morris, Jr., and V.R. Pratt, *Fast pattern matching in strings*, SIAM J. Comput., 6 (1977), pp. 323–350.

[LV] G.M. Landau and U. Vishkin, *Fast parallel and serial approximate string matching*, J. Algorithms, 10 (1989), pp. 157–169.

[ScV] B. Schieber and U. Vishkin, *On finding lowest common ancestors: Simplification and parallelization*, SIAM J. Comput., 17 (1988), pp. 1253–1262.

[Vi] U. Vishkin, *Optimal parallel pattern matching in strings*, Inform. and Control, 67 (1985), pp. 91–113.

[Vis] U. Vishkin, *Deterministic sampling—A new technique for fast pattern matching*, SIAM J. Comput., 20 (1991), pp. 22–40.

SIAM J. DISCRETE MATH.
Vol. 18, No. 2, pp. 382–387

RESOLVING THE EXISTENCE OF FULL-RANK TILINGS OF BINARY HAMMING SPACES*

PATRIC R. J. ÖSTERGÅRD† AND ALEXANDER VARDY‡

Abstract. A tiling of $\mathbb{F}_2^n$ is a pair (V, A) of subsets of $\mathbb{F}_2^n$ such that every $x \in \mathbb{F}_2^n$ can be written in exactly one way as $x = v + a$ with $v \in V$ and $a \in A$. A tiling (V, A) of $\mathbb{F}_2^n$ is said to be full-rank if $\mathrm{rank}(V) = \mathrm{rank}(A) = n$ and $\mathbf{0} \in (V \cap A)$. It is known that every tiling (V, A) decomposes into smaller tilings that are either trivial or full rank. It is furthermore known that full-rank tilings of $\mathbb{F}_2^n$ exist for all $n \geqslant 10$ and do not exist for $n \leqslant 8$. The last case $n = 9$ is resolved in this paper, thereby proving that full-rank tilings of $\mathbb{F}_2^n$ exist if and only if $n \geqslant 10$. To establish this result, we use two different methods. The first method employs group characters to show that the sets V and A in a full-rank tiling (V, A) of $\mathbb{F}_2^9$ must have a certain structure. The second method is based on the classification of $[14, 5, 3]$ binary linear codes and uses a fast algorithm for the exact cover problem. Both methods rely on a carefully designed exhaustive computer search to complete the proof.

Key words. characters, exact cover, full-rank tilings, perfect codes

AMS subject classifications. 05–04, 05A18, 05B45, 94B25

DOI. 10.1137/S0895480104435804

1. Introduction.

Let $\mathbb{F}_2^n$ be a vector space of dimension n over GF(2). A *tiling* of $\mathbb{F}_2^n$ is a pair (V, A) of subsets of $\mathbb{F}_2^n$ such that every $x \in \mathbb{F}_2^n$ can be written in exactly one way as $x = v + a$ with $v \in V$ and $a \in A$. A tiling (V, A) of $\mathbb{F}_2^n$ is *trivial* if one of the sets V, A is $\mathbb{F}_2^n$ while the other is $\{\mathbf{0}\}$, where $\mathbf{0}$ denotes the all-zero vector in $\mathbb{F}_2^n$. It is *full-rank* if $\mathrm{rank}(V) = \mathrm{rank}(A) = n$ and $\mathbf{0} \in (V \cap A)$. It is shown in [1] that any tiling of $\mathbb{F}_2^n$ decomposes into smaller tilings that are either trivial or full-rank. This reduces the classification of tilings of binary Hamming spaces to the study of full-rank tilings.

It was established in [1, 2] that full-rank tilings of $\mathbb{F}_2^n$ exist for $n = 14$ and $n \geqslant 112$, and do not exist for $n \leqslant 7$. Subsequently, it was shown in [3, Theorem 16] that if $\mathbb{F}_2^{n_0}$ admits a full-rank tiling, then so does $\mathbb{F}_2^n$ for all $n \geqslant n_0$. Then Le Van and Phelps [5] found, by computer search, a full-rank perfect binary code of length 15 with a kernel of dimension 5. By Construction D of [1, 3], this implies the existence of a full-rank tiling of $\mathbb{F}_2^{10}$. Thus it was known since 1997 that full-rank tilings of $\mathbb{F}_2^n$ exist for all $n \geqslant 10$ and do not exist for $n \leqslant 7$, leaving the cases $n = 8$ and $n = 9$ unresolved. The following problem was posed in [3, p. 220]; we quote:

> Construct full-rank tilings of $\mathbb{F}_2^n$ for $n = 8$ and $n = 9$, or prove that such tilings do not exist. This problem appears to be quite challenging, despite the small size of the sets involved.

The case $n = 8$ was recently settled in [8], where it is proved that $\mathbb{F}_2^8$ does not admit a full-rank tiling. However, the structural approach to an exhaustive search for full-rank tilings developed in [8] falls short for $n = 9$. In this paper, we extend the methods

*Received by the editors January 15, 2004; accepted for publication (in revised form) May 7, 2004; published electronically December 9, 2004. This research was supported in part by the Academy of Finland under grants 100500 and 202315, by the David and Lucile Packard Foundation, and by the National Science Foundation.

http://www.siam.org/journals/sidma/18-2/43580.html

†Department of Electrical and Communications Engineering, Helsinki University of Technology, P.O. Box 3000, 02015 HUT, Finland (patric.ostergard@hut.fi).

‡Department of Electrical and Computer Engineering, Department of Computer Science and Engineering, Department of Mathematics, University of California San Diego, 9500 Gilman Drive, La Jolla, CA 92093 (vardy@kilimanjaro.ucsd.edu).

of [8] to show that a full-rank tiling of $\mathbb{F}_2^9$ does not exist. This is the last piece of the puzzle needed to complete the proof of the following theorem.

THEOREM 1. *A full-rank tiling of $\mathbb{F}_2^n$ exists if and only if $n \geqslant 10$.*

In sections 3 and 4, we present two different proofs of Theorem 1, which were devised independently by the second and the first author, respectively. In section 3, we use the group characters of [6, pp. 132–141] to show that if (V, A) is a full-rank tiling of $\mathbb{F}_2^9$, then one can assume without loss of generality (w.l.o.g.) that the set A is "evenly distributed" on the first three positions. This reduces the search for such a tiling to a problem concerning certain configurations of $(6, 4, 3)$ binary codes. On the other hand, in section 4 we use the classification of $[14, 5, 3]$ binary linear codes, obtained in [7], to show that it suffices to consider only 2164 cases for the set V in a full-rank tiling (V, A) of $\mathbb{F}_2^9$. Each such case produces an instance of the exact cover problem, which can be solved efficiently using a recent algorithm of Knuth [4]. First, however, we need to review certain aspects of the theory developed in [8].

2. Preliminaries. Let (V, A) be a full-rank tiling of $\mathbb{F}_2^9$. Since every $x \in \mathbb{F}_2^9$ is represented uniquely as $x = v+a$ with $v \in V$ and $a \in A$, we have $|V||A| = 2^9$. Thus $|V|$ and $|A|$ are powers of 2. Since $\operatorname{rank}(V) = \operatorname{rank}(A) = 9$, both sets contain at least 2^4 vectors. Thus we can assume w.l.o.g. that $|V| = 2^4$ and $|A| = 2^5$.

LEMMA 2. *Let (V, A) be a full-rank tiling of $\mathbb{F}_2^n$, let M be an invertible $n \times n$ binary matrix, and let $\varphi_M(x) = xM$. Then $(\varphi_M(V), \varphi_M(A))$ is a full-rank tiling of $\mathbb{F}_2^n$.*

Lemma 2 was proved in [8]. It will be used as the starting point for both of our proofs in sections 3 and 4. Both proofs also assume that $|V| = 16$ and $|A| = 32$.

3. Search for a full-rank tiling of $\mathbb{F}_2^9$ using group characters. Let $\mathcal{S}_9 \stackrel{\text{def}}{=} \{\mathbf{0}, e_1, e_2, \ldots, e_9\}$ be the Hamming sphere of radius 1 about $\mathbf{0}$ in $\mathbb{F}_2^9$. Using Lemma 2, we can transform a full-rank tiling (V, A) of $\mathbb{F}_2^9$ into a full-rank tiling $(\varphi_M(V), \varphi_M(A))$ with the property that $\mathcal{S}_9 \subset \varphi_M(V)$. Thus, as in [8], we will assume later in this section that $\mathcal{S}_9 \subset V$, and write $V = \mathcal{S}_9 \cup V'$ with $|V'| = 6$. As we show next, this assumption implies that the other set A in the full-rank tiling (V, A) also has a lot of structure.

Let $d(\cdot, \cdot)$ denote the Hamming distance. Given $\mathbb{C} \subset \mathbb{F}_2^n$ such that $d(x_1, x_2) \geqslant d$ for all distinct $x_1, x_2 \in \mathbb{C}$, we write $d(\mathbb{C}) \geqslant d$ and say that $\mathbb{C}$ is *an $(n, |\mathbb{C}|, d)$ code.*

LEMMA 3. *Let (V, A) be a tiling of $\mathbb{F}_2^9$ with $\mathcal{S}_9 \subset V$. Then $d(A) \geqslant 3$.*

As shown in [8], Lemma 3 follows immediately from the definition of a tiling. However, while the analysis of [8] stops at this point, we move on, using characters.

As in [6, Chapter 5], we represent a vector $x = (x_1, x_2, \ldots, x_n) \in \mathbb{F}_2^n$ by its image $z^x = z_1^{x_1} z_2^{x_2} \cdots z_n^{x_n}$ in the group algebra QG, where G is the addition group of $\mathbb{F}_2^n$. Then for any $u, x \in \mathbb{F}_2^n$ and $\mathbb{C} \subseteq \mathbb{F}_2^n$, the group *characters* $\chi_u(\cdot)$ are defined by

$$\chi_u(z^x) \stackrel{\text{def}}{=} (-1)^{\langle u,x \rangle} \qquad \text{and} \qquad \chi_u(\mathbb{C}) \stackrel{\text{def}}{=} \sum_{x \in \mathbb{C}} \chi_u(z^x),$$

where $\langle \cdot, \cdot \rangle$ stands for the inner product (modulo 2) in $\mathbb{F}_2^n$. It is trivial that $\chi_u(\mathbb{F}_2^n) = 0$ for all nonzero $u \in \mathbb{F}_2^n$. This leads to the following proposition.

PROPOSITION 4. *Let (V, A) be a tiling of $\mathbb{F}_2^n$. Then for all nonzero $u \in \mathbb{F}_2^n$, either $\chi_u(V) = 0$ or $\chi_u(A) = 0$, or both $\chi_u(V) = \chi_u(A) = 0$.*

Proof. Since $V+A = \mathbb{F}_2^n$, in the group algebra QG we have $\chi_u(V)\,\chi_u(A) = \chi_u(\mathbb{F}_2^n)$. Thus if $u \neq \mathbf{0}$, at least one of $\chi_u(V)$ or $\chi_u(A)$ must be zero. □

From now through the end of this section, we assume that (V, A) is a full-rank tiling of $\mathbb{F}_2^9$ with $|A| = 32$ and $V = \mathcal{S}_9 \cup V'$. Proposition 4 leads to a number of

conclusions regarding A, which can be conveniently stated in terms of the following notation: given $\mathbb{C} \subseteq \mathbb{F}_2^n$, we let $[\mathbb{C}]$ denote the $|\mathbb{C}| \times n$ matrix having the codewords of $\mathbb{C}$ as its rows.

COROLLARY 5. *Every column of* $[A]$ *has exactly* 16 *ones and* 16 *zeros.*

Proof. The claim of the corollary is equivalent to the statement that $\chi_{e_i}(A) = 0$ for all i, where $e_1, e_2, \ldots, e_9$ are the vectors of weight 1 in $\mathbb{F}_2^9$. Note that $\chi_{e_i}(\mathcal{S}_9) = 8$ for all i, and therefore $\chi_{e_i}(V) \geqslant 2$. The claim thus follows from Proposition 4. □

COROLLARY 6. *In addition to* $\mathbf{0}$*, the set* A *has exactly* 15 *vectors of even weight.*

Proof. This is equivalent to $\chi_{\mathbf{1}}(A) = 0$, where $\mathbf{1}$ denotes the all-one vector in $\mathbb{F}_2^9$. Since $\chi_{\mathbf{1}}(\mathcal{S}_9) = -8$, the claim follows from Proposition 4. □

Let $\mathcal{I}$ be a subset of $[n] \stackrel{\text{def}}{=} \{1, 2, \ldots, n\}$ with $|\mathcal{I}| = t$. We say that a code $\mathbb{C} \subseteq \mathbb{F}_2^n$ is *evenly distributed* on the t positions in $\mathcal{I}$ if the corresponding t columns of $[\mathbb{C}]$ contain every binary t-tuple exactly $|\mathbb{C}|/2^t$ times.

LEMMA 7. *Let* $\mathbb{C} \subseteq \mathbb{F}_2^n$*. Let* $\mathcal{I} \leqslant [n]$ *with* $|\mathcal{I}| = t$*. Then* $\mathbb{C}$ *is evenly distributed on the* t *positions in* $\mathcal{I}$ *if and only if* $\chi_u(\mathbb{C}) = 0$ *for all nonzero* $u \in \mathbb{F}_2^n$ *with* $\mathrm{supp}(u) \subseteq \mathcal{I}$*.*

Proof. As in Corollary 5, the fact that $\chi_u(\mathbb{C}) = 0$ for all vectors u of weight 1 with $\mathrm{supp}(u) \subseteq \mathcal{I}$ implies that each of the corresponding columns of $[\mathbb{C}]$ has $|\mathbb{C}|/2$ ones and $|\mathbb{C}|/2$ zeros. Then $\chi_u(\mathbb{C}) = 0$ for all u of weight 2 with $\mathrm{supp}(u) \subseteq \mathcal{I}$ implies that each pair of relevant columns of $[\mathbb{C}]$ contains $|\mathbb{C}|/4$ occurrences of each of 00, 01, 10, 11. In other words, $\mathbb{C}$ is evenly distributed on every pair of positions in $\mathcal{I}$. Clearly, one can continue in this way up to vector(s) u of weight t, to show that $\mathbb{C}$ is evenly distributed on all the t positions in $\mathcal{I}$. The converse claim, namely, that $\chi_u(\mathbb{C}) = 0$ for all nonzero u with $\mathrm{supp}(u) \subseteq \mathcal{I}$ provided $\mathbb{C}$ is evenly distributed on $\mathcal{I}$, should be obvious. □

Since $\chi_u(\mathcal{S}_9) = 6$ for any $u \in \mathbb{F}_2^9$ with $\mathrm{wt}(u) = 2$, it follows from Proposition 4 and Lemma 7 that A is evenly distributed on any pair of positions $\{i, j\}$, unless the ith and jth columns of $[V']$ are complements of each other. In fact, a stronger property holds.

LEMMA 8. *There is a set* $\mathcal{I} = \{i, j, k\}$ *such that* A *is evenly distributed on* $\mathcal{I}$*.*

Proof. In view of Proposition 4 and Lemma 7, it would suffice to prove that there exists a set $\{i, j, k\}$ such that $\chi_u(V) \neq 0$ for all seven nonzero vectors $u \in \mathbb{F}_2^9$ whose support is confined to $\{i, j, k\}$. Equivalently, it would suffice to find three columns in $[V']$ such that no two are complements of each other and the even-weight 3-tuples 000, 011, 101, 110 occur at least twice among the six rows. It is not difficult to show that any 6×9 binary matrix with distinct rows contains some three columns with the desired property. We leave the details of this as an exercise for the reader. □

By Lemma 8, we can assume w.l.o.g. that A is evenly distributed on the first three positions. This means that A has the following structure:

$$
(1)\qquad
\begin{array}{c|c}
\begin{matrix}000\\000\\000\\000\end{matrix} & \mathbb{C}_{000} \\ \hline
\begin{matrix}100\\100\\100\\100\end{matrix} & \mathbb{C}_{100}
\end{array}
\qquad
\begin{array}{c|c}
\begin{matrix}010\\010\\010\\010\end{matrix} & \mathbb{C}_{010} \\ \hline
\begin{matrix}001\\001\\001\\001\end{matrix} & \mathbb{C}_{001}
\end{array}
\qquad
\begin{array}{c|c}
\begin{matrix}110\\110\\110\\110\end{matrix} & \mathbb{C}_{110} \\ \hline
\begin{matrix}101\\101\\101\\101\end{matrix} & \mathbb{C}_{101}
\end{array}
\qquad
\begin{array}{c|c}
\begin{matrix}011\\011\\011\\011\end{matrix} & \mathbb{C}_{011} \\ \hline
\begin{matrix}111\\111\\111\\111\end{matrix} & \mathbb{C}_{111}
\end{array}
\,,
$$

where $\mathbb{C}_{000}, \mathbb{C}_{001}, \ldots, \mathbb{C}_{111}$ are $(6, 4, 3)$ codes by Lemma 3. Note that $\mathbb{C}_{000}$ contains the vector $\mathbf{0}$, by definition. It is easy to verify that there are 14 nonisomorphic (up to coordinate permutations) choices for $\mathbb{C}_{000}$. Moreover, for any distinct $u, v \in \mathbb{F}_2^3$, we have $d(\mathbb{C}_u, \mathbb{C}_v) \geqslant 3 - d(u, v)$ by Lemma 3. This implies that the codes $\mathbb{C}_{100}$, $\mathbb{C}_{010}$, $\mathbb{C}_{001}$ are disjoint and are at distance $\geqslant 2$ from $\mathbb{C}_{000}$. The following table lists, for each of the 14

nonisomorphic choices for $\mathbb{C}_{000}$, the number of $(6,4,3)$ codes at distance $\geqslant 2$ from it:

$\mathbf{0}, 000111, 111000, 111111$:	2709 codes	$\mathbf{0}, 011011, 100011, 111100$:	2363 codes
$\mathbf{0}, 100110, 111000, 111111$:	2363 codes	$\mathbf{0}, 001111, 110011, 111100$:	2093 codes
$\mathbf{0}, 001101, 110001, 111110$:	2363 codes	$\mathbf{0}, 101010, 110011, 111100$:	2013 codes
$\mathbf{0}, 001111, 110001, 111110$:	2709 codes	$\mathbf{0}, 001110, 110010, 111100$:	2093 codes
$\mathbf{0}, 101101, 110001, 111110$:	2363 codes	$\mathbf{0}, 001101, 110010, 111100$:	2363 codes
$\mathbf{0}, 100111, 111001, 111110$:	2363 codes	$\mathbf{0}, 101001, 110010, 111100$:	2013 codes
$\mathbf{0}, 011010, 100011, 111100$:	2363 codes	$\mathbf{0}, 010101, 100110, 111000$:	2013 codes

Once the codes $\mathbb{C}_{000}, \mathbb{C}_{100}, \mathbb{C}_{010}, \mathbb{C}_{001}$ are fixed, all the possible choices (there are not too many) for the remaining vectors in A can be easily computed using (1), the fact that $d(\mathbb{C}_u, \mathbb{C}_v) \geqslant 3 - d(u,v)$, and Corollaries 5 and 6. Finally, given A, it is straightforward to check whether $\operatorname{rank}(A) = 9$ and, if so, whether the remaining six vectors in V can be completed in such a way that (V, A) is a tiling.

A computer search based on this approach did not produce a full-rank tiling of $\mathbb{F}_2^9$, thereby proving Theorem 1. The source code of our program is available by anonymous ftp to montblanc.ucsd.edu/pub. The search took 10 days on a 1.6 GHz PC.

4. Search for a full-rank tiling of $\mathbb{F}_2^9$ using classification of [14, 5, 3] codes. The key idea of this proof is to make much better use of Lemma 2 than in section 3 and [8]. Let (V, A) be a full-rank tiling of $\mathbb{F}_2^9$ with $|V| = 16$. Assuming $\mathcal{S}_9 \subset V$ as in section 3 still leaves about $4 \cdot 10^{10}$ possible choices for V (even after taking Lemma 10, below, into account). We show next that, in fact, this number can be reduced to only 2164.

A code $\mathbb{C} \subseteq \mathbb{F}_2^n$ is *linear* if it is a subspace of $\mathbb{F}_2^n$. An $[n,k,d]$ *code* is a linear code $\mathbb{C} \subseteq \mathbb{F}_2^n$ such that $\dim \mathbb{C} = k$ and $d(\mathbb{C}) \geqslant d$. An $[n,k,d]$ code can be defined in terms of an $(n-k)\times n$ full-rank *parity-check matrix* H such that $\mathbb{C} = \{x \in \mathbb{F}_2^n : Hx^t = \mathbf{0}\}$. Two $[n,k,d]$ codes $\mathbb{C}_1$, $\mathbb{C}_2$ are *equivalent* if there is a permutation π of $[n]$ with $\pi(\mathbb{C}_1) = \mathbb{C}_2$. Suppose there are exactly N inequivalent $[15,6,3]$ codes, say, $\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_N$, and let $H_1, H_2, \ldots, H_N$ be some parity-check matrices for $\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_N$.

LEMMA 9. *Let (V, A) be a full-rank tiling of $\mathbb{F}_2^9$, and let $V_{\{\emptyset\}}$ denote the set of the nonzero vectors of V. Then there is an arrangement of rows of $[V_{\{\emptyset\}}]$ and a 9×9 invertible binary matrix M such that $M[V_{\{\emptyset\}}]^t = H_i$ for some $i \in \{1, 2, \ldots, N\}$.*

Proof. Since $[V_{\{\emptyset\}}]^t$ is a full-rank 9×15 matrix, it is a parity-check matrix for some $[15,6,d]$ code $\mathbb{C}$. Moreover $d(\mathbb{C}) \geqslant 3$, as all the columns of $[V_{\{\emptyset\}}]^t$ are nonzero and distinct. Thus $\mathbb{C}$ is a $[15,6,3]$ code, and must be equivalent to one of $\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_N$. □

By Lemmas 2 and 9, in order to enumerate all possible full-rank tilings of $\mathbb{F}_2^9$, it would suffice to choose from an exhaustive set of parity-check matrices for inequivalent $[15,6,3]$ codes. Binary $[n,k,d]$ codes with $d \geqslant 3$ have been classified in [7] for all k up to length $n = 14$. One may continue this classification one step further to find all the inequivalent $[15,6,3]$ codes. However, we will use the next lemma instead.

An $(n, |\mathbb{C}|, 3)$ code $\mathbb{C}$ is *perfect* if the Hamming spheres of radius 1 about the codewords of $\mathbb{C}$ cover $\mathbb{F}_2^n$. A code $\mathbb{C} \subseteq \mathbb{F}_2^n$ is *full-rank* if $\mathbf{0} \in \mathbb{C}$ and $\operatorname{rank}(\mathbb{C}) = n$. The *kernel* of $\mathbb{C}$ is the set of all $x \in \mathbb{F}_2^n$ such that $x + \mathbb{C} = \mathbb{C}$. The next lemma is similar to Lemma 4 of [8], but requires a slightly different proof that, in fact, relies on the main result of [8].

LEMMA 10. *If (V, A) is a full-rank tiling of $\mathbb{F}_2^9$, the sum of the vectors in $V_{\{\emptyset\}}$ is $\mathbf{0}$.*

Proof. Consider the code $\mathbb{C} = \{x \in \mathbb{F}_2^{15} : [V_{\{\emptyset\}}]^t x^t \in A\}$. It follows from Theorem 5.3 and Propositions 5.4 and 5.5 of [3] that $\mathbb{C}$ is a full-rank perfect code with a kernel of dimension $6 + \dim(\ker A)$. Let v^* denote the sum of the vectors in $V_{\{\emptyset\}}$. It is shown in [1, Proposition 8.3] that $v^* \in \ker A$. Thus either $v^* = \mathbf{0}$ and we are done, or

$\dim(\ker A) \geqslant 1$. But in the latter case, $\dim(\ker \mathbb{C}) \geqslant 7$. By Proposition 5.6 of [3], this would imply that there is a full-rank tiling of $\mathbb{F}_2^8$. By the main result of [8], such a tiling does not exist. □

Lemma 10 implies that the $[15, 6, 3]$ code $\mathbb{C}$ defined by the parity-check matrix $[V_{\{\emptyset\}}]^t$ is *self-complementary*, meaning that $\mathbf{1} \in \mathbb{C}$. It is known [7] that there are precisely 17934 inequivalent $[14, 5, 3]$ codes. Let $H_1, H_2, \ldots, H_{17934}$ be a set of parity-check matrices for these codes; this set can be obtained from the first author or at http://www.hut.fi/~pat/matrices.html. Then a set of parity-check matrices for all the inequivalent self-complementary $[15, 6, 3]$ codes can be constructed as follows. For each $i = 1, 2, \ldots, 17934$, we append the sum of the 14 columns of H_i as its 15th column; we then remove any isomorphs created thereby and discard codes with minimum distance $\leqslant 2$. It turns out that there are precisely 2164 inequivalent self-complementary $[15, 6, 3]$ codes. To each of the corresponding parity-check matrices, we append the vector $\mathbf{0}$ to obtain the set V. It remains to compute, for each such V, the set of all A such that (V, A) is a tiling of $\mathbb{F}_2^9$.

The computational task above can be regarded as a special case of the exact cover problem. In general, the *exact cover problem* can be formulated as follows: given a set S and a prescribed collection of subsets of S, compute all partitions of S by these subsets. In our case, we can take $S = \mathbb{F}_2^9 \setminus V$ and for each $a \in \mathbb{F}_2^9 \setminus (V+V)$ define a subset of S consisting of $a + V$. To solve the exact cover problem, we used a fast algorithm developed by Knuth [4]. Solving the 2164 instances of exact cover in this manner produced 29823112 solutions, but none of full-rank, thereby proving Theorem 1. The search took only 18 minutes on a 1.0 GHz PC. Of course, if the classification of $[14, 5, 3]$ codes were not readily available, there would be a slight increase in the computation times.

5. The smallest full-rank tilings. It follows from Theorem 1 that a full-rank tiling of $\mathbb{F}_2^{10}$ is the *smallest possible*. As mentioned in section 1, the existence of such a tiling follows, using the results of [3], from the discovery by Le Van and Phelps [5] of a full-rank perfect code of length 15 with a kernel of dimension 5. Apparently, however, this tiling has never appeared in print, so we present it here. The set V consists of the Hamming sphere of radius 1 about $\mathbf{0}$, along with the five vectors 0111110001, 1100010111, 1011001011, 0101101111, and 1010111101. The set A is given by

(2)

000	0000000	010	1001100	110	0000100	011	0100000
000	1100110	010	1100001	110	0010010	011	1010000
000	0011001	010	1010101	110	1100010	011	0000011
000	1111111	010	0110011	110	0101001	011	0001101
000	0101101	010	0111000	110	0100111	011	1000110
000	1101000	010	1001011	110	1110100	011	0110110
000	0010110	010	0101110	110	1011001	011	1111010
000	1011010	010	0011111	110	1011110	011	1111101
100	0000011	001	1000101	101	0000110	111	1000001
100	1010000	001	0001010	101	0001001	111	0011000
100	0101010	001	0011100	101	1100000	111	1001010
100	0010101	001	0110001	101	1001100	111	0101100
100	1100101	001	1110100	101	0110010	111	0110101
100	0111100	001	1010011	101	0011111	111	1010111
100	1110011	001	0100111	101	1111001	111	0111011
100	1001111	001	1101011	101	1111110	111	1101111

It is easy to see from the results of section 3 that any full-rank tiling (V, A) of $\mathbb{F}_2^{10}$ with $|V| = 16$ must have the general structure of (2). The tiling in (2) is not unique, however. Here is another full-rank tiling of $\mathbb{F}_2^{10}$ that we have found. The set V again consists of the Hamming sphere of radius 1 about $\mathbf{0}$, along with the five vectors

1100000000, 0011100000, 0000011100, 1011011010, and 1011011001. The set A is given by

(3)

000	0000000	010	0010110	110	0001100	011	0000011
000	0000111	010	0011001	110	0010011	011	0000100
000	0101001	010	0101111	110	0101010	011	0101000
000	0111110	010	0110000	110	0111101	011	0110111
000	1011000	010	1000001	110	1001011	011	1011011
000	1011111	010	1001110	110	1010100	011	1011100
000	1101100	010	1110101	110	1100000	011	1100010
000	1110011	010	1111010	110	1100111	011	1101101
100	0010101	001	0001010	101	0001111	111	0001001
100	0011010	001	0011101	101	0010000	111	0011110
100	0100011	001	0100101	101	0111011	111	0100110
100	0100100	001	0110010	101	0111100	111	0110001
100	1000010	001	1000110	101	1001000	111	1000101
100	1001101	001	1010001	101	1010111	111	1010010
100	1110110	001	1101011	101	1100001	111	1111000
100	1111001	001	1110100	101	1101110	111	1111111

In fact, a preliminary study indicates that the sets V for (2) and (3) are, up to equivalence, the *only* two sets of size 16 that lead to full-rank tilings of $\mathbb{F}_2^{10}$.

Acknowledgments. We would like to thank Ari Trachtenberg for his contributions to this paper, and Petteri Kaski for providing an implementation of the exact cover algorithm from [4].

REFERENCES

[1] G. Cohen, S. Litsyn, A. Vardy, and G. Zémor, *Tilings of binary spaces*, SIAM J. Discrete Math., 9 (1996), pp. 393–412.

[2] T. Etzion and A. Vardy, *Perfect codes: Constructions, properties and enumeration*, IEEE Trans. Inform. Theory, 40 (1994), pp. 754–763.

[3] T. Etzion and A. Vardy, *On perfect codes and tilings: Problems and solutions*, SIAM J. Discrete Math., 11 (1998), pp. 205–223.

[4] D. E. Knuth, *Dancing links*, in Millennial Perspectives in Computer Science, J. Davies, B. Roscoe, and J. Woodcock, eds., Palgrave Macmillan, Basingstoke, UK, 2000, pp. 187–214.

[5] M. Le Van and K. T. Phelps, *unpublished private communication.*

[6] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland Elsevier, Amsterdam, 1977.

[7] P. R. J. Östergård, *Classifying subspaces of Hamming spaces*, Des. Codes Cryptogr., 27 (2002), pp. 297–305.

[8] A. Trachtenberg and A. Vardy, *Full-rank tilings of* $\mathbb{F}_2^8$ *do not exist*, SIAM J. Discrete Math., 16 (2003), pp. 390–392.

SIAM J. DISCRETE MATH.
Vol. 18, No. 2, pp. 388–396

EXTREMAL GRAPHS WITHOUT TOPOLOGICAL COMPLETE SUBGRAPHS*

M. CERA†, A. DIÁNEZ‡, AND A. MÁRQUEZ§

Abstract. The exact values of the function $ex(n;TK_p)$ are known for $\lceil \frac{2n+5}{3} \rceil \leq p < n$ (see [Cera, Diánez, and Márquez, *SIAM J. Discrete Math.*, 13 (2000), pp. 295–301]), where $ex(n;TK_p)$ is the maximum number of edges of a graph of order n not containing a subgraph homeomorphic to the complete graph of order p. In this paper, for $\lceil \frac{2n+6}{3} \rceil \leq p < n-3$, we characterize the family of extremal graphs $EX(n;TK_p)$, i.e., the family of graphs with n vertices and $ex(n;TK_p)$ edges not containing a subgraph homeomorphic to the complete graph of order p.

Key words. extremal graph theory, topological complete subgraphs

AMS subject classifications. 05C35, 05C70

DOI. 10.1137/S0895480100378677

1. Introduction.

The study of the function $ex(n;TK_p)$—i.e., the maximum number of edges of a graph of order n not containing a subgraph homeomorphic to K_p, where K_p is the complete graph with p vertices—is one of the most general extremal problems, as pointed out by Bollobas in [1]. Exact values for this function are known only in some cases, as can be seen in Table 1.1.

TABLE 1.1
Exact values of the function $ex(n;TK_p)$.

p	$ex(n;TK_p)$	Reference
3	$n-1$	
4	$2n-3$	[3]
5	$3n-6$	[4], [8], [9]
$\vdots$	$\vdots$	$\vdots$
$\lceil \frac{2n+5}{3} \rceil \leq p < \lceil \frac{3n+2}{4} \rceil$	$\binom{n}{2} - (5n-6p+3)$	[2]
$\lceil \frac{3n+2}{4} \rceil \leq p < n$	$\binom{n}{2} - (2n-2p+1)$	[2]

The aim of this work is to characterize a family of extremal graphs $EX(n;TK_p)$ for appropriate values of n and p, i.e., the set of graphs of order n, with $ex(n;TK_p)$

*Received by the editors September 28, 2000; accepted for publication (in revised form) July 1, 2004; published electronically December 9, 2004. This research was partially supported by the Ministry of Science and Technology, Spain, Research Project BMF2001-2474.
http://www.siam.org/journals/sidma/18-2/37867.html
†E.U.I.T. Agrícola, Universidad de Sevilla, Ctra. Utrera s/n, 41013-Sevilla, Spain (mcera@us.es).
‡E.T.S. Arquitectura, Universidad de Sevilla, Reina Mercedes 2, 41012-Sevilla, Spain (anadianez@us.es).
§E.T.S.I. Informática, Universidad de Sevilla, Reina Mercedes s/n, 41012-Sevilla, Spain (almar@us.es).

edges and not containing any subgraph homeomorphic to K_p. Actually, we characterize the family $EX(n;TK_p)$ for $\lceil\frac{2n+6}{3}\rceil \le p < n-3$:

$$EX(n;TK_p) = \begin{cases} (3n-4p+2)\overline{K_3} + (6p-4n-3)\overline{K_2} & \text{for } \lceil\frac{2n+6}{3}\rceil \le p < \lceil\frac{3n+2}{4}\rceil, \\ \\ K_{4p-3n-2} + (2n-2p+1)\overline{K_2} & \text{for } \lceil\frac{3n+2}{4}\rceil \le p < n-3. \end{cases}$$

2. Definitions and notation. Given a graph H and a set $\{v_1,\dots,v_q\}$ of vertices of H, we denote by $H_0 = H$ and by H_k for $k = 1,\dots,q$ the induced subgraph in H by the set of vertices $V(H) - \{v_1,\dots,v_k\}$. We denote by $\Delta(H)$ the maximum degree of the graph H and by $\delta_H(v)$ the degree of the vertex v in the graph H. The complement graph of H will be denoted by $\overline{H}$.

Let q and s be a pair of nonnegative integers; $\mathcal{C}_q^s$ denotes the set of graphs H such that there exists a set $\{v_1,\dots,v_q\}$ of vertices of H verifying the following:

(1) $\delta_{H_{j-1}}(v_j) \ge \delta_{H_j}(v_{j+1})$ for $j = 1,\dots,q-1$.

(2) For each positive integer h, if there exists $k \in \{1,\dots,q\}$ and $v \in H_k$ such that $\delta_{H_k}(v) \ge h$, then $\delta_{H_j}(v_{j+1}) \ge h$ for all $j = 1,\dots,k$.

(3) H_q has at most s edges (i.e., $|E(H_q)| \le s$).

The next results show different conditions to guarantee that a graph belongs to the family described above (see [2]).

LEMMA 2.1 (see [2]). *Let H be a graph with n vertices. Then, for any $q \le n$, there exists s such that H is in $\mathcal{C}_q^s$.*

When $s = q$, we know sufficient conditions for the edges of a graph to belong to the class $\mathcal{C}_q^q$.

LEMMA 2.2 (see [2]). *Let n and q be two positive integers, with $q < n$. If H is a graph with n vertices and $2q$ edges, then*

1. $H \in \mathcal{C}_q^q$,
2. $\delta_{H_q}(v) \le 1$ *for* $v \in V(H_q)$.

LEMMA 2.3 (see [2]). *Let q and k be two positive integers with $k \le q-2$. Let H be a graph with $4q-k+1$ vertices and $2q+k+1$ edges. Then $H \in \mathcal{C}_q^q$.*

Notation and terminology not given here can be found in [1] and [2].

3. The family of extremal graphs. In this section, we will characterize the family $EX(n;TK_p)$ for $\lceil\frac{2n+6}{3}\rceil \le p < n-3$. This problem is equivalent to characterizing $EX(n;TK_{n-q})$ for $n \ge 4q+2$ with $q \ge 4$ (case $\lceil\frac{3n+2}{4}\rceil \le p < n-3$) and $n = 4q-k+1$ with $q \ge 5$, $0 \le k \le q-5$ (the case $\lceil\frac{2n+6}{3}\rceil \le p < \lceil\frac{3n+2}{4}\rceil$).

In order to avoid excessive repetition, we define the graphs $\mathcal{H}(n;TK_{n-q})$:

$$\mathcal{H}(n;TK_{n-q}) = \begin{cases} K_{n-(4q+2)} + (2q+1)\overline{K_2} & \text{for } n \ge 4q+2, \\ \\ (k+1)\overline{K_3} + (2(q-k)-1)\overline{K_2} & \text{for } n = 4q-k+1,\ 0 \le k \le q-5. \end{cases}$$

For $n \ge 4q+2$, a graph G belongs to the family $\{\mathcal{H}(n;TK_{n-q})\}$ if G has n vertices and $\overline{G}$ is formed by $2q+1$ nonadjacent edges (see Figure 3.1).

For $n = 4q-k+1$ with $q \ge 5$ and $0 \le k \le q-5$, a graph G belongs to the family $\{\mathcal{H}(n;TK_{n-q})\}$ if it has $4q-k+1$ vertices and $\overline{G}$ is formed by $k+1$ nonadjacent triangles and $2(q-k)-1$ nonadjacent edges, as Figure 3.2 shows.

In the next two sections, we will prove the following theorem.

THEOREM 3.1. $EX(n;TK_p) = \{\mathcal{H}(n;TK_p)\}$ *for* $\lceil\frac{2n+6}{3}\rceil \le p < n-3$.

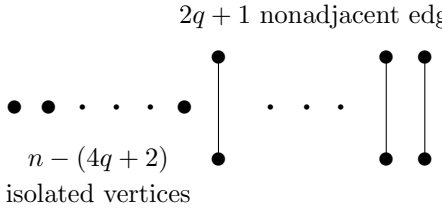


FIG. 3.1. *Structure of* $\overline{G}$ *for* $n \geq 4q+2$.

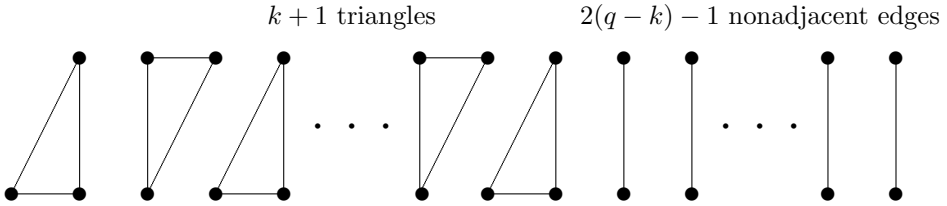


FIG. 3.2. *Structure of* $\overline{G}$ *for* $n = 4q - k + 1$.

4. Case $\lceil \frac{3n+2}{4} \rceil \leq p < n-3$. The aim of this section is to prove Theorem 3.1 when n and p are related by the expression $\lceil \frac{3n+2}{4} \rceil \leq p < n-3$.

PROPOSITION 4.1. *Let* n *and* p *be two positive integers such that* $\lceil \frac{3n+2}{4} \rceil \leq p < n-3$. *It is verified that*

$$EX(n; TK_p) = \{\mathcal{H}(n; TK_p)\}.$$

In order to provide this proposition, we need some previous results. First, we recall the following results about the function $ex(n; TK_{n-q})$ (see [2]).

THEOREM 4.2 (see [2]). *Let* n *and* q *be two positive integers. If* $n \geq 4q+2$, *then*

$$ex(n; TK_{n-q}) = \binom{n}{2} - (2q+1).$$

Also, we recall that, given a graph H and $v \in H$, the set of vertices adjacent to v in H is denoted by $\Gamma(v)$ (see [1]). Given a bipartite graph B whose classes are X and Y with $|X| \leq |Y|$, we say that B has a complete matching if there exists a set of nonadjacent edges in B with cardinality $|X|$. If we need to show the existence of a complete matching in a bipartite graph, then we can use Hall's condition.

THEOREM 4.3 (see [5]). *Given a bipartite graph with classes* X *and* Y, *if* $|\Gamma(A)| \geq |A|$ *for all* $A \subseteq X$, *where* $\Gamma(A) = \bigcup_{v \in A} \Gamma(v)$, *then there exists a complete matching.*

The next result asserts that for any graph $G \in EX(n; TK_{n-q})$ its complement graph $\overline{G}$ is extremal for $\mathcal{C}_q^{q+1}$ in the sense that $\overline{G} \in \mathcal{C}_q^{q+1}$ and $\overline{G} \notin \mathcal{C}_q^q$.

LEMMA 4.4. *Let* n *and* q *be two nonnegative integers with* $q \geq 4$ *and* $n \geq 4q+2$. *For every graph* G *from the family of graphs* $EX(n; TK_{n-q})$, *we have*

$$\overline{G} \in \mathcal{C}_q^{q+1} - \mathcal{C}_q^q.$$

Proof. Let G be a graph such that $G \in EX(n; TK_{n-q})$. The graph G does not contain a subgraph homeomorphic to K_{n-q}, so by Theorem 4.2, we know that

$$|E(G)| = \binom{n}{2} - (2q+1).$$

Hence, $|E(H)| = 2q + 1$, where $H = \overline{G}$.

By Lemma 2.1, there exists an integer s such that $H \in \mathcal{C}_q^s$. This means that there exists a subset $\{v_1, \ldots, v_q\}$ of vertices of G verifying $|E(H_q)| \leq s$, where $H_q = H - \{v_1, \ldots, v_q\}$. If $s \leq q + 1$, then $H \in \mathcal{C}_q^{q+1}$. Otherwise ($s > q + 1$), let H^* be the graph obtained from H by removing one of the edges of the subgraph H_q. The graph H^* has $n \geq 4q+2$ vertices and $2q$ edges, and applying Lemma 2.2 results in $H^* \in \mathcal{C}_q^q$. Furthermore, by the construction of the graph H^*, the set of vertices chosen to prove that H^* belongs to the class of graphs $\mathcal{C}_q^q$ is the same as the one we chose previously in H; thus $|E(H_q)| \leq q + 1$ and $H \in \mathcal{C}_q^{q+1}$.

Now we will prove that the number of edges of H_q may not be equal to or less than q, i.e., $H \notin \mathcal{C}_q^q$. Suppose that $H \in \mathcal{C}_q^q$. This means there exists a set of vertices $\{v_1, \ldots, v_q\}$ guaranteeing this assertion. Let $e_1 = (a_1, b_1), \ldots, e_s = (a_s, b_s)$ be the edges of H_q with $1 \leq s \leq q$.

We consider the bipartite graph B whose classes are $X = \{e_1, \ldots, e_s\}$ and $Y = \{v_1, \ldots, v_q\}$ such that e_i is adjacent to v_j in B if the path $a_i v_j b_i$ exists in G. We note that if there exists a complete matching in B, then we have that G contains a subgraph homeomorphic to K_{n-q}. Now Hall's condition implies the existence of a complete matching. Thus, we will prove that $|\Gamma(A)| \geq |A|$ for each $A \subseteq X$.

Let $A = \{e_i\}$ be a subset of X with $|A| = 1$ for $i \in \{1, \ldots, s\}$. If $|\Gamma(A)| = 0$, then e_i is nonadjacent to any vertex of the set $\{v_{q-2}, v_{q-1}, v_q\}$ in B. Hence, no vertex $v \in \{v_{q-2}, v_{q-1}, v_q\}$ is adjacent to both a_i and b_i in G. Consequently, $\delta_{H_{q-1}}(a_i) \geq 2$ or $\delta_{H_{q-1}}(b_i) \geq 2$ and, furthermore, $\delta_{H_{q-3}}(a_i) \geq 3$ or $\delta_{H_{q-3}}(b_i) \geq 3$. Thus, using property (2) of the definition of $\mathcal{C}_q^q$, we obtain that $\delta_{H_{j-1}}(v_j) \geq 3$ for $j = 1, \ldots, q - 2$ and $\delta_{H_{j-1}}(v_j) \geq 2$ for $j = q - 1, q$. Therefore, since $s \geq 1$ we have that

$$|E(H)| \geq 3(q - 2) + 2 \cdot 2 + s \geq 2q + 2$$

for $q \geq 3$. But this is not possible since $|E(H)| = 2q + 1$.

We consider $A = \{e_i, e_j\} \subseteq X$ for $i, j \in \{1, \ldots, s\}$ with $i \neq j$, and we suppose $|\Gamma(A)| \leq 1$. This means that at least three vertices of the set $\{v_{q-3}, v_{q-2}, v_{q-1}, v_q\}$ are nonadjacent to e_i and to e_j in B. Taking into account property (2) of the definition of $\mathcal{C}_q^q$, we have that $\delta_{H_{j-1}}(v_j) \geq 3$ for $j = 1, \ldots, q - 3$, $\delta_{H_{j-1}}(v_j) \geq 2$ for $j = q - 2, q - 1$ and $\delta_{H_{q-1}}(v_q) \geq 1$ (see Figure 4.1). Hence,

$$|E(H)| \geq 3(q - 3) + 2 \cdot 2 + 1 + s \geq 2q + 2$$

for $q \geq 4$, and this is a contradiction, as in the previous case.

Let m be an integer with $3 \leq m \leq s$. Let A be the set of vertices $\{e_{i_1}, \ldots, e_{i_m}\} \subseteq \{e_1, \ldots, e_s\}$ with $i_1 < i_2 < \cdots < i_m$. If $|\Gamma(A)| \leq m - 1$, then there

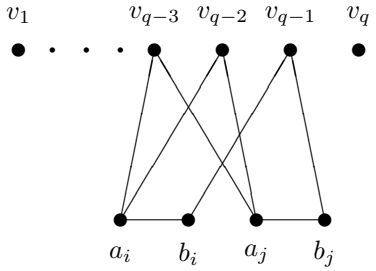


FIG. 4.1. *Possible structure of H for the most unfavorable case for $A = \{e_i, e_j\}$.*

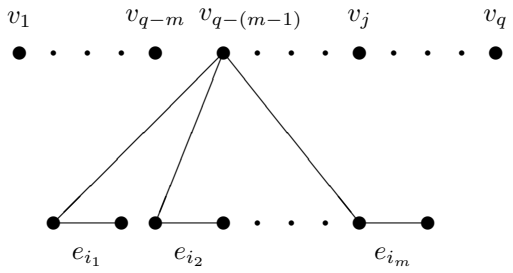


FIG. 4.2. *Possible structure of H for the most unfavorable case for $3 \leq m \leq s$.*

exists $i \in \{q-(m-1), \ldots, q\}$ in such a way that v_i is not adjacent to any vertex of the set A in the graph B. By applying condition (2) of the definition of $\mathcal{C}_q^q$, we obtain that $\delta_{H_{q-m}}(v_{q-(m-1)}) \geq m$ and, therefore, $\delta_{H_{j-1}}(v_j) \geq m$ for $1 \leq j \leq q-(m-1)$ (see Figure 4.2). Furthermore, $\delta_{H_{j-1}}(v_j) \geq 1$ for $q-(m-2) \leq j \leq q$ and $|E(H_q)| = s \geq m$. Consequently,

$$\begin{aligned} |E(H)| &\geq m(q-(m-1)) + m - 1 + s \\ &\geq mq - m^2 + 3m - 1. \end{aligned}$$

Since $E(H) = 2q+1$, we have that $2q+1 \geq mq - m^2 + 3m - 1$ and, therefore, $q \leq \frac{m^2-3m+2}{m-2} \leq m-1 < m \leq s$, but this is not possible. Therefore, $|\Gamma(A)| \geq |A|$ for each $A \subseteq X$. Thus, by Hall's condition, there exists a complete matching in B and, thereby, the graph G contains a subgraph homeomorphic to K_{n-q}. This is not possible, and the result follows. □

Now we can prove Proposition 4.1.

Proof of Proposition 4.1. It is equivalent to prove that

$$EX(n; TK_{n-q}) = \{\mathcal{H}(n; TK_{n-q})\}$$

for $q \geq 4$ and $n \geq 4q+2$.

Let G be a graph belonging to $\{\mathcal{H}(n; TK_{n-q})\}$ with $n \geq 4q+2$. It is easy to check that G does not contain a subgraph homeomorphic to K_{n-q}. Furthermore, by denoting $|E(G)|$ as the number of edges of G, we have that

$$|E(G)| = ex(n; TK_{n-q}) = \binom{n}{2} - (2q+1).$$

Thus, by Theorem 4.2, G is maximal on edges and

$$\{\mathcal{H}(n; TK_{n-q})\} \subseteq EX(n; TK_{n-q}).$$

In order to prove that $EX(n; TK_{n-q}) \subseteq \{\mathcal{H}(n; TK_{n-q})\}$, let G be a graph belonging to $EX(n; TK_{n-q})$, and we set $H = \overline{G}$. By Theorem 4.2 we have that $|E(H)| = 2q+1$. By Lemma 2.1, we know there exists s such that $H \in \mathcal{C}_q^s$. Let $\{v_1, \ldots, v_q\}$ be a set of q vertices guaranteeing this property. We know that there exists a vertex $v \in H_q$ such that $\delta_{H_q}(v) \geq 1$, because otherwise H_q is empty and $H \in \mathcal{C}_q^q$. But this is not possible because, by Lemma 4.4, we know that $H \notin \mathcal{C}_q^q$. If $\delta(v_1) \geq 2$, then $|E(H_q)| \leq 2q+1-(2+q-1) = q$ and therefore $H \in \mathcal{C}_q^q$, a contradiction. Therefore, $\delta(v_1) \leq 1$.

Thus, as v_1 is the vertex of maximum degree in H, we have that $\delta(v) \leq 1$ for all $v \in H$, and then the graph H is formed by $2q+1$ nonadjacent edges. Therefore, the result follows. □

5. Case $\lceil \frac{2n+6}{3} \rceil \leq p < \lceil \frac{3n+2}{4} \rceil$. In this section, we will characterize the family of extremal graphs $EX(n;TK_{n-q})$ for $n = 4q-k+1$ with $0 \leq k \leq q-5$ in such a way that we will show that $EX(n;TK_{n-q}) = \{\mathcal{H}(n;TK_{n-q})\}$, applying techniques based on the same ideas as in the previous section.

THEOREM 5.1. *Let n and p be two positive integers with $\lceil \frac{2n+6}{3} \rceil \leq p < \lceil \frac{3n+2}{4} \rceil$. Then*

$$EX(n;TK_p) = \{\mathcal{H}(n;TK_p)\}.$$

In order to prove this result, we also need to recall some results about the function $ex(n;TK_{n-q})$ (see [2]).

LEMMA 5.2 (see [2]). *Let k be a nonnegative integer and H be a graph with maximum degree 2 and at least $3k+1$ vertices of maximum degree. Then there exist at least $k+1$ nonadjacent vertices with degree 2.*

THEOREM 5.3 (see [2]). *Let n, k, and q be three nonnegative integers with $0 \leq k \leq q-4$ and $n = 4q-k+1$. It is verified that*

$$ex(n;TK_{n-q}) = \left(\begin{array}{c} n \\ 2 \end{array} \right) - (2q+k+2).$$

Now we will show, as in Lemma 4.4, that if $G \in EX(n;TK_{n-q})$ with $n = 4q-k+1$, then $\overline{G} \in \mathcal{C}_q^{q+1}$ but $\overline{G} \notin \mathcal{C}_q^q$.

LEMMA 5.4. *Let k, n, and q be three nonnegative integers such that $q \geq 5$, $0 \leq k \leq q-5$, and $n = 4q-k+1$. If $G \in EX(n;TK_{n-q})$, then*

$$\overline{G} \in \mathcal{C}_q^{q+1} - \mathcal{C}_q^q.$$

Proof. Let G be a graph belonging to $EX(n;TK_{n-q})$. This graph does not contain a graph homeomorphic to K_{n-q}, and by Theorem 5.3 we know that

$$|E(G)| = \left(\begin{array}{c} n \\ 2 \end{array} \right) - (2q+k+2).$$

Thus, $H = \overline{G}$ has $2q+k+2$ edges.

Let H^* be the graph obtained from H by removing one edge, similar to what we have done in Lemma 4.4. Since H^* is a graph formed by $4q-k+1$ vertices and $2q+k+1$ edges, then applying Lemma 2.3 yields $H^* \in \mathcal{C}_q^q$, and then

$$H \in \mathcal{C}_q^{q+1}.$$

Now we will show that $H \notin \mathcal{C}_q^q$. To the contrary, suppose $H \in \mathcal{C}_q^q$ and let $\{v_1, \ldots, v_q\}$ be a set of vertices of H guaranteeing that $H \in \mathcal{C}_q^q$. Let $e_1 = (a_1, b_1), \ldots, e_s = (a_s, b_s)$ be the edges of H_q with $s \leq q$. We consider the bipartite graph B constructed as in Lemma 4.4, i.e., the graph whose classes are $X = \{e_1, \ldots, e_s\}$ and $Y = \{v_1, \ldots, v_q\}$ in such a way that e_i is adjacent to v_j if the path $a_i v_j b_i$ exists in the graph G. In this case, if we show the existence of a complete matching in B, then we would have that G contains a subgraph homeomorphic to K_{n-q}. Therefore, we will show that $|\Gamma(A)| \geq |A|$ for each $A \subseteq X$.

If $|A| = m = 1$, by reasoning as in the proof of Lemma 4.4, we have that

$$|E(H)| \geq 3(q-2) + 4 + s = 3q + s - 2 \geq 3q - 1.$$

Since $k \leq q-4$, it is verified that $3q-1 \geq 2q+k+4-1 > 2q+k+2$, but this is not possible.

For $m=2$, by considering as done previously, we have that

$$|E(H)| \geq 3(q-3)+4+1+s = 3q-4+s \geq 3q-2.$$

Taking into account that $k \leq q-5$, it is verified that $|E(H)| > 2q+k+2$, and this is a contradiction.

We consider $m=3$. Let $A=\{e_{i_1}, e_{i_2}, e_{i_3}\}$ be a subset of vertices of X with $1 \leq i_1 < i_2 < i_3 \leq s$. If $|\Gamma(A)| \leq 2$, then there exists $i \in \{q-2, \ldots, q\}$ in such a way that v_i is not adjacent to any vertex of the set A in the graph B. Hence, by applying property (2) of the definition of $\mathcal{C}_q^q$, we have that $\delta_{H_{q-3}}(v_{q-2}) \geq 3$. Thus,

$$|E(H)| \geq 3(q-2)+2+s \geq 3q-1 > 2q+k+2$$

since $k \leq q-4$.

In general, if $4 \leq m \leq s$, then we consider A as the set of vertices $\{e_{i_1}, \ldots, e_{i_m}\} \subseteq \{e_1, \ldots, e_s\}$ with $i_1 < i_2 < \cdots < i_m$. If $|\Gamma(A)| \leq m-1$, then there exists $i \in \{q-(m-1), \ldots, q\}$ in such a way that v_i is not adjacent to any vertex of the set A in the graph B. Hence, as in the proof of Lemma 4.4, we have that $\delta_{H_{q-m}}(v_{q-(m-1)}) \geq m$ and, therefore,

$$|E(H)| \geq m(q-(m-1))+m-1+s \geq mq-m^2+3m-1.$$

But $|E(H)| = 2q+k+2 \leq 3q-3$ for $k \leq q-5$. Thus, $3q-3 \geq mq-m^2+3m-1$ and, thereby, $q \leq m-\frac{2}{m-3} < m$, but this is not possible.

Thus, using Hall's condition, there exists a complete matching in B, and consequently, G contains a subgraph homeomorphic to K_{n-q}, but this is not possible. Hence, $H \notin \mathcal{C}_q^q$ and the result follows. □

The next result is devoted to proving the existence of nonadjacent triangles in graphs with maximum degree 2 and the prescribed number of vertices of maximum degree.

LEMMA 5.5. *Let r be a nonnegative integer, and let H be a graph with maximum degree 2. If H has $3r+3$ vertices of degree 2 and $r+1$ of them form an independent set, then H contains $r+1$ nonadjacent triangles.*

Proof. We apply induction on r. For $r=0$ the result is obvious, because the triangle is the unique graph formed by 3 vertices of degree 2 and all of them are adjacent among themselves.

Now suppose that $r+1 \geq 2$ and the result holds for r. Let H be a graph with $3(r+1)+3 = 3(r+2)$ vertices of degree 2, and let $w_1, \ldots, w_{r+2}$ be $r+2$ nonadjacent vertices of H.

If there exist $i,\ j \in \{1, \ldots, r+2\}$ with $i \neq j$ such that $\Gamma(w_i) \cap \Gamma(w_j) \neq \emptyset$, then $|\bigcup_{k=1}^{r+2}\{\Gamma(w_k) \cup w_k\}| < 3(r+2)$. Thus, there exists $w \in H$ with degree 2 nonadjacent to w_i for all i. Hence, $\{w, w_1, \ldots, w_{r+2}\}$ is a set of $r+3$ nonadjacent vertices of degree 2, but this is a contradiction. Therefore, $\Gamma(w_i) \cap \Gamma(w_j) = \emptyset$ for all $i \neq j$. Furthermore, if $w \in H$ is adjacent to any w_i for $i \in \{1, \ldots, r+2\}$, then w has degree 2; otherwise, since the number of vertices of degree 2 is $3(r+2)$, there exists $v \in H$ with degree 2 nonadjacent to w_i for all i, and we have seen above that this is not possible.

Now, let a and b be the vertices adjacent to w_{r+2}. If the edge (a,b) does not belong to H, we have that $\{w_1, \ldots, w_{r+1}, a, b\}$ is a set of $r+3$ nonadjacent vertices of degree 2. Thus, the vertices w_1, a, and b form a triangle.

Denote by H^* the graph obtained from H, removing the previous triangle. Therefore, H^* is a graph with $3r+3$ vertices of degree 2, and $r+1$ of them are nonadjacent; by induction hypothesis, H^* contains $r+1$ nonadjacent triangles. Thus, H contains $r+2$ nonadjacent triangles. □

To finish this section, we give the proof of Theorem 5.1, using the previous results.

Proof of Theorem 5.1. It is equivalent to show that

$$EX(n;TK_{n-q}) = \{\mathcal{H}(n;TK_{n-q})\}$$

for $n = 4q-k+1$ with $q \geq 5$, $0 \leq k \leq q-5$.

Let G be a graph belonging to the set $\{\mathcal{H}(n;TK_{n-q})\}$. By checking the structure of this graph G, it is easy to prove that G does not contain a subgraph homeomorphic to K_{n-q}. Since $|E(G)| = ex(n;TK_{n-q}) = \binom{n}{2} - (2q+k+2)$, we have that $G \in EX(n;TK_{n-q})$.

In order to show that $EX(n;TK_{n-q}) \subseteq \{\mathcal{H}(n;TK_{n-q})\}$, let G be a graph belonging to $EX(n;TK_{n-q})$. We denote by $H = \overline{G}$. By Theorem 5.3, $|E(H)| = 2q+k+2$. First, we will prove that $\Delta(H) \leq 2$. Suppose the contrary, that $\Delta(H) \geq 3$.

By applying Lemma 5.4, we have $H \in \mathcal{C}_q^{q+1} - \mathcal{C}_q^q$. Hence, there exists a subset of vertices $\{v_1,\ldots,v_q\}$ of H guaranteeing this property. Furthermore, $|E(H_q)| = q+1$. We claim there exists $j \in \{1,\ldots,q\}$ such that $\Delta(H_{j-1}) \geq 3$ and $\Delta(H_j) \leq 2$, because otherwise we have $\delta_{H_{i-1}}(v_i) \geq 3$ for each $1 \leq i \leq q$, and

$$|E(H)| \geq 3q + (q+1) > 2q+k+2,$$

but this is not possible. Now we distinguish the cases $j \geq k+1$ and $j \leq k$.

For $j \geq k+1$, we consider the fact that $\Delta(H_{j-1}) \geq 3$ and $\Delta(H_j) \leq 2$. Taking into account property (2) of the definition of $\mathcal{C}_q^{q+1}$ and $|E(H_q)| > 0$, we have $\delta_{H_{i-1}}(v_i) \geq 3$ for $1 \leq i \leq j$ and $\delta_{H_{i-1}}(v_i) \geq 1$ for $j+1 \leq i \leq q$. Hence,

$$|E(H_q)| \leq 2q+k+2-(3j+(q-j)) \leq q-j+1 \leq q.$$

But this is not possible since $|E(H_q)| = q+1$.

For $j \leq k$, we have that $\delta_{H_{i-1}}(v_i) \geq 3$ for $1 \leq i \leq j$. If $\Delta(H_k) \leq 1$, then $2|E(H_k)| \leq |V(H_k)|$ and

$$4q-2k+1 = |V(H_k)| \geq 2|E(H_k)| \geq 2(q-k+q+1) = 4q-2k+2,$$

and this is a contradiction. Thus, $\Delta(H_k) = 2$ and $\delta_{H_{i-1}}(v_i) \geq 2$ for $j+1 \leq i \leq k$. Hence,

$$|E(H_q)| \leq 2q+k+2-(3j+2(k-j+1)+(q-k+1)) = q-j+1 \leq q,$$

and this not possible. Thus, $\Delta(H) \leq 2$.

Since $2|E(H)| > |V(H)|$, we have $\Delta(H) \geq 2$ and, consequently, $\Delta(H) = 2$.

Next we are going to study the structure of H. On the one hand, if H has at least $3(k+1)+1$ vertices of degree 2, then by Lemma 5.2 we have that $k+2$ of those vertices $\{w_1,\ldots,w_{k+2}\}$ are nonadjacent. Let $w_{k+3},\ldots,w_q$ be $q-(k+2)$ vertices of H such that the set $\{w_1,\ldots,w_{k+2},w_{k+3},\ldots,w_q\}$ verifies properties (1) and (2) of the definition of $\mathcal{C}_q^s$. For this set of vertices, we have that

$$|E(H_q)| \leq 2q+k+2-(2(k+2)+q-(k+2)) = q,$$

and therefore, $H \in \mathcal{C}_q^q$, a contradiction. Thus, H has at most $3k+3$ vertices of degree 2. On the other hand, if we denote by n_i the number of vertices of degree i in H, we have that

$$\left.\begin{aligned} 2n_2 + n_1 &= 2(2q+k+2) \\ n_2 + n_1 + n_0 &= 4q - k + 1 \end{aligned}\right\}.$$

Thus, $n_2 = 3k + 3 + n_0 \geq 3k + 3$ and the number of vertices of degree 2 in H is $n_2 = 3k + 3$.

Furthermore, as we have shown previously, H may not have $k+2$ nonadjacent vertices of degree 2. Since H has $3k+3 \geq 3k+1$ vertices of degree 2, by Lemma 5.2 we have that H has at least $k+1$ nonadjacent vertices. Hence, H has maximum degree 2 and $3k+3$ vertices of degree 2, and $k+1$ of them are nonadjacent. Therefore, by applying Lemma 5.5, H contains $k+1$ nonadjacent triangles. Additionally, $n_0 = 0$, $n_1 = 4q - 4k - 2$, and the result follows. □

Acknowledgment. The authors thank the referees for their helpful comments and suggestions.

REFERENCES

[1] B. Bollobas, *Extremal Graph Theory*, Academic Press, London, 1978.
[2] M. Cera, A. Diánez and A. Márquez, *The size of a graph without topological complete subgraphs*, SIAM J. Discrete Math., 13 (2000), pp. 295–301.
[3] G. A. Dirac, *In abstrakten Graphen vorhandene vollständige* 4*-Graphenund ihre Unterteilungen*, Math. Nachr., 22 (1960), pp. 61–85.
[4] G. A. Dirac, *Homeomorphism theorem for Graphs*, Math. Ann., 153 (1964), pp. 69–80.
[5] P. Hall, *On representatives of subsets*, J. London Math. Soc., 10 (1935), pp. 26–30.
[6] W. Mader, *Homomorphieegenshaften und mittlere Kantendichte von Graphen*, Math. Ann., 174 (1967), pp. 265–268.
[7] W. Mader, *Hinreichende Bedingungen für die Existenz von Teilgraphen, diezu einem vollständigen Graphen Homömorph sind*, Math. Nachr., 53 (1972), pp. 145–150.
[8] W. Mader, *Graphs without a Subdivision of K_5 of Maximum Size*, preprint, 1998.
[9] W. Mader, $3n-5$ *edges do force a subdivision of* K_5, Combinatorica, 18 (1998), pp. 569–595.
[10] C. Thomassen, *Some homomorphism properties of graphs*, Math. Nachr., 64 (1974), pp. 119–133.

SIAM J. DISCRETE MATH.
Vol. 18, No. 2, pp. 397–417

NONCROSSING PARTITIONS FOR THE GROUP D_n*

CHRISTOS A. ATHANASIADIS† AND VICTOR REINER‡

Dedicated to the memory of Rodica Simion

Abstract. The poset of noncrossing partitions can be naturally defined for any finite Coxeter group W. It is a self-dual, graded lattice which reduces to the classical lattice of noncrossing partitions of $\{1, 2, \ldots, n\}$ defined by Kreweras in 1972 when W is the symmetric group S_n, and to its type B analogue defined by the second author in 1997 when W is the hyperoctahedral group. We give a combinatorial description of this lattice in terms of noncrossing planar graphs in the case of the Coxeter group of type D_n, thus answering a question of Bessis. Using this description, we compute a number of fundamental enumerative invariants of this lattice, such as the rank sizes, number of maximal chains, and Möbius function.

We also extend to the type D case the statement that noncrossing partitions are equidistributed to nonnesting partitions by block sizes, previously known for types A, B, and C. This leads to a (case-by-case) proof of a theorem valid for all root systems: the noncrossing and nonnesting subspaces within the intersection lattice of the Coxeter hyperplane arrangement have the same distribution according to W-orbits.

Key words. noncrossing partition, nonnesting partition, reflection group, root poset, antichain, Catalan number, Narayana numbers, type D, Garside structure

AMS subject classifications. Primary, 06A07; Secondary, 05A18, 05E15, 20F55

DOI. 10.1137/S0895480103432192

1. Introduction and results.

The lattice $NC^A(n)$ of noncrossing partitions is a well-behaved and well-studied subposet inside the lattice $\Pi(n)$ of partitions of the set $[n] := \{1, 2, \ldots, n\}$. It consists of all set partitions π of $[n]$ such that if $a < b < c < d$ and a, c are contained in a block B of π while b, d are contained in a block B' of π, then $B = B'$. The lattice of noncrossing partitions arises naturally in such diverse areas of mathematics as combinatorics, discrete geometry, representation theory, group theory, probability, combinatorial topology, and mathematical biology; see the survey [21] by Simion. This paper concerns analogues of this lattice for Coxeter groups and, specifically, for the Coxeter group of type D_n.

Such analogues were suggested for the Coxeter groups of types B_n and D_n in [20] and were shown to have enumerative and order theoretic properties similar to those of $NC^A(n)$. Reiner [20, section 6] asked for a natural definition of the lattice of noncrossing partitions for any finite Coxeter group W. Although the main idea may be described as folklore (cf. [7]), only fairly recently, and in particular after the work of Bessis [4] and Brady and Watt [12], it has become apparent that such a definition is both available and useful. More precisely, for $u, w \in W$, let $u \leq w$ if there is a shortest factorization of u as a product of reflections in W which is a prefix of such a shortest factorization of w. This partial order turns W into a graded poset T^W having the identity 1 as its unique minimal element, where the rank of w is the length of the shortest factorization of w into reflections. Let γ be a Coxeter element of W. Since all Coxeter elements in W are conjugate to each other, the interval $[1, \gamma]$ in T^W

*Received by the editors July 20, 2003; accepted for publication (in revised form) April 20, 2004; published electronically Deceber 9, 2004.

http://www.siam.org/journals/sidma/18-2/43219.html

†Department of Mathematics, University of Crete, 71409 Heraklion, Crete, Greece (caa@math.uoc.gr).

‡School of Mathematics, University of Minnesota, Minneapolis, MN 55455 (reiner@math.umn.edu).

is independent, up to isomorphism, of the choice of γ. We denote this interval by NC^W or by NC^{X_n}, where X_n is the Cartan–Killing type of W. The poset NC^W plays a crucial role in the construction of new monoid structures and $K(\pi,1)$ spaces for Artin groups associated with finite Coxeter groups [4, 11, 12] and shares many of the fundamental properties of $NC^A(n)$. For instance, it is self-dual [4, section 2.3] and graded and has been verified case-by-case to be a lattice [4, Fact 2.3.1]; see also [12, section 4].

In the case of the symmetric group it is known that the poset $NC^{A_{n-1}}$ is isomorphic to the lattice $NC^A(n)$ of noncrossing partitions; see, for instance, [6, 7, 11]. Similarly, in the case of the hyperoctahedral group, the poset NC^{B_n} is isomorphic to the type B analogue $NC^B(n)$ of $NC^A(n)$ proposed in [20]; see [4, 9, 12]. However, it was observed in [12, section 4] that NC^{D_n} is *not* isomorphic to the type D analogue of $NC^A(n)$ suggested in [20]. Bessis [4, section 4.2] asked for an explicit description of the elements of NC^{D_n} as noncrossing planar graphs, similar to those which appear in the definition of $NC^B(n)$. We give such a description in section 3. Using a construction similar to that of $NC^B(n)$, we define a poset $NC^D(n)$ which we suggest as the type D analogue of $NC^A(n)$ and prove the following theorem.

THEOREM 1.1. *The poset NC^{D_n} is isomorphic to $NC^D(n)$.*

In particular, this gives a different proof that the poset NC^{D_n} is indeed a lattice [12, Theorem 4.14]; see Proposition 3.1. We should mention that, independently, Bessis and Corran [5] have generalized this construction to a class of complex reflection groups that contains D_n.

In our next main result we compute some basic enumerative invariants of NC^{D_n}. Throughout, we use the convention that $\binom{n}{k} = 0$ unless $k \in \{0,1,2,\ldots,n\}$.

THEOREM 1.2. (i) *The number of elements of NC^{D_n} of rank k is equal to the type D Narayana number*

$$\begin{aligned}\mathrm{Nar}(D_n,k) &= \binom{n}{k}^2 - \frac{n}{n-1}\binom{n-1}{k}\binom{n-1}{k-1} \\ &= \binom{n}{k}\left(\binom{n-1}{k} + \binom{n-2}{k-2}\right).\end{aligned}$$

In particular, the total number of elements of NC^{D_n} is equal to the type D Catalan number

$$\mathrm{Cat}\,(D_n) = \binom{2n}{n} - \binom{2n-2}{n-1}.$$

(ii) *More generally, for any composition $s = (s_1, s_2, \ldots, s_m)$ of the number n, the number of chains from the minimum to the maximum element in NC^{D_n} with successive rank jumps $s_1, s_2, \ldots, s_m$ is equal to*

$$2\binom{n-1}{s_1}\cdots\binom{n-1}{s_m} + \sum_{i=1}^{m}\binom{n-1}{s_1}\cdots\binom{n-2}{s_i-2}\cdots\binom{n-1}{s_m}.$$

(iii) *The zeta polynomial of NC^{D_n} is given by*

$$Z(NC^{D_n},m) = 2\binom{m(n-1)}{n} + \binom{m(n-1)}{n-1}.$$

(iv) *In particular, NC^{D_n} has $2(n-1)^n$ maximal chains, and has Möbius function between the minimum and maximum element equal to*

$$(-1)^n \left(2\binom{2n-2}{n} - \binom{2n-3}{n-1} \right).$$

The Narayana and Catalan numbers which appear in part (i) of Theorem 1.2 can be defined for any finite Coxeter group; see [2, 3] for a number of interesting combinatorial and algebraic-geometric interpretations. It is known [16, 20] that the number of elements of a given rank in NC^W and the total number of elements are equal to the corresponding Narayana and Catalan numbers, respectively, in the cases of types A and B. Thus part (i) of the theorem extends this fact to the case of type D. The statement on the cardinality of NC^{D_n} is also claimed to have been checked by Picantin [19]. We note that the type D analogue of $NC^A(n)$ suggested in [20] has the same cardinality and rank sizes as NC^{D_n} [20, Corollary 10] but different zeta polynomial, number of maximal chains, and Möbius function.

Our definition of the poset $NC^D(n)$ leads naturally to a notion of "block sizes" for noncrossing partitions of type D (see section 2). Such a notion was already suggested in [1] for nonnesting partitions for the classical root systems, which are other families of combinatorial objects counted by the corresponding Catalan numbers; see [1], [20, Remark 2], [25, Exercise 6.19 (uu)], and section 2. Our next result refines Theorem 1.2(i) and extends to the case of type D the main result of [1], stating that noncrossing and nonnesting partitions are equidistributed by block sizes for each of the classical root systems of types A, B, and C.

THEOREM 1.3. *Let λ be a partition of $n-m$ with k parts, where $m \geq 0$, and let $m_\lambda = r_1! \cdot r_2! \cdots$, where r_i is the number of parts of λ equal to i. The numbers of noncrossing or nonnesting partitions of type D_n with block sizes λ are equal to each other and are given by the formula*

$$\begin{cases} \dfrac{(n-1)!}{m_\lambda\,(n-k-1)!} & \text{if } \ m \geq 2, \\ (r_1 + 2(n-k))\ \dfrac{(n-1)!}{m_\lambda\,(n-k)!} & \text{if } \ m = 0. \end{cases}$$

Note that the type D analogue of $NC^A(n)$ proposed in [20] fails to preserve this similarity between noncrossing and nonnesting partitions [1, section 6].

Finally, we show that this equidistribution of noncrossing and nonnesting partitions for the classical types A, B, C, D leads to a case-by-case proof of a result (Theorem 6.3) valid for all (finite, crystallographic) root systems: there are embeddings of the sets of noncrossing and nonnesting partitions into the intersection lattice Π^W of the Coxeter hyperplane arrangement, and the two distributions according to W-orbits coincide.

This paper is organized as follows. Section 2 collects the necessary background and definitions related to the Coxeter group of type D_n, noncrossing partitions, and nonnesting partitions. In particular, the poset NC^{D_n} is explicitly described. We also include a few enumerative results from [1] which are used in the following sections. Theorem 1.1 is proved in section 3 after the poset $NC^D(n)$ is defined. Theorem 1.2 is proved in section 4 using Theorem 1.1 and bijective methods similar to those employed in [13, 20] in the case of $NC^A(n)$ and $NC^B(n)$. Theorem 1.3 is proved in section 5. Section 6 describes the embeddings of the sets of noncrossing and nonnesting partitions into the intersection lattice Π^W and proves Theorem 6.3 on the equidistribution of their W-orbits. Section 7 concludes with a few remarks.

2. Background and definitions. This section includes notation, definitions, and some basic background related to Coxeter groups as well as noncrossing and nonnesting partitions of types B and D.

We will mostly follow notation introduced in [1, 12, 20]. We refer the reader to the texts by Humphreys [15] and Stanley [24] for any undefined terminology related to Coxeter groups and partially ordered sets, respectively. Throughout the paper we let

$$[n] := \{1, 2, \ldots, n\},$$
$$[n]^{\pm} := \{-1, -2, \ldots, -n, 1, 2, \ldots, n\}$$

for any positive integer n.

The Coxeter group D_n. Let S_{2n} denote the symmetric group on the set $[n]^{\pm}$. For any cycle $c = (i_1, i_2, \ldots, i_k)$ in S_{2n}, we let $\bar{c} = (-i_1, -i_2, \ldots, -i_k)$. If c is the transposition (i, j) and $i \neq -j$, we denote by $(\!(i, j)\!)$ the product $c\bar{c} = (i, j)(-i, -j)$ and call $(\!(i, j)\!)$ a *D_n-reflection*, or simply a *reflection*. The Coxeter group W^{D_n} is the subgroup of S_{2n} generated by the reflections $(\!(i, j)\!)$. Any element of W^{D_n} can be expressed uniquely (up to reordering) as a product of disjoint cycles

$$c_1\bar{c}_1 \cdots c_k\bar{c}_k\, d_1 \cdots d_r, \tag{2.1}$$

each having at least two elements, where $\bar{d}_j = d_j$ for $j = 1, 2, \ldots, r$ and r is even; see, for instance, [12, Proposition 3.1]. Following [12], for a cycle $c = (i_1, i_2, \ldots, i_k)$ in S_{2n} we write

$$(\!(i_1, i_2, \ldots, i_k)\!) = c\bar{c} = (i_1, i_2, \ldots, i_k)(-i_1, -i_2, \ldots, -i_k)$$

and call $c\bar{c}$ a *paired cycle* if c is disjoint from $\bar{c}$. We also write

$$c = [i_1, i_2, \ldots, i_k]$$

if $c = \bar{c} = (i_1, \ldots, i_k, -i_1, \ldots, -i_k)$ and call c a *balanced cycle*. Note that $[i]$ denotes both the balanced cycle $(i, -i)$ and the set $\{1, 2, \ldots, i\}$. We will leave it to the reader to decide which notation is meant each time, hoping that no confusion will arise.

For $w \in W^{D_n}$ we denote by $l(w)$ the minimum number r for which w can be written as a product of r reflections and call it the *length* of w. (Note: this is *not* the usual Coxeter group length function, which is defined with respect to the *simple* reflections as generating set.) The cycle $(\!(i_1, i_2, \ldots, i_k)\!)$ has length $k - 1$. The length of any element of W^{D_n} in the form (2.1) can be written as a sum over its paired and balanced cycles, where the contribution of $(\!(i_1, i_2, \ldots, i_k)\!)$ and $[i_1, i_2, \ldots, i_k]$ to this sum is $k - 1$ and k, respectively [12, section 3]. We denote by T^{D_n} the partial order on the set W^{D_n} defined by letting $u \leq w$ if $l(w) = l(u) + l(u^{-1}w)$. The poset T^{D_n} is graded by length and has the identity element 1 as its unique minimal element. For a choice γ of a Coxeter element of W^{D_n}, which we fix as $\gamma = [1, 2, \ldots, n-1][n]$ for convenience, we denote by NC^{D_n} the interval $[1, \gamma]$ in the poset T^{D_n}. The poset NC^{D_n} is a self-dual, graded lattice of rank n [4, section 2], [12, section 4], where the rank function is the restriction of the rank function from T^{D_n}.

Noncrossing partitions. A *B_n-partition* is a partition π of the set $[n]^{\pm}$ into blocks such that (i) if B is a block of π, then its negative $-B$ is also a block of π, and (ii) there is at most one block, called the *zero block* if present, which contains both i and $-i$ for some i. The *type* of π is the integer partition λ which has a part equal to the

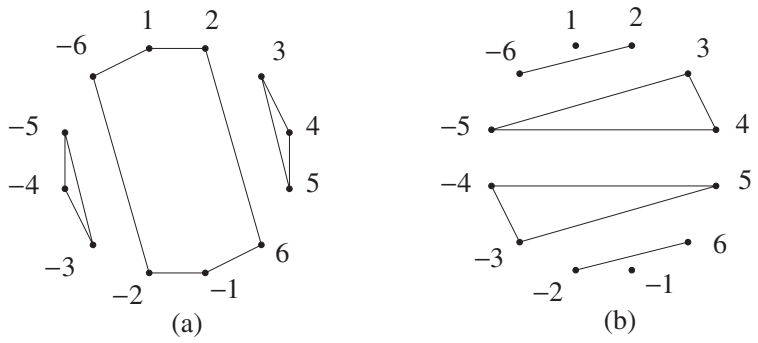


FIG. 1. *Two elements of* $NC^B(n)$ *for* $n = 6$ *with blocks* (a) $\{3,4,5\}$, $\{-3,-4,-5\}$, $\{1,2,6,-1,-2,-6\}$ *and* (b) $\{3,4,-5\}$, $\{-3,-4,5\}$, $\{2,-6\}$, $\{-2,6\}$, $\{1\}$, $\{-1\}$.

cardinality of B for each pair $\{B,-B\}$ of nonzero blocks of π. Thus λ is a partition of $n-m$, where m is half the size of the zero block of π, if present, and $m=0$ otherwise. We refer to the parts of λ as the *block sizes* of π. A *D_n-partition* is a B_n-partition π with the additional property that the zero block of π, if present, does not consist of a single pair $\{i,-i\}$. The set of all B_n-partitions, ordered by refinement, is denoted by $\Pi^B(n)$. Its subposet consisting of all D_n-partitions is denoted by $\Pi^D(n)$. The posets $\Pi^B(n)$ and $\Pi^D(n)$ are geometric lattices which are isomorphic to the intersection lattices of the B_n and D_n Coxeter hyperplane arrangements, respectively, and hence they can be considered as type B and D analogues of the partition lattice $\Pi(n)$. In particular they are graded of rank n, and the corank of an element π in either poset is the number of pairs $\{B,-B\}$ of nonzero blocks of π.

Let us label the vertices of a convex $2n$-gon as $1,2,\ldots,n,-1,-2,\ldots,-n$ clockwise, in this order. Given a B_n-partition π and a block B of π, let $\rho(B)$ denote the convex hull of the set of vertices labeled with the elements of B. We call π *noncrossing* if $\rho(B)$ and $\rho(B')$ have void intersection for any two distinct blocks B and B' of π. Two noncrossing partitions are depicted in Figure 1 for $n=6$. The subposet of $\Pi^B(n)$ consisting of the noncrossing B_n-partitions is a self-dual, graded lattice of rank n which is denoted by $NC^B(n)$ [20, section 2].

Nonnesting partitions. Let $e_1,e_2,\ldots,e_n$ be the unit coordinate vectors in $\mathbb{R}^n$ and let Φ be a root system of one of the types B_n, C_n, or D_n. In what follows, we identify Φ with its type X_n and fix the choices

$$\Phi^+ = \begin{cases} \{e_i \pm e_j : 1 \le i < j \le n\} & \text{if } \Phi = D_n, \\ D_n^+ \bigcup \{e_i : 1 \le i \le n\} & \text{if } \Phi = B_n, \\ D_n^+ \bigcup \{2e_i : 1 \le i \le n\} & \text{if } \Phi = C_n \end{cases}$$

of positive roots for Φ. The *root poset* of Φ is the set Φ^+ of positive roots partially ordered by letting $\alpha \le \beta$ if $\beta - \alpha$ is a nonnegative linear combination of the elements of Φ^+. An *antichain* in Φ^+ is a subset of Φ^+ consisting of pairwise incomparable elements.

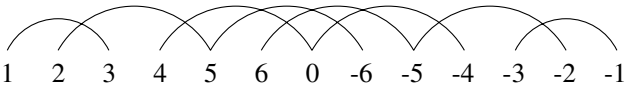


FIG. 2. *A picture of the B_6-nonnesting partition with blocks $\{1,3\}$, $\{-1,-3\}$, $\{2,5,-6\}$, $\{-2,-5,6\}$, and $\{4,-4\}$.*

FIG. 3. *A picture of the D_5-nonnesting partition with blocks $\{2,4\}$, $\{-2,-4\}$, and $\{1,3,5,-1,-3,-5\}$.*

Given an antichain A in Φ^+, we define an equivalence relation on the set $[n]^{\pm}\cup\{0\}$ if $\Phi = B_n$ or C_n, and on the set $[n]^{\pm}$ if $\Phi = D_n$, as follows. For $1 \le i < j \le n$, let

$$\begin{aligned} i \sim j \text{ and } -i \sim -j &\text{ if } e_i - e_j \in A, \\ i \sim -j \text{ and } -i \sim j &\quad \text{if } e_i + e_j \in A. \end{aligned}$$

Moreover, in the cases $\Phi = B_n$ or C_n, let

$$i \sim 0 \sim -i \quad \text{if } e_i \in A \text{ or } 2e_i \in A, \text{ respectively.}$$

Let $\pi_0(A)$ be the set of equivalence classes of the transitive closure of $\sim$. Let $\pi(A)$ be the partition of $[n]^{\pm}$ obtained from $\pi_0(A)$ by removing 0 from its class if $\Phi = B_n$ or C_n, and let $\pi(A) = \pi_0(A)$ if $\Phi = D_n$. Observe that $\pi(A)$ is a B_n-partition. Moreover, in the case $\Phi = D_n$, $\pi(A)$ has a zero block if and only if A contains both $e_i - e_n$ and $e_i + e_n$ for some $i < n$ and hence, in this event, the zero block contains $\{n,-n\}$ and at least one more pair $\{i,-i\}$. Thus in general $\pi(A)$ is a Φ-partition, where a C_n-partition is defined to be the same as a B_n-partition. A Φ-*nonnesting partition* is a Φ-partition of the form $\pi(A)$ for some antichain A in Φ^+. We denote by NN^{Φ} the set of Φ-nonnesting partitions and refer the reader to [1, section 2] and Figures 2 and 3 for the motivation behind the terminology "nonnesting," suggested by Postnikov [1], [20, Remark 2]. By definition, NN^{Φ} is in bijection with the set of antichains in the root poset Φ^+.

Block size enumeration. For an integer partition λ, we denote by $NC^B_{\lambda}(n)$ the set of elements of $NC^B(n)$ of type λ. Similarly, for $\Phi = B_n, C_n$, or D_n we denote by NN^{Φ}_{λ} the set of Φ-nonnesting partitions of type λ. The following theorem is the main result of [1].

THEOREM 2.1 (see [1]). *Let λ be a partition of $n-m$ with k parts, where $m \ge 0$, and let $m_\lambda = r_1! \cdot r_2! \cdots$, where r_i is the number of parts of λ equal to i.*

(i)

$$\#NC^B_{\lambda}(n) = \frac{n!}{m_\lambda\,(n-k)!}.$$

(ii) *The same formula holds for Φ-nonnesting partitions if $\Phi = B_n$ or C_n:*

$$\#NN^{B_n}_{\lambda} = \#NN^{C_n}_{\lambda} = \frac{n!}{m_\lambda\,(n-k)!}.$$

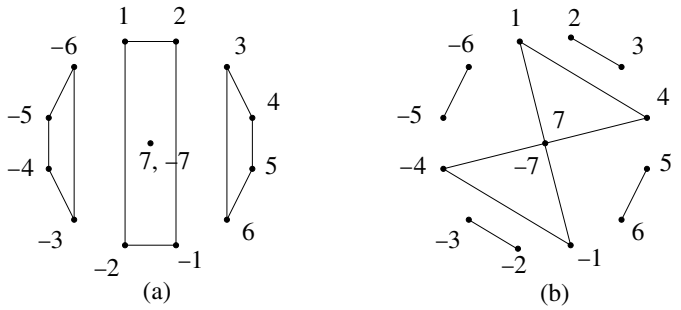


FIG. 4. *Two elements of $NC^D(n)$ for $n = 7$ with blocks* (a) $\{3,4,5,6\}$, $\{-3,-4,-5,-6\}$, $\{1,2,7,-1,-2,-7\}$ *and* (b) $\{1,4,7\}$, $\{-1,-4,-7\}$, $\{2,3\}$, $\{-2,-3\}$, $\{5,6\}$, $\{-5,-6\}$.

(iii) *For $m \geq 2$,*

$$\#NN_{\lambda}^{D_n} = \frac{(n-1)!}{m_\lambda\,(n-k-1)!}.$$

3. Noncrossing partitions of type *D*. In this section we define our type D analogue of the noncrossing partition lattice $NC^A(n)$ and prove Theorem 1.1. Let us label the vertices of a regular $(2n-2)$-gon as $1,2,\ldots,n-1,-1,-2,\ldots,-(n-1)$ clockwise, in this order, and label its centroid with both n and $-n$. Given a D_n-partition π and a block B of π, let $\rho(B)$ denote the convex hull of the set of points labeled with the elements of B. Two distinct blocks B and B' of π are said to *cross* if $\rho(B)$ and $\rho(B')$ do not coincide and one of them contains a point of the other in its relative interior. Observe that the case $\rho(B) = \rho(B')$, which we have allowed, can occur only when B and B' are the singletons $\{n\}$ and $\{-n\}$, and that if π has a zero block B, then B and the block containing n cross unless $\{n,-n\} \subseteq B$.

The poset $NC^D(n)$ is defined as the subposet of $\Pi^D(n)$ consisting of those D_n-partitions π with the property that no two blocks of π cross. Figure 4 shows two elements of $NC^D(n)$ for $n = 7$, one with a zero block and one with no zero block. Figure 5 shows the Hasse diagram of $NC^D(n)$ for $n = 3$.

PROPOSITION 3.1. *The poset $NC^D(n)$ is a graded lattice of rank n in which the corank of π is equal to the number of pairs $\{B,-B\}$ of nonzero blocks of π.*

Proof. Since $NC^D(n)$ is finite with a maximum and minimum element, to prove that it is a lattice, it suffices to show that meets in $NC^D(n)$ exist. Indeed, given elements x, y of $NC^D(n)$, one can check that the meet z of x and y in $\Pi^D(n)$ is an element of $NC^D(n)$ and hence z is also the meet of x and y in $NC^D(n)$.

As was the case for $NC^B(n)$ [20, Proposition 2], the rest of the proposition follows from the observation that given any two elements $\pi_1 \leq \pi_2$ of $NC^D(n)$, there exists a maximal chain in the interval $[\pi_1, \pi_2]$ of $\Pi^D(n)$ which passes only through elements of $NC^D(n)$, so that the grading of $NC^D(n)$ is inherited from that of $\Pi^D(n)$. □

To prove Theorem 1.1 we need to describe the covering relations in the posets T^{D_n} and $NC^D(n)$. In the case of T^{D_n}, the result of multiplying any element of D_n with a reflection $((i,j))$ is described explicitly in [12, Example 3.6]. From the computations given there we can conclude that y covers x in T^{D_n} if and only if x can be obtained from y by replacing one or two balanced cycles of y or one paired cycle of y with one

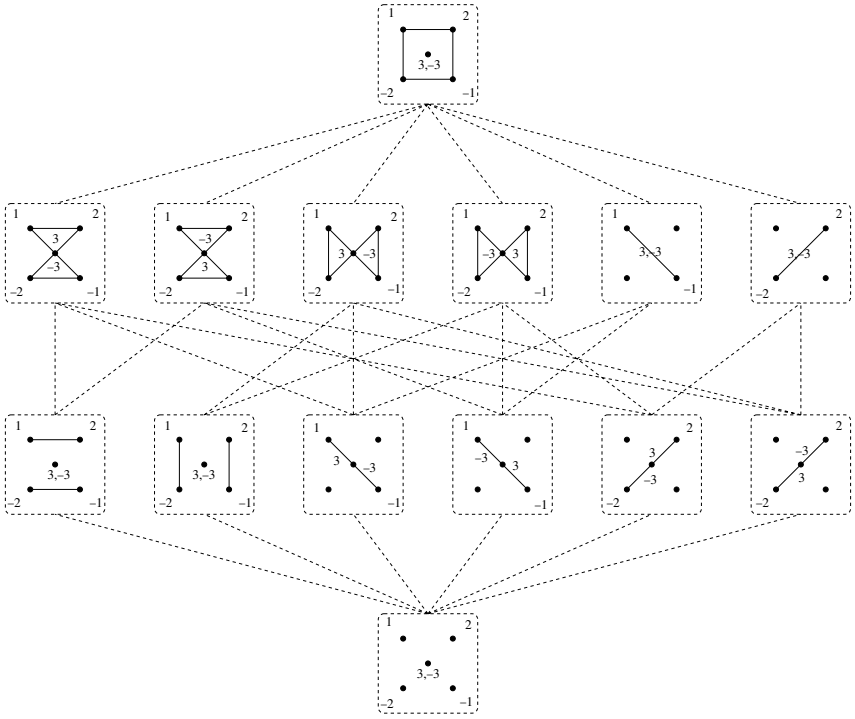


FIG. 5. *The lattice $NC^D(n)$ for $n = 3$.*

or more cycles as follows:

$$
\begin{aligned}
&[i_1, i_2, \ldots, i_k] \longrightarrow [i_1, \ldots, i_j]\,(\!(i_{j+1}, \ldots, i_k)\!),\\
&(\!(i_1, i_2, \ldots, i_k)\!) \longrightarrow (\!(i_1, \ldots, i_j)\!)\,(\!(i_{j+1}, \ldots, i_k)\!),\\
&[i_1, \ldots, i_j]\,[i_{j+1}, \ldots, i_k] \longrightarrow (\!(i_1, i_2 \ldots, i_k)\!).
\end{aligned}
\tag{3.1}
$$

In the case of $NC^D(n)$, it follows directly from the definition that y covers x in $NC^D(n)$ if and only if x can be obtained from y by one of the following:

(i) splitting the zero block of y into the zero block of x and a pair $\{B, -B\}$ of nonzero blocks,
(ii) splitting a pair of nonzero blocks $\{B, -B\}$ of y into two such pairs of x, or
(iii) splitting the zero block of y into one pair $\{B, -B\}$ of nonzero blocks of x (so that one of $B, -B$ contains n and the other contains $-n$).

Proof of Theorem 1.1. For $x \in NC^{D_n}$, let $f(x)$ denote the partition of $[n]^{\pm}$ whose nonzero blocks are formed by the paired cycles of x, and whose zero block is the union of the elements of all balanced cycles of x if such exist. We first observe that $f(x) \in NC^D(n)$. Indeed, this is clear if x is the top element $\gamma = [1, 2, \ldots, n-1]\,[n]$ of NC^{D_n}. If not, then x is covered by some element y of NC^{D_n} and we may assume, by induction on the corank of x, that y has either zero or two balanced cycles, one of which must be $[n]$ in the latter case, and that $f(y) \in NC^D(n)$. Since x can be obtained from y by one of the moves in the list (3.1), it follows with a case-by-case check that x has zero or two balanced cycles as well, one of which must be $[n]$ in the latter case, and that $f(x) \in NC^D(n)$. The map

$$f : NC^{D_n} \to NC^D(n)$$

is thus well-defined and order-preserving, since $f(x) \le f(y)$ in $NC^D(n)$ follows from (3.1) when x is covered by y in NC^{D_n}.

To define the inverse map, for $x \in NC^D(n)$, let $g(x)$ be the element of NC^{D_n}

- whose paired cycles are formed by the nonzero blocks of x, each ordered with respect to the cyclic order

$$-1, -2, \ldots, -n, 1, 2, \ldots, n, -1,$$

  and
- whose balanced cycles are $[n]$ and the cycle formed by the entries of the zero block of x other than n and $-n$, ordered in the same way, if the zero block is present in x.

We claim that $g(x) \in NC^{D_n}$. This is clear if x is the top element of $NC^D(n)$. If not, let y be any element of $NC^D(n)$ which covers x. We may assume, by induction on the corank of x, that $g(y) \in NC^{D_n}$, in other words, that $g(y) \le \gamma$ holds in T^{D_n}. It follows from the possible types of covering relations in the posets $NC^D(n)$ and T^{D_n} that $g(y)$ covers $g(x)$ in T^{D_n}. This implies that $g(x) \le \gamma$ holds in T^{D_n} or, equivalently, that $g(x) \in NC^{D_n}$. Thus the map

$$g : NC^D(n) \to NC^{D_n}$$

is well-defined and order-preserving, since $g(x) \le g(y)$ in NC^{D_n} follows from (3.1) when x is covered by y in $NC^D(n)$. Since f and g are clearly inverses of each other, they are poset isomorphisms. □

4. The zeta polynomial and chain enumeration. In this section we use bijective methods similar to those employed in [13, 20] for $NC^A(n)$ and $NC^B(n)$ to prove Theorem 1.2. We first recall a few constructions from [20, section 3]. After setting

$$P_n^B := \{(L, R) : L, R \subseteq [n], \#L = \#R\},$$

a map $\tau^B : P_n^B \to NC^B(n)$ is constructed in [20, section 3] as follows. Given $x = (L, R) \in P_n^B$, place a left parenthesis before each occurrence of i and $-i$ in the infinite cyclic sequence

$$\ldots, -1, -2, \ldots, -n, 1, 2, \ldots, n, -1, -2, \ldots \tag{4.1}$$

for $i \in L$ and a right parenthesis after each occurrence of i and $-i$ for $i \in R$. Let the strings of integers inside the lowest level matching pairs of parentheses form blocks of $\tau^B(x)$. Remove these lowest level parentheses from (4.1) and the integers they enclose and continue similarly with the remaining parenthesization until all parentheses have been removed. The remaining integers, if any, form the zero block of $\tau^B(x)$. We have the following proposition.

PROPOSITION 4.1 (see [20, Proposition 6]). *The map τ^B is a bijection from the set P_n^B to $NC^B(n)$. Moreover, for any pair $x = (L, R) \in P_n^B$, the number of pairs $\{B, -B\}$ of nonzero blocks of $\tau^B(x)$ is equal to $\#R$.*

To extend the previous proposition to the type D case, let

$$P_n^D = P_{n-1}^B \bigcup \{(L, R, \varepsilon) : L, R \subseteq [n-1], \#R = \#L + 1,\ \varepsilon = \pm 1\}.$$

For $x \in P_n^D$, we define a partition $\pi = \tau^D(x) \in \Pi^D(n)$ as follows. If $x \in P_{n-1}^B$, then π is the partition obtained from $\tau^B(x)$ by adding n and $-n$ to the zero block of $\tau^B(x)$,

if such a block exists, and by adding the singletons $\{n\}$ and $\{-n\}$ to $\tau^B(x)$ otherwise. Suppose that x is not in P^B_{n-1}, say, $x = (L, R, \varepsilon)$. We parenthesize the infinite cyclic sequence

$$\dots, -1, -2, \dots, -(n-1), 1, 2, \dots, n-1, -1, -2, \dots \tag{4.2}$$

as in the type B case and form blocks of π with the same procedure, until a right parenthesis remains after each occurrence of i and $-i$ for a unique $i \in [n-1]$. Then let B and $-B$ be blocks of π, where B consists of the integers in

$$\{-i-1, \dots, -n+1, 1, 2, \dots, i\}$$

which have not been removed from the infinite sequence together with n or $-n$, if $\varepsilon = 1$ or $\varepsilon = -1$, respectively. For instance, if $n = 9$, $L = \{2, 5, 6\}$, $R = \{1, 3, 7, 8\}$, and $\varepsilon = -1$, then π has blocks $\{2, 3\}$, $\{5, 8\}$, $\{6, 7\}$, $\{1, -4, -9\}$, and their negatives.

It is clear from the previous construction that $\pi \in NC^D(n)$; thus we have a well-defined map $\tau^D : P^D_n \to NC^D(n)$.

PROPOSITION 4.2. *The map τ^D is a bijection from the set P^D_n to $NC^D(n)$. Moreover, for any $x \in P^D_n$, the number of pairs $\{B, -B\}$ of nonzero blocks of $\tau^D(x)$ is equal to*

$$\begin{cases} \#R & \text{if } x \in P^B_{n-1} \text{ and } \tau^B(x) \text{ has a zero block,} \\ \#R+1 & \text{if } x \in P^B_{n-1} \text{ and } \tau^B(x) \text{ has no zero block,} \\ \#R & \text{if } x \notin P^B_{n-1}. \end{cases}$$

Proof. The inverse of τ^D can be defined as in the proof of [20, Proposition 6] for the map τ^B. More precisely, given $\pi \in NC^D(n)$, find a nonzero block B of π such that the elements of $B \backslash \{n, -n\}$ form a nonempty, consecutive string of integers in the sequence (4.2). If B does not contain n or $-n$, then place the absolute values of the first and last element of B, with respect to (4.2), in L and R, respectively. If it does, then place the absolute value i of the last element of $B \backslash \{n, -n\}$, with respect to (4.2), in R and let $\varepsilon = 1$ or $\varepsilon = -1$ if n or $-n$ is in the same block as i, respectively. Remove the elements of B and $-B$ from π and (4.2) and continue similarly until the zero block, or the singletons $\{n\}$ and $\{-n\}$, or no block of π remains. We leave it to the reader to check that this map is indeed the inverse of τ^D. The second statement is obvious. □

It is shown in [20, Proposition 7] that the bijection τ^B of Proposition 4.1 extends to a bijection from the set

$$P^B_{n,m} = \left\{ (L, R_1, \dots, R_{m-1}) : L, R_j \subseteq [n], \sum_{j=1}^{m-1} \#R_j = \#L \right\}$$

to the set of multichains $\pi_1 \le \pi_2 \le \cdots \le \pi_{m-1}$ in $NC^B(n)$. This bijection is defined as follows. Given $(L, R_1, \dots, R_{m-1}) \in P^B_{n,m}$, place a left parenthesis before each occurrence of i and $-i$ in the infinite cyclic sequence (4.1) for $i \in L$ and a right parenthesis labeled $)^j$ after each occurrence of i and $-i$ for $i \in R_j$. Observe that more than one right parenthesis with different labels may have been placed after some integers in (4.1). In this case order these right parentheses as

$$)^{j_1}\ )^{j_2}\ \cdots\ )^{j_t},$$

where $j_1 < j_2 < \cdots < j_t$. Read this parenthesization as in the case of the map τ^B to obtain $\pi_1 \in NC^B(n)$. Next, remove from the parenthesization all right parentheses labeled $)^1$ and their corresponding left parentheses to obtain $\pi_2 \in NC^B(n)$, and continue the process until all parentheses have been removed to obtain the multichain $\pi_1 \leq \pi_2 \leq \cdots \leq \pi_{m-1}$.

The type D analogue of this construction is given in the following proposition. To state it we introduce the following notation. Think of a multichain $\pi_1 \leq \pi_2 \leq \cdots \leq \pi_{m-1}$ in $\Pi^B(n)$ as a multichain from $\hat{0}$ to $\hat{1}$ which has m steps; in other words, set $\pi_0 := \hat{0}$ and $\pi_m := \hat{1}$. The *rank jump vector* for such a multichain c is the composition $s = (s_1, \ldots, s_m)$ of the number n (denoted $s \models n$) defined by $s_i := \mathrm{r}(\pi_i) - \mathrm{r}(\pi_{i-1})$. There is a unique step i at which a zero block is first created, meaning that π_{i-1} has no zero block but π_i does. Define $\mathrm{ind}(c)$ to be this index i.

PROPOSITION 4.3. *The bijection τ^D extends to a bijection from the union $P^D_{n,m}$ of $P^B_{n-1,m}$ with the set*

$$\left\{ (L, R_1, \ldots, R_{m-1}, \varepsilon) : L, R_j \subseteq [n-1], \sum_{j=1}^{m-1} \#R_j = \#L + 1,\ \varepsilon = \pm 1 \right\}$$

to the set of multichains $\pi_1 \leq \pi_2 \leq \cdots \leq \pi_{m-1}$ in $NC^D(n)$. Moreover, for $x \in P^D_{n,m}$, one has that

- *if $x \notin P^B_{n-1,m}$ and $x = (L, R_1, \ldots, R_{m-1}, \varepsilon)$, then the multichain in $NC^D(n)$ corresponding to x has rank jump vector*

$$s = (n - 1 - \#L, \#R_1, \ldots, \#R_{m-1}),$$

- *if $x \in P^B_{n-1,m}$ and the multichain c in $NC^B(n-1)$ corresponding to x under the generalized map τ^B has rank jump vector $s = (s_1, \ldots, s_m)$ and $\mathrm{ind}(c) = i$, then the multichain in $NC^D(n)$ corresponding to x has rank jump vector*

$$(s_1, \ldots, s_{i-1}, s_i + 1, s_{i+1}, \ldots, s_m).$$

Proof. Given $x \in P^D_{n,m}$, we construct a multichain $\pi_1 \leq \pi_2 \leq \cdots \leq \pi_{m-1}$ in $NC^D(n)$ as follows. If $x \in P^B_{n-1,m}$, let $\pi'_1 \leq \pi'_2 \leq \cdots \leq \pi'_{m-1}$ be the multichain in $NC^B(n-1)$ corresponding to x under the bijection of [20, Proposition 7]. Let π_i be the partition obtained from π'_i by adding n and $-n$ to the zero block of π'_i if such a block exists, and by adding the singletons $\{n\}$ and $\{-n\}$ to π'_i otherwise. It is then clear that $\pi_i \in NC^D(n)$ and that $\pi_1 \leq \pi_2 \leq \cdots \leq \pi_{m-1}$ is a multichain in $NC^D(n)$. Suppose now that x is not in $P^B_{n-1,m}$, say, $x = (L, R_1, \ldots, R_{m-1}, \varepsilon)$. Place a left parenthesis before each occurrence of i and $-i$ in the infinite cyclic sequence (4.2) for $i \in L$ and a right parenthesis labeled $)^j$ after each occurrence of i and $-i$ for $i \in R_j$, using the same rules as in the type B case described earlier for placing multiple right parentheses. Read this parenthesization as in the case of the map τ^D to obtain $\pi_1 \in NC^D(n)$. Observe that the singletons $\{n\}$ and $\{-n\}$ may be blocks of π_1 if $m - 1 \geq 2$. Next, remove from the parenthesization all right parentheses labeled $)^1$ and their corresponding left parentheses, if any, to obtain $\pi_2 \in NC^D(n)$, and continue the process until all parentheses have been removed. This results in a multichain $\pi_1 \leq \pi_2 \leq \cdots \leq \pi_{m-1}$ in $NC^D(n)$ in which n belongs to a nonsingleton, nonzero block of π_j for at least one index j.

To define the inverse of this map, let $\pi_1 \leq \pi_2 \leq \cdots \leq \pi_{m-1}$ be a multichain in $NC^D(n)$. Parenthesize the sequence (4.2) by applying the inverse of τ^D to π_{m-1} and

label all right parentheses by $)^{m-1}$. Repeat the process with π_{m-2} and label all right parentheses by $)^{m-2}$, but include neither the new pairs of parentheses that would produce more than one left parenthesis before the occurrence of a single integer in (4.2), nor a new unmatched right parenthesis, if one exists already. Continue similarly with the remaining elements of the multichain to get an element of $P^D_{n,m}$. We leave it again to the reader to check that this map is well-defined and that the two maps are indeed inverses of each other. The "moreover" statement is obvious from the construction. □

In [20, Proposition 7], the bijection from $P^B_{n,m}$ to multichains in $NC^B(n)$ was used to deduce that there are $\binom{n}{s_1}\cdots\binom{n}{s_m}$ chains in $NC^B(n)$ with rank jump vector $s=(s_1,\ldots,s_m)$. In order to perform the analogous chain enumeration for $NC^D(n)$, we will need the following refinement of this type B result, keeping track of the extra statistic $\mathrm{ind}(c)$.

LEMMA 4.4. *Let $s=(s_1,\ldots,s_m)\models n$. Then among the $\binom{n}{s_1}\cdots\binom{n}{s_m}$ chains c in $NC^B(n)$ having rank jump vector s, the fraction of those having $\mathrm{ind}(c)=i$ is equal to $\frac{s_i}{n}$.*

Unlike our other enumerative results, our proof of Lemma 4.4 is not bijective. For this reason, we have relegated it to the appendix.

Proof of Theorem 1.2. In view of Theorem 1.1, it suffices to prove the theorem for the poset $NC^D(n)$ instead.

(i) Clearly, the set P^D_n has

$$\binom{2n-2}{n-1}+2\binom{2n-2}{n}=\binom{2n}{n}-\binom{2n-2}{n-1}$$

elements. Hence the statement on the total number of elements of $NC^D(n)$ follows from the first statement in Proposition 4.2. By an easy computation, the statement on the number of elements of rank k is equivalent to the case $m=2$ in (ii).

(ii) This follows from Proposition 4.3 and Lemma 4.4. The summand $2\binom{n-1}{s_1}\cdots\binom{n-1}{s_m}$ counts the chains coming from $x\in P^D_{n,m}-P^B_{n-1,m}$. Within the summation, the ith term

$$\begin{aligned}&\binom{n-1}{s_1}\cdots\binom{n-2}{s_i-2}\cdots\binom{n-1}{s_m}\\&\quad=\frac{s_i-1}{n-1}\binom{n-1}{s_1}\cdots\binom{n-1}{s_i-1}\cdots\binom{n-1}{s_m}\end{aligned}$$

counts the chains coming from $x\in P^B_{n-1,m}$ that correspond to chains c in $NC^B(n-1)$ with rank jump vector $(s_1,\ldots,s_{i-1},s_i-1,s_{i+1},\ldots,s_m)$ and $\mathrm{ind}(c)=i$.

(iii) Observe (as for $P^B_{n,m}$ in the proof of [20, Proposition 7]) that the set $P^D_{n,m}$ has

$$2\binom{m(n-1)}{n}+\binom{m(n-1)}{n-1}$$

elements, and recall that the value $Z(P,m)$ of the zeta polynomial for a poset P is defined to be the number of multichains in P of cardinality $m-1$. The formula in (iii) for the zeta polynomial of $NC^D(n)$ then follows from Proposition 4.3.

(iv) Both assertions follow from the zeta polynomial calculated in (iii), via [24, Proposition 3.11.1]. □

Finally, we briefly discuss how Theorem 1.2(ii) leads to a nice expression for $F_{NC^D(n)}$, where F_P denotes Ehrenborg's quasi-symmetric function associated with a ranked poset P; we refer the reader to [14, 26] for the definitions.

As mentioned in section 1, the posets $NC^A(n), NC^B(n)$, and $NC^D(n)$ are self-dual by virtue of a result of Bessis [4, section 2.3], stating that the poset NC^W is always self-dual. A case-free proof of a stronger statement, namely, that all intervals of NC^W are self-dual, was outlined by McCammond [17, section 3]. Alternatively, it is easy to check from the explicit descriptions of $NC^A(n), NC^B(n)$, and $NC^D(n)$ that any interval in one of these posets is isomorphic to a Cartesian product of posets lying in the union of these three families; see also [18], [20, Remark 1], and the appendix. Hence their intervals are also self-dual, which implies that the posets themselves are *locally rank-symmetric* and their quasi-symmetric functions are actually *symmetric* functions.

In [27], Stanley used the known explicit expressions for the numbers of chains in $NC^A(n)$ and $NC^B(n)$ with given rank jump vector to compute nice formulas for $F_{NC^A(n)}$ and $F_{NC^B(n)}$ (and to connect them with symmetric group actions on parking functions of types A and B; see also Biane [8]). He proved that

$$F_{NC^A(n)} = \frac{1}{n}\left[t^{n-1}\right] E(t)^n$$

and

$$F_{NC^B(n)} = [t^n]\, E(t)^n,$$

where $E(t) := \prod_{i\geq 1}(1+x_i t)$ and $[t^n]\,\psi(t)$ denotes the coefficient of t^n in a formal power series $\psi(t)$ in the variable t. An equally easy computation (which we omit) shows that Theorem 1.2(ii) is equivalent to the following proposition.

PROPOSITION 4.5. *We have*

$$F_{NC^D(n)} = [t^n]\, E(t)^{n-1}\left(2 + \sum_{i\geq 1} \frac{x_i^2 t^2}{1+x_i t}\right).$$

5. Enumeration by block sizes. For an integer partition λ, let $NC^D_\lambda(n)$ denote the set of elements of $NC^D(n)$ with block sizes λ. To enumerate the elements of $NC^D(n)$ by block sizes we define a map

$$\tau : NC^D(n) \to NC^B(n-1)$$

as follows. Let $\pi \in NC^D(n)$. If π has a zero block B, then simply remove n and $-n$ from B to obtain $\tau(\pi)$. Otherwise n and $-n$ are in distinct blocks B and $-B$ of π. Then either remove B and $-B$ from π if they are singletons, or if not, replace them with the zero block $B \cup (-B) \setminus \{n, -n\}$ to obtain $\tau(\pi)$. It should be clear that $\tau(\pi) \in NC^B(n-1)$.

LEMMA 5.1. *The map $\tau : NC^D(n) \to NC^B(n-1)$ has the following property: if $x \in NC^B_\lambda(n-1)$, with $\lambda \vdash n-m-1$, then the set $\tau^{-1}(x)$ consists of $2m+1$ elements. Moreover, $2m$ of these have type $\lambda \uplus \{m+1\}$, and the remaining element has type*

$$\begin{cases} \lambda \uplus \{1\} & \text{if } m = 0, \\ \lambda & \text{if } m \geq 1. \end{cases}$$

Proof. If $x \in NC^B_\lambda(n-1)$ has no zero block, then $\tau^{-1}(x)$ consists of a single element π, obtained from x by adding the singletons $\{n\}$ and $\{-n\}$. Suppose that x has a zero block B of size $2m$. A partition in $\tau^{-1}(x)$ is obtained either by adding $n, -n$ to the zero block B, or by splitting B into two parts C and $-C$ and replacing B with the pair of blocks $C \cup \{n\}$ and $-C \cup \{-n\}$. There are $2m$ ways to do the latter so that the resulting partition is in $NC^D(n)$. □

COROLLARY 5.2. *If λ is a partition of $n-m$, where $m \geq 0$, then $\#NC^D_\lambda(n)$ is given by the formula in Theorem* 1.3.

Proof. Lemma 5.1 implies that

$$\#NC^D_\lambda(n) = \begin{cases} \#NC^B_\lambda(n-1) & \text{if } m \geq 2, \\ \#NC^B_{\lambda\backslash 1}(n-1) + \sum_{p\geq 2}(2p-2)\ \#NC^B_{\lambda\backslash p}(n-1) & \text{if } m = 0, \end{cases}$$

and the result follows from Theorem 2.1(i). Note that, in the above formula, we interpret $NC^B_{\lambda\backslash p}(n-1)$ as empty for any integer $p \geq 1$ that does not appear as a part of λ. □

In the remainder of this section we show that nonnesting partitions of type D have the same distribution by block sizes as noncrossing partitions of the same type.

Assume that $\lambda \vdash n$, so that $NN^{B_n}_\lambda \subseteq NN^{D_n}_\lambda$, and observe that the inclusion is strict for $n \geq 3$ since $\{e_i + e_n, e_j - e_n\}$ is an antichain in D^+_n for $i < j < n$ but not in B^+_n. Let $\pi \in NN^{D_n}_\lambda$. Since π does not have a zero block, n and $-n$ belong to distinct blocks B and $-B$ of π, respectively. Let π' denote the partition obtained from π by exchanging n and $-n$ in the blocks B and $-B$, and let

$$\sigma : NN^{D_n}_\lambda \to NN^{B_n}_\lambda$$

be defined by

$$\sigma(\pi) = \begin{cases} \pi & \text{if } \pi \in NN^{B_n}, \\ \pi' & \text{otherwise.} \end{cases}$$

One can check directly from the definitions that σ is well-defined. Let $T_\lambda(n)$ be the set of partitions $\pi \in NN^{B_n}_\lambda$ such that if B is the block of π containing n, then $B \setminus \{n\}$ contains both positive and negative elements, and let $T^+_\lambda(n)$, $T^-_\lambda(n)$ be the sets of those $\pi \in NN^{B_n}_\lambda$ for which $B \setminus \{n\}$, if nonempty, contains only positive elements and only negative elements, respectively.

LEMMA 5.3. *Let $\lambda \vdash n$.*

(i) *The map $\sigma : NN^{D_n}_\lambda \to NN^{B_n}_\lambda$ induces a bijection between $NN^{D_n}_\lambda \setminus NN^{B_n}_\lambda$ and $T_\lambda(n)$.*

(ii) *We have*

$$\#T^+_\lambda(n) = \#T^-_\lambda(n) = \sum_{p\geq 1} \#NN^{B_{n-1}}_{\lambda\backslash p}$$

and

$$\#(T^+_\lambda(n) \bigcap T^-_\lambda(n)) = \#NN^{B_{n-1}}_{\lambda\backslash 1}.$$

Proof.

(i) If $\pi \in NN^{D_n}_\lambda \setminus NN^{B_n}_\lambda$, then there exist integers $i < j < n$ such that j and n are in a block B of π, while i and $-n$ are in a different block, which must be $-B$. Thus

$\{i, -j, -n\}$ is contained in a block of π and hence $\{i, -j, n\}$ is contained in a block of $\sigma(\pi)$. This implies that $\sigma(\pi) \in T_\lambda(n)$, so that the map $\sigma : NN_\lambda^{D_n} \setminus NN_\lambda^{B_n} \to T_\lambda(n)$ is well-defined. The inverse map again switches n and $-n$ in a partition in $T_\lambda(n)$ and is checked to be well-defined by reversing the previous argument.

(ii) The first two equalities follow from the fact that either $T_\lambda^+(n)$ or $T_\lambda^-(n)$ is in bijection with the set of B_{n-1}-nonnesting partitions whose type is obtained from λ by removing one of its parts, where the bijection removes the blocks B and $-B$ containing n and $-n$ of an element in $T_\lambda^+(n)$ or $T_\lambda^-(n)$ if these blocks are singletons, or replaces them with the zero block $B \cup (-B)\backslash\{n, -n\}$ if they are not. The last equality is obvious. □

COROLLARY 5.4. *If λ is as in Corollary* 5.2*, then $\#NN_\lambda^{D_n} = \#NC_\lambda^{D_n}$; that is, $\#NN_\lambda^{D_n}$ is given by the formula of Theorem* 1.3.

Proof. For $m \geq 2$ the statement is the content of Theorem 2.1(iii). Suppose that $m = 0$, i.e., $\lambda \vdash n$. Lemma 5.3(i) implies that

$$\#NN_\lambda^{D_n} = \#NN_\lambda^{B_n} + \#T_\lambda(n).$$

Since $T_\lambda(n) = NN_\lambda^{B_n} \setminus (T_\lambda^+(n) \cup T_\lambda^-(n))$, part (ii) of the same lemma gives

$$\#T_\lambda(n) = \#NN_\lambda^{B_n} - \#NN_{\lambda\backslash 1}^{B_{n-1}} - 2\sum_{p\geq 2} \#NN_{\lambda\backslash p}^{B_{n-1}},$$

and the result follows from Theorem 2.1(ii). □

The next corollary also follows from the main result of [3] and the computations in the type D case carried out there in section 5.

COROLLARY 5.5 (see [3, section 5]). *The number of elements of NN^{D_n} with k pairs $\{B, -B\}$ of nonzero blocks is equal to*

$$\binom{n}{k}^2 - \frac{n}{n-1}\binom{n-1}{k}\binom{n-1}{k-1}.$$

Proof. This follows from Corollary 5.4 and Theorem 1.2(i). □

6. Block sizes and root systems. The goal of this section is to generalize the results of [1] and Theorem 1.3 on the classical root systems to an arbitrary (finite, crystallographic) root system (Theorem 6.3). To this end we begin by recalling some facts about noncrossing and nonnesting partitions for arbitrary finite Coxeter groups and root systems.

For a finite Coxeter group (W, S), acting with its natural reflection representation on a Euclidean space V, we denote by Π^W the poset of all subspaces of V which are intersections of reflecting hyperplanes of W, ordered by reverse inclusion. Thus Π^W is a graded (geometric) lattice of rank $\#S$ which is isomorphic to the lattice $\Pi(n)$, $\Pi^B(n)$, or $\Pi^D(n)$, defined in the first two sections, when W has type A_{n-1}, B_n, or D_n, respectively.

There is a natural embedding of the lattice of noncrossing partitions NC^W into Π^W. Recall from section 1 that NC^W is defined to be the interval $[1, \gamma]$ in a certain partial order T^W on the group W, where γ is any Coxeter element of W. It follows from results of Brady and Watt (see [4, Proposition 1.6.4]) that the map

$$\begin{aligned} NC^W &\to \Pi^W, \\ w &\mapsto V^w := \{v \in V : w(v) = v\} \end{aligned}$$

is a rank- and order-preserving embedding.

Now assume that W is the finite Weyl group associated to a crystallographic root system Φ. Let Φ^+ be a choice of positive roots, equipped with the standard root order, and let Π be the corresponding set of simple roots. Let $\mathcal{A}^\Phi$ be the collection of all antichains in Φ^+, meaning subsets of pairwise incomparable elements. It turns out that, like NC^W, the set $\mathcal{A}^\Phi$ has a natural embedding into Π^W, endowing it with a poset structure. The crucial fact needed is a recent result of Sommers [23].

THEOREM 6.1 (see [23, page 1]). *Given an antichain A of positive roots, there exists $w \in W$ such that $w(A) \subseteq \Pi$.*

COROLLARY 6.2. *If Φ is a crystallographic root system with Weyl group W, then the map*

$$\begin{aligned} \mathcal{A}^\Phi &\to \Pi^W, \\ A &\mapsto \bigcap_{\alpha \in A} \alpha^\perp \end{aligned}$$

is an injection, sending A to an element of rank $\#A$.

Proof. The image of A has rank $\#A$ because Theorem 6.1 implies that any antichain in $\mathcal{A}^\Phi$ is linearly independent (since Π is also). That the map is injective will follow from a stronger assertion about the interaction between the linear independence and convexity structure of Φ^+. Given $B \subseteq \Phi^+$, let $\overline{B}$ denote the *matroid closure* of B, meaning the subset of vectors in Φ^+ lying in the linear span of B. Let $\mathrm{ext}(B)$ denote the set of *extreme vectors* within the convex cone spanned by B. We then claim that for an antichain A in Φ^+,

$$A = \mathrm{ext}(\overline{A}).$$

This will show that the map is injective, since one has the alternate characterization of $\overline{A}$ as

$$\overline{A} = \left\{ \beta \in \Phi^+ : \bigcap_{\alpha \in A} \alpha^\perp \subseteq \beta^\perp \right\}.$$

To prove the claim note that, since A is linearly independent, $\mathrm{ext}(\overline{A})$ has at least as many elements as A. Consequently it suffices to show the inclusion $\mathrm{ext}(\overline{A}) \subseteq A$. To this end, given $\beta \in \mathrm{ext}(\overline{A})$, express β (uniquely) as

$$\beta = \sum_{\alpha \in A} c_\alpha \alpha. \tag{6.1}$$

Since β and all elements of A are positive roots, at least one of the coefficients c_α must be positive. Using Theorem 6.1 again, one can find $w \in W$ so that $w(A) \subset \Pi$. As $w(\beta)$ lies in Φ, it has a unique expression in terms of simple roots with all coefficients of the same sign. Hence the expression

$$w(\beta) = \sum_{\alpha \in A} c_\alpha w(\alpha),$$

obtained by applying w to (6.1), forces all of the other coefficients c_α to be nonnegative. Therefore (6.1) shows that β lies in the convex cone spanned by A. Since β is an extreme vector of the larger cone spanned by $\overline{A}$, it is extreme in this smaller cone, so it must lie in A. ∎

We require a notion that generalizes the "block sizes" of a type A, B, or D partition to an arbitrary intersection subspace in Π^W. This is supplied by the orbit map, sending a subspace to its W-orbit:

$$\begin{aligned}\Pi^W &\to \Pi^W/W,\\ U &\mapsto W\cdot U := \{w(U) : w \in W\}.\end{aligned}$$

We can now state the main result of this section.

THEOREM 6.3. *Let W be the Weyl group of a crystallographic root system Φ and consider the two composite maps*

$$\begin{aligned}f : NC^W \hookrightarrow &\Pi^W \to \Pi^W/W,\\ g : \mathcal{A}^\Phi \hookrightarrow &\Pi^W \to \Pi^W/W.\end{aligned}$$

Then for each W-orbit $x \in \Pi^W/W$, we have

$$\#\, f^{-1}(x) = \#\, g^{-1}(x).$$

Proof. The general statement follows from the corresponding statement for irreducible root systems, so one may proceed case-by-case via the classification.

For types A, B, and C, the statement follows from the results of [1] and the fact that the W-orbits of Π^W are precisely the sets of intersection subspaces whose corresponding type A or B partition has given block sizes. In the type D case, a slight complication arises due to the fact that the W-orbit of an intersection subspace is not always determined by the nonzero block sizes λ of its associated D_n-partition. In fact this occurs exactly when λ is a partition of n having only even parts (so that, in particular, n must be even). In this case the set of intersection subspaces in Π^W whose corresponding D_n-partitions have block sizes λ decomposes further into exactly two W-orbits, determined by one extra (parity) piece of data: pick arbitrarily one block B out of each pair $\{B, -B\}$ of blocks in the partition and compute the parity (even or odd) of the total number of negative elements in the union of these blocks. We claim, however, that the preimages under either f or g of two such parity orbits have the same number of elements, so that the result follows from Corollaries 5.2 and 5.4. To check the claim, simply observe that, for a partition λ of n with even parts, the swap of n and $-n$ gives rise to fixed-point free involutions on both $NC^D_\lambda(n)$ and $NN^{D_n}_\lambda$, which switch parity. This is obvious in the case of $NC^D_\lambda(n)$ and should be clear from the discussion preceding Lemma 5.3 in the case of $NN^{D_n}_\lambda$.

The exceptional types E_6, E_7, E_8, F_4, G_2 have been checked one by one with computer calculations, using software in Mathematica available from the second author. ∎

7. Remarks.

1. It would be interesting to find a conceptual, case-free proof of Theorem 6.3.

2. The set of maximal chains in the poset NC^W is in bijection with the set of factorizations of shortest possible length, henceforth called *minimal factorizations*, of a Coxeter element of W into reflections. Hence Theorem 1.2(iv) implies the following statement.

COROLLARY 7.1. *The number of minimal factorizations of a Coxeter element of the group W^{D_n} into reflections is equal to $2(n-1)^n$.*

A direct proof of this fact, analogous to the proofs of the corresponding statements by Biane [8] for the symmetric and hyperoctahedral group, is possible. More

precisely, let $\gamma = [1, 2, \ldots, n-1]\,[n]$ be as in section 2 and let $\mathcal{M}_n$ be the set of tuples $(t_1, t_2, \ldots, t_n)$ of reflections in W^{D_n} such that $\gamma = t_1 t_2 \cdots t_n$. Let us label the reflections in W^{D_n} as follows:

$$\ell(t) = \begin{cases} i & \text{if } \ t = ((i, \pm n)) \ \text{ and } \ 1 \le i \le n-1, \\ i & \text{if } t = ((i, j)) \ \text{ and } \ 1 \le i < j \le n-1, \\ j & \text{if } t = ((i, -j)) \ \text{ and } \ 1 \le i < j \le n-1. \end{cases}$$

It has been shown by the first author that the map which assigns to any element $(t_1, t_2, \ldots, t_n)$ of $\mathcal{M}_n$ the sequence of labels $(\ell(t_1), \ell(t_2), \ldots, \ell(t_n))$ is a two-to-one map from the set $\mathcal{M}_n$ to $[n-1]^n$.

3. It is natural to conjecture that the poset $NC^D(n)$ is shellable. However, the EL-labellings given by Edelman and Björner [10] in the case of $NC^A(n)$ and Reiner [20] in the case of $NC^B(n)$ do not seem to extend to that of $NC^D(n)$.

4. It has been shown by Eleni Tzanaki (private communication) that the poset $NC^D(n)$ has a symmetric chain decomposition analogous to those of $NC^A(n)$ [22, Theorem 2] and $NC^B(n)$ [20, Theorem 13].

Appendix. Proof of Lemma 4.4. We recall the statement of the lemma.

LEMMA 4.4. *Let $s = (s_1, \ldots, s_m) \models n$. Then among the $\binom{n}{s_1} \cdots \binom{n}{s_m}$ chains c in $NC^B(n)$ having rank jump vector s, the fraction of those having $\mathrm{ind}(c) = i$ is equal to $\frac{s_i}{n}$.*

The proof will utilize the type B generalization [20, Theorem 16] of a result of Nica and Speicher [18] on incidence algebras, which we recall here.

Let R be any ring with unit and let P be a poset. The (R-valued) incidence algebra for P consists of R-valued functions f on the set of intervals $[a, b]$ of P, with pointwise addition and multiplication by convolution:

$$(f * g)[a, c] := \sum_{b \in P:\, a \le b \le c} f[a, b]\, g[b, c].$$

In [20, Remark 1] a certain multiplicative subgroup $I^0_{\mathrm{mult}}(NC^B; R)$ of the union of all R-valued incidence algebras of type A and B noncrossing partition lattices was defined. As observed in [18, 20], every interval $[a, b]$ in $NC^A(n)$ or $NC^B(n)$ has a canonical isomorphism to a Cartesian product

$$NC^B(n_0) \times NC^A(n_1) \times NC^A(n_2) \times \cdots \times NC^A(n_r)$$

for some integers $n_0, n_1, \ldots, n_r$, where the factor $NC^B(n_0)$ need not be present. The multiplicative subgroup $I^0_{\mathrm{mult}}(NC^B; R)$ consists of those elements f in the incidence algebra which take the value 1 on $NC^A(1)$ and which are *multiplicative*, in the sense that

$$f[a, b] = f(NC^B(n_0)) \prod_{i=1}^{r} f(NC^A(n_i)).$$

Define a map

$$I^0_{\mathrm{mult}}(NC^B; R) \xrightarrow{\mathcal{F}} R[[t, u]]/(u^2) \ (\cong R[u]/(u^2)[[t]]),$$
$$f \mapsto \mathcal{F}(f) := \frac{\phi_f^{\langle -1 \rangle}}{t},$$

where

$$\phi_f := \sum_{n\geq 1} f(NC^A(n))t^n + f(NC^B(n))t^n u \tag{7.1}$$

and $\phi_f^{\langle -1\rangle}$ denotes the compositional inverse of ϕ_f with respect to the variable t. This map gives an isomorphism of $I^0_{\text{mult}}(NC^B; R)$ onto the multiplicative subgroup of power series in $R[[t,u]]/(u^2)$ whose coefficient of t^0 equals 1 [20, Theorem 16].

Proof of Lemma 4.4. We will perform generating function calculations in these rings, choosing

$$R = \mathbb{Z}[[x_1, x_2, \ldots, y_1, y_2, \ldots]].$$

Define $f \in I^0_{\text{mult}}(NC^B; R)$ by

$$\begin{aligned} f(NC^A(n)) &:= \sum_{s\models n-1} \sum_{\substack{\text{chains in } NC^A(n)\\ \text{with rank jump vector } s}} x^s, \\ f(NC^B(n)) &:= \sum_{s\models n} \sum_{\substack{\text{chains } c \text{ in } NC^B(n)\\ \text{with rank jump vector } s}} x^s \, \frac{y_{\text{ind}(c)}}{x_{\text{ind}(c)}}, \end{aligned} \tag{7.2}$$

where $x^s := x_1^{s_1}\cdots x_m^{s_m}$. A little thought shows that f coincides with the convolution $f = f_1 * f_2 * \cdots$, where

$$\begin{aligned} f_i(NC^A(n)) &:= x_i^{n-1}, \\ f_i(NC^B(n)) &:= x_i^{n-1} y_i. \end{aligned}$$

From this, one calculates that

$$\phi_{f_i} = \sum_{n\geq 1} x_i^{n-1}t^n + x_i^{n-1}y_i t^n u = \frac{t(1+y_i u)}{1-tx_i},$$

and hence, by computing the compositional inverse,

$$\begin{aligned} \phi_{f_i}^{\langle -1\rangle} &= \frac{t}{1+tx_i+uy_i}, \\ \mathcal{F}(f_i) &= \frac{\phi_{f_i}^{\langle -1\rangle}}{t} = \frac{1}{1+tx_i+uy_i}. \end{aligned}$$

Therefore

$$\begin{aligned} \mathcal{F}(f) &= \textstyle\prod_{i\geq 1} \mathcal{F}(f_i) = \prod_{i\geq 1} \frac{1}{1+tx_i+uy_i}, \\ \phi_f^{\langle -1\rangle} &= t\mathcal{F}(f) = \frac{t}{\prod_{i\geq 1}(1+tx_i+uy_i)}. \end{aligned}$$

One can apply the Lagrange inversion formula [25, Theorem 5.4.2] to this last expression. Letting $[t^k]\,\psi(t)$ denote the coefficient of t^k in any formal power series $\psi(t)$ in a

variable t, one has that

$$\begin{aligned}
[t^n]\,\phi_f &= \frac{1}{n}[T^{n-1}]\prod_{i\geq 1}(1+x_iT+y_iu)^n \\
&= \frac{1}{n}[T^{n-1}]\prod_{i\geq 1}\sum_{k=0}^{n}\binom{n}{k}(x_iT+y_iu)^k \\
&= \frac{1}{n}[T^{n-1}]\prod_{i\geq 1}\sum_{k=0}^{n}\binom{n}{k}(x_i^kT^k+k\cdot x_i^{k-1}y_iT^{k-1}u) \\
&= \sum_{s\models n-1}\frac{1}{n}\binom{n}{s_1}\cdots\binom{n}{s_m}x^s + u\sum_{s\models n}\binom{n}{s_1}\cdots\binom{n}{s_m}\sum_{i=1}^{m}\frac{s_i}{n}x^s\frac{y_i}{x_i}.
\end{aligned}$$

Comparing (7.1), (7.2) with this last expression gives the result. □

The proof of the lemma gives an alternative derivation for [13, Theorem 3.2] and, by setting $y_i = x_i$ for all i, also one for [20, Proposition 7].

Acknowledgment. The authors thank Jon McCammond for helpful conversations and for originally bringing references [4, 11, 12] to their attention.

REFERENCES

[1] C. A. Athanasiadis, *On noncrossing and nonnesting partitions for classical reflection groups*, Electron. J. Combin., 5 (1998), Research Paper 42, 16 pp. (electronic).
[2] C. A. Athanasiadis, *Generalized Catalan numbers, Weyl groups and arrangements of hyperplanes*, Bull. London Math. Soc., 36 (2004), pp. 294–302.
[3] C. A. Athanasiadis, *On a refinement of the generalized Catalan numbers for Weyl groups*, Trans. Amer. Math. Soc., 357 (2005), pp. 179–196.
[4] D. Bessis, *The dual braid monoid*, Ann. Sci. École Norm. Sup. (4), 36 (2003), pp. 647–683.
[5] D. Bessis and R. Corran, *Non-crossing partitions of type* (e, e, r), preprint, 2004 (arXiv:math.GR/0403400).
[6] D. Bessis, F. Digne, and J. Michel, *Springer theory in braid groups and the Birman-Ko-Lee monoid*, Pacific J. Math., 205 (2002), pp. 287–309.
[7] P. Biane, *Some properties of crossings and partitions*, Discrete Math., 175 (1997), pp. 41–53.
[8] P. Biane, *Parking functions of types A and B*, Electron. J. Combin., 9 (2002), Note 7, 5 pp. (electronic).
[9] P. Biane, F. Goodman, and A. Nica, *Non-crossing cumulants of type B*, Trans. Amer. Math. Soc., 355 (2003), pp. 2263–2303.
[10] A. Björner, *Shellable and Cohen-Macaulay partially ordered sets*, Trans. Amer. Math. Soc., 260 (1980), pp. 159–183.
[11] T. Brady, *A partial order on the symmetric group and new K(π, 1)'s for the braid groups*, Adv. Math., 161 (2001), pp. 20–40.
[12] T. Brady and C. Watt, *K(π, 1)'s for Artin groups of finite type*, in Proceedings of the Conference on Geometric and Combinatorial Group Theory, Part I (Haifa, 2000), Geom. Dedicata, 94 (2002), pp. 225–250.
[13] P. H. Edelman, *Chain enumeration and noncrossing partitions*, Discrete Math., 31 (1980), pp. 171–180.
[14] R. Ehrenborg, *On posets and Hopf algebras*, Adv. Math., 119 (1996), pp. 1–25.
[15] J. E. Humphreys, *Reflection Groups and Coxeter Groups*, Cambridge Studies in Advanced Mathematics 29, Cambridge University Press, Cambridge, UK, 1990.
[16] G. Kreweras, *Sur les partitions non-croisées d'un cycle*, Discrete Math., 1 (1972), pp. 333–350.
[17] J. McCammond, *Noncrossing partitions in surprising locations*, preprint, University of California–Santa Barbara, Santa Barbara, CA, 2003.
[18] A. Nica and R. Speicher, *A "Fourier transform" for multiplicative functions on non-crossing partitions*, J. Algebraic Combin., 6 (1997), pp. 141–160.

[19] M. Picantin, *Explicit presentations for the dual braid monoids*, C. R. Math. Acad. Sci. Paris, 334 (2002), pp. 843–848.
[20] V. Reiner, *Non-crossing partitions for classical reflection groups*, Discrete Math., 177 (1997), pp. 195–222.
[21] R. Simion, *Noncrossing partitions*, Discrete Math., 217 (2000), pp. 367–409.
[22] R. Simion and D. Ullman, *On the structure of the lattice of non-crossing partitions*, Discrete Math., 98 (1991), pp. 193–206.
[23] E. Sommers, *B-stable ideals in the nilradical of a Borel subalgebra*, Canad. Math. Bull., to appear.
[24] R. P. Stanley, *Enumerative Combinatorics*, Vol. 1, Wadsworth & Brooks/Cole Advanced Books & Software, Monterey, CA, 1986; second printing, Cambridge Studies in Advanced Mathematics 49, Cambridge University Press, Cambridge, UK, 1996.
[25] R. P. Stanley, *Enumerative Combinatorics*, Vol. 2, Cambridge Studies in Advanced Mathematics 62, Cambridge University Press, Cambridge, UK, 1999.
[26] R. P. Stanley, *Flag-symmetric and locally rank-symmetric partially ordered sets*, Electron. J. Combin., 3 (1996), Research Paper 6, 22 pp. (electronic).
[27] R. P. Stanley, *Parking functions and noncrossing partitions*, Electron. J. Combin., 4 (1997), Research Paper 20, 14 pp. (electronic).

SIAM J. DISCRETE MATH.
Vol. 18, No. 2, pp. 418–435

ON THE MULTIPLICITY OF PARTS IN A RANDOM COMPOSITION OF A LARGE INTEGER*

PAWEŁ HITCZENKO† AND CARLA D. SAVAGE‡

Abstract. In this paper we study the following question posed by H. S. Wilf: what is, asymptotically as $n \to \infty$, the probability that a randomly chosen part size in a random composition of an integer n has multiplicity m? More specifically, given positive integers n and m, suppose that a composition λ of n is selected uniformly at random and then, out of the set of part sizes in λ, a part size j is chosen uniformly at random. Let $\mathbb{P}(A_n^{(m)})$ be the probability that j has multiplicity m. We show that for fixed m, $\mathbb{P}(A_n^{(m)})$ goes to 0 at the rate $1/\ln n$. A more careful analysis uncovers an unexpected result: $(\ln n)\mathbb{P}(A_n^{(m)})$ does not have a limit but instead oscillates around the value $1/m$ as $n \to \infty$.

This work is a counterpart of a recent paper of Corteel, Pittel, Savage, and Wilf, who studied the same problem in the case of partitions rather than compositions.

Key words. compositions of an integer, random compositions, geometric random variables

AMS subject classifications. 05A16, 60C05

DOI. 10.1137/S0895480199363155

1. Introduction.

In this paper we consider the multiplicity of a randomly chosen part size in a random composition of an integer n. Let us recall that a multiset $\lambda = \{\lambda_1, \ldots, \lambda_k\}$ is a *partition* of an integer n if the λ_j are positive integers, called *parts*, such that $\sum \lambda_j = n$. *Compositions* are merely partitions in which the order of parts is significant. Thus, for example, the integer 3 admits three partitions, $\{1,1,1\}$, $\{2,1\}$, and $\{3\}$, and four compositions, namely $(1,1,1)$, $(1,2)$, $(2,1)$, and (3).

Integer partitions (as deterministic objects) have been studied for quite some time, but Erdös and Lehner [6] were apparently the first to study integer partitions from the probabilistic perspective; namely, they considered the set of all partitions, $P(n)$, of an integer n, as a probability space equipped with the uniform probability measure. Quantities of interest are treated as random variables, and one can study their probabilistic properties, most typically the limiting properties as $n \to \infty$. Erdös and Lehner, for example, considered the limiting distribution of the total number of parts in a partition. Their paper opened a new line of investigation.

Goh and Schmutz [11] obtained the central limit theorem for the number of different part sizes in a random partition; that is, they proved that the number of different part sizes, appropriately normalized, has, approximately, the standard Gaussian distribution. (Several years earlier, Wilf [18] found an asymptotic formula for the expected number of distinct part sizes.) This approach culminated in an important paper by Fristedt [10], who proved that the joint distribution of the multiplicities of

*Received by the editors November 2, 1999; accepted for publication (in revised form) October 8, 2003; published electronically December 9, 2004. Research supported in part by National Science Foundation grant DMS9622772.

http://www.siam.org/journals/sidma/18-2/36315.html

†Department of Mathematics, Drexel University, Philadelphia, PA 19104 (phitczenko@mcs.drexel.edu).

‡Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206 (savage@cayley.csc.ncsu.edu).

part sizes is that of independent geometric random variables (Y_k), with parameters $(1-p^k)$, conditioned on the event $\{\sum kY_k = n\}$.

Fristedt's work, in turn, opened new possibilities and resulted in further progress in our understanding of the structure of random partitions. A good example is a paper of Pittel [17] substantiating two well-known conjectures concerning integer partitions. Utilizing Fristedt's result, Corteel, Pittel, Savage, and Wilf [4] quite recently provided an answer to the following question. Consider the following two-step sampling procedure: first choose uniformly at random a partition λ of n. Then, out of all different part sizes in λ pick one uniformly at random. What is the asymptotic *unconditional* probability that this part size has a certain specified multiplicity, say, m? For example, partition $\lambda = \{3,2,2,1,1,1\}$ of the number 10 has three different part sizes 1, 2, and 3, and only one of them has multiplicity three, namely 1. Thus, for this particular partition, the probability of choosing a part that has multiplicity three is 1/3. In order to find the unconditional probability of randomly choosing a part of multiplicity three in a randomly chosen partition of 10, one would have to average similar probabilities over all partitions of 10. Corteel, Pittel, Savage, and Wilf showed that in general the probability in question approaches $1/(m(m+1))$ (in particular, the probability that the randomly chosen part size in a random partition is unrepeated approaches 1/2 as $n \to \infty$).

Wilf then asked the same question for random compositions: what is the asymptotic value of the probability that a randomly chosen part size in a random composition of an integer n has multiplicity m? Our aim here is to provide an answer as complete as we can. On the "first level" of precision the answer is simple: for every fixed m this probability approaches zero. One would then like to know the rate of this convergence. We will show that the rate is $1/\ln n$. Specifically, if $A_n^{(m)}$ is the event that a randomly chosen part size in a random composition of n has multiplicity m, then there exist constants $c_1(m) \le c_2(m)$ such that $c_1(m) \le (\ln n)\mathbb{P}(A_n^{(m)}) \le c_2(m)$ for $n \ge 2$.

The next natural step is to find possibly tight bounds on $c_1(m)$ and $c_2(m)$, or to show that the limit $(\ln n)\mathbb{P}(A_n^{(m)})$ exists as $n \to \infty$. This is the place where things become a bit tricky. In order to describe the difficulties let us briefly discuss the argument. Letting $U_n^{(m)}$ and D_n denote the number of parts of multiplicity m and the number of distinct part sizes, respectively, we have $\mathbb{P}(A_n^{(m)}) = \mathbb{E}(U_n^{(m)}/D_n)$. In the case of partitions, Corteel, Pittel, Savage, and Wilf used Fristedt's result to argue that D_n is heavily concentrated around its expectation, and therefore, $\mathbb{P}(A_n^{(m)})$ is asymptotic to the ratio of expectations $\mathbb{E}U_n^{(m)}/\mathbb{E}D_n$ and one needs to find asymptotic values of these two expectations. In the case of compositions, much of the story is the same, with one crucial exception: the expected value of $U_n^{(m)}$ does not have a limit, but exhibits oscillations around $1/(m\ln 2)$. (This phenomenon is not new and was observed in the context of head runs in coin tossing; see, e.g., [3], [12], or [13].) Since, as we will show, the behavior of $(\ln n)\mathbb{P}(A_n^{(m)})$ is governed by the behavior of $\mathbb{E}U_n^{(m)}$, it will follow that $(\ln n)\mathbb{P}(A_n^{(m)})$ oscillates around the value $1/m$ as $n \to \infty$.

The rest of the paper is organized as follows: in the next section we will introduce notation and state our result precisely. In section 3 we will describe the probabilistic set-up. In section 4 we estimate the number of distinct part sizes and show that D_n is heavily concentrated about its expectation. In section 5, we give an estimate for the expected number of parts of given multiplicity. In section 6, we compute bounds on the oscillation.

2. Notation and statement of the result. A *composition* κ of an integer n is an ordered tuple $(\gamma_1, \ldots, \gamma_k)$, where $\gamma_1, \ldots, \gamma_k$ are positive integers such that $\sum_{i=1}^{k} \gamma_i = n$. The numbers $\gamma_1, \ldots, \gamma_k$ are called *parts*, k is the total number of parts, and the elements of the set $\{\gamma_1, \ldots, \gamma_k\}$ are the *part sizes* of κ. For example, $(2, 1, 2, 3, 1, 1)$ is a composition of the number 10 into six parts with part sizes 1, 2, and 3, where part size 1 has multiplicity 3, 2 has multiplicity 2, and 3 has multiplicity 1. We denote the set of all compositions of n by $C(n)$ and note that $|C(n)| = 2^{n-1}$. For a composition $\kappa = (\gamma_1, \ldots, \gamma_k)$ we let $D_n(\kappa)$ denote the number of distinct part sizes and, for fixed integer m, $U_n^{(m)}(\kappa)$ will denote the number of part sizes of κ that have multiplicity m. More formally,

$$D_n(\kappa) = 1 + \sum_{i=2}^{k} I_{\{\gamma_i \neq \gamma_j,\ j=1,\ldots,i-1\}},$$

where I_A, the indicator of event A, is 1 if A takes place and 0, otherwise. Similarly,

$$U_n^{(m)}(\kappa) = \sum_{i=1}^{k} I_{B_i},$$

where

$$B_i = \{\gamma_i \neq \gamma_j,\ j < i \text{ and } \operatorname{card}\{\ell > i :\ \gamma_\ell = \gamma_i\} = m - 1\}.$$

We equip $C(n)$ with the uniform probability measure $\mathbb{P}$ (i.e., $\mathbb{P}(\kappa) = |C(n)|^{-1} = 2^{-n+1}$ for every $\kappa \in C(n)$), and we will denote the expectation with respect to that measure by $\mathbb{E}$.

Throughout the paper the letter c is reserved for an absolute constant whose value is of no relevance and may change from line to line.

We consider the following experiment. First, a composition is chosen at random. Then, out of all distinct part sizes one is selected uniformly at random. We would like to know what the unconditional probability is that this part size has multiplicity m. We will denote this event by $A_n^{(m)}$. Since for a given composition κ the probability that a randomly chosen part size has multiplicity m is given by the ratio

$$\frac{U_n^{(m)}(\kappa)}{D_n(\kappa)},$$

the unconditional probability that a randomly chosen part size in a random composition has this multiplicity is just the expected value of that ratio. That is,

$$\mathbb{P}(A_n^{(m)}) = \mathbb{E}\frac{U_n^{(m)}}{D_n}.$$

Thus, our goal is to approximate this expectation. Our result is as follows.

THEOREM 1. *Under the above notation we have the following: for a fixed integer* m

$$(\ln n)\mathbb{P}(A_n^{(m)}) = \Theta(1),$$

i.e., there exist two positive constants $c_1(m)$ *and* $c_2(m)$ *such that for all* $n \geq 2$,

$$c_1(m) \leq (\ln n)\mathbb{P}(A_n^{(m)}) \leq c_2(m).$$

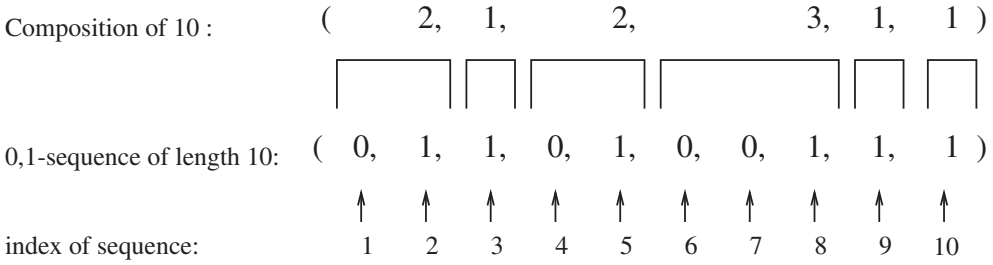


FIG. 1. *Correspondence between compositions of n and $0,1$-sequences of length n which end with* 1.

More precisely, as $n \to \infty$,

$$(\ln n)\mathbb{P}(A_n^{(m)}) = \frac{1}{m} + H^{(m)}(c \ln n) + o(1),$$

where $H^{(m)}$ is a mean-zero function of period 1 *whose Fourier coefficients are given by*

$$\phi_\ell = \frac{1}{m!}\Gamma\left(m - \frac{2\ell\pi i}{\ln 2}\right), \qquad \ell \neq 0.$$

3. Probabilistic set-up. Much of our proof relies on an appropriate interpretation of a composition, found, e.g., in Andrews [2]. This interpretation allows us to connect the study of random compositions to another much investigated topic, namely the study of runs of successes in independent Bernoulli trials (see, for example, Erdös and Rényi [7] or Erdös and Révész [8]). In order to describe this connection we interpret compositions as follows: consider a composition $\kappa = (\gamma_1, \dots, \gamma_k)$ of n into parts $\gamma_1, \gamma_2, \dots, \gamma_k$ (for example, $(2,1,2,3,1,1)$ is a composition of the number 10 into six parts $2,1,2,3,1,1$.) Such a composition is associated with a $\{0,1\}$-valued sequence $(x_1, \dots, x_n)$ in which $x_i = 1$ for $i \in \{\gamma_1, \gamma_1+\gamma_2, \dots, \gamma_1+\cdots+\gamma_k\}$ and otherwise $x_i = 0$. (Note that this forces $x_n = 1$.) For example, the composition $(2,1,2,3,1,1)$ is associated with the sequence $(0,1,1,0,1,0,0,1,1,1)$, as illustrated in Figure 1. Clearly there is a one-to-one correspondence between compositions of n and $\{0,1\}$-sequences $(x_1, \dots, x_n)$ with $x_n = 1$.

To say that a composition is chosen at random is to say that the 0's and 1's occur with probability $1/2$ at each of the first $n-1$ positions, and the occurrences at different positions are independent of each other. In other words, the number of 1's in the first $n-1$ positions is a binomial random variable, $\mathrm{Bin}(n-1, 1/2)$, with parameters $n-1$ and $1/2$. With this interpretation the total number of parts is just the number of 1's (including the one in the nth position) and thus it is equidistributed with $1 + \mathrm{Bin}(n-1, 1/2)$. (This contrasts with the case of "unordered" partitions where the exact distribution of the number of parts is unknown and it took a considerable effort to find a limiting distribution of the total number of parts; see Erdös and Lehner [6].) Furthermore, the numbers $\gamma_1, \dots, \gamma_k$ can be viewed as "waiting times" for the first, second, ... , and kth appearance of 1 in the associated $\{0,1\}$-sequence $(x_1, \dots, x_n)$. (In our example, 1 appears in the second, third, fifth, eighth, ninth, and, of course, tenth positions.) It is well known and easy to check that in an infinite sequence of independent Bernoulli trials with the probability of success p, waiting times for

successes are independent and identically distributed (i.i.d.) random variables whose common distribution is that of a geometric random variable with parameter p. Since we are considering only $n-1$ trials, this is no longer true. But we have the following fact.

PROPOSITION 2. *Let $\Gamma_1, \Gamma_2 \dots$ be i.i.d. geometric random variables with parameter $1/2$ (that is, $\mathbb{P}(\Gamma_1 = j) = 2^{-j}$, $j = 1, 2 \dots$) and define*

$$\tau = \inf\{k \geq 1 : \ \Gamma_1 + \Gamma_2 + \cdots + \Gamma_k \geq n\}.$$

Then we have the following: if the set $C(n)$ of all compositions of an integer n is equipped with the uniform probability measure, then the distribution of a randomly chosen composition is given by

$$\left(\Gamma_1, \Gamma_2, \dots, \Gamma_{\tau-1}, n - \sum_{j=1}^{\tau-1} \Gamma_j\right).$$

4. The number of distinct parts. In this section we will study certain aspects of the behavior of D_n. For the purpose of approximating $\mathbb{P}(A_n^{(m)})$ we will work with the ratio $U_n^{(m)}/D_n$, but it will be clear from our argument, for example, that

$$\frac{\mathbb{E}D_n}{\log_2 n} \to 1 \quad \text{as} \quad n \to \infty.$$

We will proceed in the following fashion: we will establish the existence of two sequences of natural numbers (ℓ_n) and (k_n) which increase to infinity and are asymptotically the same, i.e.,

$$\lim_{n\to\infty} \frac{\ell_n}{k_n} = 1,$$

and such that both probabilities

$$\mathbb{P}(D_n \leq \ell_n), \qquad \mathbb{P}(D_n \geq k_n)$$

tend to zero as $n \to \infty$ at a rate faster than $1/\log_2 n$. This will allow us to replace the D_n in the denominator by either of the sequences (ℓ_n) or (k_n), and then in the next section, we will approximate the expected value of $U_n^{(m)}$. We begin with establishing the existence of (ℓ_n). For a composition $\kappa = (\gamma_1, \dots, \gamma_k)$, let $S_n(\kappa)$ denote the number of consecutive part sizes (starting with size 1) in κ. That is,

$$S_n(\kappa) = \max\{\ell : \ \forall j \leq \ell \ \exists i \leq k : \gamma_i = j\}.$$

Consider for a moment an arbitrary integer ℓ_n. Since $S_n(\kappa) \leq D_n(\kappa)$ we have

$$\mathbb{P}(D_n \leq \ell_n) \leq \mathbb{P}(S_n \leq \ell_n) \leq \mathbb{P}(\exists j \leq \ell_n, \ \forall i < \tau, \ : \ \Gamma_i \neq j), \tag{1}$$

where we purposely ignored the last part $n - \sum_{j=1}^{\tau-1} \Gamma_i$ by writing "$i < \tau$." In order to bound the last probability we first notice that, since τ is equidistributed with the random variable $1 + \mathrm{Bin}(n-1, 1/2)$, we have

$$\mathbb{E}\tau = 1 + (n-1)/2 = (n+1)/2.$$

Moreover, τ is well concentrated around its mean. Namely (see, for example, [1, section A.1]), for every $t > 0$ we have

$$\mathbb{P}(|\tau - \mathbb{E}\tau| \geq t) \leq 2\exp\left\{-\frac{2t^2}{n-1}\right\}.$$

In particular, letting $t_n = \sqrt{\alpha(n-1)\ln n}$, we get

$$\mathbb{P}(|\tau - \mathbb{E}\tau| \geq t_n) \leq 2\exp\{-2\alpha \ln n\} = \frac{2}{n^{2\alpha}}. \tag{2}$$

(The value of α plays a minimal role in the argument, so we will set it to be 1 for the rest of this section; we just want to mention that by increasing this value as necessary we can get arbitrary polynomial rate of convergence to zero of this probability. This will be useful in the next section.) Let $q_n^- = \mathbb{E}\tau - t_n = (n+1)/2 - \sqrt{(n-1)\ln n}$. Then we can bound (1) by

$$\begin{aligned}\mathbb{P}(\exists j \leq \ell_n,\ \forall i < \tau,\ :\ \Gamma_i \neq j) &\leq \mathbb{P}(|\tau - \mathbb{E}\tau| > t_n) \\ &\quad + \mathbb{P}(\{\exists j \leq \ell_n,\ \forall i < \tau,\ :\ \Gamma_i \neq j\} \cap \{|\tau - \mathbb{E}\tau| \leq t_n\}).\end{aligned} \tag{3}$$

From (2), the first probability in the right-hand side (rhs) of (3) goes to 0 at a polynomial rate, so we concentrate on the second. Since $|\tau - \mathbb{E}\tau| \leq t_n$ implies that $\tau \geq q_n^- = (n+1)/2 - o(n)$, we bound the second term in the rhs of (3) by

$$\begin{aligned}&\mathbb{P}(\{\exists j \leq \ell_n,\ \forall i < \tau,\ :\ \Gamma_i \neq j\} \cap \{|\tau - \mathbb{E}\tau| \leq t_n\}) \\ &\quad\leq \mathbb{P}\left(\bigcup_{j=1}^{\ell_n} \bigcap_{i=1}^{\tau-1} \{\Gamma_i \neq j\} \cap \{\tau > q_n^-\}\right) \leq \mathbb{P}\left(\bigcup_{j=1}^{\ell_n} \bigcap_{i=1}^{q_n^-} \{\Gamma_i \neq j\}\right) \\ &\quad\leq \sum_{j=1}^{\ell_n} \mathbb{P}\left(\bigcap_{i=1}^{q_n^-} \{\Gamma_i \neq j\}\right) = \sum_{j=1}^{\ell_n} (\mathbb{P}(\Gamma_1 \neq j))^{q_n^-} = \sum_{j=1}^{\ell_n} \left(1 - \frac{1}{2^j}\right)^{q_n^-} \\ &\quad\leq \sum_{j=1}^{\ell_n} \exp\left\{-\frac{q_n^-}{2^j}\right\} = \sum_{j=1}^{\ell_n} \exp\left\{-\frac{q_n^-}{2^{\ell_n}} 2^{\ell_n - j}\right\} \\ &\quad\leq \sum_{k=0}^{\infty} \exp\left\{-\frac{q_n^-}{2^{\ell_n}} 2^k\right\} \leq \sum_{k=1}^{\infty} \exp\left\{-\frac{q_n^-}{2^{\ell_n}} k\right\} \leq 2\exp\left\{-\frac{q_n^-}{2^{\ell_n}}\right\}\end{aligned}$$

as long as $q_n^-/2^{\ell_n} \geq 1$. Furthermore, the upper bound will go to 0 as $n \to \infty$ if $q_n^-/2^{\ell_n} \to \infty$. For that it is enough to let $\ell_n \sim \log_2(q_n^-/\phi(q_n^-))$, where $q_n^-/\phi(q_n^-) \to \infty$ as $n \to \infty$. For our purpose, the choice $\phi(q_n^-) = \log_2(q_n^-)$ will be convenient. With this choice, we conclude that

$$\mathbb{P}(D_n \leq \ell_n) \leq 2\exp\left\{-\frac{q_n^-}{q_n^-/\ln(q_n^-)}\right\} \leq \frac{2}{q_n^-} = O\left(\frac{1}{n}\right).$$

Using the fact that $0 \leq U_n^{(m)}/D_n \leq 1$, we infer that

$$\mathbb{E}\frac{U_n^{(m)}}{D_n} = \mathbb{E}\frac{U_n^{(m)}}{D_n} I_{D_n \leq \ell_n} + \mathbb{E}\frac{U_n^{(m)}}{D_n} I_{D_n > \ell_n} \leq \mathbb{P}(D_n \leq \ell_n) + \frac{\mathbb{E}U_n^{(m)}}{\ell_n}.$$

As we will see in the next section, $\mathbb{E}U_n^{(m)} = \Theta(1)$, so that the second term in the last sum is dominating.

As for the lower bound, consider a sequence (k_n) which will be specified later. We then have

$$\mathbb{E}\frac{U_n^{(m)}}{D_n} \geq \mathbb{E}\frac{U_n^{(m)}}{D_n} I_{D_n \leq k_n} \geq \frac{1}{k_n}\mathbb{E}U_n^{(m)} I_{D_n \leq k_n} = \frac{1}{k_n}(\mathbb{E}U_n^{(m)} - \mathbb{E}U_n^{(m)} I_{D_n > k_n}).$$

We will choose (k_n) so that the term $\mathbb{E}U_n^{(m)} I_{D_n > k_n}$ will be of lower order than $\mathbb{E}U_n^{(m)}$. Since the latter term will be shown to be bounded away from zero, this means that it suffices to choose (k_n) so that

$$\mathbb{E}U_n^{(m)} I_{D_n > k_n} \to 0 \qquad \text{as} \qquad n \to \infty.$$

Since the number of distinct part sizes is no larger than the largest part size, letting Γ_n^* and Γ_τ^* denote $\max\{\Gamma_1, \ldots, \Gamma_n\}$ and $\max\{\Gamma_1, \ldots, \Gamma_\tau\}$, respectively, we have

$$U_n^{(m)} \leq D_n \leq \Gamma_\tau^* \leq \Gamma_n^*.$$

(The second inequality is valid since the size of the last part is no more than Γ_τ.) It follows that

$$\{D_n > k_n\} \subset \{\Gamma_n^* \geq k_n\},$$

and thus

$$\mathbb{E}U_n^{(m)} I_{D_n > k_n} \leq \mathbb{E}\Gamma_n^* I_{\Gamma_n^* \geq k_n}.$$

To find a choice of (k_n) that would make this latter expectation go to 0 we write

$$\mathbb{E}\Gamma_n^* I_{\Gamma_n^* \geq k_n} = \sum_{t=k_n}^{\infty} t\mathbb{P}(\Gamma_n^* = t) \leq n \sum_{t=k_n}^{\infty} t\mathbb{P}(\Gamma_1 = t) = n \sum_{t=k_n}^{\infty} \frac{t}{2^t} = \frac{n(2+2k_n)}{2^{k_n}}.$$

Choosing $k_n \sim \log_2(n\psi(n))$, we get

$$\mathbb{E}\Gamma_n^* I_{\Gamma_n^* \geq k_n} \leq \frac{2n + 2n\log_2(n\psi(n))}{n\psi(n)},$$

which goes to 0 for $\psi(n) = \log_2^2 n$, for example. Thus one can set $k_n \sim \log_2(n \log_2^2 n)$. With these choices of (ℓ_n) and (k_n) we obtain that

$$\begin{aligned}\mathbb{P}(A_n^{(m)}) = \mathbb{E}\frac{U_n^{(m)}}{D_n} &= \mathbb{E}\frac{U_n^{(m)}}{D_n} I_{\ell_n \leq D_n \leq k_n} + \mathbb{E}\frac{U_n^{(m)}}{D_n} I_{\{\ell_n \leq D_n \leq k_n\}^c} \\ &= \mathbb{E}\frac{U_n^{(m)}}{\log_2 n \pm o(\log n)} + \mathbb{E}\frac{U_n^{(m)}}{D_n} I_{D_n < \ell_n} + \mathbb{E}\frac{U_n^{(m)}}{D_n} I_{D_n > k_n}.\end{aligned}$$

By the choice of (ℓ_n) and (k_n) the last two expectations are bounded above by

$$\begin{aligned}\mathbb{P}(D_n < \ell_n) + \mathbb{P}(D_n > k_n) &\leq 2 \cdot 2^{-q_n^-/2^{\ell_n}} + \mathbb{P}(\Gamma_n^* > k_n) \\ &\leq 2 \cdot 2^{-\phi(q_n^-)} + n\mathbb{P}(\Gamma_1 > k_n) \leq O\left(\frac{1}{n}\right) + \frac{n}{2^{k_n}} \leq O\left(\frac{1}{\log_2^2 n}\right),\end{aligned}$$

and we see that

$$(\ln n)\mathbb{P}(A_n^{(m)}) = \frac{\mathbb{E}U_n^{(m)}}{\log_2 e + o(1)} + o(1), \tag{4}$$

provided that $\mathbb{E}U_n^{(m)} = \Theta(1)$. Thus, the asymptotic behavior of $(\ln n)\mathbb{P}(A_n^{(m)})$ is determined completely by the behavior of $\mathbb{E}U_n^{(m)}$, and to complete the proof we need to estimate $\mathbb{E}U_n^{(m)}$.

5. Parts of multiplicity m. In this section we will approximate $\mathbb{E}U_n^{(m)}$. Let $\mathcal{U}^{(m)} = \mathcal{U}^{(m)}(\kappa)$ denote the set of part sizes in κ that have multiplicity m, and let us write $j \in \mathcal{U}^{(m)}$ to indicate that size "j" has multiplicity m. We have

$$\mathbb{E}U_n^{(m)} = \mathbb{E}\sum_{j\le n/m} I_{j\in\mathcal{U}^{(m)}} = \sum_{j\le n/m} \mathbb{P}(j \in \mathcal{U}^{(m)}).$$

Therefore, we need to estimate the sum of $\mathbb{P}(j \in \mathcal{U}^{(m)})$. The degree of difficulty of this approximation increases with the accuracy that one desires to achieve. Furthermore, since, as we will see, $\mathbb{E}U_n^{(m)}$ is an oscillatory function, explicit bounds on $\mathbb{E}U_n^{(m)}$, no matter how tight, cannot be used to show that $(\ln n)\mathbb{P}(A_n^{(m)})$ converges. Thus, one may consider devoting too much attention to an accurate approximation to be a questionable investment. We will present the detailed argument for the fairly precise bound on $\mathbb{E}U_n^{(m)}$, but the reader interested in just the fact that this expectation is $\Theta(1)$ (which is all that is needed to establish (4)) will notice that the argument may be simplified. To make this point more transparent, let $\tilde{\Gamma}_i(\kappa)$, $i = 1, \dots, \tau(\kappa)$, denote the parts of a composition κ, i.e.,

$$\tilde{\Gamma}_i(\kappa) = \Gamma_i(\kappa) \quad \text{for} \quad i < \tau(\kappa) \quad \text{and} \quad \tilde{\Gamma}_{\tau(\kappa)}(\kappa) = n - \sum_{i=1}^{\tau(\kappa)-1} \Gamma_i(\kappa).$$

It is much more convenient to work with Γ's rather than with $\tilde{\Gamma}$'s, because the last part, $\tilde{\Gamma}_\tau$, complicates the dependence structure. As a result, a nonnegligible part of our argument is to show that "tildes" can be neglected. This is, of course, not an issue if one is interested merely in a $\Theta(1)$ result; tildes may be dropped since the single part $\tilde{\Gamma}_\tau$ can be ignored without affecting $U_n^{(m)}$ by more than 1. To estimate $\mathbb{P}(j \in \mathcal{U}^{(m)})$ write

$$\begin{aligned}
\mathbb{P}(j \in \mathcal{U}^{(m)}) =& \mathbb{P}\left(\sum_{i=1}^{\tau} I_{\tilde{\Gamma}_i=j} = m\right) = \mathbb{P}\left(\{\tilde{\Gamma}_\tau = j\} \cap \left\{\sum_{i=1}^{\tau-1} I_{\Gamma_i=j} = m-1\right\}\right) \\
&+ \mathbb{P}\left(\{\tilde{\Gamma}_\tau \neq j\} \cap \left\{\sum_{i=1}^{\tau-1} I_{\Gamma_i=j} = m\right\}\right) \\
&= \mathbb{P}\left(\sum_{i=1}^{\tau-1} I_{\Gamma_i=j} = m\right) + \mathbb{P}\left(\{\tilde{\Gamma}_\tau = j\} \cap \left\{\sum_{i=1}^{\tau-1} I_{\Gamma_i=j} = m-1\right\}\right) \\
&- \mathbb{P}\left(\{\tilde{\Gamma}_\tau = j\} \cap \left\{\sum_{i=1}^{\tau-1} I_{\Gamma_i=j} = m\right\}\right).
\end{aligned} \tag{5}$$

We begin by estimating the first probability in (5), and then we will show that the sums over j of the last two probabilities are negligible. Let $q_n^{\pm} = (n+1)/2 \pm t_n$. As in

the previous section, let $t_n = \sqrt{\alpha(n-1)\ln n}$, but we now choose $\alpha = 2$ so that from (2) we get $n\mathbb{P}(|\tau - \mathbb{E}\tau| \geq t_n) \leq 2/n^3$. To get an upper bound on the first term in (5) write

$$\mathbb{P}\left(\sum_{i=1}^{\tau-1} I_{\Gamma_i=j} = m\right) \leq \mathbb{P}\left(\{|\tau - \mathbb{E}\tau| \leq t_n\} \cap \left\{\sum_{i=1}^{\tau-1} I_{\Gamma_i=j} = m\right\}\right) + \mathbb{P}(|\tau - \mathbb{E}\tau| > t_n). \tag{6}$$

The second probability in the rhs of (6) is $O(1/n^4)$ and for the first one we have

$$\begin{aligned}
&\mathbb{P}\left(\left\{\sum_{i=1}^{\tau-1} I_{\Gamma_i=j} = m\right\} \cap \{|\tau - \mathbb{E}\tau| \leq t_n\}\right) \\
&\quad = \mathbb{P}\Bigg(\bigcup_{1\leq i_1<\cdots<i_m\leq\tau} \left(\bigcap_{\ell=1}^{m}\{\Gamma_{i_\ell} = j\} \cap \bigcap_{\substack{i=1\\ i\neq i_1,\ldots,i_m}}^{\tau-1} \{\Gamma_i \neq j\}\right) \cap \{|\tau - \mathbb{E}\tau| \leq t_n\}\Bigg) \\
&\quad \leq \mathbb{P}\Bigg(\bigcup_{1\leq i_1<\cdots<i_m\leq q_n^+} \left(\bigcap_{\ell=1}^{m}\{\Gamma_{i_\ell} = j\} \cap \bigcap_{\substack{i=1\\ i\neq i_1,\ldots,i_m}}^{q_n^- -1} \{\Gamma_i \neq j\}\right) \cap \{|\tau - \mathbb{E}\tau| \leq t_n\}\Bigg) \\
&\quad \leq \mathbb{P}\Bigg(\bigcup_{1\leq i_1<\cdots<i_m\leq q_n^+} \left(\bigcap_{\ell=1}^{m}\{\Gamma_{i_\ell} = j\} \cap \bigcap_{\substack{i=1\\ i\neq i_1,\ldots,i_m}}^{q_n^- -1} \{\Gamma_i \neq j\}\right)\Bigg) \\
&\quad \leq \binom{q_n^+}{m}\frac{1}{2^{jm}}\left(1-\frac{1}{2^j}\right)^{q_n^- -1-m}.
\end{aligned}$$

Similarly, to get a lower bound for the first term of (5) we have

$$\begin{aligned}
&\mathbb{P}\left(\sum_{i=1}^{\tau-1} I_{\Gamma_i=j} = m\right) \geq \mathbb{P}\left(\left\{\sum_{i=1}^{\tau-1} I_{\Gamma_i=j} = m\right\} \cap \{|\tau - \mathbb{E}\tau| \leq t_n\}\right) \\
&\quad \geq \mathbb{P}\Bigg(\bigcup_{1\leq i_1<\cdots<i_m\leq q_n^-} \left(\bigcap_{\ell=1}^{m}\{\Gamma_{i_\ell} = j\} \cap \bigcap_{\substack{i=1\\ i\neq i_1,\ldots,i_m}}^{q_n^+} \{\Gamma_i \neq j\}\right) \cap \{|\tau - \mathbb{E}\tau| \leq t_n\}\Bigg) \\
&\quad \geq \mathbb{P}\Bigg(\bigcup_{1\leq i_1<\cdots<i_m\leq q_n^-} \left(\bigcap_{\ell=1}^{m}\{\Gamma_{i_\ell} = j\} \cap \bigcap_{\substack{i=1\\ i\neq i_1,\ldots,i_m}}^{q_n^+} \{\Gamma_i \neq j\}\right)\Bigg) - \mathbb{P}(|\tau - \mathbb{E}\tau| > t_n) \\
&\quad \geq \binom{q_n^-}{m}\frac{1}{2^{jm}}\left(1-\frac{1}{2^j}\right)^{q_n^+ -m} - O\left(\frac{1}{n^4}\right).
\end{aligned}$$

It remains to bound the sum over j of the terms

$$\mathbb{P}\left(\{\tilde{\Gamma}_\tau = j\} \cap \left\{\sum_{i=1}^{\tau-1} I_{\Gamma_i=j} = m\right\}\right) \tag{7}$$

and

$$\mathbb{P}\left(\{\tilde{\Gamma}_\tau = j\} \cap \left\{\sum_{i=1}^{\tau-1} I_{\Gamma_i=j} = m-1\right\}\right) \tag{8}$$

in (5) and to show that they are negligible compared to the sum over j of the first term in (5). Since

$$\{\tilde{\Gamma}_\tau = j\} \subset \{\tilde{\Gamma}_\tau \geq j\} \subset \{\Gamma_\tau \geq j\},$$

for the probability in (7) we have

$$\begin{aligned}
\mathbb{P}\left(\{\tilde{\Gamma}_\tau = j\} \cap \left\{\sum_{i=1}^{\tau-1} I_{\Gamma_i=j} = m\right\}\right) &\leq \mathbb{P}\left(\{\Gamma_\tau \geq j\} \cap \left\{\sum_{i=1}^{\tau-1} I_{\Gamma_i=j} = m\right\}\right) \\
&\leq \mathbb{P}\left(\{|\tau - \mathbb{E}\tau| \leq t_n\} \cap \{\Gamma_\tau \geq j\} \cap \left\{\sum_{i=1}^{\tau-1} I_{\Gamma_i=j} = m\right\}\right) + O\left(\frac{1}{n^4}\right) \\
&\leq \sum_{k=q_n^-}^{q_n^+} \mathbb{P}\left(\{\tau = k\} \cap \{\Gamma_k \geq j\} \cap \left\{\sum_{i=1}^{k-1} I_{\Gamma_i=j} = m\right\}\right) + O\left(\frac{1}{n^4}\right) \\
&\leq \sum_{k=q_n^-}^{q_n^+} \mathbb{P}\left(\{\Gamma_k \geq j\} \cap \left\{\sum_{i=1}^{k-1} I_{\Gamma_i=j} = m\right\}\right) + O\left(\frac{1}{n^4}\right) \\
&\leq \sum_{k=q_n^-}^{q_n^+} \frac{1}{2^{j-1}} \binom{k-1}{m} \frac{1}{2^{jm}} \left(1 - \frac{1}{2^j}\right)^{k-1-m} + O\left(\frac{1}{n^4}\right) \\
&\leq c\sqrt{n \ln n} \binom{q_n^+}{m} \frac{1}{2^{j(m+1)}} \left(1 - \frac{1}{2^j}\right)^{q_n^- - m} + O\left(\frac{1}{n^4}\right)
\end{aligned}$$

by the definition of q_n^+ and q_n^-. Thus, summing up over j, we get

$$\begin{aligned}
&c\sqrt{n \ln n} \binom{q_n^+}{m} \sum_{j=1}^{n} \frac{1}{2^{j(m+1)}} \left(1 - \frac{1}{2^j}\right)^{q_n^- - m} + O\left(\frac{1}{n^3}\right) \\
&\quad \leq c\sqrt{n \ln n} \binom{q_n^+}{m} \int_1^\infty \frac{1}{2^{(m+1)x}} \left(1 - \frac{1}{2^x}\right)^{q_n^- - m} dx + O\left(\frac{1}{n^3}\right) \\
&\quad \leq c\sqrt{n \ln n} \binom{q_n^+}{m} \int_0^1 u^m (1-u)^{q_n^- - m} du + O\left(\frac{1}{n^3}\right) \\
&\quad = c\sqrt{n \ln n} \binom{q_n^+}{m} \frac{\Gamma(m+1)\Gamma(q_n^- - m + 1)}{\Gamma(q_n^- + 2)} + O\left(\frac{1}{n^3}\right) \\
&\quad = \Theta\left(\sqrt{\frac{\ln n}{n}}\right).
\end{aligned}$$

For the second probability (8) the argument is essentially the same:

$$\begin{aligned}
\mathbb{P}\left(\{\tilde{\Gamma}_\tau = j\} \cap \left\{\sum_{i=1}^{\tau-1} I_{\Gamma_i=j} = m-1\right\}\right) &\le \mathbb{P}\left(\{\Gamma_\tau \ge j\} \cap \left\{\sum_{i=1}^{\tau-1} I_{\Gamma_i=j} = m-1\right\}\right) \\
&\le \sum_{k=q_n^-}^{q_n^+} \mathbb{P}\left(\{\tau = k\} \cap \{\Gamma_k \ge j\} \cap \left\{\sum_{i=1}^{k-1} I_{\Gamma_i=j} = m-1\right\}\right) + O\left(\frac{1}{n^4}\right) \\
&\le \sum_{k=q_n^-}^{q_n^+} \frac{1}{2^{j-1}}\binom{k-1}{m-1}\frac{1}{2^{j(m-1)}}\left(1-\frac{1}{2^j}\right)^{k-1-m+1} + O\left(\frac{1}{n^4}\right) \\
&\le c\sqrt{n\ln n}\binom{q_n^+}{m-1}\frac{1}{2^{jm}}\left(1-\frac{1}{2^j}\right)^{q_n^- - m} + O\left(\frac{1}{n^4}\right),
\end{aligned}$$

and in the same fashion as before we see that the sum over j of these terms does not exceed $\Theta(\sqrt{(\ln n)/n})$.

We now observe that for $q = q_n \sim n/2$ the sum

$$\binom{q}{m}\sum_{j=1}^{\infty} 2^{-jm}(1-2^{-j})^{q-m}$$

is easily seen to be $\Theta(1)$ (it suffices to compare it to the integral $\int_1^\infty 2^{-mx}(1-2^{-x})^{q-m}dx$). Therefore, since q_n^+ and q_n^- are asymptotically the same, m is fixed, and

$$\sum_{j>n} 2^{-jm}(1-2^{-j})^{q_n^\pm - m} \le \sum_{j>n} 2^{-j} = \frac{1}{2^n},$$

we conclude that

$$\mathbb{E}U_n^{(m)} \sim \binom{q}{m}\sum_{j=1}^{\infty} 2^{-jm}(1-2^{-j})^{q-m}. \tag{9}$$

A more detailed analysis reveals a quite interesting and unexpected phenomenon. The rhs in (9) does not have a limit, but exhibits oscillations about $1/(m\ln 2)$. To see this, one approach is as follows (for convenience we will replace $q-m$ with q in (9)—this does not affect asymptotics): expanding $(1-2^{-j})^q$ using the binomial formula, and summing over j, gives

$$\sum_{j=1}^{\infty} 2^{-jm}(1-2^{-j})^q = \sum_{k=0}^{q}(-1)^k\binom{q}{k}\frac{1}{2^{k+m}-1}.$$

Alternating sums of this type appear surprisingly often in the analysis of certain algorithms and can be approximated using methods of complex analysis. Since the standard method, attributed to Rice, has been described recently in several papers, we will not reproduce the details here. Rather, we refer to [16, section 5.2.2], [9], [14], or [15] for some examples of applications and illustration of the method. In particular, these last two papers explicitly treat the asymptotics of the sum

$$\sum_{k=0}^{q}(-1)^k\binom{q}{k}\frac{1}{2^{k+m}-1}.$$

We would like to indicate an alternative approach to approximating (9) shown to the first author by Bennett Eisenberg and Gilbert Stengle [5]. Although it seems less general than the Rice method, in the case of our sum it gives a more elementary and direct proof of the asymptotics. Consider a sequence (q_s) such that for some $1 \le \beta < 2$, $q_s/2^s \to \beta = 2^x$, $0 \le x < 1$. Then, for s large, replacing q_s with 2^{x+s} and j with $s + r$ we get

$$\begin{aligned}\binom{q_s}{m}\sum_{j=1}^{\infty}\frac{1}{2^{jm}}\left(1-\frac{1}{2^j}\right)^{q_s} &\sim \frac{2^{m(x+s)}}{m!}\sum_{r=-s+1}^{\infty}2^{-sm}2^{-rm}\left(1-\frac{1}{2^s2^r}\right)^{2^x2^s}\\ &\sim \frac{1}{m!}\sum_{r=-\infty}^{\infty}2^{m(x-r)}e^{-2^{x-r}},\end{aligned}$$

where the "legality" of passing to the limits is easily checked (see [5]). The latter expression defines a 1-periodic function, and its Fourier coefficients are easily found:

$$\begin{aligned}\phi_\ell =&\frac{1}{m!}\int_0^1\sum_{r=-\infty}^{\infty}2^{m(x-r)}e^{-2^{x-r}}e^{-2\pi i\ell x}dx\\ =&\frac{1}{m!}\int_{-\infty}^{\infty}2^{mx}e^{-2^x}e^{-2\pi i\ell x}dx,\end{aligned}$$

which, upon substitution $u = 2^x$, becomes

$$\phi_\ell = \frac{1}{m!\ln 2}\int_0^\infty u^{m-1-2\pi i\ell/\ln 2}e^{-u}du = \frac{1}{m!\ln 2}\Gamma\left(m-\frac{2\pi i\ell}{\ln 2}\right).$$

Note that $\phi_0 = 1/(m\ln 2)$, so if we let

$$H^{(m)}(x) = \frac{\ln 2}{m!}\sum_{r=-\infty}^{\infty}2^{m(x-r)}e^{-2^{x-r}} - \frac{1}{m},$$

then $H^{(m)}$ satisfies the conditions of Theorem 1. Combining this with (4) completes the proof of the theorem.

6. Bounding the oscillation. The sum below is used in section 5 of the paper to approximate $\mathbb{E}U_n^{(m)}$:

$$\binom{q}{m}\sum_{j=1}^{\infty}2^{-jm}(1-2^{-j})^{q-m}, \tag{10}$$

where $q = \lfloor (n/2)\rfloor$. The data displayed in Figures 2, 3, and 4 indicate that $1/(m\ln 2)$ is a reasonable approximation to the actual value $\mathbb{E}U_n^{(m)}$ for small m and that the sum (10) is a good approximation to $\mathbb{E}U_n^{(m)}$ as n gets large.

As noted in the previous section, the sum (10) oscillates about $1/(m\ln 2)$. We would like to note that the oscillation is not an artifact of our approximation. The data show that the actual value of $\mathbb{E}U_n^{(m)}$ does itself oscillate about $1/(m\ln 2)$. This is illustrated in Figures 5, 6, and 7 for $m = 1, 5, 10$, respectively. These plots use successively coarser scales and show how the amplitude of the oscillation of $\mathbb{E}U_n^{(m)}$ about $1/(m\ln n)$ increases as m increases.

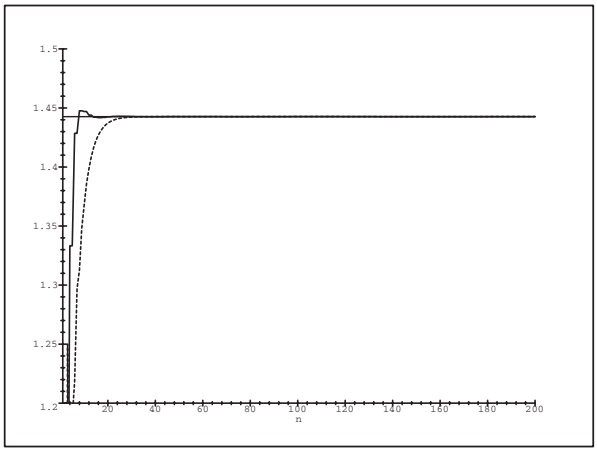


FIG. 2. *Comparison of* $\mathbb{E}U_n^{(1)}$ *(dotted) the approximating sum* (10) *and* $1/(\ln 2)$ *(bold).*

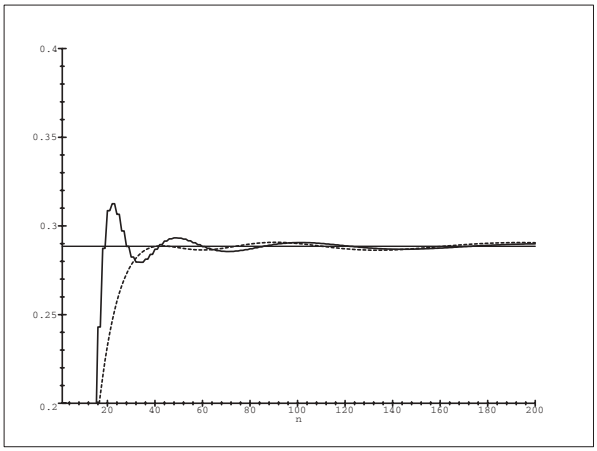


FIG. 3. *Comparison of* $\mathbb{E}U_n^{(5)}$ *(dotted), the approximating sum* (10)*, and* $1/(5\ln 2)$ *(bold).*

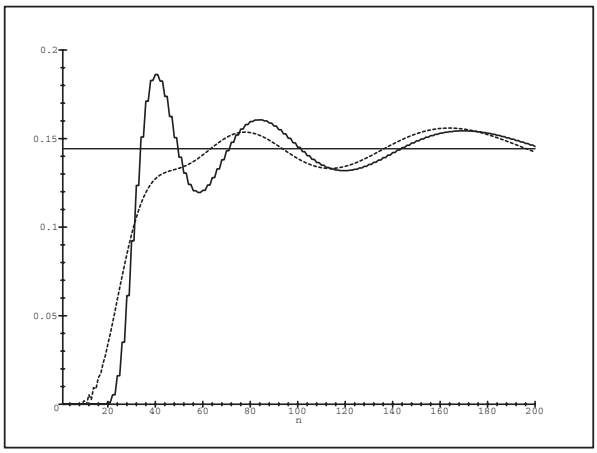


FIG. 4. *Comparison of* $\mathbb{E}U_n^{(10)}$ *(dotted) the approximating sum* (10) *and* $1/(10\ln 2)$ *(bold).*

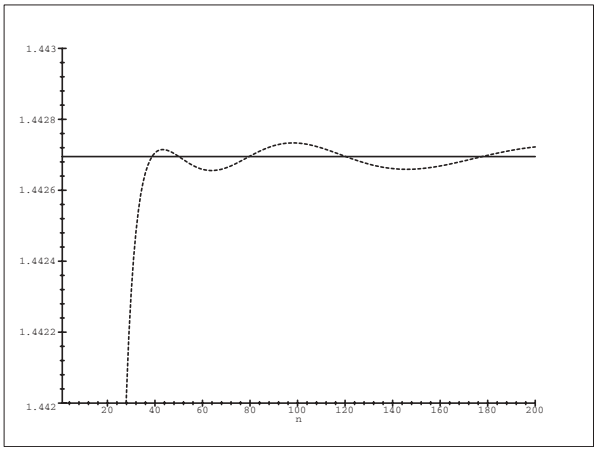


FIG. 5. *The oscillation of* $\mathbb{E}U_n^{(1)}$ *about* $1/(\ln 2)$.

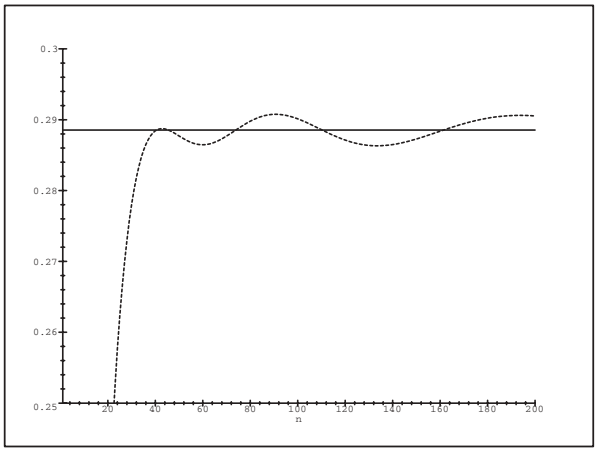


FIG. 6. *The oscillation of* $\mathbb{E}U_n^{(5)}$ *about* $1/(5\ln 2)$.

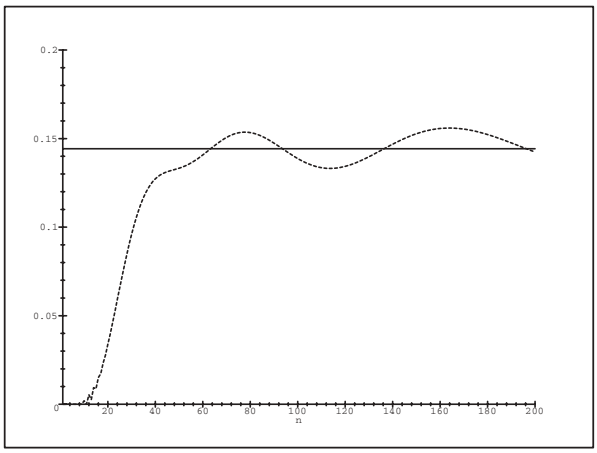


FIG. 7. *The oscillation of* $\mathbb{E}U_n^{(10)}$ *about* $1/(10\ln 2)$.

In order to bound the amplitude of the oscillation, note that the coefficients of Fourier expansion of the function $H^{(m)}/\ln 2$ (which is asymptotic to $\mathbb{E}U_n^{(m)} - 1/(m\ln 2)$) satisfy

$$\sum_{\ell\neq 0}|\phi_\ell| = \frac{1}{m!\ln 2}\sum_{\ell\neq 0}\left|\Gamma\left(m-\frac{2\pi i\ell}{\ln 2}\right)\right|,$$

and therefore,

$$\limsup_n\left|\mathbb{E}U_n^{(m)} - \frac{1}{m\ln 2}\right| \le \left|\frac{H^{(m)}(c\ln n)}{\ln 2}\right| \le \frac{1}{m!\ln 2}\sum_{\ell\neq 0}\left|\Gamma\left(m-\frac{2\pi i\ell}{\ln 2}\right)\right|.$$

Using the properties of gamma function,

$$|\Gamma(it)| = \sqrt{\frac{\pi}{t\sinh(\pi t)}} \quad\text{and}\quad \Gamma(z+1) = z\Gamma(z),$$

and letting $\rho = 2\pi/\ln 2$, we get a bound on the oscillation:

$$\begin{aligned}\frac{1}{m!\ln 2}\sum_{\ell\neq 0}\left|\Gamma(m-\rho\ell i)\right| &\le \frac{2}{m!\ln 2}\sum_{\ell=1}^{\infty}\left(\prod_{k=0}^{m-1}|k-\rho\ell i|\right)\sqrt{\frac{\pi}{\rho\ell\sinh(\pi\rho\ell)}}\\ &= \frac{\sqrt{2}}{m!\sqrt{\ln 2}}\sum_{\ell=1}^{\infty}\left(\prod_{k=0}^{m-1}\sqrt{k^2+\rho^2\ell^2}\right)\frac{1}{\sqrt{\ell\sinh(\pi\rho\ell)}}. \qquad (11)\end{aligned}$$

For small m, this bound is very good, but as illustrated in Table 1, as m increases it becomes increasingly weaker. In fact, for m exceeding the value 52, it becomes useless, as this bound on the amplitude exceeds the mean value, $1/(m\ln 2)$. (Thus, from this bound on the oscillation, one could not even conclude that the quantity (10) is positive.) A more detailed analysis of the nature of these fluctuations is perhaps an interesting question but we do not pursue it much further in this paper. One thing worth pointing out is that oscillations of $\mathbb{E}U_n^{(m)}$ are highly nonsymmetric around $1/(m\ln 2)$. On one hand, considering q of the form $m2^p$ and replacing the sum over j by its largest term (corresponding to $j=p$) we find that

$$\limsup_n \mathbb{E}U_n^{(m)} \ge \binom{q}{m}\left(\frac{m}{q}\right)^m\left(1-\frac{m}{q}\right)^{q-m} \sim \frac{m^m}{m!}e^{-m} \ge \frac{e^{-1/(12m)}}{\sqrt{2\pi m}}$$

by Stirling's formula. On the other hand, we have

$$\liminf_n \mathbb{E}U_n^{(m)} \ge \frac{e^{-2m}}{m\ln 2}.$$

To see this, let

$$f(x) = \binom{q}{m}2^{-mx}(1-2^{-x})^{q-m},$$

so that

$$\mathbb{E}U_n^{(m)} \sim \sum_{j=1}^{\infty} f(j).$$

TABLE 1
Comparison of the bound (11) *with the mean value as* m *increases.*

m	The bound (11) on oscillation	$1/(m \ln 2)$
1	.00001426024765	1.442695041
2	.00006502473820	.7213475205
3	.0002012028112	.4808983470
4	.0004802854938	.3606737603
5	.0009517428766	.2885390082
6	.001642131452	.2404491735
7	.002550173579	.2060992916
8	.003650969724	.1803368801
9	.004904708738	.1602994490
10	.006265585898	.1442695041
⋮	⋮	⋮
50	.02756514237	.02885390082
51	.02757480454	.02828813806
52	.02757887758	.02774413540
53	.02757781675	.02722066115
54	.02757203860	.02671657484
55	.02756192443	.02623081892
56	.02754782372	.02576241145
57	.02753005703	.02531043932
58	.02750891844	.02487405243
59	.02748467827	.02445245832
60	.02745758499	.02404491735
⋮	⋮	⋮
100	.02546701322	.01442695041
150	.02295420798	.009617966940
200	.02098570878	.007213475205
250	.01942401432	.005770780164
300	.01813918847	.004808983470
350	.01704834346	.004121985831
400	.01610016200	.003606737603

Since

$$f'(x) = \binom{q}{m} \frac{\ln 2}{2^{mx}} \left(1 - \frac{1}{2^x}\right)^{q-m-1} \left(\frac{q}{2^x} - m\right),$$

f is increasing for $x < x_0$ and decreasing for $x > x_0$, where $x_0 = \log_2(q/m)$. Therefore, letting $k_0 = [\log_2(q/m)]$ be the integer part of x_0, we see that

$$\begin{aligned}\sum_{j=1}^{\infty} f(j) &= \sum_{j=1}^{k_0} f(j) + \sum_{j=k_0+1}^{\infty} f(j) \geq \int_0^{k_0} f(x)dx + \int_{k_0+1}^{\infty} f(x)dx \\ &= \int_0^{\infty} f(x)dx - \int_{k_0}^{k_0+1} f(x)dx.\end{aligned}$$

The first integral upon substitution $u = 2^{-x}$ is easily seen to be equal to $1/(m \ln 2)$,

while for the second one, letting $\delta = x_0 - k_0$ and then $x = x_0 + t$, we get

$$\begin{aligned}\int_{k_0}^{k_0+1} f(x)dx &= \int_{x_0-\delta}^{x_0+1-\delta} f(x)dx = \int_{-\delta}^{1-\delta} f(x_0+t)dt \\ &= \binom{q}{m}\int_{-\delta}^{1-\delta} \frac{1}{2^{mx_0}}\frac{1}{2^{mt}}\left(1-\frac{1}{2^{x_0+t}}\right)^{q-m} dt \\ &= \binom{q}{m}\left(\frac{m}{q}\right)^m \int_{-\delta}^{1-\delta} \frac{1}{2^{mt}}\left(1-\frac{m}{2^t q}\right)^{q-m} dt,\end{aligned}$$

which upon substitution $u = 2^{-t}$ becomes

$$\binom{q}{m}\left(\frac{m}{q}\right)^m \frac{1}{\ln 2}\int_{2^{\delta-1}}^{2^\delta} u^{m-1}\left(1-\frac{mu}{q}\right)^{q-m} du.$$

Using $\binom{q}{m} \le q^m/m!$ and letting $q \to \infty$ we see that the latter expression is bounded above by

$$\frac{m^m}{m!\ln 2}\int_{2^{\delta-1}}^{2^\delta} u^{m-1}e^{-mu}du \le \frac{1}{m!\ln 2}\int_0^{m2^\delta} u^{m-1}e^{-u}du,$$

and by a successive partial integration, and because $0 \le \delta \le 1$, we see that the last integral is no more than

$$\frac{1}{m!\ln 2}(m-1)!(1-e^{-m2^\delta}) \le \frac{1}{m\ln 2}(1-e^{-2m}).$$

Thus,

$$\int_{k_0}^{k_0+1} f(x)dx \le \frac{1}{m\ln 2}(1-e^{-2m}),$$

that is,

$$\mathbb{E}U_n^{(m)} \ge \frac{e^{-2m}}{m\ln 2},$$

and the argument is completed.

Acknowledgment. We would like to express our gratitude to Herbert Wilf for bringing to our attention the paper of Kirschenhofer and the work of Knuth on the alternating sums that are used in our analysis. Part of the research of the first author was carried out while he was visiting the Department of Mathematics of Lehigh University in the spring semester of 1999. He would like to thank the department for its hospitality.

REFERENCES

[1] N. Alon, and J. H. Spencer, *The Probabilistic Method*, 2nd ed., Wiley-Interscience, New York, 2000.
[2] G. E. Andrews, *The Theory of Partitions*, Addison–Wesley, Reading, MA, 1976.
[3] D. W. Boyd, *Losing Runs in Bernoulli Trials*, unpublished manuscript, 1972.
[4] S. Corteel, B. Pittel, C. D. Savage, and H. S. Wilf, *On the multiplicity of parts in a random partition*, Random Structures Algorithms, 14 (1999), pp. 185–197.

[5] B. Eisenberg, G. Stengle, and G. Strang, *The asymptotic probability of a tie for first place*, Ann. Appl. Probab., 3 (1993), pp. 731–745.
[6] P. Erdös and J. Lehner, *The distribution of the number of summands in the partitions of positive integer*, Duke Math. J., 8 (1941), pp. 335–345.
[7] P. Erdös and A. Rényi, *On a new law of large numbers*, J. Anal. Math., 23 (1970), pp. 103–111.
[8] P. Erdös and P. Révész, *On the length of the longest head–run*, in Topics in Information Theory, Colloq. Math. Soc. Janos Bolyai 16, North-Holland, Amsterdam, 1975, pp. 219–228.
[9] P. Flajolet and R. Sedgewick, *Mellin transform and asymptotics: Finite differences and Rice's integrals*, Theoret. Comput. Sci., 144 (1995), pp. 101–124.
[10] B. Fristedt, *The structure of random partitions of large integers*, Trans. Amer. Math. Soc., 337 (1993), pp. 703–735.
[11] W. M. Y. Goh and E. Schmutz, *The number of distinct part sizes in a random integer partition*, J. Combin. Theory Ser. A, 69 (1995), pp. 149–158.
[12] L. Gordon, M. F. Schilling, and M. S. Waterman, *An extreme value theory for long head runs*, Probab. Theory Related Fields, 72 (1986), pp. 279–287.
[13] L. J. Guibas and A. M. Odlyzko, *Long repetitive patterns in random sequences*, Z. Wahrsch. Verv. Gebiete, 53 (1980), pp. 241–262.
[14] P. Kirschenhofer, *A note on alternating sums*, Electron. J. Combin., 3 (1996), Paper #R7.
[15] P. Kirschenhofer and H. Prodinger, *The number of winners in a discrete geometrically distributed sample*, Ann. Appl. Probab., 6 (1996), pp. 687–694.
[16] D. E. Knuth, *The Art of Computer Programming.* Vol. III. *Sorting and Searching*, Addison-Wesley, Reading, MA, 1973.
[17] B. Pittel, *Confirming two conjectures about the integer partitions*, J. Combin. Theory Ser. A, 88 (1999), pp. 123–135.
[18] H. S. Wilf, *Three problems in combinatorial asymptotics*, J. Combin. Theory Ser. A, 35 (1983), pp. 199–207.

SIAM J. DISCRETE MATH.
Vol. 18, No. 3, pp. 437–450

SECURE HYPERGRAPHS: PRIVACY FROM PARTIAL BROADCAST*

MATTHEW FRANKLIN† AND MOTI YUNG‡

Abstract. A "partial broadcast channel" enables one processor to send the same message—simultaneously and privately—to a fixed subset of processors. Suppose that a collection of processors are connected by an arbitrary network of partial broadcast channels (a hypergraph). We initiate the study of necessary and sufficient conditions, complexity bounds, and protocols for individual processors to exchange private messages across this network. Private message exchange, in turn, enables the realization of general secure computation primitives. The model (motivated by various environments such as multicast network architectures and group communication in distributed systems) is an intermediate setting between the private channels model and the full information model, both of which have been investigated extensively in the last few years. We assume a computationally unlimited adversary (i.e., the information theoretic notion of security), and our techniques are combinatorial. Both the possibility and the polynomial-time feasibility of private message exchange are investigated.

Key words. secure communication, unconditional security, partial broadcast

AMS subject classifications. 94A60, 68M10, 94C15

DOI. 10.1137/S0895480198335215

1. Introduction.

This paper examines private communication derived from a seemingly "unprivate" environment. A collection of processors can communicate among themselves only via "partial broadcasts," i.e., by sending the same message simultaneously from one processor to a subset of processors. The processors can cooperate to execute protocols, but some of them may be dishonest. We initiate the study of secure computation in such an environment, focusing on the enabling primitive of secure message transmission. Given a partial broadcast network, can individual processors exchange messages privately?

The answers that we find depend on several factors: (a) the particular partial broadcast channels in the network; (b) the number and nature of dishonest processors; and (c) the efficiency of communication. We restrict our attention to passive attacks on privacy: No information about message contents is revealed to any (sufficiently small) coalition of t (gossiping, honest-but-curious) processors, assuming all processors execute protocols correctly. The protocols that we present are all extremely simple, have zero probability of error, and guarantee privacy in an information theoretic sense (even when the adversary is computationally unlimited). We model partial broadcast networks as hypergraphs, and rely on combinatorial techniques for our results.

An example of a partial broadcast channel is a local area network (LAN) like an ethernet bus or a token ring. More modern network architectures implement multicast channels, enabling "one to many" ("point-to-multipoint") communication. In these architectures (e.g., [18]), a processor may belong to various such channels.

*Received by the editors March 6, 1998; accepted for publication (in revised form) August 6, 2003; published electronically December 30, 2004. A preliminary version of this work appeared in *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, Las Vegas, NV, 1995.
http://www.siam.org/journals/sidma/18-3/33521.html
†Department of Computer Science, University of California–Davis, 100 Shields Ave., Davis, CA 95616 (franklin@cs.ucdavis.edu).
‡Computer Science Department, Columbia University, New York, NY 10027 (moti@cs.columbia.edu).

When a processor member of a multicast channel sends a message, it is heard by all other processors on that channel. In our examples we assume architectures where the message is heard exclusively by its designated group. Since LANs are often somewhat structured, they may not be ideal for realizing complex networks of partial broadcast channels. Virtual private networks (VPNs), where the substructures are arbitrary, may be more appropriate.

Another example of a partial broadcast channel is a shared cryptographic key. By publishing an encrypted message, a processor initiates a partial broadcast to the subset of processors that are able to decrypt it. A partial broadcast network (against a polynomial adversary) arises from a collection of shared keys, e.g., from a key distribution scheme for conferences [6], where each processor has a subset of the set of keys (thus, this processor belongs to a subset of designated conferences, each conference corresponding to a key). An efficient system that uses cryptography to provide for "broadcast encryption" to a subset of users was suggested by Fiat and Naor [13].

Many distributed operation systems are being constructed currently to support partial broadcast, which is called "group communication primitive" [9]; this constitutes yet another example of a partial broadcast environment (see survey [7]). In particular, Dolev and Malki [11] solve basic distributed tasks in an environment supporting broadcast communication.

Radio networks are a fourth example of multicast channels. We remark that the "radio network model" studied by Alon et al. [2] is similar to one of the partial broadcast networks that we consider (our "neighbor network"). However, a main difference of their model is that a processor receives no messages (i.e., "hears only noise") if it is a recipient of two or more partial broadcasts simultaneously. Their work addresses issues of coordination and scheduling that arise in packet radio networks and does not consider privacy.[1]

The use of private channels as a primitive for general secure computation is shown by Ben-Or, Goldwasser, and Wigderson [3] and Chaum, Crépeau, and Damgård [8], and is further explored by many others (see survey [16]). Dolev et al. [10] consider distributed protocols for private communication over an incomplete network of private channels in a variety of fault scenarios. At the other extreme, the full information model of Ben-Or and Linial [4] assumes an environment with only full broadcast (from any processor to all other processors). Originally considered for the problem of flipping a global coin with bounded bias, it was extended to the question of general secure computation by Goldreich, Goldwasser, and Linial [17]. Though the full broadcast model cannot support unbiased secure computations, minimizing the influence of a coalition of misbehaving parties is the goal there. The partial broadcast model of this work addresses the area in between the private channels model and the full information model. Our problem can also be viewed as the inverse of the construction of a broadcast channel from private channels, namely, the Byzantine agreement problem [19], which has been a fundamental question in distributed computing theory. Note that much of this related work considers an active (Byzantine) adversary, unlike the passive (honest-but-curious) adversary that we consider in this paper.[2]

[1] See [12] for a general overview of the related notion of sensor networks, i.e., self-organizing networks of computers that are small, cheap, and low-power and communicate by radio-link.

[2] Subsequently, the partial broadcast model with an active adversary has been studied for secure communication [15] and for Byzantine agreement [14].

1.1. Simple motivating example. Consider a ring of four processors, labeled A, B, C, D, such that each processor can broadcast to its two neighbors. The partial broadcast network consists of only these four partial broadcast channels: $A \to \{B, D\}$; $B \to \{A, C\}$; $C \to \{B, D\}$; $D \to \{A, C\}$. Since every partial broadcast has either B or D as a participant, it is impossible, via any conceivable protocol, for A to send a message to C while keeping complete privacy from the coalition $\{B, D\}$ (unless the sender and receiver share secret information before the start of the protocol, or the coalition of listeners is computationally bounded, or senders can be anonymous [1]; our model excludes all of these possibilities). Using the terminology introduced in the next section, we say that 2-private communication from A to C is impossible on this partial broadcast network, because any attempt to send a message from A to C leaks information to a coalition of size two.

The following protocol enables A to send to C the bit string m 1-privately. Processor B generates a random bit string r_B (of the same length as m, from the uniform distribution) and partially broadcasts it to $\{A, C\}$; processor D does the same with r_D. Now A computes $m' = m \oplus r_B \oplus r_D$, and m' is propagated from A to C with the help of either (or both) B or D. At the end of this protocol, processor C can compute $m = m' \oplus r_B \oplus r_D$. From the point of view of processor B (D), all messages are indistinguishable, since every m is consistent with exactly one choice of r_D (r_B).

1.2. Our results. In section 3, we characterize when an arbitrary partial broadcast network can support private point-to-point communication, based on the directed and undirected connectivity of the hypergraph. One interpretation of this result is that privacy depends more on the "shape" of the hypergraph than on the directionality of its hyperedges. We then consider "subgraph networks" and "neighbor networks," which arise naturally from an arbitrary undirected graph connecting pairs of processors. For subgraph networks, privacy depends on the connectivity of the underlying graph. For neighbor networks, however, connectivity is not sufficient to determine privacy; we analyze several important graph topologies. We also show that deciding privacy is co-NP-complete, but is in polynomial time when only a constant number of processors are dishonest.

The protocols in section 3 have communication costs (round and bit complexity) that are polynomial in the length of the message and the size of the network. However, the size of the network (number of hyperedges) might not be polynomial in the number of processors. In section 4, we restrict our attention to protocols for which communication costs are polynomial in the number n of participating processors. For the case where up to t processors may be faulty, and where all partial broadcasts of size k are allowed, we prove bounds relating t, k, and n. Our most general upper and lower bounds, constraining the product tk, are within a log factor of each other.

2. Model and definitions. *Notation.* We write $|S|$ to denote the cardinality of the set S. When the universe of elements V is clear from context, we may write $\bar{U}$ to denote the set $V - U$. We write $ng(x)$ to denote the neighborhood of a vertex x in a graph $G = (V, E)$, i.e., $ng(x) = \{y \in V : (x, y) \in E\}$. A "star" centered at x is the set $\{x\} \cup ng(x)$.

We assume that the messages to be sent privately are elements of some finite group. We say that $m_1, \ldots, m_l$ are "random additive shares" of m if $\sum_{i=1}^{l} m_i = m$ and $m_1, \ldots, m_l$ are otherwise random. For example, $m_1, \ldots, m_{l-1}$ can be chosen from the group according to the uniform distribution, and $m_l = m - \sum_{i=1}^{l-1} m_i$. It is an elementary observation that no proper subset of random additive shares of m reveals

any information about m.

A "partial broadcast channel" enables any message to be sent from some sender x to some collection of receivers S. Only the members of S learn the message that was sent by x, while all parties outside of $S \cup \{x\}$ learn nothing about the contents of the message. A "simple channel" from x to y is a partial broadcast channel from x to the singleton set $S = \{y\}$.

We can represent a partial broadcast channel from x to S as a directed hyperedge connecting x to S. We can represent a collection of partial broadcast channels among n parties $[1 \cdots n]$ as a directed hypergraph on $V = [1 \cdots n]$. We say that a protocol among the n parties $[1 \cdots n]$ "simulates" a simple channel from x to y, $x, y \in [1 \cdots n]$ if the protocol begins with x choosing a message m and the protocol ends with y outputting m (with zero probability of error). The protocol may be randomized; i.e., the processors can flip coins during execution. We are interested in using a collection of partial broadcast channels on $[1 \cdots n]$ to simulate a complete network of simple channels on $[1 \cdots n]$.

We say that a protocol is t-private, $1 \leq t \leq n-2$, if no collection $L \subseteq [1 \cdots n] - \{x, y\}$, $|L| = t$, learns anything about the message m from the information it receives after honestly participating in the protocol. More formally, let the "view" of a participant be the record of its activity throughout the protocol: messages received and sent, coins flipped, and computations performed. Then the distribution of combined views of any t processors in $[1 \cdots n] - \{x, y\}$ at the end of the protocol should be independent of the message m that was chosen by x.

We may consider these t views to be seen at the end of the protocol by a single adversary of unbounded computational power. The adversary does not know the message m to be sent and has no advantage a priori in guessing the coin flips any processor will make during the protocol. The adversary can be assumed to know the identity of sender and receiver, the specification of the protocol, and auxiliary memory contents of all processors at the start of the protocol (i.e., other than coin flips or m itself). Since the adversary has unbounded computational power, this excludes protocols that rely on complexity theoretic assumptions for their privacy. All protocols have zero probability of error; i.e., correct message transmission is guaranteed.

We say that hypergraph H can simulate a complete network of simple channels t-privately if there exists a protocol using only the communication channels given by H to simulate the simple channel from x to y t-privately for every $x, y \in [1 \cdots n]$. Equivalently, we say that H "supports t-private point-to-point communication." In this paper, we investigate necessary and sufficient conditions on H and t for this task.

We write a directed hyperedge over a nodeset V as (v, S), where $v \in V$ and $S \subseteq V - \{v\}$. We say that there is a "directed link" from node v to node w if there exists a hyperedge (v, S) such that $w \in S$. We say that there is an "undirected link" from x to y if there is a directed link from x to y or a directed link from y to x. If there is a directed (undirected) link from v_i to v_{i+1} for every $i, 0 \leq i < k$, then we say that there is a "directed path" ("undirected path") from v_0 to v_k. A hypergraph is "strongly k-connected" ("weakly k-connected") if for all nodes $x, y \in V$, and for all $U \subset V - \{x, y\}, |U| < k$, there remains a directed (undirected) path from x to y after the removal of U and all hyperedges (v, S) such that $U \cap (S \cup \{v\}) \neq \emptyset$. An ordinary (nonhyper) undirected graph is k-connected if there are at least k vertex disjoint paths connecting any two nodes.

3. Achieving privacy. In this section, we consider the conditions under which t-private point-to-point communication can be simulated on an underlying network

H of partial broadcast channels. We establish a necessary and sufficient condition for private communication that depends more on the "shape" of the communication network than on its "direction." Then we consider private communication on subgraph networks and neighbor networks.

We are not concerned in this section with the efficiency of our simulations. In fact, all of our protocols will have round complexity proportional to the diameter of the network, and bit complexity polynominal in the size of the network. However, the size of the network is not necessarily polynominal in the number of processors: A network of n processors can have up to $n2^{n-1}$ distinct partial broadcast channels. In a later section, we treat the case of private simulation using resources that are polynominal in the number of processors.

3.1. Shape versus direction. The following theorem characterizes those partial broadcast networks that can support point-to-point private communication.

THEOREM 3.1. *t-private point-to-point communication is possible on H if and only if H is strongly 1-connected and weakly $(t+1)$-connected.*

LEMMA 3.2. *If H is not strongly 1-connected and weakly $(t+1)$-connected, then t-private point-to-point communication on H is impossible.*

Proof. If H is not strongly 1-connected, then there exist x and y such that any communication—private or not—from x to y is impossible.

If H is not weakly $(t+1)$-connected, we argue toward a contradiction of Shannon's impossibility result [20] for information theoretic secure two-party communication without a priori information. There exist nodes x, y and subset $S \subseteq V - \{x, y\}$, $|S| = t$, such that every undirected path from x to y hits S. Suppose there were a t-private message transmission protocol from x to y when the adversary controls S. We can find subsets V_x, V_y such that $V_x \cup V_y = \bar{S}$ and every undirected path from (any node in) V_x to (any node in) V_y hits S. The sender x could simulate the behavior of every party in V_x, without any communication to y. The receiver y could simulate the behavior of every party in V_y, without any communication to x. The simulation of the parties in S could be maintained jointly by x and y, with public conversations to agree on the coin flips of S and the messages received from outside S. By Shannon's result, this cannot be a secure two-party communication; i.e., a computationally unbounded eavesdropper listening to the public conversations could learn the message that x is sending to y. Thus the multiparty protocol cannot be secure. □

Lemma 3.3 establishes the reverse direction of the theorem. Central to the proof is the following protocol, which suffices for any party x to send a message m to any party y whenever the condition of the lemma is met.

1. For every hyperedge $e = (v, S)$ and for every $u \in S$, $v \to S : (u, r_{eu})$. That is, v broadcasts to every node in S the message "(u, r_{eu})." The second element of the broadcast message is r_{eu}, a random group element chosen by v from the message space according to the uniform distribution. The first element of the broadcast message is a "label" indicating that u is the "intended target" of r_{eu} (even though the message is of course seen by all of S). We say that v is the "originator" of r_{eu}. Note that a single hyperedge (v, S) causes v to broadcast $|S|$ distinct messages to S.
2. Every node $u \neq x$ computes the sum of messages for which it was the intended target and subtracts the sum of messages for which it was the originator: $T_u = (\sum_{e=(v,S),u\in S} r_{eu}) - (\sum_{e=(u,S),s\in S} r_{es})$. T_x is computed similarly and includes the message $m : T_x = m + (\sum_{e=(v,S),x\in S} r_{ex}) - (\sum_{e=(x,S),s\in S} r_{es})$.

3. For every $u \neq y$, T_u is communicated to y through any series of partial broadcasts, without concern for privacy. Then y can compute $m = \sum_u T_u$.

LEMMA 3.3. *If H is strongly 1-connected and weakly $(t+1)$-connected, then the above protocol achieves t-private point-to-point communication on H.*

Proof. Consider the relevant information in the combined views of $L \subseteq V - \{x, y\}$: any values r_{eu} sent or received by a party in L, and any sums T_u propagated through L on their way to y. In the worst case from a security standpoint, it is possible that every T_u except T_y passes through L on its way to y. It suffices to show that, for every L such that $|L| = t$, the distribution of WORST-CASE-VIEW$_L(m) = \{(u, r_{eu}) : u \in L\}$, $\{(u, r_{eu}) : e = (v, S), v \in L\}$, $\{(u, T_u) : u \neq y\}$ is independent of the choice of message m.

By weak $(t+1)$-connectivity, there exists an undirected path from x to y that misses L. Let $x_0 = x, x_1, \ldots, x_k = y$ be the nodes on this path, and let $e_1, \ldots, e_k$ be the corresponding hyperedges. The ith link from x_{i-1} to x_i is "forward" if $e_i = (v, S)$ where $v = x_{i-1}$ and $x_i \in S$. Otherwise, $v = x_i$ and $x_{i-1} \in S$ and the ith link is "backward."

For any execution of the protocol with message m, and for any group element Δ, consider the following modified execution: Add Δ to $r_{e_i x_i}$ if the ith link is forward, subtract Δ from $r_{e_i x_{i-1}}$ if the ith link is backward, and otherwise don't change the execution at all. The modification yields an execution of the protocol with message $m + \Delta$ for which the values of $\{(u, r_{eu}) : u \in L\}$, $\{(u, r_{eu}) : e = (v, S), v \in L\}$, $\{(u, T_u) : u \neq y\}$ are unchanged. Thus the distribution of WORST-CASE-VIEW$_L(m)$ is independent of the choice of message. □

3.2. Subgraph networks. In this section we consider t-private point-to-point communication on "subgraph networks." A subgraph network is a directed hypergraph derived from an undirected connected graph as follows. Let G be an undirected graph on n nodes. The (G, k)-subgraph network is the set of hyperedges such that there is an edge to S from $x \notin S$ if and only if $S \cup \{x\}$ is a connected subgraph of G of size k. Note that if G is connected, then the (G, k)-subgraph network is strongly 1-connected (unless $k > |V|$). The following are straightforward corollaries of Theorem 3.1.

COROLLARY 3.4. *t-private point-to-point communication on a (G, k)-subgraph network is possible if and only if $t + k \leq n$ and G is $(t+1)$-connected.*

Proof. Suppose G is $(t+1)$-connected and $t + k \leq n$. For every L of size t, and for every $x, y \notin L$, we need to show a path from x to y in the (G, k)-subgraph network that misses L. By $(t+1)$-connectivity, there is a path in G from x to y that misses L. If this path has length less than k, then a single hyperedge in the subgraph network connects x to y (consisting of the path together with enough neighboring nodes of $\bar{L}$). If the path from x to y in G has length at least k, then every length-k subsequence of the path corresponds to a hyperedge in the (G, k)-subgraph network. From these hyperedges we can construct the path from x to y in the subgraph network. The other direction is clear. □

A "complete k-ary network" is a (K_n, k)-subgraph network, where K_n is the complete graph on n nodes. It enables every processor to broadcast to any other $k-1$ processors. This network will be further studied in section 4.

COROLLARY 3.5. *t-private point-to-point communication is possible on a complete k-ary network if and only if $t + k \leq n$.*

3.3. Neighbor networks. In this section, we consider t-private point-to-point communication on "neighbor networks." Let $G = (V, E)$ be an undirected graph

on n nodes. The G-neighbor network is defined to be the set of hyperedges such that there is a hyperedge from x to $\{y : (x, y) \in E\}$ for every $x \in V$ (sometimes called a "star" centered at x). After giving some general technical conditions, we analyze some particular neighbor networks with quite different privacy properties. Although efficiency is not the focus of this section, we point out that the protocol from section 3.1 is always efficient when executed on a neighbor network; i.e., round complexity and bit complexity are polynominal in the number of processors n and the size of the message (since the size of a neighbor network is linear in the number of processors).

It is easy to show that t-private point-to-point communication on a G-neighbor network requires that G be $(t + 1)$-connected, as for subgraph networks. Unlike subgraph networks, however, that necessary condition is not, in general, sufficient. In the next few sections, we will illustrate how privacy can vary with respect to connectivity. Note that if G is connected, then its neighbor network is always strongly 1-connected.

3.3.1. Neighbor networks with minimal privacy. Although connectivity provides an upper bound on privacy, it does not, in general, imply any lower bound whatsoever. The simplest example of minimal privacy is the complete graph K_n, which cannot support 1-private communication. Moreover, *any* single node can prevent *every* pair of nodes from communicating privately.

For another example, let H^{2d} be the $2d$-dimensional hypercube, whose nodes are identified with $2d$-bit strings in the standard way. Let G be H^{2d} together with a special node z, where edges connect z to all hypercube nodes of weight $d - 1$, d, and $d + 1$. This graph is $(2d)$-connected, while even 1-privacy is unachievable on its neighbor network. Notice that although a single listener at z can prevent certain private communications (e.g., from low weight senders to high weight receivers), no single listener can cut off one node from *all* private communications.

3.3.2. Neighbor networks with maximal privacy. In this section, we show that certain graphs yield neighbor networks with maximal privacy, i.e., t-private communication when the graph is $(t+1)$-connected. The 4-node example from section 1.1 can be easily generalized to obtain 1-private communication between any two nodes on an arbitrary ring. It is straightforward to see that a ring neighbor network is strongly 1-connected and weakly 2-connected. To generalize to graphs of arbitrary connectivity $k \geq 3$, we use a construction in which a number of trees are "glued" together at the leaves in an unusual way.

THEOREM 3.6. *For every $k \geq 2$, there exists a k-connected graph G such that its neighbor network is weakly k-connected and strongly* 1*-connected.*

Proof. Let $T_{k,m}$ be a full tree of depth m and degree $k \geq 3$; i.e., the root has k children, every internal node has $k - 1$ children, and there are m levels excluding the root, $m \geq 2$. Let $G_{k,m}$ be k copies of $T_{k,m}$ with leaves identified ("glued together into superleaves") so that each superleaf corresponds to one leaf from each tree, and so that the following property holds: Every pair of superleaves are siblings in at most one tree. There are many ways to achieve such a mapping.

For example, the mapping of leaves to superleaves can be created as follows. Let $T^1, \ldots, T^k$ be the k copies of $T_{k,m}$. Choose k primes $p_1, \ldots, p_k$ between $k + 1$ and $k(k - 1)^{m-2}$. Label the leaves of T^i from left to right with increasing multiples of p_i mod $k(k-1)^{m-1}$, so that the jth leaf from the left gets the label jp_i mod $k(k-1)^{m-1}$. Let all leaves with label i get mapped to the ith superleaf of $G_{k,m}$. This mapping is a bijection of leaves to superleaves for each tree, since each p_i is relatively prime to

$k(k-1)^{m-1}$. Suppose that two superleaves are mapped from siblings in T^i. Then the difference between the labels of the superleaves must be $\lambda_i p_i \bmod k(k-1)^{m-1}$ for some $\lambda_i, 1 \leq \lambda_i \leq k-2$ (since siblings must be among $k-1$ adjacent leaves in T^i). If the same two superleaves are mapped from siblings in some other tree T^j, then we would have $\lambda_i p_i = \lambda_j p_j \bmod k(k-1)^{m-1}$, $1 \leq \lambda_i, \lambda_j \leq k-2$. This is impossible, since the two sides of the equality are both less than the modulus, and each contains a prime factor that is missing from the other.

To complete the proof, it suffices to show that $G_{k,m}$ is k-connected, and that the $G_{k,m}$-neighbor network is weakly k-connected. The details are in Appendix A. ∎

3.3.3. Hypercube neighbor networks: "Halfway-between" privacy. In this section, we consider neighbor networks in which the underlying graph is a hypercube. Let H^d denote the d-dimensional hypercube, with 2^d nodes and $d2^{d-1}$ edges. Although H^d is d-connected, t-private communication is possible only when $t < \lceil \frac{d+1}{2} \rceil$, i.e., roughly halfway between the minimal and maximal levels of privacy from a d-connected graph.

Let each node of H^d have a d-bit label in the standard way. Let $L = \{0^{d-1}1\} \cup \{0^{2i}110^{d-2i-2} : 0 \leq i \leq \lceil \frac{d-1}{2} \rceil - 1\}$. Then $|L| = \lceil \frac{d+1}{2} \rceil$, and L separates 0^d from the rest of H^d. The following theorem makes this bound tight. Proof details are in Appendix B.

THEOREM 3.7. *The H^d-neighbor network is strongly 1-connected and weakly $\lceil \frac{d+1}{2} \rceil$-connected for sufficiently large d.*

3.4. Decision complexity. In this section, we consider the complexity of deciding whether t-private point-to-point communication is possible for a given hypergraph H, sender x, and receiver y. The size of a problem instance depends on t and the size of H.

THEOREM 3.8. *Deciding the possibility of private point-to-point communication is co-NP-complete.*

Proof. Consider the complement problem of the impossibility of privacy. If there is no directed path from x to y, or if removing some subset of t nodes leaves no undirected path from x to y, then there is a short witness of this fact (e.g., the disconnecting set, together with the computation of the reachability partition of the nodeset). Thus the decision problem for impossibility can be shown to be in NP.

NP-hardness for the complement problem is shown by a reduction from vertex cover. Given a graph $G = (V, E)$, create a hypergraph H as follows. The nodes of H are $V \cup \{x, y\}$ for new nodes x, y. There is a hyperedge from x to $\{u, v\}$ for every $(u, v) \in E$, and a hyperedge from u to $\{y\}$ for every $u \in V$. A subset $L \subseteq V - \{x, y\}$ can weakly disconnect x and y in H if and only if L is a vertex cover for G. ∎

Notice that this reduction uses only hyperedges of size two and three. Notice also that the decision problem is polynominal time for constant t (e.g., the undirected connectivity of x and y can be tested for every subset L of size t in $V - \{x, y\}$).

The decision problem remains co-NP-complete when H is restricted to being a neighbor network by reducing dominating set to the impossibility problem. Given a graph $G = (V, E)$, let $G' = (V', E')$, where $V' = V \cup \{x, y\}$, and where $E' = E \cup \{(x, u) : u \in V\} \cup \{(u, y) : u \in V\}$. L is a dominating set of size t for G if and only if L weakly disconnects x from y in the G'-neighbor network.

4. Efficient privacy. We say that a simulation is "efficient" if private communication requires only a polynominal overhead in round and bit complexity, with respect to the number of processors n and the length of the message. The protocol

from section 3.1 is efficient only when the size of the partial broadcast network is polynominal in the number of processors. In this section, we analyze when t-private efficient point-to-point communication is possible given the ability to make partial broadcasts of size k. Specifically, the assumption throughout this section is that the underlying hypergraph H contains hyperedge (x, S) for every $|S| = k - 1$ and every $x \notin S$.

4.1. Requirements for k-ary networks.

THEOREM 4.1. *Efficient t-private point-to-point communication from partial broadcasts of size k requires that $tk = O(n \log n)$.*

Proof. Assume t is not a constant; otherwise the theorem is trivially true. Suppose that x sends a message to y by participating in a t-private protocol among the n parties wherein a total of l partial broadcasts (of size k) are sent. Let $\mathcal{S}' = \{S'_1, \ldots, S'_l\}$ denote the parties involved in these partial broadcasts. Let $\mathcal{S} = \{S_1, \ldots, S_l\}$ denote the parties remaining in the partial broadcasts after x and y are removed from consideration, i.e., $S_i = S'_i - \{x, y\}$. Then $k - 2 \leq |S_i| \leq k$ for all $1 \leq i \leq l$. Privacy is violated if there exists a set of parties $R \subseteq [1 \cdots n] - \{x, y\}$, $|R| = t$, such that $R \cap S_i \neq \emptyset$ for all $1 \leq i \leq l$. Thus for every $R, |R| = t$, there must exist an $S_j \in \mathcal{S}$ such that $R \subseteq \bar{S}_j$. There are $\binom{n-2}{t}$ choices for R. Privacy requires that some S_j must satisfy $R \subseteq \bar{S}_j$ for each possible R. However, each S_i can satisfy $R \subseteq \bar{S}_i$ for at most $\binom{n-2-|S_i|}{t} \leq \binom{n-k}{t}$ choices of R (i.e., the number of subsets of size t that miss x, y and the parties in S_i). Thus we have shown

$$l \geq \binom{n-2}{t} \Big/ \binom{n-k}{t.}. \tag{4.1}$$

Expanding (4.1), and ignoring constants that don't affect superpolynomiality, we have $|\mathcal{S}| \geq \frac{n!/(n-t)!t!}{(n-k)!/(n-t-k)!t!} = \frac{n!(n-t-k)!}{(n-t)!(n-k)!} \geq (\frac{n}{n-k})^t$, since $\frac{x}{y} < \frac{x-\lambda}{y-\lambda}$ whenever $x > y$ and $\lambda > 0$. If $(\frac{n}{n-k})^t = (1 + \frac{k}{n-k})^t$ is polynomial in n and t, then $\frac{k}{n-k} = O(\frac{\log n}{t})$. That is, there exists a $c > 0$ such that $\frac{k}{n-k} \leq \frac{c \log n}{t}$. Then $kt \leq c(n-k) \log n$, and thus $kt = O(n \log n)$. ∎

We observe that condition (4.1) also implies that $k \leq n - \Theta(\frac{n}{\log n})$ when t is not a constant.

4.2. Protocols for k-ary networks. Now we give some protocols for efficient private communication given a complete k-ary partial broadcast network. In fact, all of our protocols for private communication in this case are noninteractive. All messages are initiated by x and include y as a recipient. We emphasize that Corollary 3.5 in section 3.1 considers the same environment as the results of this section, but *without* requiring that communication costs be polynomial in n. The following two claims treat the cases where t or k is a constant.

CLAIM 1. *t-private point-to-point communication on a complete k-ary network is efficient whenever k is a constant and $t + k \leq n$.*

Proof. The protocol from Lemma 3.3 is efficient in this case, but we give an even more efficient protocol. The message m is split into $l = \binom{n-2}{k-2}$ random additive shares $m_1, \ldots, m_l$ by the sender x. Let $S_1, \ldots, S_l$ be all subsets of $[1 \cdots n] - \{x, y\}$ of size $k-2$. Then x sends m_i to $S_i \cup \{y\}$ for each $i, 1 \leq i \leq l$. This scheme is efficient, since the number of partial broadcasts is $\binom{n-2}{k-2} < n^k$, which is polynomial in n when k is a constant. The scheme is private: Each subset $L \in [1 \cdots n] - \{x, y\}$ of size t is disjoint with at least one subset $S_i \in [1 \cdots n] - \{x, y\}$ of size $k-2$, and thus an adversary controlling L misses the additive share m_i. ∎

CLAIM 2. *t-private point-to-point communication on a complete k-ary network is efficient whenever t is a constant and* $t + k \leq n$.

Proof. Suppose that x wishes to transmit to y. The message m is split into $l = \binom{n-2}{t}$ random additive shares $m, \ldots, m_l$. Let $S_1, \ldots, S_l$ be all subsets of $[1 \cdots n] - \{x, y\}$ of size t. Let $\hat{S}_i \subseteq [1 \cdots n] - (S_i \cup \{x, y\})$ be an arbitrary subset of size $k-2$; existence is guaranteed since $t + k \leq n$. Now x sends m_i to $\hat{S}_i \cup \{y\}$, for each $i, 1 \leq i \leq l$. This scheme is efficient, since the number of partial broadcasts is $l = \binom{n-2}{t} < n^t$, where t is a constant. The scheme is private, since an adversary controlling any S_i misses the additive share m_i. □

The following bound is within a logarithmic factor of the bound of Theorem 4.1.

THEOREM 4.2. *t-private point-to-point communication on a complete k-ary network is efficient whenever* $t + k \leq n$ *and* $tk = O(n)$.

Proof. We can assume that t and k are not constant, or else one of the preceding claims can be applied. Thus we can assume that $t < cn$ and $k < cn$ for every $c > 0$. Suppose that x wants to send a message to y. First x finds a positive integer m such that $tk < (m-1)n$. Then x chooses λm random additive shares of his message, where $\lambda = \lfloor \frac{n-2}{k+m-3} \rfloor$. Now x finds a partition of $[1 \cdots n] - \{x, y\}$ into λ disjoint subsets $S_1, \ldots, S_\lambda$ of size $k + m - 3$ each. Then x partially broadcasts a different share to $S_i^* \cup \{y\}$ for all i such that $1 \leq i \leq \lambda$, and for all $S_i^* \subseteq S_i$ such that $|S_i^*| = k-2$. The number of messages sent is $\lambda\binom{k+m-3}{k-2} < \lambda(k+m-3)^{m-1} < n(k+m-3)^{m-1}$, which is polynominal in n and k.

The adversary cannot learn any information about the message unless it intercepts every random additive share. Let $S_i' \subseteq S_i$ be the processors in S_i controlled by the adversary. If $|S_i'| < m$, then there exists at least one $S_i^* \subseteq S_i - S_i'$ such that $|S_i^*| = k-2$, and the adversary cannot intercept the share sent from x to $S_i^* \cup \{y\}$. Thus the adversary must control at least m processors in each $S_i, 1 \leq i \leq \lambda$. It suffices, then, to show that $m\lambda > t$. Since $tk < (m-1)n$, and since t is not constant, we have that $mn > tk + n > tk + n + (m^2 - m - 3t)$. Since $t, k < cn$ for all $c > 0$, we have that $tk + n + (m^2 - m - 3t) > tk + m(t+k) + (m^2 - m - 3t)$. Thus $mn - mk - m^2 + m > tk + tm - 3t$, and so $t < m\frac{n-k-m+1}{k+m-3} = m(\frac{n-2}{k+m-3} - 1) \leq m\lfloor \frac{n-2}{k+m-3} \rfloor = m\lambda$ as required. □

5. Conclusions. In conclusion, we have analyzed the possibility of achieving private point-to-point communication on a given network of partial broadcast channels. We have given exact characterizations for different sorts of partial broadcast networks, including those derived from subgraphs and neighborhoods of an arbitrary underlying graph. We have also considered simulations for which communication costs must be polynomial in the number of processors, and have given various bounds on relevant parameters for this case. We also investigated the computational complexity of deciding privacy.

There are many questions left unexplored, since the space of possible partial broadcast networks is vast. Finding better characterizations for natural classes of partial broadcast networks is a challenging problem, as is the simulation of secure channels on these networks versus stronger adversaries. There are also natural generalizations of neighbor networks. Define a (G, k)-neighbor network to be derived from an undirected graph G as follows: There is a hyperedge to S from x if and only if S are $k-1$ neighbors of x. What conditions on G, k, t allow t-private point-to-point communication? The same question could be asked for a network that allows broadcast from any processor x to all nodes in G at distance k or less from x.

Appendix A. Details of the proof of Theorem 3.6.

LEMMA A.1. *$G_{k,m}$ is k-connected.*

Proof. We show that there are k vertex disjoint paths connecting any pair of nodes x, y. Let T^x denote the tree containing x and let T^y denote the tree containing y. We divide our analysis into cases.

Case 1 is that x, y are in the same copy of $T_{k,m}$ and neither is a superleaf. Then there is a path from x to y, $k-1$ paths from x to superleaves $l_1, \ldots, l_{k-1}$, and $k-1$ paths from y to superleaves $l'_1, \ldots, l'_{k-1}$ such that all are vertex disjoint, and such that all are within that single tree. Then the remaining $k-1$ trees can complete the paths, e.g., a path from l_i to l'_i contained in the ith remaining tree.

Case 2 is that x, y are in the same tree and y is a superleaf. Then there is a path from x to y, and $k-1$ paths from x to superleaves $l_1, \ldots, l'_{k-1}$ such that all are vertex disjoint and all are within that single tree. Then the ith remaining tree can complete the path from l_i to y.

Case 3 is that x, y are both leaves. Then each of the k trees has its own path from x to y.

Case 4 is that x, y are nonleaves from different trees, and neither is a root. Find two superleaves l, l' such that exactly one is a descendant of x and exactly one is a descendant of y; this is always possible, no matter what mapping of leaves to superleaves has been used. There are vertex disjoint paths from x to l and from x to l' entirely within T^x; there are vertex disjoint paths from y to l and from y to l' entirely within T^y. Two vertex disjoint paths from x to y can be constructed from these four partial paths. There are $k-2$ remaining disjoint paths from x to leaves and $k-2$ disjoint paths from y to leaves, all disjoint and disjoint with the first two paths. Each of the remaining $k-2$ trees can join pairs of leaves to complete these paths.

Case 5 is that x, y are nonleaves from different trees, and exactly one of them is a root; without loss of generality, x is a root. Then there are two vertex disjoint paths in T^x from x to superleaves l, l', such that l is a descendant of y and l' is not. Then complete the two paths in T^y as in the previous case. Then find the $k-2$ remaining path as in the previous case.

The remaining case is that x, y are both roots. Then there are two vertex disjoint paths entirely within T^x to superleaves that have y as their least common ancestor in T^y. These paths can be completed in T^y. Then find the $k-2$ remaining paths as in the previous two cases. ∎

LEMMA A.2. *The $G_{k,m}$-neighbor network is weakly k-connected.*

Proof. Let L be any set of nodes of size $k-1$. We need to show that for any two nodes $x, y \in V - L$ there remains a weak path from x to y in the neighbor network after the removal of L (and all affected hyperedges).

We first claim that there remains a weak path from any nonleaf $x \in V - L$ to some leaf. Let T be the tree containing x. The proof is by induction on the (shortest) distance from x to a superleaf in $G_{k,m}$. The base case is when x is the parent of a superleaf. For any two children of x, their neighborhoods have only x in common (by the "tangled" mapping of leaves to superleaves). If L is disjoint with a star centered at a child of x, then this single star is a weak path from x to a leaf. If L hits the star centered at every child of x, then every element of L is a child of x or in a different tree than x. Choose a node $y \neq x$ in T such that y is the parent of a leaf. Then there is a weak path in T connecting x to the children of y, beginning with the star centered at y and ending with the star centered at the parent of x.

Suppose that x is at a distance d from a leaf, $d > 1$, and that x is not the root of T. If the star centered at every child of x hits L, then this accounts for all of L (since these neighborhoods have only x in common). Let y be any leaf that is not a descendant of x in T. Then there is a weak path in T from x to y, beginning with the star centered at the parent of y in T and ending with the star centered at the parent of x. Otherwise, the single star centered at some child of x is a weak path from x to nodes that are closer to leaves than x. By the induction hypothesis, this weak path can be completed. Lastly, if x is the root of T, then the star centered at one of its k children is disjoint with L. By the induction hypothesis, this weak path can be completed to a leaf.

Next, we claim that there remains a weak path between any two leaves $x, y \in V - L$. Suppose that L consists of λ leaves and μ nonleaves, $\lambda + \mu = k - 1$. The neighborhoods of any two parents of x have only x in common; the same is true for y. Thus $k - \lambda$ (or more) parents of x have no children in L; the same is true for y. For at least $k - \lambda - \mu \geq 1$ of these parents of x, the tree containing the parent has no nonleaves in L; the same is true for y. Let u be a parent of x in tree T_u such that the star centered at u is disjoint with L and such that T_u has no nonleaves in L; find a parent v of y in tree T_v similarly. For any tree in $G_{k,m}$, the stars centered at (at least) $k(k-1)^{m-2} - \lambda - \mu$ leaf parents are disjoint with L. Thus, when $k \geq 3$ and $m \geq 3$, the stars centered at more than half of the leaf parents are disjoint with L (i.e., $k(k-1)^{m-2} - \lambda - \mu \geq k > \lambda + \mu$). Thus there exists a leaf parent u' in T_u and a leaf parent v' in T_v such that their neighborhoods are disjoint with L but not disjoint with each other (i.e., have a leaf in common). Then the weak path from x to y consists of a weak path entirely in T_u (beginning with the star centered at u and ending with the star centered at u') concatenated to a path entirely within T_v (beginning with the star centered at v' and ending with the star centered at v).

To complete the proof, if $x, y \in V - L$ are both nonleaves, then a weak path between them can be formed by combining a weak path from x to a leaf, a weak path from y to a leaf, and a weak path between the leaves. □

Appendix B. Details of the proof of Theorem 3.7. For the proof of Theorem 3.7, we need the following four facts.

FACT 1 (Bollobas [5, Theorem 10.5]). *Let $S \subset H^d$, where $|S| = \sum_{i=0}^{r} \binom{d}{i}$. Then $|ng(S)| \geq \binom{d}{r+1}$.*

FACT 2. *Let $S, S' \subset H^d$, where $S' = S \cup \{x\}$ for some $x \in H^d$. Then* (a) $|ng(S')| \geq |ng(S)| - 1$ *and* (b) $|ng(S')| + |ng(S' \cup ng(S'))| \geq |ng(S)| + |ng(S \cup ng(S))| - 1$.

Proof of Fact 2. If $y \in ng(S)$, then $y \notin S$, and there exists $z \in S$ such that y and z are adjacent. Then $y \in ng(S')$ also, unless $y = x$. This suffices to prove part (a).

If $y \in ng(S \cup ng(S))$, then $y \notin S \cup ng(S)$, and there exists $z \in ng(S)$ such that y and z are adjacent. Then $y \in ng(S' \cup ng(S'))$ as well, unless $y = x$ or $y \in ng(x)$. Furthermore, $y \in ng(S')$ when $y \in ng(x)$.

Thus $ng(S) \cup ng(S \cup ng(S)) - \{x\} \subseteq ng(S') \cup ng(S' \cup ng(S'))$. Then $|ng(S')| + |ng(S' \cup ng(S'))| \geq |ng(S)| + |ng(S \cup ng(S))| - 1$. This proves part (b). □

FACT 3. $|ng(S)| + |ng(S \cup ng(S))| \geq d^2 - d$ for all $S \subset H^d$, $|S| = 2$, for sufficiently large d.

Proof of Fact 3. We prove for $d \geq 4$. Let $S = \{x, y\}$. By symmetry, the size of the neighborhoods will depend only on the distance from x to y. The relevant neighborhoods of x and y are disjoint when the distance from x to y is greater than four,

so only four cases need to be considered: $\{0^d, 0^{d-1}1\}$, $\{0^d, 0^{d-2}11\}$, $\{0^d, 0^{d-3}111\}$, $\{0^d, 0^{d-4}1111\}$.

When $S = \{0^d, 0^{d-1}1\}$, $ng(S) \cup ng(S \cup ng(S))$ contains $(d-1)$ nodes of weight one (all but one), $\frac{1}{2}d(d-1)$ nodes of weight two (all), and $\frac{1}{2}(d-1)(d-2)$ nodes of weight three (all ending in 1). The sum is $d^2 - d$.

When $S = \{0^d, 0^{d-2}11\}$, $ng(S) \cup ng(S \cup ng(S))$ contains d nodes of weight one (all), $\frac{1}{2}d(d-1) - 1$ nodes of weight two (all but one), $(d-2)$ nodes of weight three (all ending in 11), and $\frac{1}{2}(d-2)(d-3)$ nodes of weight four (all ending in 11). The sum is $d^2 - d$.

When $S = \{0^d, 0^{d-3}111\}$, $ng(S) \cup ng(S \cup ng(S))$ contains d nodes of weight one (all), $\frac{1}{2}d(d-1)$ nodes of weight two (all), $3(d-3)$ nodes of weight three (all ending in 011, 101, 110), $(d-3)$ nodes of weight four (all ending in 111), and $\frac{1}{2}(d-3)(d-4)$ nodes of weight five (all ending in 111). The sum is $d^2 + d - 6 \geq d^2 - d$.

When $S = \{0^d, 0^{d-4}1111\}$, $ng(S) \cup ng(S \cup ng(S))$ contains d nodes of weight one (all), $\frac{1}{2}d(d-1)$ nodes of weight two (all), 4 nodes of weight three (all starting 0^{d-4}), $4(d-4)$ nodes of weight four (all ending 0111, 1011, 1101, 1110), $(d-4)$ nodes of weight five (all ending in 1111), and $\frac{1}{2}(d-4)(d-5)$ nodes of weight six (all ending in 1111). The sum is $d^2 + d - 6 \geq d^2 - d$. □

FACT 4. *$|ng(S)| + |ng(S \cup ng(S))| > (d+1)\left\lceil \frac{d+1}{2} \right\rceil$, for all $S \subset H^d$, $1 < |S| < 2^{d-1}$, for sufficiently large d.*

Proof of Fact 4. We prove for $d \geq 7$. The case $|S| = 2$ is handled by Fact 3 above. When $2 < |S| \leq d$, let $R \subset S$, $|R| = 2$. By Fact 3 together with Fact 2(b), we have $|ng(S)| + |ng(S \cup ng(S))| \geq |ng(R)| + |ng(R \cup ng(R))| - |S - R| \geq d^2 - d + 2 - |S|$. Thus $|ng(S)| + |ng(S \cup ng(S))| \geq d^2 - d + 2 - d > (d+1)\left\lceil \frac{d+1}{2} \right\rceil$ for $d \geq 7$.

Otherwise $\sum_{i=0}^{j} \binom{d}{i} \leq |S| < \sum_{i=0}^{j+1} \binom{d}{i}$ for some j, $1 \leq j \leq \frac{d}{2} - 1$. Let $R \subset S$, $|R| = \sum_{i=0}^{j} \binom{d}{i}$. Then $|ng(S)| + |ng(S \cup ng(S))| \geq |ng(R)| + |ng(R \cup ng(R))| - |S - R|$ by Fact 2(b). By Fact 1, $|ng(R)| = \binom{d}{j+1} + \Delta$ for some $\Delta \geq 0$. By Fact 1 together with Fact 2(a), $|ng(R)| + |ng(R \cup ng(R))| \geq \binom{d}{j+1} + \Delta + \binom{d}{j+2} - \Delta = \binom{d}{j+1} + \binom{d}{j+2}$. Thus $|ng(S)| + |ng(S \cup ng(S))| \geq \binom{d}{j+i} + \binom{d}{j+2} - |S - R| > \binom{d}{j+2} \geq \binom{d}{3} = \frac{1}{6}d(d-1)(d-2) > (d+1)\left\lceil \frac{d+1}{2} \right\rceil$ for $d \geq 7$. □

Proof of Theorem 3.7. Let L be any subset of nodes of H^d, $|L| < \left\lceil \frac{d+1}{2} \right\rceil$. Toward a contradiction, suppose that no path remains from x to y after the removal from H^d of L and all hyperedges that meet L. Let S be a maximal subset of $V - L$ such that $x \in S$, such that S is either a singleton or a union of stars in H, and such that no path remains from z to y for any $z \in S$. Without loss of generality, $|S| < 2^{d-1}$ (otherwise use y instead of x in the construction of S).

If $S = \{x\}$, then suppose that L contains k neighbors of x, $0 \leq k \leq d$. Then stars centered at the remaining $d - k$ neighbors of x must meet elements of L in $ng(\{x\} \cup ng(x))$. But every element of $ng(\{x\} \cup ng(x))$ belongs to exactly two stars with centers in $ng(x)$. Thus we need $t \geq k + \left\lceil \frac{d-k}{2} \right\rceil \geq 1 + \left\lceil \frac{d-1}{2} \right\rceil$.

If $S = \{x\} \cup ng(x)$ (a single star), then suppose that L contains k elements of $ng(S)$, $0 \leq k \leq |ng(S)| = \frac{1}{2}d(d-1)$. Then stars centered at the remaining $\frac{1}{2}d(d-1) - k$ elements of $ng(S)$ must hit elements of L in $ng(S \cup ng(S))$. But every element of $ng(S \cup ng(S))$ belongs to exactly three stars with centers $ng(S)$. Thus we need $t \geq k + \left\lceil \frac{1}{3}(\frac{1}{2}d(d-1) - k) \right\rceil \geq 1 + \left\lceil \frac{1}{3}(\frac{1}{2}d(d-1) - 1) \right\rceil \geq \left\lceil \frac{d+1}{2} \right\rceil$, $d \geq 5$.

When S is a union of more than one star, Fact 4 implies that the attempted disconnection must fail when $|ng(S)| + |ng(V-S)| > |L| + |ng(L)|$. But $|L| + |ng(L)| \leq (d+1)|L|$, so it suffices to show that $|ng(S)| + |ng(V - S)| \geq (d+1)\left\lceil \frac{d+1}{2} \right\rceil$. Thus it

suffices to show that $|ng(C)| + |ng(C \cup ng(C))| \geq (d+1)\left\lceil \frac{d+1}{2} \right\rceil$ for all nodesets C, $1 < |C| < 2^{d-1}$; here C is the set of centers of stars in S. By Fact 4, this is true for $d \geq 7$, completing the proof of the theorem. □

Acknowledgment. We thank Peter Winkler for his contributions to the decision complexity results of section 3.4.

REFERENCES

[1] B. Alpern and F. Schneider, *Key exchange using "keyless" cryptography*, Inform. Process. Lett., 16 (1983), pp. 79–82.

[2] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg, *A lower bound for radio broadcast*, J. Comput. System Sci., 43 (1991), pp. 290–298.

[3] M. Ben-Or, S. Goldwasser, and A. Wigderson, *Completeness theorems for non-cryptographic fault-tolerant distributed computation*, in Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC), Chicago, IL, 1988, pp. 1–9.

[4] M. Ben-Or and N. Linial, *Collective coin flipping and other models of imperfect randomness*, in Randomness and Computation, S. Micali, ed., JAI Press, Greenwich, CT, 1989, pp. 91–115.

[5] B. Bollobas, *Combinatorics*, Cambridge University Press, Cambridge, UK, 1986.

[6] M. Burmester and Y. Desmedt, *A secure and efficient conference key distribution system*, in Advances in Cryptology-Eurocrypt '94, Lecture Notes in Comput. Sci. 1189, Springer-Verlag, Berlin, 1995, pp. 275–286.

[7] S. Chanson, G. Neufeld, and L. Liang, *A bibliography on multicast and group communications*, ACM Operating Systems Review, 23 (1989), pp. 20–25.

[8] D. Chaum, C. Crépeau, and I. Damgård, *Multiparty unconditionally secure protocols*, in Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC), Chicago, IL, 1988, pp. 11–19.

[9] D. Cheriton and W. Zwaenepoel, *Distributed process group in the V kernel*, ACM Trans. Comput. Systems, 3 (1985), pp. 77–107.

[10] D. Dolev, C. Dwork, O. Waarts, and M. Yung, *Perfectly secure message transmission*, J. ACM, 40 (1993), pp. 17–47.

[11] D. Dolev and D. Malki, *On distributed algorithms in a broadcast domain*, in Proceedings of the 20th International Colloquium on Automata, Languages and Programming (ICALP), Lecture Notes in Comput. Sci. 700, Springer-Verlag, Heidelberg, Germany, 1993, pp. 371–387.

[12] D. Estrin, D. Culler, K. Pister, and G. Sukhatme, *Connecting the physical world with pervasive networks*, IEEE Pervasive Computing, 1 (2002), pp. 59–69.

[13] A. Fiat and M. Naor, *Broadcast encryption*, in Advances in Cryptology-Crypto '93, Lecture Notes in Comput. Sci. 773, Springer-Verlag, New York, 1994, pp. 480–491.

[14] M. Fitzi and U. Maurer, *From partial consistency to global broadcast*, in Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (STOC), Portland, OR, 2000, pp. 494–503.

[15] M. Franklin and R. Wright, *Secure communication in minimal connectivity models*, J. Cryptology, 13 (2000), pp. 9–30.

[16] M. Franklin and M. Yung, *Varieties of secure distributed computing*, in Proceedings of the PTOC, Sequences II, Methods in Communications, Security and Computer Science, Positano, Italy, 1991, pp. 392–417.

[17] O. Goldreich, S. Goldwasser, and N. Linial, *Fault-tolerant computation in the full information model*, SIAM J. Comput., 27 (1998), pp. 506–544.

[18] I. Gopal and J. Jaffe, *Point-to-multipoint communication over broadcast links*, IEEE Trans. Communications, 32 (1982), pp. 1034–1044.

[19] L. Lamport, R. Shostak, and M. Pease, *The Byzantine generals problem*, ACM Trans. Programming Lang. Systems (1982), pp. 382–401.

[20] C. Shannon, *Communication theory of secrecy systems*, Bell Sys. Tech. J., 30 (1949), pp. 656–715.

SIAM J. DISCRETE MATH.
Vol. 18, No. 3, pp. 451–461

COLORING POWERS OF CHORDAL GRAPHS*

DANIEL KRÁL'†

Abstract. We prove that the kth power G^k of a chordal graph G with maximum degree Δ is $O(\sqrt{k}\Delta^{(k+1)/2})$-degenerate for even values of k and $O(\Delta^{(k+1)/2})$-degenerate for odd values. In particular, this bounds the chromatic number $\chi(G^k)$ of the kth power of G. The bound proven for odd values of k is the best possible. Another consequence is the bound $\lambda_{p,q}(G) \leq \lfloor \frac{(\Delta+1)^{3/2}}{\sqrt{6}} \rfloor (2q-1) + \Delta(2p-1)$ on the least possible span $\lambda_{p,q}(G)$ of an $L(p,q)$-labeling for chordal graphs G with maximum degree Δ. On the other hand, a construction of such graphs with $\lambda_{p,q}(G) \geq \Omega(\Delta^{3/2}q + \Delta p)$ is found.

Key words. chordal graphs, graph powers, graph coloring, $L(p,q)$-labeling

AMS subject classifications. 05C15, 05C62, 68R10

DOI. 10.1137/S0895480103424079

1. Introduction.

The concept of an $L(p,q)$-labeling of graphs, in particular that of an $L(2,1)$-labeling, is an important graph-theoretical model for assignment of radio frequencies, which is intensively studied both from the combinatorial and algorithmic points of view [1, 4, 8, 9, 14]. An *$L(p,q)$-labeling*, $p \geq q \geq 1$, is an assignment of numbers $0, \ldots, K$ to the vertices of an input graph G such that each two adjacent vertices receive numbers which differ by at least p and each two vertices at distance two receive numbers which differ by at least q. The number K is called the *span*, and the minimum span for which a proper labeling exists is denoted by $\lambda_{p,q}(G)$.

A study of the relation between the minimum span of an $L(2,1)$-labeling and the maximum degree Δ of a graph G was stated in the paper of Griggs and Yeh [11]. They conjectured that each graph G has an $L(2,1)$-labeling with the span at most Δ^2 and proved the upper bound of $\Delta^2 + 2\Delta$. This bound was later improved to $\Delta^2 + \Delta$ by Chang et al. [6]. The best currently known upper bound of $\Delta^2 + \Delta - 1$ is a consequence of a recent result for the channel assignment problem of the author and Škrekovski [13]. The conjecture of Griggs and Yeh remains unsettled but is known to be true for several special classes of graphs, among them chordal graphs. The upper bound of $(\Delta+3)^2/4$ for chordal graphs was proved by Sakai [15]. The general bound of $(\Delta + 2p - 1)^2/4$ for the case of $L(p,1)$-labelings of chordal graphs can be found in [7]. In the present paper, we prove asymptotically optimal bounds of $O(\Delta^{3/2})$ on the minimum span of an $L(2,1)$-labeling and $O(\Delta^{3/2}q + \Delta p)$ on the minimum span of an $L(p,q)$-labeling for chordal graphs with maximum degree Δ.

1.1. Results.

We are actually concerned with a more general problem to bound the chromatic number of powers of a chordal graph G in terms of the maximum degree of G. A graph G is said to be *chordal* if it contains no induced cycle of length four or more. It is well known that a graph is chordal if and only if it has a perfect elimination sequence. A *perfect elimination sequence* is an ordering $v_1, \ldots, v_n$ of vertices of G such that for each i, the neighbors of the vertex v_i preceding it in the sequence form a clique

*Received by the editors March 7, 2003; accepted for publication (in revised form) March 11, 2004; published electronically December 30, 2004.
http://www.siam.org/journals/sidma/18-3/42407.html

†Institute for Theoretical Computer Science (ITI), Charles University, Malostranské náměstí 25, 118 00 Prague, Czech Republic (kral@kam.mff.cuni.cz). The Institute for Theoretical Computer Science is supported by the Ministry of Education of Czech Republic as project LN00A056.

in G. The *kth power* G^k of a graph G is the graph on the same vertex set such that two vertices in G^k are joined by an edge if their distance in G is at most k. Note that $\chi(G^2) = \lambda_{1,1}(G) + 1$. It is known [3] that if the girth of G is at least 7, then $\chi(G^2) \leq O(\Delta^2/\log \Delta)$, and otherwise the chromatic number of G may reach $\Theta(\Delta^2)$ (this is witnessed by bipartite incidence graphs of finite projective planes). Our results establish that $\chi(G^2) \leq O(\Delta^{3/2})$ if G is chordal.

More generally, we show in Theorem 3.1 that if G is a chordal graph with maximum degree Δ, then G^k is $O(\sqrt{k}\Delta^{(k+1)/2})$-degenerate, i.e., each subgraph H of G^k contains a vertex of degree at most $O(\sqrt{k}\Delta^{(k+1)/2})$. In Theorem 4.1, a better and sharp bound for odd k's is proven, namely, that G^k is $O(\Delta^{(k+1)/2})$-degenerate. It is quite surprising that similar upper bounds also hold for the class of planar graphs [2].

Theorems 3.1 and 4.1 have several interesting corollaries. First, if G is a chordal graph with maximum degree Δ, then the chromatic number of its kth power G^k is at most $O(\sqrt{k}\Delta^{(k+1)/2})$ if k is even and at most $\Delta^{(k+1)/2}/4 + O(\Delta^{(k-1)/2})$ if k is odd. Second, the minimum span of an $L(2,1)$-labeling of G is at most $O(\Delta^{3/2})$ and, generally, the minimum span of an $L(p,q)$-labeling of G is at most $O(\Delta^{3/2}q + \Delta p)$. All our results are asymptotically tight (for a fixed k) and the presented upper bound for odd powers of chordal graphs is the best even possible (cf. Theorems 4.1 and 5.1). In section 5, we describe a construction of chordal graphs G on n vertices with maximum degree Δ such that G^k is a clique for $n = \Omega(\Delta^{(k+1)/2})$. The lower bound $\Omega(\Delta^{3/2}q + \Delta p)$ for the minimum span of an $L(p,q)$-labeling of chordal graphs is then presented in Corollary 5.4.

The shown upper and lower bounds match for odd powers of chordal graphs, but they do not match for even powers, although in the case of the second power and $L(2,1)$-labeling, the proven leading coefficients are quite close. In Corollary 3.2, we establish the following estimate on the chromatic number of the second power of a chordal graph G with maximum degree Δ:

$$\chi(G^2) \leq \left\lfloor \frac{(\Delta+1)^{3/2}}{\sqrt{6}} \right\rfloor + \Delta + 1 \approx 0.4082\Delta^{3/2} + O(\Delta).$$

And, in Theorem 3.3, we show the following bound on the span of its $L(2,1)$-labeling:

$$\lambda_{2,1}(G) \leq \frac{\Delta^{3/2}}{\sqrt{6}} + O(\Delta) \approx 0.4082\Delta^{3/2} + O(\Delta).$$

On the other hand, there exists a chordal graph G with maximum degree Δ such that (see Theorem 5.3)

$$\lambda_{2,1}(G) \geq \chi(G^2) - 1 \geq \frac{2\Delta^{3/2}}{3\sqrt{3}} - O(\Delta^{107/84}) \approx 0.3849\Delta^{3/2} - O(\Delta^{107/84}).$$

We conjecture that the lower bound is tight.

CONJECTURE 1.1. *If G is a chordal graph with maximum degree Δ, then*

$$\chi(G^2) \leq \lambda_{2,1}(G) + 1 \leq \frac{2\Delta^{3/2}}{3\sqrt{3}} + O(\Delta).$$

1.2. Separators in chordal graphs. The proof of the upper bound in Theorem 3.1 is based on a partial separator lemma which we prove in section 2. The lemma seems to be of independent interest. We call Lemma 2.1 a *partial separator*

lemma because we cut only a part of the set U (of size between κ and 2κ) from the rest of a graph. We remark that the following separator theorem for chordal graphs was proved by Gilbert, Rose, and Edenbrandt [10]: For each chordal graph G and each set $U \subseteq V(G)$, there is a clique C in G such that every component of $G \setminus C$ contains at most $|U|/2$ vertices of U. However, the proof of their result does not seem adaptable to our case.

2. A partial separator lemma. It is well known [5] that the class of chordal graphs is precisely the class of intersection graphs of subtrees of a tree, i.e., for each chordal graph G, there exists a tree $\mathcal{T}$ and a mapping which assigns to each vertex v of G a subtree T_v of $\mathcal{T}$ which have the following property: The vertices v and v' of G are joined by an edge if and only if $T_v \cap T_{v'} \neq \emptyset$. Such a representation of a chordal graph can be modified to a little more restricted representation which we will call a *nice tree representation* of a chordal graph. We leave details of the straightforward proof of the next proposition to the reader ($A \div B$ denotes the symmetric difference of sets A and B).

If G is a chordal graph, then there exists a rooted tree $\mathcal{T}$ and a mapping which assigns to each vertex $v \in V(G)$ a subtree T_v of $\mathcal{T}$. For a vertex w of the tree $\mathcal{T}$, let $\tau_w = \{v \in V(G) \mid w \in T_v\}$. The rooted tree $\mathcal{T}$ and the mapping have the following properties:

(i) $T_v \cap T_w \neq \emptyset$ for $v, w \in V(G)$ if and only if $vw \in E(G)$, i.e., $\mathcal{T}$ and the mapping is an intersection representation of G.

(ii) Each vertex of $\mathcal{T}$ has at most two children.

(iii) If w is a leaf or the root of $\mathcal{T}$, then $\tau_w = \emptyset$.

(iv) If w is a vertex of $\mathcal{T}$ with a single child w', then $|\tau_w \div \tau_{w'}| \leq 1$.

(v) If w is a vertex of $\mathcal{T}$ with two children w' and w'' and a parent w_0, then $\tau_w = \tau_{w_0} = \tau_{w'} = \tau_{w''}$.

Sketch of proof. Let $\mathcal{T}_0$ be a tree representation of the chordal graph G. We modify $\mathcal{T}_0$ into another tree representation of G. First, replace sequentially each vertex w of $\mathcal{T}_0$ of degree $d \geq 4$ by a tree S with d leaves such that all internal vertices of S have degree three, and identify the leaves of S with the neighbors of w. In addition, add to each leaf a pending vertex contained in no tree T_v. Let $\mathcal{T}_1$ be the resulting tree. Root $\mathcal{T}_1$ at any of its leaves. If T_v contains a vertex w of $\mathcal{T}_0$, then it contains all the internal vertices of S in $\mathcal{T}_1$. Note that $\mathcal{T}_1$ already has the properties (i)–(iii).

Now replace each edge ww' of $\mathcal{T}_1$ by a path of length $|\tau_w \div \tau_{w'}|$. Let $\mathcal{T}_2$ be the resulting tree and ww' a fixed edge of $\mathcal{T}_1$. In addition, let $v_1, \ldots, v_k$ be all the vertices of $\tau_w \setminus \tau_{w'}$ and let $v'_1, \ldots, v'_{k'}$ be all the vertices of $\tau_{w'} \setminus \tau_w$. If T_v contains in $\mathcal{T}_1$ both the vertices w and w', it contains the entire path between w and w' in $\mathcal{T}_2$. The tree T_{v_i} contains the first i vertices of the path from w to w' and $T_{v'_i}$ contains the last i vertices. In this way, we make sure that $\mathcal{T}_2$ has the property (iv).

In the final step, for each vertex w of degree three in $\mathcal{T}_2$, we subdivide all the edges incident with w. Let $\mathcal{T}_3$ be the resulting tree. If T_v contains a vertex w in $\mathcal{T}_2$, then it contains w and all its three new neighbors in $\mathcal{T}_3$. The tree $\mathcal{T}_3$ and the subtrees T_v have all five properties from the statement of the lemma. □

Proposition 2 makes the proof of the following lemma quite simple.

LEMMA 2.1. *Let G be a chordal graph, U a subset of $V(G)$, and $1 \leq \kappa \leq |U|$ a real number. The graph G contains a clique C and a subgraph G_0 which is a union of some of the components of $G \setminus C$ with the property $\kappa \leq |U \cap V(G_0)| \leq 2\kappa$.*

Proof. Let $\mathcal{T}$ be a nice tree representation of G, i.e., a tree representation having the properties described in Proposition 2. Note that $\mathcal{T}$ is a rooted tree. We can

assume that each tree T_v, $v \in V(G)$, is nonempty. Let $\mathcal{T}(w)$ be the maximal subtree of $\mathcal{T}$ rooted at a vertex w, $w \in V(\mathcal{T})$, and let $\sigma(w)$ be the number of subtrees T_u with $u \in U$ which are entirely contained in $\mathcal{T}(w)$ and which satisfies that $w \notin T_u$, i.e., $\sigma(w) = |\{u \in U \mid T_u \subseteq \mathcal{T}(w) \setminus \{w\}\}|$. The properties of (ii)–(v) from the statement of Proposition 2 imply the following:

(i) If w is the root of $\mathcal{T}$, then $\sigma(w) = |U|$.
(ii) If w is a leaf of $\mathcal{T}$, then $\sigma(w) = 0$.
(iii) If w has a single child w', then $\sigma(w)$ is either $\sigma(w')$ or $\sigma(w') + 1$.
(iv) If w has two children w' and w'', then $\sigma(w) = \sigma(w') + \sigma(w'')$.

Choose now a vertex w of $\mathcal{T}$ with $\sigma(w) \geq \kappa$ but $\sigma(w') < \kappa$ for each child w' of w in $\mathcal{T}$. Note that the vertex w is not a leaf of $\mathcal{T}$ because $\kappa > 0$. Let C be the clique consisting of the vertices v whose tree T_v contains w, i.e., $w \in T_v$. Further, let G_0 be the subgraph of G induced by the vertices v such that $T_v \subseteq \mathcal{T}(w) \setminus \{w\}$. G_0 is clearly a union of components of $G \setminus C$. In addition, $|U \cap V(G_0)| = \sigma(w) \geq \kappa$. It remains to show that $|U \cap V(G_0)| \leq 2\kappa$. If w has a single child w', then $|U \cap V(G_0)| < \kappa + 1 \leq 2\kappa$ because $|\tau_w \div \tau_{w'}| \leq 1$ and $\sigma(w') < \kappa$. On the other hand, if w has two children w' and w'', then $|U \cap V(G_0)| = \sigma(w) = \sigma(w') + \sigma(w'') < 2\kappa$ by the choice of the vertex w. □

The proved factor 2 in Lemma 2.1 is the best possible. Consider three paths consisting of m vertices and identify their ends such that a tree with a single vertex v of degree three is obtained. The resulting graph G is a tree and hence it is chordal. Let U be all the vertices of G except for v. The tightness is witnessed for the choice $\kappa = m + 1$ as m goes to infinity.

3. Powers of chordal graphs and $L(p, q)$-labelings. The result of the following theorem is improved for odd values of k in the next section, but we state the theorem in its general form.

THEOREM 3.1. *Let G be a chordal graph with maximum degree $\Delta \geq 2$ and let $k \geq 2$. Each subgraph H of the kth power G^k of G contains a vertex of degree at most*

$$\left\lfloor \sqrt{\frac{91k - 118}{384}} (\Delta + 1)^{(k+1)/2} \right\rfloor + \Delta.$$

Proof. We may assume that G is connected. Suppose that the statement of the theorem is false. Let H be a subgraph of G^k with the minimum degree at least $d_0 := \lfloor \sqrt{\frac{91k-118}{384}} (\Delta + 1)^{(k+1)/2} \rfloor + \Delta + 1$. Note that $d_0 \geq \Delta + 3$. Apply Lemma 2.1 for $\kappa = (d_0 - \Delta)/3 \geq 1$ and the set $U = V(H)$. Let C be a clique and G_0 an union of some components of $G \setminus C$ as described in Lemma 2.1. Let $U_0 = U \cap V(G_0)$, $G' = G \setminus (G_0 \cup C)$, and $U' = U \cap V(G')$. Set κ_0 to be $|U_0| = |U \cap V(G_0)|$. Note that $\kappa \leq \kappa_0 \leq 2\kappa$ and the order of the clique C is at most Δ. The latter follows from the fact that $\kappa_0 \geq 1$.

An induced path in G from a vertex $u \in U_0$ to a vertex $w \in U'$ which passes through a vertex $v \in C$ is called an *interconnecting* path. We estimate the number of interconnecting paths in G of length at most k. The length of a path is the number of its edges. Since the minimum degree of H in G^k is at least d_0 and $|U_0| = \kappa_0$, G^k contains at least $\kappa_0(d_0 - |C| - \kappa_0)$ edges between a vertex $u \in U_0$ and a vertex $w \in U'$. For such an edge uw, G contains an interconnecting path from u to w of length at most k. Therefore, the number of interconnecting paths of length at most k is at least $\kappa_0(d_0 - |C| - \kappa_0) \geq \kappa_0(d_0 - \Delta - \kappa_0) \geq \frac{2(d_0-\Delta)^2}{9}$. Next, we bound the number of interconnecting paths from above.

Let $\deg_0(v)$, $\deg_C(v)$, and $\deg'(v)$ be the number of neighbors of $v \in C$ among the vertices of $V(G_0)$, C, and $V(G')$, respectively. Note that $\deg_0(v)+\deg_C(v)+\deg'(v) \leq \Delta$. The number of interconnecting paths of length l which contain exactly one vertex of C is at most $\sum_{v\in C} \deg_0(v) \deg'(v)(l-1)(\Delta-1)^{l-2}$. A vertex v of the clique C can be one of $l-1$ inner vertices of the path, and there are $\deg_0(v)$ choices for its neighbor in G_0, $\deg'(v)$ choices for its neighbor in G', and at most $\Delta-1$ choices at each vertex in G_0 or in G' of how to continue the path. Note that not all such paths need to join a vertex of U_0 and a vertex of U'. Hence, the number of interconnecting paths of length at most k which contain one vertex of C is bounded by the sum

$$\sum_{l=2}^{k} \sum_{v\in C} \deg_0(v) \deg'(v)(l-1)(\Delta-1)^{l-2}$$

$$= \left(\sum_{l=2}^{k} (l-1)(\Delta-1)^{l-2}\right) \left(\sum_{v\in C} \deg_0(v) \deg'(v)\right)$$

$$\leq (k-1) \left(\sum_{l=2}^{k} (\Delta-1)^{l-2}\right) \left(\sum_{v\in C} \deg_0(v)(\Delta - \deg_C(v) - \deg_0(v))\right)$$

$$\leq (k-1)\Delta^{k-2} \sum_{v\in C} \frac{(\Delta - \deg_C(v))^2}{4} = (k-1)\Delta^{k-2} \sum_{v\in C} \frac{(\Delta+1-|C|)^2}{4}$$

$$= (k-1)\Delta^{k-2} \frac{|C|(\Delta-|C|+1)^2}{4} \leq \frac{(k-1)\Delta^{k-2}(\Delta+1)^3}{27}. \tag{3.1}$$

Let $D_0 = \sum_{v\in C} \deg_0(v)$ and $D' = \sum_{v\in C} \deg'(v)$ be the number of edges between the clique C and G_0 and G', respectively. Note that $D_0+D'+|C|(|C|-1) \leq |C|\Delta$. An interconnecting path cannot contain three or more vertices of C because it is induced. Moreover, if it contains two vertices of C, the two vertices of C are consecutive. Therefore, the number of interconnecting paths of length l with two or more vertices of C is equal to the number of interconnecting paths of length l containing an edge of C. Hence, the number of such interconnecting paths is at most $D_0D'(l-2)(\Delta-1)^{l-3}$. An edge of C can be one of $l-2$ inner edges of the interconnecting paths, there are at most D_0 edges between C and G_0 and at most D' edges between C and G', and there are at most $\Delta-1$ choices at each vertex in G_0 or $G \setminus (C \cup G_0)$ of how to continue the path. Again, not all the paths counted in this way need to be interconnecting. Hence, the following expression bounds the number of interconnecting paths of length at most k:

$$\sum_{l=2}^{k} D_0D'(l-2)(\Delta-1)^{l-3} = D_0D' \sum_{l=2}^{k} (l-2)(\Delta-1)^{l-3}$$

$$\leq \frac{(|C|\Delta - |C|(|C|-1))^2}{4}(k-2) \sum_{l=2}^{k} (\Delta-1)^{l-3}$$

$$\leq \frac{(|C|(\Delta+1-|C|))^2}{4}(k-2)\Delta^{k-3} \leq \frac{(\Delta+1)^4(k-2)\Delta^{k-3}}{64}. \tag{3.2}$$

The upper bound on the number of interconnecting paths of length at most k is equal to the sum of (3.1) and (3.2):

$$\frac{(\Delta+1)^3\Delta^{k-2}(k-1)}{27} + \frac{(\Delta+1)^4\Delta^{k-3}(k-2)}{64}$$

$$\leq \left(\frac{k-1}{27} + \frac{k-2}{64}\right)(\Delta+1)^{k+1} = \frac{91k-118}{1728}(\Delta+1)^{k+1}.$$

We now compare the obtained upper and lower bounds on the number of interconnecting paths of length at most k. This will yield a contradiction with the inequality $d_0 > \sqrt{\frac{91k-118}{384}}(\Delta+1)^{(k+1)/2} + \Delta$ following from the choice of d_0:

$$\begin{aligned}\frac{2(d_0-\Delta)^2}{9} &\leq \frac{91k-118}{1728}(\Delta+1)^{k+1},\\ (d_0-\Delta)^2 &\leq \frac{91k-118}{384}(\Delta+1)^{k+1},\\ d_0 &\leq \sqrt{\frac{91k-118}{384}(\Delta+1)^{k+1}} + \Delta. \qquad \square\end{aligned}$$

An immediate corollary of Theorem 3.1 is the following upper bound on the chromatic number of powers of chordal graphs.

COROLLARY 3.2. *If G is a chordal graph with maximum degree $\Delta \geq 2$, then*

$$\chi(G^k) \leq \left\lfloor \sqrt{\frac{91k-118}{384}}(\Delta+1)^{(k+1)/2} \right\rfloor + \Delta + 1 = O(\sqrt{k}\Delta^{(k+1)/2}).$$

In particular,

$$\chi(G^2) \leq \left\lfloor \frac{(\Delta+1)^{3/2}}{\sqrt{6}} \right\rfloor + \Delta + 1 \approx 0.4082\Delta^{3/2} + O(\Delta).$$

Another corollary of Theorem 3.1 is an $O(\Delta^{3/2})$-bound on the minimum span of the $L(2,1)$-labeling and the $L(p,q)$-labeling.

THEOREM 3.3. *If G is a chordal graph with maximum degree $\Delta \geq 2$, then*

$$\lambda_{p,q}(G) \leq \left\lfloor \frac{(\Delta+1)^{3/2}}{\sqrt{6}} \right\rfloor (2q-1) + \Delta(2p-1) = \frac{\Delta^{3/2}}{\sqrt{6}}(2q-1) + O(\Delta p).$$

In particular,

$$\lambda_{2,1}(G) \leq \left\lfloor \frac{(\Delta+1)^{3/2}}{\sqrt{6}} \right\rfloor + 3\Delta = \frac{\Delta^{3/2}}{\sqrt{6}} + O(\Delta) \approx 0.4082\Delta^{3/2} + O(\Delta).$$

Proof. Since G is $\lfloor\sqrt{\frac{91k-118}{384}}(\Delta+1)^{(k+1)/2}+\Delta\rfloor$-degenerate by Theorem 3.1, there is an ordering $v_1,\ldots,v_n$ of the vertices of G such that v_i has at most $\lfloor\frac{(\Delta+1)^{3/2}}{\sqrt{6}}\rfloor + \Delta$ neighbors among the vertices $v_1,\ldots,v_{i-1}$ in G^2. Label the vertices of G in this order

by labels from 0 to $\lfloor \frac{(\Delta+1)^{3/2}}{\sqrt{6}} \rfloor (2q-1) + \Delta(2p-1)$ in a greedy fashion. Consider a step when a label to the vertex v_i is to be assigned. Each of the neighbors of v_i in G^2 among the vertices $v_1, \ldots, v_{i-1}$ prevents assigning at most $2q-1$ different labels to v_i. In addition, each of at most Δ neighbors of v_i in G among the preceding vertices prevents assigning at most $2p-1-(2q-1) = 2p-2q$ additional labels to v_i. Altogether, at most $\lfloor \frac{(\Delta+1)^{3/2}}{\sqrt{6}} \rfloor (2q-1) + \Delta(2p-1)$ labels are forbidden. Hence, there exists a label which can be assigned to the vertex v_i. □

4. Odd powers of chordal graphs. The proof of degeneracy in the case of odd powers of chordal graphs is based on a simple proof of the well-known fact that odd powers of chordal graphs are chordal. As we noted in the introduction, the proven upper bound is the best possible, i.e., it matches the lower bound which we show in Theorem 5.1.

THEOREM 4.1. *Let G be a chordal graph with maximum degree $\Delta \geq 2$ and let $k \geq 3$ be an odd integer. The kth power G^k of G is d-degenerate for the following choice of d:*

$$d = \begin{cases} k & \text{if } \Delta = 2, \\ \left\lfloor \frac{(\Delta+2)^2}{4} \right\rfloor - 1 & \text{if } \Delta \geq 3 \text{ and } k = 3, \\ \left\lceil \frac{\Delta+1}{2} \right\rceil + \left\lfloor \frac{(\Delta+1)^2}{4} \right\rfloor \frac{(\Delta-1)^{(k-1)/2}-1}{\Delta-2} - 1 & \text{otherwise.} \end{cases}$$

Proof. If $\Delta = 2$, then G is a disjoint union of paths and triangles and hence G^k is k-degenerate. We assume that $\Delta \geq 3$ in the rest of the proof. Let $\mathcal{T}$ be a (not necessarily nice) tree representation of G and let T_v be the subtree corresponding to a vertex v of G. We construct a tree representation of G^k with the same underlying tree $\mathcal{T}' = \mathcal{T}$: A vertex v of G^k is represented by a tree $T'_v = \bigcup_{w \in N_{(k-1)/2}(v)} T_w$, where $N_i(v)$ is the set of all vertices of G whose distance from v is at most i in G. It is easy to check that each of the subtrees T'_v is connected, i.e., T'_v is a subtree of $\mathcal{T}'$. In addition, $T'_u \cap T'_v \neq \emptyset$ if and only if the distance between u and v is at most $2 \cdot \frac{k-1}{2} + 1 = k$ in G. Hence, we obtained a tree representation of G^k. In particular, G^k is chordal.

If $\omega(G^k)$ is the order of the largest clique of G^k, then the graph G^k is d-degenerate for $d = \omega(G^k) - 1$ (consider a perfect elimination sequence for G^k). Hence, it is enough to prove the inequality

$$\omega(G^k) \leq \begin{cases} \left\lfloor \frac{(\Delta+2)^2}{4} \right\rfloor & \text{if } k = 3, \\ \left\lceil \frac{\Delta+1}{2} \right\rceil + \left\lfloor \frac{(\Delta+1)^2}{4} \right\rfloor \frac{(\Delta-1)^{(k-1)/2}-1}{\Delta-2} & \text{otherwise.} \end{cases}$$

Let C_0 be a largest clique of G^k. Since subtrees of a tree satisfy a so-called Helly-property, there exists a vertex u_0 of $\mathcal{T}'$ such that $u_0 \in T'_v$ for each $v \in C_0$. Let C be the clique of G comprised by all the vertices v with $u_0 \in T_v$ and let p be its order. The distance of a vertex v from C (in G) is the smallest distance of the vertex v from some vertex of C. The clique C_0 consists (precisely) of all the vertices which are at distance at most $(k-1)/2$ from C (recall the definition of $\mathcal{T}'$). There are at most $p(\Delta-(p-1))$ vertices at distance one and additionally at most $p(\Delta-(p-1))(\Delta-1)$ vertices at distance two from C. In general, the number of vertices at distance i from C is at most $p(\Delta-(p-1))(\Delta-1)^{i-1}$. Hence, the order of the clique C_0 is bounded from above by the sum

$$p + p(\Delta+1-p) + p(\Delta+1-p)(\Delta-1) + \cdots + p(\Delta+1-p)(\Delta-1)^{(k-3)/2}.$$

If $k = 3$, then the order of C_0 is at most

$$p + p(\Delta + 1 - p) = p(\Delta + 2 - p) \leq \left\lfloor \frac{\Delta + 2}{2} \right\rfloor \left\lceil \frac{\Delta + 2}{2} \right\rceil = \left\lfloor \frac{(\Delta + 2)^2}{4} \right\rfloor.$$

In the general case $k \geq 5$, the order of C_0 is at most

$$p + p(\Delta + 1 - p)\frac{(\Delta - 1)^{(k-1)/2} - 1}{\Delta - 2}$$

$$\leq \left\lceil \frac{\Delta + 1}{2} \right\rceil + \left\lfloor \frac{\Delta + 1}{2} \right\rfloor \left\lceil \frac{\Delta + 1}{2} \right\rceil \frac{(\Delta - 1)^{(k-1)/2} - 1}{\Delta - 2}$$

$$= \left\lceil \frac{\Delta + 1}{2} \right\rceil + \left\lfloor \frac{(\Delta + 1)^2}{4} \right\rfloor \frac{(\Delta - 1)^{(k-1)/2} - 1}{\Delta - 2}. \qquad \square$$

An immediate corollary of Theorem 4.1 is the following bound for the chromatic number of odd powers of chordal graphs.

COROLLARY 4.2. *If G is a chordal graph with maximum degree $\Delta \geq 1$ and $k \geq 3$ is an odd integer, then*

$$\chi(G^k) \leq \begin{cases} 2 & \text{if } \Delta = 1, \\ k + 1 & \text{if } \Delta = 2, \\ \left\lfloor \frac{(\Delta+2)^2}{4} \right\rfloor & \text{if } \Delta \geq 3 \text{ and } k = 3, \\ \left\lceil \frac{\Delta+1}{2} \right\rceil + \left\lfloor \frac{(\Delta+1)^2}{4} \right\rfloor \frac{(\Delta-1)^{(k-1)/2}-1}{\Delta-2} & \text{otherwise.} \end{cases}$$

5. Lower bounds. We first prove that the bound shown in Theorem 4.1 is tight.

THEOREM 5.1. *Let $k \geq 3$ be an odd integer and let $\Delta \geq 1$ be an integer. There is a chordal graph G of order n whose maximum degree is Δ such that the kth power G^k of G is a clique for the following choice of n:*

$$n = \begin{cases} 2 & \text{if } \Delta = 1, \\ k + 1 & \text{if } \Delta = 2, \\ \left\lfloor \frac{(\Delta+2)^2}{4} \right\rfloor & \text{if } \Delta \geq 3 \text{ and } k = 3, \\ \left\lceil \frac{\Delta+1}{2} \right\rceil + \left\lfloor \frac{(\Delta+1)^2}{4} \right\rfloor \frac{(\Delta-1)^{(k-1)/2}-1}{\Delta-2} & \text{otherwise.} \end{cases}$$

Proof. If $\Delta = 1$, then set G to be K_2. If $\Delta = 2$, set G to be a path of length k. If $\Delta \geq 3$ and $k = 3$, then we choose G to be a clique of order $\lceil \frac{\Delta+2}{2} \rceil$ such that each vertex of the clique is adjacent to $\lfloor \frac{\Delta}{2} \rfloor$ vertices of degree one.

We may now assume that $\Delta \geq 3$ and $k \geq 5$. The graph G contains a clique of order $\lceil \frac{\Delta+1}{2} \rceil$. Each vertex of the clique has $\lfloor \frac{\Delta+1}{2} \rfloor$ neighbors such that each of them has degree one. These vertices of degree one form the first level. Each vertex of the first level has $\Delta - 1$ neighbors which form the second level, each vertex of the second level has $\Delta - 1$ neighbors which form the third level, etc. The graph G has altogether $(k - 1)/2$ levels. The vertices of the same level form an independent set. This completes the construction of the graph G.

It is easy to verify that the maximum degree of G is Δ and G^k is a clique. The number of vertices of the graph G is equal to the sum

$$\left\lceil \frac{\Delta + 1}{2} \right\rceil + \left\lfloor \frac{\Delta + 1}{2} \right\rfloor \left\lceil \frac{\Delta + 1}{2} \right\rceil + \left\lfloor \frac{\Delta + 1}{2} \right\rfloor \left\lceil \frac{\Delta + 1}{2} \right\rceil (\Delta - 1) + \cdots$$

$$+\left\lfloor \frac{\Delta+1}{2} \right\rfloor \left\lceil \frac{\Delta+1}{2} \right\rceil (\Delta-1)^{(k-3)/2}$$

$$= \left\lceil \frac{\Delta+1}{2} \right\rceil + \left\lfloor \frac{(\Delta+1)^2}{4} \right\rfloor \frac{(\Delta-1)^{(k-1)/2}-1}{\Delta-2}. \qquad \square$$

The following result of Iwaniec and Pintz [12], which was drawn to our attention by Martin Klazar, is used in the lower bound construction for the case of even powers of chordal graphs.

PROPOSITION 5.2. *For each $n \geq 2$, there is a prime p such that $n - n^{23/42} \leq p \leq n$.*

THEOREM 5.3. *If $k \geq 2$ is an even integer and $\Delta \geq 12$, then there exists a chordal graph G of order n whose maximum degree is Δ such that the kth power G^k of G is a clique for the following choice of n:*

$$n = \begin{cases} \frac{2\Delta^{3/2}}{3\sqrt{3}} - O(\Delta^{107/84}) \approx 0.3849\Delta^{3/2} - O(\Delta^{107/84}) & \text{if } k = 2, \\ \frac{1}{2\sqrt{2}}\Delta^{(k+1)/2} - O(\Delta^{k/2+23/84}) & \text{if } k \geq 4. \end{cases}$$

In particular, there is a chordal graph G with maximum degree Δ such that

$$\lambda_{2,1}(G) \geq \chi(G^2) - 1 \geq \frac{2\Delta^{3/2}}{3\sqrt{3}} - O(\Delta^{107/84}) \approx 0.3849\Delta^{3/2} - O(\Delta^{107/84}).$$

Proof. We first consider the case that $k = 2$. Let $q_0 = \lfloor \sqrt{\frac{\Delta}{3}} \rfloor - 1 \geq 1$. If $q_0 \geq 2$, let p be a prime between $q_0 - q_0^{23/42}$ and q_0. Such a prime p exists by Proposition 5.2. And further, let (X, Π) be a finite projective plane of order p (thus $|X| = p^2 + p + 1$). If $q_0 = 1$, let $p = 1$, and let (X, Π) be a pair of sets such that $|X| = 3$ and Π contains all three pairs of elements of X. In the rest of the proof, it does not matter whether (X, Π) is a projective plane or a "triangle" formed by three "segments."

Let $m = \lfloor 2\sqrt{\frac{\Delta}{3}} \rfloor \geq 4$. The desired graph G consists of $p^2 + p + 1$ vertices v_x for each $x \in X$ and m vertices $v_\pi^1, \ldots, v_\pi^m$ for each of the $p^2 + p + 1$ lines π of Π. Hence, the order of the graph G is $(p^2 + p + 1)(m + 1) = \frac{2\Delta^{3/2}}{3\sqrt{3}} - O(\Delta^{107/84})$ vertices. We now describe the edges contained in G. The vertices v_x, $x \in X$, form a clique of order $|X| = p^2 + p + 1$ and each vertex v_π^i, $\pi \in \Pi$, and $1 \leq i \leq m$, is joined by an edge to the vertex v_x, $x \in X$, for each $x \in \pi$. The sequence of vertices containing the vertices v_x first and then the vertices v_π^i is a perfect elimination sequence and hence the obtained graph G is chordal.

In order to show that the square G^2 of G is a clique, we need to verify that each pair of nonadjacent vertices has a common neighbor. This is clear for every pair of vertices v_x and v_π^i. Consider now a pair of vertices v_π^i and $v_{\pi'}^{i'}$. Since Π is a projective plane, there exists a point $x \in X$ such that $x \in \pi \cap \pi'$. The sought common neighbor of v_π^i and $v_{\pi'}^{i'}$ is the vertex v_x. Therefore, G^2 is a clique.

It remains to verify that the maximum degree of G is at most Δ. The degree of a vertex v_x contained in the central clique is

$$p^2 + p + (p+1)m \leq (q_0 + 1)(q_0 + m) \leq \sqrt{\frac{\Delta}{3}} \cdot \left(\sqrt{\frac{\Delta}{3}} + 2\sqrt{\frac{\Delta}{3}} \right) = \Delta.$$

The degree of a vertex v_π^i is even smaller, namely, $(p+1)$.

If $k \geq 4$, we first proceed as in the case $k = 2$. We obtain a graph G for the choice of parameters $q_0 = \lfloor\sqrt{\frac{\Delta}{2}}\rfloor - 1$ and $m = \lfloor\sqrt{\frac{\Delta}{2}}\rfloor$. The number p is again a prime between $q_0 - q_0^{23/42}$ and q_0 if $q_0 \geq 2$ and it is one if $q_0 = 1$. Note that the degrees of the vertices contained in the clique are bounded by $p^2 + p + (p+1)m \leq (p+1)^2 + (p+1)m \leq (q_0+1)^2 + (q_0+1)m \leq \Delta$. In the second part of the construction, we proceed similarly as in the proof of Theorem 5.1. We consider the vertices v_π^i to form the first level. Each of these vertices has $\Delta - (p+1)$ neighbors which form the second level. Additional $(k-4)/2$ levels are formed by adding $\Delta - 1$ new vertices adjacent to each vertex of the preceding level. The vertices of each level form an independent set and their degrees are bounded by Δ. The maximum degree of the graph G is Δ and the kth power G^k of G is a clique. The number of vertices of G forming the last level is equal to the product

$$(p^2 + p + 1)m(\Delta - (p+1))(\Delta - 1)^{(k-4)/2}$$

$$= \frac{\Delta}{2} \cdot \sqrt{\frac{\Delta}{2}} \cdot \Delta \cdot (\Delta - 1)^{(k-4)/2} - O(\Delta^{k/2+23/84})$$

$$= \frac{1}{2\sqrt{2}}\Delta^{(k+1)/2} - O(\Delta^{k/2+23/84}).$$

Since the number of vertices of the last level dominates the number of vertices of the remaining levels, we may then conclude that the order of G is $\frac{1}{2\sqrt{2}}\Delta^{(k+1)/2} - O(\Delta^{k/2+23/84})$. □

The construction presented in Theorem 5.3 also gives a good lower bound on the minimum span of an $L(p,q)$-labeling of a chordal graph.

COROLLARY 5.4. *There exists a chordal graph G with maximum degree Δ such that $\lambda_{p,q}(G) \geq \Omega(\Delta^{3/2}q + \Delta p)$.*

Proof. Consider a chordal graph G from Theorem 5.3 for $k = 2$. The order of G is $\frac{2\Delta^{3/2}}{3\sqrt{3}} - O(\Delta^{107/84})$ and its second power is a clique. Hence, $\lambda_{p,q}(G) \geq (\frac{2\Delta^{3/2}}{3\sqrt{3}} - 1)q - O(\Delta^{107/84}q)$. On the other hand, the graph G contains a clique of order $q_0^2 + q_0 + 1 = \Theta(\Delta)$, where q_0 is as in the proof of Theorem 5.3. Therefore, $\lambda_{p,q}(G) \geq \Omega(\Delta p)$. Combining both the lower bounds on $\lambda_{p,q}(G)$ yields the claimed bound. □

Acknowledgments. The author is very indebted to Jan Kratochvíl and Riste Škrekovski for fruitful discussions on $L(2,1)$-labeling of graphs and on the structure of chordal graphs, during which he discovered a lot of new insights, as well as for their helpful comments on a preliminary version of this paper. This research was started during the author's stay at SFU and PIMS in November 2002. The author would like to thank Pavol Hell and Riste Škrekovski for making his stay there pleasant and fruitful. The comments of both the anonymous referees who pointed out several flaws and helped to improve the clarity of the presentation are strongly appreciated.

REFERENCES

[1] G. Agnarsson, R. Greenlaw, and M. M. Halldórsson, *On Powers of chordal graphs and their coloring*, Congr. Numer., 144 (2000), pp. 41–65.

[2] G. Agnarsson and M. M. Halldórsson, *Coloring powers of planar graphs*, SIAM J. Discrete Math., 16 (2003), pp. 651–662.

[3] N. Alon and B. Mohar, *The chromatic number of graph powers*, Combin. Probab. Comput., 11 (2002), pp. 1–10.

[4] H. L. Bodlaender, T. Kloks, R. B. Tan, and J. van Leeuwen, *λ-coloring of graphs*, in Proceedings of the STACS'00, Lecture Notes in Comput. Sci. 1770, G. Goos, J. Hartmanis, and J. van Leeuwen, eds., Springer-Verlag, Berlin, 2000, pp. 395–406.

[5] A. Brandstädt, *Special Graph Classes—A Survey*, Schriften-Mathematik Uni.-Duisburg, Duisburg, Germany, 1991.

[6] G. J. Chang and D. Kuo, *The $L(2,1)$-labeling problem on graphs*, SIAM J. Discrete Math., 9 (1996), pp. 309–316.

[7] G. J. Chang, W.-T. Ke, D. D.-F. Liu, and R. K. Yeh, *On $L(d,1)$-labellings of graphs*, Discrete Math., 220 (2000), pp. 57–66.

[8] J. Fiala, J. Kratochvíl, and T. Kloks, *Fixed-parameter complexity of λ-labelings*, Discrete Appl. Math., 113 (2001), pp. 59–72.

[9] D. A. Fotakis, S. E. Nikoletseas, V. G. Papadopoulou, and P. G. Spirakis, *NP-completeness results and efficient approximations for radiocoloring in planar graphs*, in Proceedings of the MFCS'00, Lecture Notes in Comput. Sci. 1893, B. Rovan, ed., Springer-Verlag, New York, 2000, pp. 363–372.

[10] J. R. Gilbert, D. J. Rose, and A. Edenbrandt, *A separator theorem for chordal graphs*, SIAM J. Algebraic Discrete Methods, 5 (1984), pp. 306–313.

[11] J. R. Griggs and R. K. Yeh, *Labelling graphs with a condition at distance* 2, SIAM J. Discrete Math., 5 (1992), pp. 586–595.

[12] H. Iwaniec and J. Pintz, *Primes in short intervals*, Monatsh. Math., 98 (1984), pp. 115–143.

[13] D. Král' and R. Škrekovski, *A theorem about the channel assignment problem*, SIAM J. Discrete Math., 16 (2003), pp. 426–437.

[14] C. McDiarmid, *Discrete mathematics and radio channel assignment*, in Recent Advances in Algorithms and Combinatorics, C. Linhares-Salas and B. Reed, eds., Springer, New York, 2003, pp. 27–63.

[15] D. Sakai, *Labeling chordal graphs: Distance two condition*, SIAM J. Discrete Math., 7 (1994), pp. 133–140.

SIAM J. DISCRETE MATH.
Vol. 18, No. 3, pp. 462–471

THE BAR VISIBILITY NUMBER OF A GRAPH*

YI-WU CHANG†, JOAN P. HUTCHINSON‡, MICHAEL S. JACOBSON§, JENŐ LEHEL¶, AND DOUGLAS B. WEST‖

Abstract. The *bar visibility number* of a graph G, denoted $b(G)$, is the minimum t such that G can be represented by assigning each vertex x the set S_x of points in at most t horizontal segments in the plane so that $uv \in E(G)$ if and only if some point of S_u sees some point of S_v via a vertical segment of positive width unobstructed by assigned points. Among our results are the following:

(1) Every planar graph has bar visibility number at most 2, which is sharp.
(2) $r \le b(K_{m,n}) \le r+1$, where $r = \left\lceil \frac{mn+4}{2m+2n} \right\rceil$.
(3) $b(K_n) = \lceil n/6 \rceil$.
(4) If G has n vertices, then $b(G) \le \lceil n/6 \rceil + 2$.

Key words. bar visibility graph, interval number, planar graph, thickness, splitting number

AMS subject classifications. 05C62, 05C35, 05C10

DOI. 10.1137/S0895480198343455

1. Introduction.

In computational geometry, graphs are used to model visibility relations in the plane. For example, we may say that two vertices of a polygon "see" each other if the segment joining them lies inside the polygon. In the *visibility graph* on the vertex set, vertices are adjacent if they see each other. Similarly, we can define visibility on a set of line segments; two segments see each other if some segment joining them crosses no other segment in the set. Dozens of papers have been written concerning the computation and the recognition of visibility graphs and discussing applications to search problems and motion planning.

We consider a simpler model where all visibility is vertical. Let S be a family of pairwise disjoint horizontal segments (henceforth "bars") in the plane. The *bar visibility graph* of S is the graph with vertex set S in which two vertices are adjacent if and only if there is an unobstructed vertical channel (a strip of positive width) joining those bars. Requiring a channel of positive width is realistic, and it permits bars $[(a,y),(x,y)]$ and $[(x,z),(c,z)]$ to block visibility at x without seeing each other.

Tamassia and Tollis [18] and Wismath [22] characterized bar visibility graphs as the planar graphs that are embeddable with all cut-vertices on a common face. Similar questions have been studied for other models. Hutchinson, Shermer, and Vince [12] studied graphs generated by horizontal and vertical visibility of rectangles in the plane (see also [4, 5]). Hutchinson [11] studied polar visibility of arcs on concentric circles centered at the origin. An introduction to bar visibility and other models appears in [16].

*Received by the editors July 15, 1998; accepted for publication (in revised form) January 9, 2004; published electronically December 30, 2004. The United States Government is authorized to reproduce and distribute reprints notwithstanding any copyright notation herein.
http://www.siam.org/journals/sidma/18-3/34345.html

†National Politics University, Taipei, Taiwan (chang@math.nccu.edu.tw). Portions of this material appeared in the thesis of this author in 1994.
‡Macalester College, St. Paul, MN 55105 (hutchinson@macalester.edu).
§University of Colorado at Denver, Denver, CO 80217 (msj@cudenver.edu).
¶University of Memphis, Memphis, TN 38152 (jlehel@memphis.edu), on leave from the Computer and Automation Research Institute of the Hungarian Academy of Sciences.
‖University of Illinois, Urbana, IL 61801 (west@math.uiuc.edu.). The work of this author was partially supported by the National Security Agency under grant MDA904-03-1-0037.

We study a generalization of visibility graphs analogous to a well-studied generalization of interval graphs. The *interval graph* of a family S of intervals on the real line is the graph with vertex set S in which two vertices are adjacent if and only if as intervals they intersect. Bar visibility graphs provide a geometric analogue of interval graphs; visibility replaces intersection as the adjacency relation, and the intervals may be at various heights.

To generalize interval representations, we let a *t-interval* be the set of points in a union of (at most) t intervals on a line. A *t-interval representation* of a graph G assigns t-intervals to vertices so that vertices are adjacent if and only if their assigned t-intervals intersect. The *interval number* $i(G)$ of G is the minimum t such that G has a t-interval representation. A recent nice result and further references on $i(G)$ appear in [2].

Here we similarly generalize the bar visibility model. A *t-bar* is the set of points in a union of (at most) t horizontal bars in the plane. A *t-bar representation* of G is an assignment of t-bars to vertices of G so that vertices are adjacent if and only if some vertical channel of positive width links their t-bars without intersecting any other assigned t-bar. The *bar visibility number* $b(G)$ of a graph G is the minimum t such that G has a t-bar representation. For simplicity, in this paper we abbreviate the term to *visibility number*. Note that bar visibility graphs are the graphs with visibility number equal to 1.

For graphs without large cliques, visibility number tends to be smaller than interval number, because bars can block visibility and because the upper and lower "sides" of a bar can be used independently to establish edges. In section 2, we observe that it follows easily from the results of [18] or [22] that every planar graph has visibility number at most two, and this is sharp (planar graphs have interval number at most three [17]).

For other families, our lower bounds arise from an easy lemma based on the maximum number of edges in N-vertex planar graphs. Constructions are more difficult; we study complete bipartite graphs, complete graphs, and general n-vertex graphs. The visibility number of the complete bipartite graph $K_{m,n}$ is roughly half its interval number, being either $\lceil \frac{mn+4}{2m+2n} \rceil$ or one more than this (section 3), but the complete graph K_n has interval number 1 and visibility number $\lceil n/6 \rceil$ (section 4).

Every planar graph without cut-vertices is a visibility graph. Thus in some sense visibility number is a measure of how far a graph is from being planar. It is related to other such parameters, such as "thickness" (section 2) and "splitting number" (sections 3 and 5). The optimal upper bound on the visibility number of K_n arises from the solution of "Heawood's empire problem" (section 4).

We conjecture that $b(G) \leq \lceil n/6 \rceil$ when G has n vertices; equality holds for K_n. In support of this conjecture, in section 5 we show that $b(G) \leq \lceil n/6 \rceil + 2$. We use the result of Lovász [15] that every m-vertex graph decomposes into at most $\lfloor m/2 \rfloor$ paths and cycles. Gallai's conjecture [6] that $\lceil m/2 \rceil$ paths suffice would yield $b(G) \leq \lceil n/6 \rceil + 1$.

2. Planar graphs and thickness. The extremal problem for visibility number of planar graphs is solved by expressing an arbitrary planar graph as the union of two bar visibility graphs.

REMARK 1. $b(G \cup H) \leq b(G) + b(H)$.

Proof. Bar visibility representations of G and H can be placed in disjoint vertical strips to represent $G \cup H$. □

There are two ways to use earlier results to show that all planar graphs have vis-

ibility number at most 2. The most immediate uses a result of Wismath [22] showing that every planar graph is a *rectangle-visibility graph*, meaning that vertices can be assigned rectangles so that two vertices are adjacent if and only if the corresponding rectangles can see each other along a horizontal or a vertical strip of positive width.

REMARK 2. *If G is a planar graph, then $b(G) \leq 2$.*

Proof. By [22], G has a rectangle-visibility representation. The projections in the horizontal and vertical directions yield two bar visibility graphs whose union is G. □

As mentioned earlier, a graph is a bar visibility graph if and only if it is a planar graph embeddable so that all cut-vertices lie on a single face [18, 22]. The minimal planar graphs not embeddable with every vertex on a single face are K_4 and $K_{2,3}$. Adding a pendant edge at each vertex of such a graph produces a planar graph that is not a bar visibility graph because the cut-vertices cannot lie on a single face. Thus the bound in Remark 2 is sharp.

The characterization of bar visibility graphs also yields another proof of the upper bound via an inductive proof of a technically stronger statement. The *block-cutpoint tree* of a connected graph G is the bipartite graph whose partite sets are the cut-vertices and the blocks of G, with vertex v adjacent to block B if $v \in V(B)$. The block-cutpoint graph was introduced by Harary and Prins [8], and it is an elementary exercise that it is a tree.

THEOREM 3. *Every planar graph has a 2-bar representation in which every vertex that is not a cut-vertex is assigned a 1-bar.*

Proof. If H is a disjoint union of planar graphs with at most one cut-vertex in each component, then the characterization in [18, 22] yields $b(H) = 1$. We express a planar graph G as the union of two such graphs G_0 and G_1, and then Remark 1 applies.

We may assume that G is connected. We allocate the blocks of G to G_0 and G_1. Let B_0 be an arbitrary block of G. The distance of each block from B_0 in the block-cutpoint tree $B(G)$ is even; place a block in G_i if its distance from B_0 is congruent to $2i$ modulo 4. Two blocks wind up in the same component of G_0 or G_1 if and only if they share the same cut-vertex as neighbor on their paths to B_0 in $B(G)$. Hence a component of G_0 or G_1 consists of one or more such blocks of G sharing a single cut-vertex. □

Our subsequent lower bounds on visibility number use an easy counting argument based on the number of edges in planar graphs.

LEMMA 4. *The visibility number of a graph G with n vertices and e edges is at least $\lceil \frac{e+6}{3n} \rceil$. If the graph is triangle-free, then $b(G) \geq \lceil \frac{e+4}{2n} \rceil$.*

Proof. Consider a t-bar representation of G. Let N be the total number of bars used, so $N \leq nt$. In the plane, draw one vertical segment joining each pair of bars that see each other. Now shrink each bar so that it becomes a single point. The added segments remain, covering the edges of G. The result is a simple planar graph G' with N vertices and at least e edges. Since it also has at most $3N - 6$ edges, we have the desired bound.

If G is triangle-free, then the graph G' will remain simple and triangle-free after we contract edges joining points assigned to the same vertex of G. Now G' has at most $2N - 4$ edges, and again these cover all edges of G. □

The *thickness* of a graph G is the minimum number of planar graphs needed to decompose it. Each graph in a decomposition has at most $3n - 6$ edges when G has n vertices, and is bipartite when G is bipartite. Hence the thickness of an n-vertex

graph with e edges is at least $\lceil \frac{e}{3n-6} \rceil$, and it is at least $\lceil \frac{e}{2n-4} \rceil$ if the graph is bipartite.

These lower bounds on thickness are slightly larger than the lower bounds of Lemma 4 on visibility number. If G has an optimal decomposition using 2-connected planar graphs, then the thickness of G becomes an upper bound on visibility number. This bound need not be optimal, since the lower bound on visibility number is smaller. In general, we can do better for bar visibility representations because the planar pieces can "interact".

For the complete graph K_n, with two exceptions, the thickness equals the counting bound, as shown in [1]. The bound simplifies to $\lceil (n+2)/6 \rceil$. When n is 9 or 10, the thickness is 3 (see [3, 20]), although the general formula suggests 2. For $n \geq 7$, we improve these upper bounds by showing in section 4 that $b(K_n) = \lceil n/6 \rceil$.

A graph represented with one bar per vertex must be planar, so the upper bound using thickness cannot be improved when $5 \leq n \leq 6$. However, it can be improved when $n = 9$.

CONSTRUCTION 5. $b(K_9) = 2$.

Proof. Since K_9 is nonplanar, $b(K_9) \geq 2$. Although the thickness of K_9 is 3, we can express $K_9 - ws$ as the union of two planar graphs when ws is an edge in K_9. We can put a representation of one of these above the other and extend bars for w and s as in Figure 1 to obtain the missing visibility for ws. This establishes $b(K_9) = 2$. ∎

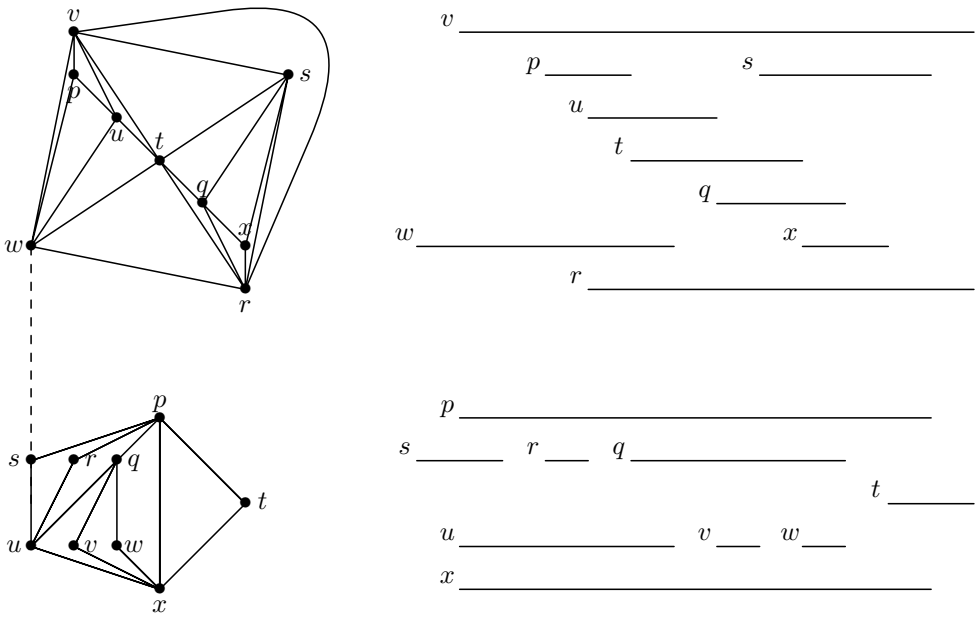


FIG. 1. 2-*bar representation of* K_9.

Thickness also yields an upper bound for visibility number of the complete bipartite graph $K_{m,n}$. The thickness is conjectured to equal the counting bound $\lceil \frac{mn}{2m+2n-4} \rceil$. For $K_{10,6}$, the lower bound on thickness is 3. Nevertheless, Construction 6 shows that $b(K_{10,6}) = 2$. In section 3, we show that the counting bound on $b(K_{m,n})$ can be achieved within 1. Our upper bound $\lceil \frac{mn+4}{2m+2n} \rceil + 1$ is generally less than the thickness bound.

CONSTRUCTION 6. $b(K_{10,6}) = 2$.

Proof. Since $(mn + 4)/(2m + 2n)$ exactly equals 2, achieving the lower bound is quite delicate. Our partite sets are $X = \{1, \ldots, 10\}$ and $Y = \{a, \ldots, f\}$. The construction is shown in Figure 2, with the bars for Y in bold. Neighboring bars on a line meet at their endpoints; there is no visibility through the gap. □

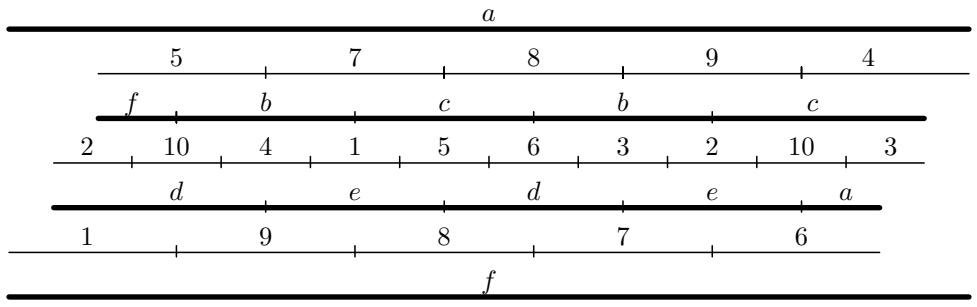


FIG. 2. 2-*bar representation of* $K_{10,6}$.

3. Complete bipartite graphs and splitting number. Lemma 4 implies that $b(K_{m,n}) \geq \lceil \frac{mn+4}{2m+2n} \rceil$; Construction 6 gives hope for equality. Trotter and Harary [19] proved that $i(K_{m,n}) = \lceil \frac{mn+1}{m+n} \rceil$. Our lower bound for $b(K_{m,n})$ always equals $\lceil i(K_{m,n})/2 \rceil$ or $\lceil i(K_{m,n})/2 \rceil + 1$. By using the tops and bottoms of bars separately, we prove constructively that $b(K_{m,n})$ is within one of the lower bound from Lemma 4.

Before presenting our construction, we compare $b(K_{m,n})$ with another parameter. Let a *vertex split* of a graph be the replacement of a vertex by two nonadjacent vertices, with each edge incident to the deleted vertex becoming incident instead to exactly one of the new vertices. The *splitting number* $s(G)$ of a graph G is the minimum number of vertex splits needed to turn G into a planar graph. Each vertex of G becomes an independent set in the resulting planar graph G'. If G' is 2-connected, then it has a bar visibility representation using $n(G) + s(G)$ bars. Thus $b(G) \geq 1 + s(G)/n(G)$.

Jackson and Ringel [14] proved that if m and n are both at least 2, then $s(K_{m,n}) = \lceil (m-2)(n-2)/2 \rceil$. Although $1 + \frac{(m-2)(n-2)}{2(m+n)} = \frac{mn+4}{2m+2n}$, this does not yield $b(K_{m,n}) \leq \lceil \frac{mn+4}{2m+2n} \rceil$, because the independent sets corresponding to vertices of $b(K_{m,n})$ in its optimal split need not have the same size. Indeed, the construction of [14] splits vertices in only one partite set.

We have observed that a bipartite graph G has at most $2N - 4$ edges if it has a t-bar representation using altogether N bars. Since $K_{m,n}$ has mn edges, this means that when $\frac{mn+4}{2m+2n}$ is an integer, achieving $b(K_{m,n}) = \frac{mn+4}{2m+2n}$ requires an $\frac{mn+4}{2m+2n}$-bar representation in which every vertex is assigned exactly $\frac{mn+4}{2m+2n}$ bars, and every face in the planar graph that results from turning the visibilities into edges and shrinking the bars has length exactly 4. It may be that $b(K_{m,n}) = \lceil \frac{mn+4}{2m+2n} \rceil$ always, but we have not proved this. Instead, we prove that allowing one more bar per vertex provides enough flexibility for a general construction.

THEOREM 7. *If* $r = \lceil \frac{mn+4}{2m+2n} \rceil$, *then* $r \leq b(K_{m,n}) \leq r + 1$.

Proof. We may take $m \geq n$ and let the partite sets be X and Y with $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_n\}$. As m grows, r increases to $\lceil n/2 \rceil$. When $r = \lceil n/2 \rceil$, we construct an r-bar representation using r vertical strips, where the jth strip consists of one bar each for y_j and y_{n+1-j} with one bar for each vertex of X between them.

We may therefore assume that $r \le \lfloor (n-1)/2 \rfloor$. Let $s = \lfloor (n-1)/2 \rfloor - r$; note that $s \ge 0$. Since $r > n/4$, we have $r > s$. We construct an $(r+1)$-bar representation of $K_{m,n}$.

We start with (up to) $2(r+1)$ rows of bars for vertices of Y as in Figure 3 (in bold, with some labels dropped for clarity). The first row has bars for $y_1, \ldots, y_{\lceil n/2 \rceil}$, the second for $y_{\lceil n/2 \rceil + 1}, \ldots, y_n$, and thereafter these two types alternate. In each row, the ith bar extends from horizontal coordinate $i-1$ to i, except that when n is odd the bars for y_n extend from $(n-3)/2$ to $(n+1)/2$.

We add rows of up to $\lceil n/2 \rceil - 1$ bars for X between successive rows for Y. Consecutive bars in a row share endpoints, so each bar sees only bars for vertices of the opposite partite set in the two neighboring rows.

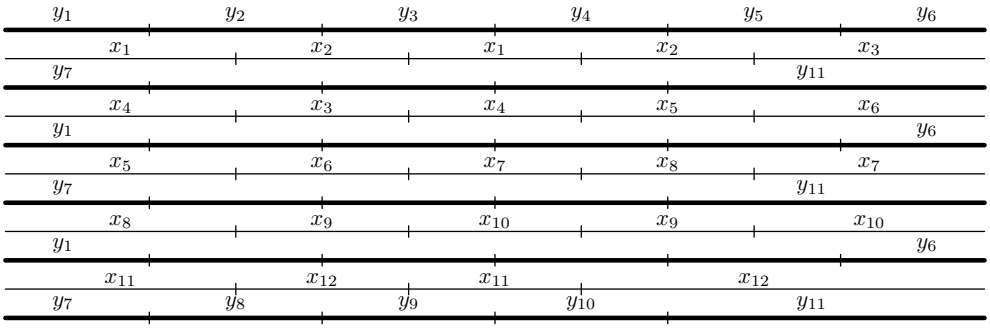


FIG. 3. *Part of a 4-bar representation of $K_{12,11}$; $r = 3$ and $s = 2$.*

Reading from left to right within successive rows for X from top to bottom, we alternate x_1, x_2 until we have s bars for each, then we alternate x_3, x_4, etc. We do this until we reach $x_{2\lfloor m/2 \rfloor}$, using $2s\lfloor m/2 \rfloor$ positions for X. The last of these bars extends out to the right, filling the row to avoid visibilities within Y. Having enough positions available requires $2s\lfloor m/2 \rfloor \le (2r+1)(\lceil n/2 \rceil - 1)$; we prove this. From $\frac{mn+4}{2m+2n} \le r$ we obtain $m(n/2 - r) \le rn - 2$, and hence

$$2s\left\lfloor \frac{m}{2} \right\rfloor \le ms \le m\left(\frac{n-1}{2} - r\right) \le rn - 2 - \frac{m}{2} \le 2r\left\lfloor \frac{n-1}{2} \right\rfloor + 2r - 2 - \frac{m}{2}.$$

Since $r \le \lceil n/2 \rceil \le \lceil m/2 \rceil$, we have $2r - 1 - m/2 \le \lceil n/2 \rceil$, which yields the desired inequality.

If m is odd, then we add one or two long bars for x_m. If we ended with $x_{2\lfloor m/2 \rfloor}$ after an odd number of rows for X, as in Figure 3, then we put one bar for x_m above the top row for Y and another below the bottom row. If we ended with $x_{2\lfloor m/2 \rfloor}$ after an even number of rows for X, then we have another row for $y_{\lceil n/2 \rceil + 1}, \ldots, y_n$ available to add and put one bar for x_m between the bottom two rows for Y. This establishes all visibilities for x_m.

We complete the construction by adding $r + 1 - s$ more bars for each $x \in \{x_1, \ldots, x_{2\lfloor m/2 \rfloor}\}$. Partition Y into $\lceil n/2 \rceil$ sets: the pairs of the form $\{y_j, y_{j+\lceil n/2 \rceil}\}$, plus the singleton $y_{\lceil n/2 \rceil}$ if n is odd. Via the s bars already placed, x already sees bars for $2s$ of these sets. Since $2s < s + r = \lfloor (n-1)/2 \rfloor$, the s bars for x cannot stretch far enough to cover two neighboring "columns" or the same "column" twice, and hence the $2s$ sets seen by x are distinct.

Since $r + 1 - s = \lceil n/2 \rceil - 2s$, it suffices to insert the remaining bars for x so that each such bar sees another of these sets. At a place where bars for a needed set

appear (or at the right end when the set is $\{y_{\lceil n/2 \rceil}\}$), we shrink the intervening bars for vertices of X and insert small bars for vertices of X that need to add visibility to this set. (When $s = 0$, there are no bars to shrink, and the inserted bars block all the vertical space.) □

4. Complete graphs and Heawood's empire problem. Heawood generalized the Four Color Problem by considering maps where many regions belong to a single "empire". Regions in a single empire must get the same color. Gardner [7] coined the term *m-pire* for an empire consisting of m regions. Jackson and Ringel [13] defined an *m-pire map* to be a map in the plane in which every empire consists of at most m regions. Heawood [10] proved that every m-pire map can be colored with $6m$ colors. He conjectured that the bound is sharp for $m > 1$; this became *Heawood's empire problem*.

Heawood's conjecture is proved by building a map with $6m$ pairwise adjacent m-pires. The adjacency graph is then K_{6m}, which requires $6m$ colors. Heawood did this for $m = 2$, and Taylor (see [7]) did it for $m = 3$ and $m = 4$. For larger m, it was first done by Jackson and Ringel [13]. Wessel [21] later gave a short proof. The result determines $b(K_n)$.

THEOREM 8. *If $n \geq 7$, then $b(K_n) = \lceil n/6 \rceil$.*

Proof. For $n \geq 7$, Lemma 4 yields $b(K_n) \geq \left\lceil \frac{n-1}{6} + \frac{2}{n} \right\rceil = \lceil n/6 \rceil$. For the upper bound, we may assume that n is divisible by 6, because the absence of unwanted visibilities in the complete graph implies that deleting the bars for a vertex in an m-bar representation of K_n yields an m-bar representation of K_{n-1}.

Consider an m-pire map with $6m$ pairwise adjacent m-pires. When we associate a vertex with each region, we obtain a dual graph with at most $6m^2$ vertices. The dual graph is a plane graph, and we may assume that it is 2-connected because the m-pires are pairwise adjacent. That is, a cut-vertex in the dual would correspond to an annular region R that separates one set of regions from another. In cutting a channel through R to establish a common boundary for the two sets that had been separated, we do not change the set of regions neighboring R.

Being a 2-connected plane graph, the dual is a bar visibility graph. Associating the bars arising from each m-pire with one vertex of K_{6m} yields an m-bar representation of K_{6m}. □

The m-pire maps used to prove Heawood's conjecture are quite complex, even in Wessel's proof. A surprisingly simple visibility construction produces a representation using at most $b(K_n)+1$ bars per vertex. It will motivate the construction in Theorem 10.

CONSTRUCTION 9. $b(K_n) \leq \lceil n/6 \rceil + 1$.

Proof. As in Theorem 8, we may assume that n is divisible by 6. Let $n = 6m$. We partition the vertex set into three sets A_1, A_2, A_3, each of size $2m$. A complete graph with $2m$ vertices has a decomposition into m spanning paths, consisting of the m rotations of a zigzag path when the vertices are placed around a circle (see Figure 4).

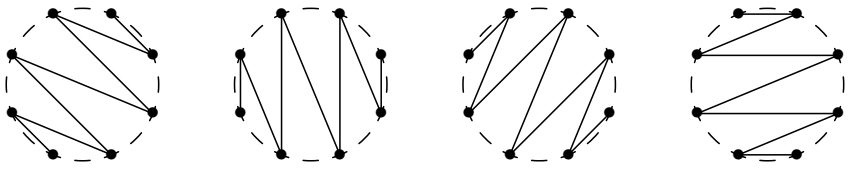


FIG. 4. *Path decomposition of K_8.*

Our representation of K_n has $3m$ modules; each is a bar visibility representation of $P_{2m} \vee 2K_1$ (the *join* $G \vee H$ of graphs G and H is the graph formed from the disjoint union of G and H by adding edges to make each vertex of G adjacent to each vertex of H). We represent the path P_{2m} by a staircase of bars; each sees the bar before and after it. We add one long bar above and one long bar below; each sees the entire path (see Figure 5).

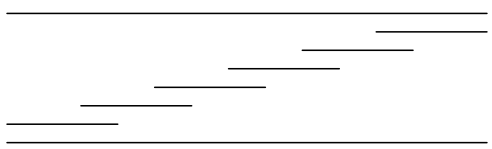


FIG. 5. *Bar visibility representation of* $P_{2m} \vee 2K_1$.

To each path in the decomposition of A_i, we assign two vertices in A_{i+1} (indices modulo 3). This produces m pairwise edge-disjoint copies of $P_{2m} \vee 2K_1$. Their union covers the complete graph on A_i and all edges from A_i to A_{i+1}. Doing this for each A_i yields $3m$ modules whose union represents K_n.

A vertex of A_i appears in each path drawn from A_i, and it appears once as a top or bottom bar in a module for a path in A_{i-1}. Thus each vertex is assigned $m + 1$ bars. □

5. *n*-vertex graphs. When bounding $b(G)$, it is tempting to use Remark 1 and express G as a union of bar visibility graphs. Unfortunately, as we have noted, the number of bars per vertex in the resulting representation is at least the thickness of G, which is too large.

Splitting number is more promising. If $H \subseteq G$, then $s(H) \leq s(G)$, since any sequence of splits that reduces G to a planar graph also reduces H to a planar graph. Thus it may be possible to prove the conjecture that $b(G) \leq \lceil n/6 \rceil$ by using a map with $6m$ pairwise adjacent m-pires (where $m = \lceil n/6 \rceil$), convert it to a splitting of K_{6m}, and delete adjacencies to reach a splitting of G without introducing a need for extra bars.

In addition to the splittings that result from Wessel's construction, one might also start with splittings of K_n studied by Hartsfield, Jackson, and Ringel [9]; they proved that $s(K_n) = \lceil (n-3)(n-4)/6 \rceil$. If their construction split vertices equally often, then it would like Wessel's construction yield an $\lceil n/6 \rceil$-bar representation of K_n.

In both results, the splittings of K_n are hard to describe, and it is not clear how to do the other steps. Instead, we generalize Construction 9 to prove directly that $b(G) \leq \lceil n/6 \rceil + 2$.

It is hard to modify Construction 9 directly to delete arbitrary edges. For example, let u, v, w appear consecutively on some path in the decomposition of A_1, and let y, z be the vertices of A_2 whose bars surround this path. Extending the bar for u or w can block v from seeing y or z. Deleting the bar for v and extending those for u and w to the same vertical line can delete all these edges. However, how can we delete vy, vz, uv and keep vw?

If all edges of the path in A_i were present, then we could delete arbitrary edges to y and z by extending the bars for vertices on the path. If t_k is the maximum number of paths needed to partition the edges of a k-vertex graph, we could thus obtain $b(G) \leq t_{n/3} + 1$. Gallai [6] conjectured that $t_k = \lceil k/2 \rceil$, which would yield

$b(G) \le \lceil n/6 \rceil + 1$.

We do almost as well by using the result of Lovász [15] that every k-vertex graph can be decomposed into $\lfloor k/2 \rfloor$ paths and cycles. Each vertex of odd degree must be an endpoint of some path in such a decomposition. Thus the decomposition must consist entirely of paths when G has at most one vertex of even degree.

THEOREM 10. *If G has n vertices, then $b(G) \le \lceil n/6 \rceil + 2$.*

Proof. By adding isolated vertices, we may assume that n is divisible by 6. Let $n = 6m$, and again partition $V(G)$ into sets A_1, A_2, A_3 of size $2m$. To the subgraph $G[A_i]$ induced by A_i, add one vertex w adjacent to all vertices with even degree in $G[A_i]$; call this graph G_i'. Since G_i' has at most one vertex of even degree, G_i' has a decomposition into m paths, since $\lfloor (2m+1)/2 \rfloor = m$.

With each such path P we associate two "special" vertices y and z of A_{i+1}, using different special vertices for different paths. We design a module that establishes the edges of P and the edges of G from A_i to y and z. We use at most one bar for each vertex of A_i and at most two bars each for y and z. Doing this for each i and each P in the decomposition of G_i' produces an $(m+2)$-bar representation of G (see Figure 6), since each vertex serves as a special vertex only once.

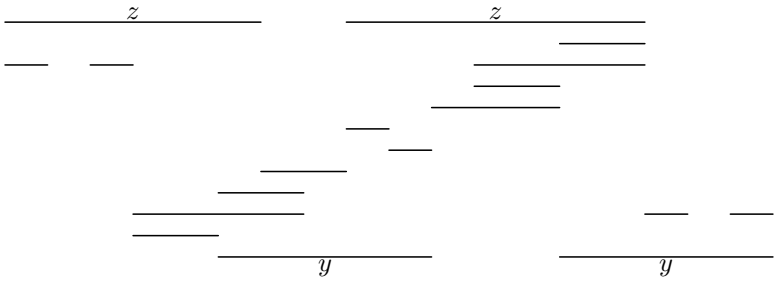


FIG. 6. *Module for representation of general graph.*

Let I_v denote the set assigned to v in such a module. We begin by representing $P \vee \{y, z\}$ as in Figure 5: a staircase plus a bar for y underneath and a bar for z above. The edges on P not involving the added vertex w belong to G, so we do not block these visibilities. Erasing I_w (if $w \in V(P)$) produces a gap that may cause us to break I_y and I_z.

Beginning at the upper right end of P, we block visibilities between y and P as needed. When the current vertex v of P is not adjacent to y, we extend the bar for the next lower vertex of P to the right end of the current I_v. If the last vertex v before w is not adjacent to y, then we cannot extend a lower bar to block I_v from I_y; instead, we break I_y and shorten the left end of the right portion to the right endpoint of I_v.

Delete I_w; note that the bars for the neighbors of w on P do not see each other. Block bars in the lower part of P from seeing I_y as needed, in the same manner as before. Visibilities up to I_z are corrected in the same manner, working from the bottom left end of P.

For $q \in A_i - V(P)$, we also must consider edges from q to $\{y, z\}$. If $y, z \notin N(q)$, then we add no bar for q in this module. For all $q \in N(y) - N(z)$, we add a bar at the right of P; none of these see each other, and I_y extends to the right to see them all. Similarly, bars for vertices of $A_i - V(P)$ in $N(z) - N(y)$ are added at the left of the bars for P.

Now consider common neighbors of y and z in $A_i - V(P)$. If $w \in V(P)$, then we add bars in the gap between the left and right portions of P where I_w was deleted.

Together they fill this gap so that I_y does not see I_z. The left portion of I_y and the right portion of I_z see these bars. If there are no vertices of $V(A) - V(P)$ in $N(y) \cap N(z)$, then we shorten the left portion of I_y and the right portion of z so that they will not see each other. If $w \notin V(P)$, then we did not break I_y or I_z into two bars. We therefore can place another bar for y and z at the right end of the module and use these to establish visibilities for the edges from $(V(A_i) - V(P)) \cap N(y) \cap N(z)$ to $\{y, z\}$, as above.

We have established and/or deleted all the desired adjacencies, using the desired number of bars for each vertex. □

Note added in proof. W. Cao has improved the upper bound in Theorem 7 by eliminating the "+1" when m and n are both even, thereby computing $b(K_{m,n})$ exactly.

REFERENCES

[1] V. B. Alekseev and V.S. Gončakov, *The thickness of an arbitrary complete graph*, Mat. Sb. (N.S.), 101 (1976), pp. 212–230 (in Russian).

[2] J. Balogh and A. Pluhár, *A sharp edge bound on the interval number of a graph*, J. Graph Theory, 32 (1999), pp. 153–159.

[3] J. Battle, F. Harary, and Y. Kodama, *Every planar graph with nine points has a nonplanar complement*, Bull. Amer. Math. Soc. (N.S.), 68 (1962), pp. 569–571.

[4] P. Bose, A. M. Dean, J. P. Hutchinson, and T. Shermer, *On rectangle visibility graphs*, Lecture Notes in Comput. Sci. 1190, S. North, ed., Springer-Verlag, Berlin, 1997, pp. 25–44.

[5] A. M. Dean and J. P. Hutchinson, *Rectangle-visibility representations of bipartite graphs*, Discrete Appl. Math., 75 (1997), pp. 9–25.

[6] T. Gallai, *Über extreme Punkt- und Kantenmengen*, Ann. Univ. Sci. Budapest. Eötvös Sect. Math., 2 (1959), pp. 133–138.

[7] M. Gardner, *Mathematical games*, Sci. Amer., 242 (1980), pp. 14–19.

[8] F. Harary and G. Prins, *The block-cutpoint-tree of a graph*, Publ. Math. Debrecen 13 (1966), pp. 103–107.

[9] N. Hartsfield, B. Jackson, and G. Ringel, *The splitting number of the complete graph*, Graphs Combin., 1 (1985), pp. 311–329.

[10] P. J. Heawood, *Map colour theorem*, Quart. J. Math., 24 (1890), pp. 332–338.

[11] J. P. Hutchinson, *Arc- and circle-visibility graphs*, Australas. J. Combin., 25 (2002), pp. 241–262.

[12] J. P. Hutchinson, T. Shermer, and A. Vince, *On representations of some thickness-two graphs*, Comput. Geom., 13 (1999), pp. 161–171.

[13] B. Jackson and G. Ringel, *Solution of Heawood's empire problem in the plane*, J. Reine Angew. Math., 347 (1984), pp. 146–153.

[14] B. Jackson and G. Ringel, *The splitting number of the complete bipartite graphs*, Arch. Math. (Basel), 42 (1984), pp. 178–184.

[15] L. Lovász, *On covering of graph*, in Theory of Graphs, P. Erdős and G. Katona, eds., Academic Press, New York, 1968, pp. 231–236.

[16] J. O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, New York, 1987.

[17] E. R. Scheinerman and D. B. West, *The interval number of a planar graph: Three intervals suffice*, J. Combin. Theory Ser. B, 35 (1983), pp. 224–239.

[18] R. Tamassia and I. G. Tollis, *A unified approach to visibility representations of planar graphs*, Discrete Comput. Geom., 1 (1986), pp. 321–341.

[19] W. T. Trotter and F. Harary, *On double and multiple interval graphs*, J. Graph Theory, 3 (1979), pp. 205–211.

[20] W. T. Tutte, *On the non-biplanar character of the complete 9-graph*, Canad. Math. Bull., 6 (1963), pp. 319–330.

[21] W. Wessel, *A short solution of Heawood's empire problem in the plane*, Discrete Math., 191 (1998), pp. 241–245.

[22] S. Wismath, *Characterizing bar line-of-sight graphs*, in Proceedings of the 1st ACM Symposium Comput. Geom. Assoc. Comp. Mach., Baltimore, MD, 1985, ACM, New York, pp. 147–152.

SIAM J. DISCRETE MATH.
Vol. 18, No. 3, pp. 472–478

THE RADON TRANSFORM ON $\mathbb{Z}_n^{k*}$

MICHELLE R. DEDEO† AND ELINOR VELASQUEZ†

Abstract. The Radon transform on $\mathbb{Z}_n^k$ averages a function over its values on a translate of a fixed subset S in $\mathbb{Z}_n^k$. We discuss invertibility conditions and computer inverse formulas based on the Moore–Penrose inverse and on linear algorithms. We expect the results to be of use in directional and toroidal time series.

Key words. Radon transform, invertibility, Fourier transform

AMS subject classifications. Primary, 44A15; Secondary, 42A38

DOI. 10.1137/S0895480103430764

1. Introduction. Suppose G is a finite group and fix $S \subset G$. Let $C(G)$ be the space of real-valued maps on G. A finite analogue of the Radon transform may be described as follows: For all $f \in C(G)$, define the Radon transform based on translates of $S \subset G$ as

$$\overline{f}(k) = \sum_{j \in S+a} f(j).$$

Diaconis and Graham [2] discuss the cases where $G = \mathbb{Z}_2^k$, the group of binary k-tuples, and where $G = S_n$, the symmetric group on n letters, in order to provide an exposition on discrete Radon transforms which appear in applied statistics. Fill [4] examines the case $G = \mathbb{Z}_n$, the group of integers modulo n. This case arises in directional data analysis and circular time series. Other finite analogues of the Radon transform occur, but we shall restrict ourselves to the case $G = \mathbb{Z}_n^k$, the group of k-tuples of the integers modulo n, as these results are of use in k-dimensional toroidal time series.

In section 1, we use representation theory as described in DeDeo and Velasquez [1] to describe the Radon transform on $\mathbb{Z}_n^k$ and its invertibility conditions. In section 2, we discuss a specific example, the Radon transform based on a translate of a fixed $S_r \subset \mathbb{Z}_n^k$, where S_r denotes a sphere of radius r. In this situation, $\mathbb{Z}_n^k$ is associated with a Cayley graph with a Hamming metric. The injectivity of the Radon transform is seen to depend on the zeros of particular Krawtchouk polynomials. The proof of this includes a counting argument rather than computing the appropriate spherical functions for the graph. Inversion formulas are presented in section 3. First, explicit formulas are computed using the Fourier transform, its inverse, and a Moore–Penrose generalized inverse transform. Then inverse algorithms, which do not rely on the Fourier transform, are described for the Radon transform based on translates of the fixed spheres S_r in $\mathbb{Z}_n^k$.

1.1. Motivation. This is an extension of DeDeo and Velasqeuz [1] which attempts to extend directional data and time series to discretized analogues of manifolds. Two-dimensional manifolds are homomorphically either spheres with handles

*Received by the editors July 2, 2003; accepted for publication (in revised form) April 6, 2004; published electronically December 30, 2004.
http://www.siam.org/journals/sidma/18-3/43076.html
†Department of Mathematics and Statistics, University of North Florida, Jacksonville, FL 32224 (mdedeo@unf.edu, velasque@sfsu.edu).

or spheres with crosscaps. For dimensions greater than two, the situation is more complex. In this paper we shall inspect the discrete analogue of the sphere with k-handles, i.e., the discretized k-dimensional torus denoted by $\mathbb{Z}_n^k$.

Functions relevant to time series are the maps $f : T^k \to \mathbb{R}$ or $f : \mathbb{Z}_n^k \to \mathbb{R}$, with $\mathbb{Z}_n^k$ the proposed discretization of the k-fold torus. The act of employing a certain linear filter to functions on $\mathbb{Z}_n^k$ may be considered to be the mapping of f to $\overline{f}$, its Radon transform, where

$$\overline{f}(a) = \sum_{j \in \mathbb{Z}_n^k} \varphi(j-a) f(j)$$

for some $\varphi : \mathbb{Z}_n^k \to \mathbb{R}$. If φ is the characteristic function of a fixed S in $\mathbb{Z}_n^k$, the mapping above reduces to the usual Radon transform based on translates of S in $\mathbb{Z}_n^k$.

2. Fourier analysis on $\mathbb{Z}_n^k$. Throughout the next two sections, we set $G = \mathbb{Z}_n^k$, the group of k-tuples with elements in $\mathbb{Z}_n$. Explicitly, $\mathbb{Z}_n^k = (\mathcal{A}, +)$, where $\mathcal{A} = \{x | x = (x_1, x_2, \ldots, x_k)^t$ for x_i in $\mathbb{Z}_n$, where $i = 1, \ldots, k\}$. Assign the natural inner product to $\mathbb{Z}_n^k$ such that, for all x and y in $\mathbb{Z}_n^k$, $x \cdot y = \sum_{i=1}^k x_i y_i$. Thus $\mathbb{Z}_n^k$ is an abelian group under addition and its group representations all have degree one since $\mathbb{Z}_n^k \cong \prod^k \mathbb{Z}_n$. We have that the characters are the homomorphisms $\chi_j : \mathbb{Z}_n^k \to \mathbb{C}^\times$, where

$$\chi_j(z) = \prod_{i=1}^k \chi_{j_i}(z_i)$$

for all $z = (z_1, z_2, \ldots, z_k)^t$ and $j = (j_1, j_2, \ldots, j_k)^t$ in $\mathbb{Z}_n^k$, $\mathbb{C}^\times$ denotes the nonzero complex numbers, and χ_j denotes the character of the jth copy of $\mathbb{Z}_n$. Recall that the characters of $\mathbb{Z}_n$ are explicitly the functions $\chi_m : \mathbb{Z}_n \to \mathbb{C}^\times$ defined by $\chi_m(y) = \omega^{m \cdot y}$ for $m \in \{0, \ldots, n-1\}$, where $\omega = e^{\frac{2\pi i}{n}}$, an nth root of unity. Thus we have that $\chi_j : \mathbb{Z}_n^k \to \mathbb{C}^\times$ defined by $\chi_j(z) = \omega^{j \cdot z}$ indexed by $j \in \mathbb{Z}_n^k$ are the inequivalent, irreducible unitary representations of $\mathbb{Z}_n^k$. We denote the Hilbert spaces of functions on $\mathbb{Z}_n^k$ and $\widehat{\mathbb{Z}_n^k}$, the space of characters of $\mathbb{Z}_n^k$, by $\mathcal{L}^2(\mathbb{Z}_n^k)$, and the space of square-differentiable functions on $\mathbb{Z}_n^k$ by $\mathcal{L}^2(\widehat{\mathbb{Z}_n^k})$.

The Fourier transform for $\mathbb{Z}_n^k$ from $\mathcal{L}^2(\mathbb{Z}_n^k) \to \mathcal{L}^2(\widehat{\mathbb{Z}_n^k})$ is

$$(\mathcal{F}f) = \widehat{f}(x) = \widehat{f}(\chi_x) = \sum_{f(y) \in \mathbb{Z}_n^k} \chi_x(y) = \sum_{f(y) \in \mathbb{Z}_n^k} f(y) \omega^{x \cdot y}$$

with the Fourier inverse from $\mathcal{L}^2(\widehat{\mathbb{Z}_n^k}) \to \mathcal{L}^2(\mathbb{Z}_n^k)$ as

$$\begin{aligned}(\mathcal{F}^{-1}\widehat{f}) = f(y) &= \frac{1}{n^k} \sum_{\widehat{f}(y) \in \widehat{\mathbb{Z}_n^k}} \chi(y^{-1}) \widehat{f}(\chi) \\ &= \frac{1}{n^k} \sum_{x \in \mathbb{Z}_n^k} \chi_x(y^{-1}) \widehat{f}(\chi_x) \\ &= \frac{1}{n^k} \sum_{x \in \mathbb{Z}_n^k} \omega^{-x \cdot y} \widehat{f}(x).\end{aligned}$$

For all functions $f \in C(\mathbb{Z}_n^k)$, the Radon transform

$$(\mathcal{R}f) = \overline{f}(a) = \sum_{j \in S+a} f(j) = \sum_{j \in \mathbb{Z}_n^k} \varphi_S(j-a) f(j),$$

where $\varphi_S(\cdot)$ is the characteristic function of S in $\mathbb{Z}_n^k$. Therefore, we can identify the Radon transform $\mathcal{R}$ with the matrix

$$(\mathcal{R})_{k,j} = \varphi_S(j-k)$$

and the finite Fourier transform $\mathcal{F}$ and its inverse with the matrices

$$(\mathcal{F})_{l,m} = \omega^{l\cdot m} \text{ and } \left(\mathcal{F}^{-1}\right)_{l,m} = \frac{1}{n^k}\left(\mathcal{F}\right)^*_{m,l} = \frac{1}{n^k}\omega^{l\cdot m},$$

respectively, where ω is a fixed root of unity and * denotes the conjugate transpose. We wish to describe the invertibility of the Radon matrix. The group $\left(\mathbb{Z}_n^k, +\right)$ is abelian; therefore, singular value decompression via Fourier matrices results in a diagonal matrix.

PROPOSITION 2.1. *$\left(\mathcal{F}\mathcal{R}\mathcal{F}^*\right)_{j,l} = \delta_{j,l} n^k \widehat{\varphi}_s(-l)$, where $\delta_{j,l} = 1$ if $j = l$ and 0 otherwise.*

Proof. We refer the reader to DeDeo and Velasquez [1]. □

We note that the Radon matrix is not invertible if $S = \mathbb{Z}_n^k$ as it leads to a matrix of all 1's, and that the matrix is always invertible if S has one element.

3. The Krawtchouk polynomial and invertibility. Consider a subset of $\mathbb{Z}_n^k$. Specifically, for a fixed $r \in \mathbb{N}$, set $S = S_r = \{x \in \mathbb{Z}_n^k | H(x) = r\} = H(x)$, where $H(x)$ is the Hamming distance of x from the origin of $\mathbb{Z}_n^k$. In other words, we associate $\mathbb{Z}_n^k$ with the graph $X = X(V, E)$ with vertex set $V = \mathbb{Z}_n^k$ and edge set $E = \{(x, y) \in V \times V | H(x, y) = 1\}$, and where $H(x)$ is the Hamming metric, the number of coordinates in which x and y differ.

DEFINITION 3.1. *Fix the following integers r in $0, \ldots, k$ and let q be a prime. The Krawtchouk polynomial (MacWilliams and Sloane* [6]*) is*

$$p_r^k(\nu; q) = \sum_{l \in \mathbb{Z}_r} (-1)^l (q-1)^{k-1} \binom{\nu}{l} \binom{k-\nu}{r-l},$$

where $\binom{a}{b}$ denotes the usual binomial coefficient. Setting $q = 2$ results in the form we will be using for the remainder of the paper:

$$p_r^k(\nu) = \sum_{l \in \mathbb{Z}_r} (-1)^l \binom{\nu}{l} \binom{k-\nu}{r-l}.$$

PROPOSITION 3.2 (DeDeo and Velasquez [1]). *For $x \in \mathbb{Z}_n^k$, $\widehat{\varphi_{S_r}}(x) = p_r^k(H(x))$.*

Proof. For $x \in \mathbb{Z}_n^k$, we have $\widehat{\varphi_{S_r}}(x) = \sum_{s \in S_r} \omega^{x \cdot s}$, where ω is a primitive nth root of unity. Since $\widehat{\varphi_{S_r}}(x)$ depends on $x = (x_1, \ldots, x_k)^t$ in $\mathbb{Z}_n^k$ only through the unordered set $\{x_1, \ldots, x_k\}$, then we may assume, without loss of generality, that $x_i \neq 0$ for $i = 1, \ldots, h$, where $h := H(x)$ and $x_i = 0$ for $i = h+1, \ldots, k$. Now

$$\widehat{\varphi_{S_r}}(x) = \sum_{l=0}^{r} \sum_{\{\alpha_1, \ldots, \alpha_l\} \subset \{1, \ldots, k\}} \sum_{s \in S_r(\{\alpha_1, \ldots, \alpha_l\})} \omega^{x \cdot s}$$

with $S_r(\{\alpha_1, \ldots, \alpha_l\}) := \{s \in S | s_i \neq 0 \text{ for } i \in \{\alpha_1, \ldots, \alpha_l\} \text{ and } s_i = 0 \text{ for } i \in \{1, \ldots, k\} / \{\alpha_1, \ldots, \alpha_l\}\}$.

Then

$$\begin{aligned}\sum_{s\in S_r(\{\alpha_1,\ldots,\alpha_l\})}\omega^{x\cdot s} &= \binom{k-h}{r-l}\sum_{S_{\alpha_1}\neq 0}\cdots\sum_{S_{\alpha_l}\neq 0}\omega^{\sum_{j=1}^{l}x_{\alpha_j}\cdot s_{\alpha_j}}\\ &= \binom{k-h}{r-l}\prod_{j=1}^{l}\left[\sum_{s=1}^{k}\omega^{x_{\alpha_j}\cdot s}\right]\\ &= \binom{k-h}{r-l}\prod_{j=1}^{l}\left[0-\omega^{x_{\alpha_j}\cdot 0}\right]\\ &= (-1)^l\binom{k-h}{r-l}\end{aligned}$$

since ω is a primitive root of unity and $x_{\alpha_j}\neq 0$.

Hence $\widehat{\varphi_{S_r}}(x)=\sum_{l=0}^{r}\binom{H(x)}{l}(-1)^l\binom{k-H(x)}{r-l}=p_r^k(H(x))$. □

4. Inversion algorithms. We now consider the case of an invertible Radon matrix along with a typically noninvertible Radon matrix. Inversion formulas using standard Fourier methods are constructed. Then follows an exposition of inversion algorithms which do not use Fourier transforms.

DEFINITION 4.1. *If T is a finite-dimensional matrix, let the Moore–Penrose generalized inverse matrix of T be the unique matrix U satisfying* (1) $TUT=T$; (2) $UTU=U$; (3) $(TU)^*=TU$; (4) $(UT)^*=UT$, *where * is the conjugate transpose of T.*

We shall denote U by $T^{\div}$.

PROPOSITION 4.2 (DeDeo and Velasquez [1]). *Suppose $f\in C(\mathbb{Z}_n^k)$. Then we have the following.*

i. *If $\varphi_S(x)\neq 0$ for all $x\in\mathbb{Z}_n^k$, then, for all $z\in\mathbb{Z}_n^k$,*

$$f(x)=\sum_{z\in\mathbb{Z}_n^k}\overline{f}(z)\cdot\frac{1}{n^k}\sum_{y\in\mathbb{Z}_n^k}\frac{\omega^{(z-x)\cdot y}}{\widehat{\varphi_{S_r}}(-y)}.$$

ii. *If $\varphi_S(x)=0$ for some $x\in\mathbb{Z}_n^k$, let matrix Λ be given by*

$$(\Lambda)_{j,l}=\delta_{jl}n^k\sum_{s\in S}\omega^{-l\cdot s},$$

which implies that

$$(\Lambda^{\div})_{jl}=\delta_{jl}\lambda_l^{\div}\ with\ \lambda_l^{\div}=\begin{cases}\frac{1}{n^k}\left(\sum_{s\in S}\omega^{-l\cdot s}\right)^{-1} & if\ \sum_{s\in S}\omega^{-l\cdot s}\neq 0.\\ 0 & otherwise\end{cases}$$

Then $f=\mathcal{R}^{\div}\overline{f}$ with $\mathcal{R}^{\div}\equiv\mathcal{F}^*\Lambda^{\div}\mathcal{F}$. In other words, the reconstruction of $f\in C(\mathbb{Z}_n^k)$ for $\widehat{\varphi_S}(x)$ is

$$f(x)\overset{def}{=}\mathcal{R}^{\div}(\overline{f}(x))=\frac{1}{n^k}\sum_{y,z\in\mathbb{Z}_n^k}\lambda_y^{\div}\omega^{y\cdot(z-x)}\overline{f}(x).$$

Fill [4] estimates the possibility of the accurate reconstruction of a function by a least squares error discussion. This argument is completely valid for $\mathbb{Z}_n^k$. Hence, we provide a brief sketch here and refer the reader to Fill [4] for the details.

Given g in $C(\mathbb{Z}_n^k)$, we need to redefine the residual vector $E(f)$ for all f in $C(\mathbb{Z}_n^k)$ such that

$$E(f) = g - \mathcal{R}f \text{ in } C(\mathbb{Z}_n^k).$$

Then

$$\|E(f)\|^2 = \|g - \mathcal{R}f\|^2 = \sum_{x \in \mathbb{Z}_n^k} |g(x) - (\mathcal{R}f)(x)|^2 .$$

If $f_0 = \mathcal{R}^{\div}\overline{f}$ and $f = f_0 + h$, where $h \in C(\mathbb{Z}_n^k)$, then the least squares error is

$$\|E_0\|^2 = \|f - f_0\|^2 = \left\|(I - \mathcal{R}\mathcal{R}^{\div})g\right\|,$$

where I is the identity transform.

We now consider inversion algorithms based on a linear equations approach rather than the use of Fourier transforms. Diaconis and Graham [2] consider algorithms for shells and balls of Hamming radius 1. We shall consider shells to indicate the general scheme.

PROPOSITION 4.3. *Suppose f is in $C(\mathbb{Z}_n^k)$. For $m \in \{0, \ldots, k\}$, define*

$$g(m) = \sum_{H(x)=m} f(x) \quad \textit{and} \quad \overline{g}(m) = \sum_{H(x)=m} \overline{f}(x),$$

where $\overline{f}$ is the Radon transform of f on translates of a shell of Hamming radius 1. Then

$$\overline{g}(m) = (n-1)(k-m+1) \cdot g(m-1) + (n-2)m \cdot g(m) + (m+1) \cdot g(m+1)$$

with $\overline{g}(-1) \equiv \overline{g}(k+1) \equiv 0$.

Proof. Given $f \in C(\mathbb{Z}_n^k)$, note that the Radon transform of f on a shell of radius 1 is

$$\overline{f}(x) = \sum_{y \in S_1 + x} f(y) = \sum_{y : H(x,y)=1} f(y).$$

Given x in $\mathbb{Z}_n^k$, we examine y in $\mathbb{Z}_n^k$ such that $H(x,y) = 1$. Suppose $x \in S_m \overset{def}{=} \{x \in \mathbb{Z}_n^k | H(x) = m\}$. Then there are three possible radii for $y : H(y) = m-1, m$, or $m+1$. We shall consider each case separately.

First, we discuss some notation. Fix $m \in \{0, \ldots, k\}$ and consider $w = w_{i_1} \ldots w_{i_m}$ in S_m. Then w has m nonzero coordinates w_{i_α} for $\alpha \in \{1, \ldots, m\}$. We choose $\{i_\alpha\}_1^m \subset \{j\}_1^k$.

1. $H(y) = m-1$. Let $x = x_{i_1} \ldots x_{i_m} \in S_m$. It is clear that there exists a y in S_{m-1} such that $y = x_{i_1} \ldots x_{i_{m-1}}$. For a fixed y, partition S_m into subsets where

   $$\mathcal{F}_y = \{x_{i_1} \ldots x_{i_{m-1}} x_{i_m} | y = x_{i_1} \ldots x_{i_{m-1}}\}.$$

   Since x_{i_m} has $k - (m-1)$ possible coordinate positions in the k-tuple and $n-1$ possible nonzero values to assume at each position, we have that the cardinality of $\mathcal{F}_y$ is $(k-m+1)(n-1)$ for a fixed y. It is also clear that there is a one-to-one correspondence between the $y \in S_m$ and $\mathcal{F}_y$.

2. $H(y) = m$. Let $x = x_{i_1} \dots x_{i_m} \in S_m$. Each x_{i_a} has some value in $\mathbb{Z}^\times$. Thus each coordinate is a map $x_{i_\nu} : \mathcal{A} \subset \mathbb{Z}_n^k \to \mathbb{Z}_n^\times$, where $x_{i_\nu} \to x_{i_\nu}(\beta_\nu) \equiv x_{i_\nu}^{\beta_\nu}$ for $\nu = 1, \dots, m$. Fix x_0 in S_m. Then x_0 looks like $x_{i_1}^{\alpha_1} \dots x_{i_m}^{\alpha_m}$ for fixed coordinate positions $\{i_\nu\}_1^m$ and fixed values $\{\alpha_\nu\}_1^m$. If $y \in S_m \cap \{y | H(x,y) = 1\}$, then y can have the forms

$$x_{i_1}^{\beta_1} x_{i_2}^{\alpha_2} x_{i_3}^{\alpha_3} \dots x_{i_m}^{\alpha_m}, \quad x_{i_1}^{\alpha_1} x_{i_2}^{\beta_2} x_{i_3}^{\alpha_3} \dots x_{i_m}^{\alpha_m}, \dots, \quad x_{i_1}^{\alpha_1} \dots x_{i_{m-1}}^{\alpha_{m-1}} x_{i_m}^{\beta_m},$$

where $\beta_j \neq \alpha_j$ for $j = 1, \dots, m$. Outputting other x's of the form

$$x_{i_1}^{\gamma_1} x_{i_2}^{\alpha_2} x_{i_3}^{\alpha_3} \dots x_{i_m}^{\alpha_m}, \quad x_{i_1}^{\alpha_1} x_{i_2}^{\gamma_2} x_{i_3}^{\alpha_3} \dots x_{i_m}^{\alpha_m}, \dots, x_{i_1}^{\alpha_1} \dots x_{i_{m-1}}^{\alpha_{m-1}} x_{i_m}^{\gamma_m},$$

where $\gamma_j \neq \alpha_j$, results in the y's associated with each outputted x. For example, consider $x_{i_1}^{\gamma_1} x_{i_2}^{\alpha_2} x_{i_3}^{\alpha_3} \dots x_{i_m}^{\alpha_m}$, where $\gamma_1 \neq \alpha_1$. We already know that the associated y's are

$$\begin{aligned} &x_{i_1}^{\beta_1 \neq \gamma_1} x_{i_2}^{\beta_2 = \alpha_2} x_{i_3}^{\beta_3 = \alpha_3} \dots x_{i_m}^{\beta_m = \alpha_m}, \\ &x_{i_1}^{\beta_1 = \alpha_1} x_{i_2}^{\beta_2 \neq \alpha_2} x_{i_3}^{\beta_3 = \alpha_3} \dots x_{i_m}^{\beta_m = \alpha_m}, \\ &\dots, x_{i_1}^{\beta_1 = \alpha_1} \dots x_{i_{m-1}}^{\beta_{m-1} = \alpha_{m-1}} x_{i_m}^{\beta_m \neq \alpha_m} \end{aligned}$$

for β_δ in $1, \dots, d-1$ if $\delta = 1, \dots, m$ unless stated otherwise. We need only examine y in $\{x_{i_1}^{\beta_1 \neq \gamma_1} x_{i_2}^{\beta_2 = \alpha_2} x_{i_3}^{\beta_3 = \alpha_3} \dots x_{i_m}^{\beta_m = \alpha_m}\}_{\beta_1 = 1}^{\beta_1 = d-1}$. This set has cardinality $d-2$ and contains a unique y_0 such that $y_0 = x_{i_1}^{\alpha_1} \dots x_{i_m}^{\alpha_m}$. In other words, there exists β_1 such that $\beta_1 = \alpha_1 \neq \gamma_1$ for some β_1 in $1, \dots, d-1$. For each fixed γ_1, we can find a copy of x_0 which we denote as y_0. Thus there are $d-2$ copies of x_0 among the y associates if we output x in S_m such that $x = x_{i_1}^{\gamma_1} x_{i_2}^{\alpha_2} x_{i_3}^{\alpha_3} \dots x_{i_m}^{\alpha_m}$ for all $\gamma_1 = 1, \dots, d-1$, where $\gamma_1 \neq \alpha_1$.
If we output all forms of x as described above, we get $(d-2) * m$ copies of x_0. No more repetitions of x_0 can occur amongst the y associates because we have exhausted all possible forms of x with which to compute y associates. Since x_0 was arbitrary, outputting all x from S_m results in $(d-2) * m$ copies of each y in S_m.
3. $H(y) = m+1$. Fix y_0 in S_{m+1}. Then $y_0 = y_{i_1}^{\alpha_1} y_{i_2}^{\alpha_2} \dots y_{i_{m+1}}^{\alpha_{m+1}}$ for fixed coordinate positions $\{i_\nu\}_1^{m+1}$ and fixed values $\{\alpha_\nu\}_1^{m+1}$. Partition S_m into families which are projections of y in S_{m+1}. (For example, y_0 has the projection $\mathcal{F}_{y_0} = \{y_{i_1}^{\alpha_1} y_{i_2}^{\alpha_2} \dots y_{i_m}^{\alpha_m}, y_{i_1}^{\alpha_1} y_{i_2}^{\alpha_2} \dots y_{i_{m-1}}^{\alpha_{m-1}} y_{i_{m+1}}^{\alpha_{m+1}}, \dots, y_{i_2}^{\alpha_2} y_{i_3}^{\alpha_3} \dots y_{i_{m+1}}^{\alpha_{m+1}}\}$.) Then given y in S_{m+1}, the cardinality of $\mathcal{F}_{y_0}$ is $m+1$. It is clear that there exists a one-to-one correspondence between y in S_{m+1} and $\mathcal{F}_{y_0}$ as a subset of S_{m+1}. □

We use the results of Proposition 4.2 to describe the system of equations $\bar{g}(p) = (G)_{m,p} g(m)$ for $0 \leq m, p \leq k$ and

$$G \stackrel{def}{=} (G)_{m,p} \stackrel{def}{=} \begin{cases} (n-1) \cdot (k-m+1) & \text{when } m = p-1, \\ (n-2) \cdot m & \text{when } m = p, \\ m+1 & \text{when } m = p+1, \\ 0 & \text{otherwise.} \end{cases}$$

Then G is a singular, nonsymmetric tridiagonal matrix. (Note that when $r \geq 1$, G is a singular, nonsymmetric band-limited matrix with bandwidth proportional to r.)

Therefore, in order to describe the system of equations $g(m) = ((G)_{m,p})^{-1}\bar{g}(p)$, we need to set $(G)^{-1} \equiv G^{+}$, the generalized Moore–Penrose inverse of G.

PROPOSITION 4.4. *Suppose f is in $C(\mathbb{Z}_n^k)$. If $\widehat{\chi}_{S_1} \not\equiv 0$, then*

$$f(y) = \sum_{H(x,y)=0}^{k} (G^{+})_{1,H(x,y)}\overline{f}(x),$$

where

$$\overline{f}(x) = \sum_{H(x,y)=1} f(y)$$

and G^{+} is the Moore–Penrose inverse of G.

Proof. We note that from DeDeo and Velasquez [1], the inversion problem has become one of inverting singular, nonsymmetric band-limited matrices. For $r = 1$, band-limited means inverting the tridiagonal matrix G computed in Proposition 4.2. Using the definitions and results from Proposition 4.2, we have that

$$g(0) = \sum_{\beta=0}^{k} (G^{+}(s,t))_{1,\beta} g(\beta)$$

with $(G^{+}(s,t))_{\alpha,\beta} = G^{+}(s,t)$, the Moore–Penrose inverse of $G = G(s,t)$. Then, since $f(0) = g(0)$, we have that

$$f(0) = \sum_{H(x)=0}^{k} (G^{+}(s,t))_{1,H(x,y)}\overline{f}(x)$$

and, by a shift action,

$$f(y) = \sum_{H(x,y)=0}^{k} (G^{+}(s,t))_{1,H(x,y)}\overline{f}(x)\ . \qquad \square$$

REFERENCES

[1] M. R. DeDeo and E. Velasquez, *An introduction to the Radon transform on $\mathbb{Z}_2^k$*, Congr. Numer., 156 (2002), pp. 201–209.

[2] P. Diaconis and R. Graham, *The Radon transform on $\mathbb{Z}_2^k$*, Pacific J. Math., 118 (1985), pp. 323–345.

[3] C. Dunkel, *A Krawtchouk polynomial addition theorem and wreath products of symmetric groups,* Indiana Univ. Math. J., 25 (1976), pp. 335–358.

[4] J. A. Fill, *The Radon transform on $\mathbb{Z}_n$*, SIAM J. Discrete Math., 2 (1989), pp. 262–283.

[5] G. M. L. Gladwell and N. B. Williams, *A discrete Gel'fand-Levitan method for band-matrix inverse eigenvalue problems*, Inverse Problems, 5 (1989), pp. 165–279.

[6] J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes*, North Holland, Amsterdam, 1977.

[7] P. Rózsa, R. Bevilacqua, F. Romani, and P. Favati, *On band matrices and their inverses*, Linear Algebra Appl., 150 (1991), pp. 287–295.

[8] D. Stanton, *Orthogonal polynomials and Chevalley groups*, in Special Functions: Group Theoretical Aspects and Applications, R. A. Askey et al., eds., D. Reidel, Boston, MA, 1984, pp. 87–128.

[9] D. Stanton, *An introduction to group representations and orthogonal polynomials*, in Orthogonal Polynomials, P. Nevai, ed., Kluwer Academic, Norwell, MA, 1990, pp. 419–433.

SIAM J. DISCRETE MATH.
Vol. 18, No. 3, pp. 479–487

ORIENTABLE AND NONORIENTABLE GENERA FOR SOME COMPLETE TRIPARTITE GRAPHS*

KEN-ICHI KAWARABAYASHI†, CHRIS STEPHENS‡, AND XIAOYA ZHA§

Abstract. In this paper, we obtain three general reduction formulas to determine the orientable and nonorientable genera for complete tripartite graphs. As corollaries, we (1) reduce the determination of the orientable (nonorientable, respectively) genera of 75 percent (85 percent, respectively) of nonsymmetric (with respect to l, m, and n) $K_{l,m,n}$ to that of $K_{m,m,n}$, and (2) determine the orientable and nonorientable genera for several classes of complete tripartite graphs.

Key words. complete tripartite graph, orientable genus, nonorientable genus

AMS subject classification. 05C10

DOI. 10.1137/S0895480103429319

1. Introduction. The surfaces appearing in this paper are closed compact 2-manifolds without boundaries. The orientable surface with h handles is denoted by $S_h, h \geq 0$. The nonorientable surface with k crosscaps is denoted by $N_k, k \geq 1$. The *orientable genus* $\gamma(G)$ of a graph G is the minimum h such that G has an embedding into the surface S_h. Likewise, the *nonorientable genus* $\overline{\gamma}(G)$ of G is the minimum k such that G has an embedding into N_k [GT, MT].

While the orientable and nonorientable genera of the whole family of complete bipartite graphs were determined by Ringel [Ri1, Ri2] as early as the 1960s, only partial results for the genera of complete tripartite graphs are known. Let $K_{l,m,n}$, $l \geq m \geq n$, denote the family of complete tripartite graphs. By applying Euler's formula, Stahl and White [SW] provided the following lower bounds for genera of $K_{l,m,n}$:

$$\gamma(K_{l,m,n}) \geq \left\lceil \frac{(l-2)(m+n-2)}{4} \right\rceil, \ l \geq m \geq n \geq 1, \tag{1.1}$$

$$\overline{\gamma}(K_{l,m,n}) \geq \left\lceil \frac{(l-2)(m+n-2)}{2} \right\rceil, \ l \geq m \geq n \geq 1. \tag{1.2}$$

They also made the following conjecture.

Conjecture 1.1 (Stahl and White [SW]). Equality holds in (1.1) and (1.2).

White [Wh1] proved that the orientable conjecture is true for $K_{l,m,n}$, where $m + n \leq 6$, and for $K_{mn,n,n}$, where $m, n \in \mathbf{N}$ [Wh2].

Ringel and Youngs [RY] also proved the orientable conjecture for $K_{n,n,n}$. Stahl and White [SW] proved that the orientable conjecture holds for $K_{n,n,n-2}$ when $n \geq 2$ is even, and for $K_{2n,2n,n}$ for all $n \geq 1$. They also showed that the nonorientable conjecture holds for $K_{n,n,n-2}$ for all $n \geq 3$, and for $K_{n,n,n-4}$ when $n \geq 4$ is even.

*Received by the editors June 6, 2003; accepted for publication (in revised form) March 30, 2004; published electronically December 30, 2004.

http://www.siam.org/journals/sidma/18-3/42931.html

†Graduate School of Information Sciences, Tohoku University, Aoba-ku Sendai, Miyagi 980-8597, Japan (k_keniti@dais.is.tohoku.ac.jp).

‡Department of Mathematics, 1326 Stevenson Center, Vanderbilt University, Nashville, TN 37240 (cstephen@math.vanderbilt.edu). The research of this author was supported by NSF grant DMS-0070613.

§Department of Mathematical Sciences, Middle Tennessee State University, Murfreesboro, TN 37132 (xzha@mtsu.edu). The research of this author was supported by NSF grant DMS-0070430.

Nothing further was discovered until 1998, when Craft [Cr1] proved that the orientable conjecture holds for $K_{l,m,n}$ when $l \geq 2(m+n-2)$ and when $m+n$ and l are both even. (Craft also determined the orientable genera for more complete tripartite graphs in his dissertation [Cr2].) Recently, Ellingham, Stephens, and Zha [ESZ1] proved that, in fact, the nonorientable conjecture is not true for $K_{3,3,3}$, $K_{4,4,1}$, and $K_{4,4,3}$. They also showed that these are the only exceptions to either conjecture for $K_{l,m,n}$ with $l \geq m \geq n$ and $l \leq 5$. Together with K_7 being the only counterexample among the complete graphs to the possible nonorientable genus values obtained from Euler's formula, these counterexamples seem to form an interesting phenomenon: Suppose $\mathcal{T}$ is a family of graphs. If the possible values for orientable genera obtained by the Euler formula are, in fact, right for all graphs in $\mathcal{T}$, then the possible values for nonorientable genera obtained by Euler's formula may also be right for all graphs in $\mathcal{T}$, except for a few small cases. Counterexamples obtained in [ESZ1] may also be considered as graphs called *orientably simple.* (A graph is called orientably simple if its orientable genus is g and its nonorientable genus is $2g+1$, which is an obvious upper bound for the nonorientable genus (see [SB] and [HMR] for details).) It would be nice to see that these three graphs are the only counterexamples to Conjecture 1.1.

In this paper, we first provide some constructions which reduce the determination of genera of some complete tripartite graphs to the determination of the genera of other complete tripartite graphs. In particular, we reduce the verification of the genera of $K_{l,m,n}$ for over 75 percent of all generally nonsymmetric triples (l,m,n) to that of semisymmetric triples (m,m,n). We hope this reduction will shed some light on the verification of Conjecture 1.1. Parallel results for nonorientable genera are also obtained. As corollaries of the general reduction formulas, we confirm Conjecture 1.1 for several larger subclasses of $K_{l,m,n}$.

Since submitting the original version of this paper, two of the authors, together with Ellingham, have shown [ESZ2] that the nonorientable portion of Conjecture 1.1 is true with only the three exceptions described in [ESZ1]; the constructions in this paper form the step for the induction argument used in [ESZ2].

2. Technical lemmas. In this section we prove two technical lemmas about the ceiling function.

LEMMA 2.1. *Let $f(x_1,\ldots,x_s)$ and $g(x_1,\ldots,x_s)$ be polynomials with integer coefficients on multivariables $x_1,\ldots,x_s$, and let $n_1,\ldots,n_s$ be integers. Suppose p is a positive integer. Then*

$$\left\lceil \frac{f(n_1,\ldots,n_s)}{p} \right\rceil + \left\lceil \frac{g(n_1,\ldots,n_s)}{p} \right\rceil = \left\lceil \frac{f(n_1,\ldots,n_s)+g(n_1,\ldots,n_s)}{p} \right\rceil \tag{2.1}$$

if and only if

$$\left\lceil \frac{f(n_1^*,\ldots,n_s^*)}{p} \right\rceil + \left\lceil \frac{g(n_1^*,\ldots,n_s^*)}{p} \right\rceil = \left\lceil \frac{f(n_1^*,\ldots,n_s^*)}{p} + \frac{g(n_1^*,\ldots,n_s^*)}{p} \right\rceil,$$

where $n_i \equiv n_i^* \bmod p, i = 1,\ldots,s$.

Proof. We first observe that if n is an integer and x is a real number, then

$$\lceil n+x \rceil = n + \lceil x \rceil. \tag{2.2}$$

We use LHS and RHS to represent the left-hand side and the right-hand side of (2.1), respectively.

Suppose $n_i = pq_i + n_i^*,\ i = 1, \ldots, s$. Then

$$\begin{aligned}\text{LHS} &= \left\lceil \frac{f(pq_1 + n_1^*, \ldots, pq_s + n_s^*)}{p} \right\rceil + \left\lceil \frac{g(pq_1 + n_1^*, \ldots, pq_s + n_s^*)}{p} \right\rceil \\ &= \left\lceil \frac{f(n_1^*, \ldots, n_s^*) + pA_f}{p} \right\rceil + \left\lceil \frac{g(n_1^*, \ldots, n_s^*) + pA_g}{p} \right\rceil \\ &= A_f + \left\lceil \frac{f(n_1^*, \ldots, n_s^*)}{p} \right\rceil + A_g + \left\lceil \frac{g(n_1^*, \ldots, n_s^*)}{p} \right\rceil,\end{aligned}$$

where pA_f and pA_g are the parts of the binomial expansions of $f(pq_1+n_1^*, \ldots, pq_s+n_s^*)$ and $g(pq_1 + n_1^*, \ldots, pq_s + n_s^*)$, respectively, that contain p as a factor.

Similarly,

$$\text{RHS} = A_f + A_g + \left\lceil \frac{f(n_1^*, \ldots, n_s^*)}{p} + \frac{g(n_1^*, \ldots, n_s^*)}{p} \right\rceil.$$

Therefore Lemma 2.1 is true. □

LEMMA 2.2. *Let $l \geq k \geq m \geq n \geq 1$ be four positive integers. Then*

(i)

$$\left\lceil \frac{(l-k)(m+n-2)}{2} \right\rceil + \left\lceil \frac{(k-2)(m+n-2)}{2} \right\rceil = \left\lceil \frac{(l-2)(m+n-2)}{2} \right\rceil + \epsilon_1, \tag{2.3}$$

(ii)

$$\left\lceil \frac{(l-k)(m+n-2)}{4} \right\rceil + \left\lceil \frac{(k-2)(m+n-2)}{4} \right\rceil = \left\lceil \frac{(l-2)(m+n-2)}{4} \right\rceil + \epsilon_2, \tag{2.4}$$

where

$$\epsilon_1 = \begin{cases} 1 & \text{if } l \text{ even, } k \text{ odd, } m+n \text{ odd,} \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\epsilon_2 = \begin{cases} 1 & \text{if } (l, k, m+n) \in S \text{ in Table 2.1,} \\ 0 & \text{otherwise.} \end{cases}$$

Proof. We first prove (2.3). If k is even, or if $m+n$ is even, then $\frac{(k-2)(m+n-2)}{2}$ is an integer. By (2.2), the lemma is true with $\epsilon_1 = 0$. Similarly, if both l and k are odd, then $\frac{(l-k)(m+n-2)}{2}$ is an integer, and therefore the lemma is also true with $\epsilon_1 = 0$.

This leaves the case that l is even and both k and $m+n$ are odd. By Lemma 2.1, we only need to consider the case that $l = 0$, $k = m + n = 1$. It is easy to verify that the lemma is true in this case with $\epsilon_1 = 1$. Therefore (2.3) is true.

We now prove (2.4). Let η_1 and η_2 be the least nonnegative residue (mod 4) of $\frac{(l-k)(m+n-2)}{4}$ and $\frac{(k-2)(m+n-2)}{4}$, respectively. It is clear that the following is true.

Observation. (2.4) holds with $\epsilon_2 = 0$ if and only if either one of η_1 and η_2 is 0, or $\eta_1 + \eta_2 > 1$.

By Lemma 2.1, we only need to consider the cases with $0 \leq l, k, m+n \leq 3$.

Case 1. Both k and $m+n$ are even. In this case, $\eta_2 = 0$, and by the observation, (2.4) is true with $\epsilon_2 = 0$.

TABLE 2.1
$S = \{(l, k, m+n)\}$ *(modulo 4).*

l	k	$m+n$
0	1	0
	3	0
	1	1
	3	3
1	0	3
	3	3
2	1	0
	3	0
	0	1
	0	3
	1	1
	1	3
	3	3
	3	1
3	0	1
	1	1

Case 2. k is odd and $m+n$ is even. If l is also odd or $m+n=2$, then $\eta_1 = 0$, and hence (2.4) is true with $\epsilon_2 = 0$.

Now we assume that $l = 0$ or 2, and $m+n=0$. In all four cases of $(l, k, m+n)$, $\eta_1 = \eta_2 = \frac{1}{2}$. Hence by the observation, (2.4) holds with $\epsilon_2 = 1$.

Case 3. k is even and $m+n$ is odd. If $k=2$ or $l=k$, then either $\eta_2 = 0$ or $\eta_1 = 0$, and hence (2.4) holds with $\epsilon_2 = 0$.

Now we assume $k=0$ and $l = 1, 2$, or 3. If $m+n=1$ and $l=2$, or if $m+n=3$ and $l=2$, then $\eta_1 = \eta_2 = \frac{1}{2}$. If $m+n=1$ and $l=3$, or if $m+n=3$ and $l=1$, then $\eta_1 = \frac{1}{4}$. Therefore (2.4) holds with $\epsilon_2 = 1$.

Case 4. k and $m+n$ are both odd. There are 16 cases of $(l, k, m+n)$ with $l = 0, 1, 2, 3,\ k = 1, 3$, and $m+n = 1, 3$.

If $l = k = 1$ or 3, and $m+n = 1$ or 3, then $\eta_1 = 0$. By (2.2), (2.4) holds with $\epsilon_2 = 0$.

If $k = 1, m+n = 3$, and $l = 0$ or 3, or if $k = 3, m+n = 1$, and $l = 0$ or 1, then $\eta_1 \geq \frac{2}{4}$ and $\eta_2 = \frac{3}{4}$. By the observation, (2.4) holds with $\epsilon_2 = 0$.

If $k = 1, m+n = 3$, and $l = 2$, or if $k = 3, m+n = 1$, and $l = 2$, then $\eta_1 = \frac{1}{4}$. If $k = 1, m+n = 1$, and $l = 0, 2$, or 3, or if $k = 3, m+n = 3$, and $l = 0, 1$, or 2, then $\eta_2 = \frac{1}{4}$. Therefore $\eta_1 + \eta_2 \leq 1$, and hence by the observation, (2.4) holds with $\epsilon_2 = 1$ for these cases. This finishes the verification of (2.4). Therefore Lemma 2.2 is true. □

3. Main results. The following operation is an extension of Mohar and Thomassen's [MT, Theorem 4.4.7, pp. 117–118] interpretation of a construction by Bouchet [Bo]. Bouchet used his construction to give a new proof for the genus of a complete bipartite graph.

OPERATION 3.1. *Let $\Psi_1 : G_1 \to \Sigma_1$ and $\Psi_2 : G_2 \to \Sigma_2$ be two embeddings of G_1 and G_2 into the surfaces Σ_1 and Σ_2, respectively. Suppose $u \in V(G_1)$ and $v \in V(G_2)$ are two vertices, each adjacent to n vertices $u_1, u_2, \ldots, u_n \in V(G_1)$ and $v_n, v_{n-1}, \ldots, v_1 \in V(G_2)$, respectively, in this clockwise order. Let D_1 be a closed*

disk contained in a small neighborhood of the star $st(u) = \{u\} \cup \{uu_1, uu_2, \ldots, uu_n\}$ *that contains* $st(u)$ *and intersects* G_1 *only at* $u_1, u_2, \ldots, u_n$. *Choose* D_2 *in a small neighborhood of the star* $\{v\} \cup \{vv_1, vv_2, \ldots, vv_n\}$ *in a similar way. Remove* D_1° *and* D_2° *(the interior of* D_1 *and* D_2, *respectively) from* Σ_1 *and* Σ_2, *respectively, and identify the boundaries of* $\Sigma_1 \backslash D_1$ *and* $\Sigma_2 \backslash D_2$ *to obtain a new embedding* Ψ *of a new graph* G *in the surface* $\Sigma_1 \circ \Sigma_2$, *where* $\Sigma_1 \circ \Sigma_2$ *is the disk sum of* Σ_1 *and* Σ_2, *and* G *is obtained from* $G_1 \backslash \{u\}$ *and* $G_2 \backslash \{v\}$ *by identifying* u_i *with* $v_i, i = 1, 2, \ldots, n$. *We use the notation*

$$(G_1, u) \diamondsuit (G_2, v)$$

for the operation on the graph, and

$$\Psi_1(G_1, u) \diamondsuit \Psi_2(G_2, v)$$

for the operation on the embedding. We call these operations the diamond sum of graphs and the diamond sums of embeddings, respectively.

Remark 3.2. If one of $\Psi_1(G_1)$ and $\Psi_2(G_2)$ is a nonorientable embedding, then $\Psi_1(G_1, u) \diamondsuit \Psi_2(G_2, v)$ is also a nonorientable embedding.

THEOREM 3.3. *Suppose* $l \geq k \geq m \geq n$.

(i) *If* $\gamma(K_{k,m,n}) = \lceil \frac{(k-2)(m+n-2)}{4} \rceil$ *and* $(l, k, m+n) \notin S$ *of Table* 2.1, *then* $\gamma(K_{l,m,n}) = \lceil \frac{(l-2)(m+n-2)}{4} \rceil$.

(ii) *If* $\gamma(K_{k,m,n}) = \lceil \frac{(k-2)(m+n-2)}{4} \rceil$ *and*

$$2\left\lceil \frac{(k-2)(m+n-2)}{4} \right\rceil + \left\lceil \frac{(l-k)(m+n-2)}{2} \right\rceil = \left\lceil \frac{(l-2)(m+n-2)}{2} \right\rceil, \tag{3.1}$$

then $\overline{\gamma}(K_{l,m,n}) = \lceil \frac{(l-2)(m+n-2)}{2} \rceil$ *if* $l > m$.

(iii) *If* $\overline{\gamma}(K_{k,m,n}) = \lceil \frac{(k-2)(m+n-2)}{2} \rceil$ *and* l *is odd or one of* k *or* $m+n$ *is even, then* $\overline{\gamma}(K_{l,m,n}) = \lceil \frac{(l-2)(m+n-2)}{2} \rceil$ *if* $l > m$.

Proof. We state the genera for complete bipartite graphs first [Ri1, Ri2]:

$$\gamma(K_{s,t}) = \left\lceil \frac{(s-2)(t-2)}{4} \right\rceil,$$

$$\overline{\gamma}(K_{s,t}) = \left\lceil \frac{(s-2)(t-2)}{2} \right\rceil.$$

To obtain Theorem 3.3(i), we construct the diamond sum (Operation 3.1) of an embedding of $K_{k,m,n}$ in $S_{\lceil \frac{(k-2)(m+n-2)}{4} \rceil}$ and an embedding of $K_{l-k+2,m+n}$ in $S_{\lceil \frac{(l-k)(m+n-2)}{4} \rceil}$, and then apply Lemma 2.2(ii) with $\epsilon_2 = 0$.

Similarly, to obtain Theorem 3.3(iii), we construct the diamond sum of an embedding of $K_{k,m,n}$ in $N_{\lceil \frac{(k-2)(m+n-2)}{2} \rceil}$ and an embedding of $K_{l-k+2,m+n}$ in $N_{\lceil \frac{(l-k)(m+n-2)}{2} \rceil}$, and then apply Lemma 2.2(i) with $\epsilon_1 = 0$.

To obtain Theorem 3.3(ii), we construct the diamond sum of an embedding of $K_{k,m,n}$ in $S_{\lceil \frac{(k-2)(m+n-2)}{4} \rceil}$ and an embedding of $K_{l-k+2,m+n}$ in $N_{\lceil \frac{(l-k)(m+n-2)}{2} \rceil}$. Notice that in the genus counting, each handle in the orientable surface is traded for two

crosscaps in the nonorientable surface. By Remark 3.2 and the assumption (3.1), we have Theorem 3.3(ii). □

If we let $k = m$ and apply Theorem 3.3, we obtain the following corollary.

COROLLARY 3.4. *Suppose* $l \geq m \geq n$.

(i) *If* $\gamma(K_{m,m,n}) = \lceil \frac{(m-2)(m+n-2)}{4} \rceil$, *then* $\gamma(K_{l,m,n}) = \lceil \frac{(l-2)(m+n-2)}{4} \rceil$ *if* $l \geq m \geq n$ *and* $(l, m, m+n) \notin S$ *in Table* 2.1.

(ii) *If* $\gamma(K_{m,m,n}) = \lceil \frac{(m-2)(m+n-2)}{4} \rceil$ *and*

$$2\left\lceil \frac{(m-2)(m+n-2)}{4} \right\rceil + \left\lceil \frac{(l-m)(m+n-2)}{2} \right\rceil = \left\lceil \frac{(l-2)(m+n-2)}{2} \right\rceil,$$

then $\overline{\gamma}(K_{l,m,n}) = \lceil \frac{(l-2)(m+n-2)}{2} \rceil$ *if* $l > m \geq n$.

(iii) *If* $\overline{\gamma}(K_{m,m,n}) = \lceil \frac{(m-2)(m+n-2)}{2} \rceil$, *then* $\overline{\gamma}(K_{l,m,n}) = \lceil \frac{(l-2)(m+n-2)}{2} \rceil$ *if* $l > m \geq n$ *and* n *is odd or one of* l *or* m *is even.*

Remark 3.5. Corollary 3.4(i) reduces the verification of the orientable genera of $K_{l,m,n}$ for three-fourths of all generally nonsymmetric triples (l, m, n) to that of semisymmetric triples (m, m, n). Corollary 3.4(iii) reduces the verification of the nonorientable genera of $K_{l,m,n}$ for seven-eighths of generally nonsymmetric triples (l, m, n) to that of semisymmetric triples (m, m, n). It is hoped that the determination of genera of $K_{m,m,n}$ is easier.

Continuing to apply Theorem 3.3, we are able to determine the genera for the following complete tripartite graphs.

COROLLARY 3.6.

(i)

$$\gamma(K_{l,n,n}) = \left\lceil \frac{(l-2)(n-1)}{2} \right\rceil, \ l \geq n \geq 2, \tag{3.2}$$

$$\overline{\gamma}(K_{l,n,n}) = (l-2)(n-1), \ l > n \geq 2. \tag{3.3}$$

(ii)

$$\gamma(K_{l,m,n}) = \left\lceil \frac{(l-2)(m+n-2)}{4} \right\rceil, \ m+n \text{ even}, \ l \geq 2(m+n-2), \tag{3.4}$$

$$\overline{\gamma}(K_{l,m,n}) = \left\lceil \frac{(l-2)(m+n-2)}{2} \right\rceil, \ m+n \text{ even}, \ l > 2(m+n-2). \tag{3.5}$$

(iii)

$$\gamma(K_{l,n+2,n}) = \left\lceil \frac{n(l-2)}{2} \right\rceil, \ l \geq n+2 \geq 2 \text{ and } n \text{ is even}, \tag{3.6}$$

$$\overline{\gamma}(K_{l,n+2,n}) = n(l-2), \ l \geq n+2 \geq 3. \tag{3.7}$$

(iv)

$$\gamma(K_{l,2n,n}) = \left\lceil \frac{(l-2)(3n-2)}{4} \right\rceil, \ l \geq 2n \geq 2, \tag{3.8}$$

$$\overline{\gamma}(K_{l,2n,n}) = \left\lceil \frac{(l-2)(3n-2)}{2} \right\rceil, \ l > 2n \geq 2. \tag{3.9}$$

(v)

$$\overline{\gamma}(K_{l,n+4,n}) = (l-2)(n+1), \ l \geq n+4 \geq 4 \text{ and } n \text{ is even.} \tag{3.10}$$

In Corollary 3.6, (3.2) extends the orientable genus results of Ringel and Youngs [RY] on $K_{n,n,n}$ and White [Wh2] on $K_{mn,n,n}$ (Craft also obtained the same result as in (3.2) in his dissertation [Cr2]), (3.4) extends a result of Craft [Cr1] on the orientable genus of $K_{l,m,n}$, where $l \geq 2(m+n-2)$ and $m+n$ and l are both even (Craft had a stronger result than in (3.4) in his dissertation [Cr2]), (3.6) and (3.8) extend results of Stahl and White [SW] on orientable genera of $K_{n+2,n+2,n}$ and $K_{2n,2n,n}$, respectively, and (3.7) and (3.10) extend results of Stahl and White [SW] on nonorientable genera of $K_{n+2,n+2,n}$ and $K_{n+4,n+4,n}$, respectively.

Proof of Corollary 3.6. Since $\gamma(K_{n,n,n}) = \frac{1}{2}(n-1)(n-2)$ [RY, Wh2], by Theorem 3.3(i) (with $k = m = n$) we obtain (3.2), and by Theorem 3.3(ii) we obtain (3.3).

Since $\gamma(K_{2m+2n-4,m,n}) = \frac{1}{2}(m+n-3)(m+n-2)$ [Cr1], by Theorem 3.3(i) we obtain (3.4) and by Theorem 3.3(ii) we obtain (3.5).

By [SW], for $n \geq 4$ and even, $\gamma(K_{n+2,n+2,n}) = \lceil \frac{n^2}{2} \rceil$. Let $k = m = n+2$, where n is even. Then $(k,m,n) = (n+2, n+2, n) \notin S$ of Table 2.1. Applying Theorem 3.3(i) with $k = m = n+2$, we obtain (3.6).

By [SW], $\overline{\gamma}(K_{n+2,n+2,n}) = n^2$. Applying Theorem 3.3(iii) with $k = m = n+2$ (hence $m+n$ is even), we obtain (3.7).

Also by [SW], $\gamma(K_{2n,2n,n}) = \frac{(n-1)(3n-2)}{2}$. Apply Theorem 3.3(i) with $k = m = 2n$. Since $(l,m,n) = (l, 2n, 3n) \notin S$ of Table 2.1, we obtain (3.8). Apply Theorem 3.3(ii) with $k = m = 2n$. Since $2\lceil \frac{(2n-2)(2n+n-2)}{4} \rceil + \lceil \frac{(l-2n)(2n+n-2)}{2} \rceil = (n-1)(3n-2) + \lceil \frac{(l-2n)(3n-2)}{2} \rceil = \lceil \frac{(l-2)(3n-2)}{2} \rceil$, we have (3.9).

Since $\overline{\gamma}(K_{n+4,n+4,n}) = (n+1)(n+2)$, applying Theorem 3.3(iii) with $k = m = n+4$, we have (3.10). This finishes the proof of Corollary 3.6. □

We also have the following bounds for $\gamma(K_{l,m,n})$ (respectively, $\overline{\gamma}(K_{l,m,n})$).

THEOREM 3.7. *Suppose* $l \geq k \geq m \geq n$.

(i) *If* $\gamma(K_{k,m,n}) = \lceil \frac{(k-2)(m+n-2)}{4} \rceil$, *then*

$$\left\lceil \frac{(l-2)(m+n-2)}{4} \right\rceil \leq \gamma(K_{l,m,n}) \leq \left\lceil \frac{(l-2)(m+n-2)}{4} \right\rceil + 1. \tag{3.11}$$

(ii) *If* $\overline{\gamma}(K_{k,m,n}) = \lceil \frac{(k-2)(m+n-2)}{2} \rceil$, *then*

$$\left\lceil \frac{(l-2)(m+n-2)}{2} \right\rceil \leq \overline{\gamma}(K_{l,m,n}) \leq \left\lceil \frac{(l-2)(m+n-2)}{2} \right\rceil + 1. \tag{3.12}$$

Proof. The first inequalities of (3.11) and (3.12) are just (1.1) and (1.2), respectively.

To obtain the second inequality of (3.11), construct the diamond sum of an embedding of $K_{k,m,n}$ in $S_{\lceil \frac{(k-2)(m+n-2)}{4} \rceil}$ and an embedding of $K_{l-k+2,m+n}$ in $S_{\lceil \frac{(l-k)(m+n-2)}{4} \rceil}$, and then apply Lemma 2.2(ii).

To obtain the second inequality of (3.12), construct the diamond sum of an embedding of $K_{k,m,n}$ in $N_{\lceil \frac{(k-2)(m+n-2)}{2} \rceil}$ and an embedding of $K_{l-k+2,m+n}$ in $N_{\lceil \frac{(l-k)(m+n-2)}{2} \rceil}$, and then apply Lemma 2.2(i). □

We make the following remark to conclude our paper. As shown in section 3, the simple yet useful diamond sum of a complete tripartite graph with a complete bipartite graph (Operation 3.1) helps us obtain a large percentage of genus embeddings (orientable and nonorientable) of complete tripartite graphs (Theorem 3.3). It also helps us reduce the determination of genera (orientable and nonorientable) of at least 75 percent of $K_{l,m,n}$'s to that of $K_{m,m,n}$.

On the other hand, many conjectured genus embeddings of complete tripartite graphs cannot be obtained by this operation. One reason for this is that in some cases the sum of two ceiling functions may result in a number which is one more than we need, as indicated in Lemma 2.2. One might be tempted to try to modify Operation 3.1 by choosing the deleted vertex u from some part other than the largest part of the 3-partition of the vertices of $K_{l,m,n}$. This will not work. The result will, of course, be an embedding, but not a genus embedding; it is only possible to construct genus embeddings by choosing u from the largest part. The reason for this is that for an embedding to be a genus embedding, the number of faces must be as large as possible. In particular, the number of triangular faces must be as large as possible. In fact, the known lower bound on the genus of $K_{l,m,n}$, $l \geq m \geq n$, is obtained by assuming that each edge between the part containing m vertices and the part containing n vertices corresponds to a side of two different triangular faces. Such triangular faces must be preexisting in the embedding of the complete tripartite graph, for the faces from the complete bipartite graph have a size of at least four and therefore the diamond sum will not add any new triangular faces.

REFERENCES

[Bo] A. Bouchet, *Orientable and nonorientable genus of the complete bipartite graph*, J. Combin. Theory Ser. B, 24 (1978), pp. 24–33.

[Cr1] D. L. Craft, *On the genus of joins and compositions of graphs*, Discrete Math., 178 (1998), pp. 25–50.

[Cr2] D. L. Craft, *Surgical Techniques for Constructing Minimal Orientable Imbeddings of Joins and Compositions of Graphs*, Ph.D. dissertation, Western Michigan University, Kalamazoo, MI, 1991.

[ESZ1] M. N. Ellingham, C. Stephens, and X. Zha, *Counterexamples to the nonorientable genus conjecture for complete tripartite graphs*, European J. Combin., to appear.

[ESZ2] M. N. Ellingham, C. Stephens, and X. Zha, *The nonorientable genus of complete tripartite graphs*, J. Combin. Theory Ser. B, submitted.

[GT] J. L. Gross and T. W. Tucker, *Topological Graph Theory*, Wiley-Interscience, New York, 1987.

[HMR] J. P. Huneke, D. McQuillan, and R. B. Richter, *Embeddings of* 8*-Vertex Graphs and Orientably Simple Graphs*, preprint, 1996.

[MT] B. Mohar and C. Thomassen, *Graphs on Surfaces*, The Johns Hopkins University Press, Baltimore, MD, 2001.

[Ri1] G. Ringel, *Das Geschlecht des vollständigen paaren Graphen*, Abh. Math. Sem. Univ. Hamburg, 28 (1965), pp. 139–150.

[Ri2] G. Ringel, *Der vollständige paare Graph auf nichtorientierbaren Flächen*, J. Reine Angew. Math., 220 (1965), pp. 88–93.

[RY] G. Ringel and J. W. T. Youngs, *Das Geschlecht des symmetrischen vollständigen dreifarbbaren Graphen*, Comment. Math. Helv., 45 (1970), pp. 152–158.

[SB] S. Stahl and L. Beineke, *Bolcks and the nonorientable genus of graphs*, J. Graph Theory, 1 (1977), pp. 75–78.

[SW] S. Stahl and A. T. White, *Genus embeddings for some complete tripartite graphs*, Discrete Math., 14 (1976), pp. 279–296.

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

[Wh1] A. T. White, *The Genus of Cartesian Products of Graphs*, Ph.D. thesis, Michigan State University, Kalamazoo, MI, 1969.

[Wh2] A. T. White, *The genus of the complete tripartite graph* $K_{mn,n,n}$, J. Graph Theory, 7 (1969), pp. 283–285.

SIAM J. DISCRETE MATH.
Vol. 18, No. 3, pp. 488–500

FINDING LARGE INDEPENDENT SETS IN GRAPHS AND HYPERGRAPHS*

HADAS SHACHNAI† AND ARAVIND SRINIVASAN‡

Abstract. A basic problem in graphs and hypergraphs is that of finding a *large* independent set—one of guaranteed size. Understanding the parallel complexity of this and related independent set problems on hypergraphs is a fundamental open issue in parallel computation. Caro and Tuza [*J. Graph Theory*, 15 (1991), pp. 99–107] have shown a certain lower bound $\alpha_k(H)$ on the size of a maximum independent set in a given k-uniform hypergraph H and have also presented an efficient sequential algorithm to find an independent set of size $\alpha_k(H)$. They also show that $\alpha_k(H)$ is the size of the maximum independent set for various hypergraph families. Here, we show that an *RNC* algorithm due to Beame and Luby [in *Proceedings of the ACM–SIAM Symposium on Discrete Algorithms*, 1990, pp. 212–218] finds an independent set of expected size $\alpha_k(H)$ and also derandomizes it for certain special cases. (An intriguing conjecture of Beame and Luby implies that understanding this algorithm better may yield an *RNC* algorithm to find a maximal independent set in hypergraphs, which is among the outstanding open questions in parallel computation.) We also present lower bounds on independent set size for *nonuniform* hypergraphs using this algorithm. For graphs, we get an *NC* algorithm to find independent sets of size essentially that guaranteed by the general (degree-sequence based) version of Turán's theorem.

Key words. independent sets, parallel algorithms, randomized algorithms

AMS subject classifications. 05C65, 60C05, 68R10, 68W10, 68W20

DOI. 10.1137/S0895480102419731

1. Introduction. Finding large/maximal independent sets (ISs) in (hyper)graphs, defined formally below, is a fundamental problem in parallel combinatorial optimization. An outstanding open question in parallel computation is whether a maximal IS in a given hypergraph can be found in $(R)NC$ [13]. The work of Karp and Wigderson [15] on finding maximal independent sets (MISs) in graphs in NC was a breakthrough that inspired several graph-theoretic NC algorithms and also led to a rich theory of derandomization. The corresponding problems on hypergraphs have applications, e.g., to feasible communication in channelized cellular telephone systems [19] but seem much harder than in the case of graphs. In this work we consider an RNC algorithm for finding large ISs in hypergraphs due to Beame and Luby [3][1] and derandomize it for various special cases. We show that this algorithm finds IS that have been shown to exist for uniform hypergraphs via a *sequential* algorithm in [5]; we also use the algorithm to give lower bounds on IS size for certain families of nonuniform hypergraphs. For graphs, our derandomization yields the first NC algorithm to

*Received by the editors December 15, 2002; accepted for publication (in revised form) March 8, 2004; published electronically December 30, 2004. Preliminary versions of parts of this work appeared in: (i) A. Srinivasan, *New approaches to covering and packing problems*, in Proceedings of the ACM–SIAM Symposium on Discrete Algorithms, 2001, pp. 567–576; and (ii) H. Shachnai and A. Srinivasan, *Finding large independent sets of hypergraphs in parallel*, in Proceedings of the ACM Symposium on Parallel Algorithms and Architectures, 2001, pp. 163–168.

http://www.siam.org/journals/sidma/18-3/41973.html

†Department of Computer Science, The Technion, Haifa 32000, Israel (hadas@cs.technion.ac.il). Part of this work was done while the author was on leave at Bell Laboratories, Lucent Technologies, 600 Mountain Ave., Murray Hill, NJ 07974.

‡Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742 (srin@cs.umd.edu). Most of this work was done while the author was at Bell Laboratories and Lucent Technologies; part of this work was supported by the National Science Foundation under grant 0208005.

[1]We describe the algorithm in section 2.1.

find ISs of size essentially that guaranteed by the degree-sequence version of Turán's theorem [28].

Recall that a *hypergraph* $H = (V, E)$ consists of a vertex set V and a collection E of subsets of V; each element of E is called a (hyper)edge. We will consider only finite hypergraphs here and will throughout denote the number of vertices and edges in a given hypergraph by n and m, respectively. An IS in H is a subset S of V that does not contain any edge. S is an MIS if S is an IS and if no proper superset of S is an IS. It is easy to find an MIS sequentially, but efficient parallel algorithms appear much harder. There has been much work on finding MISs in parallel in graphs (see, e.g., [8, 15, 17, 21]) and in hypergraphs (e.g., [1, 3, 6, 16, 14, 12]); as mentioned above, it is a major open question in parallel computation whether these can be found in RNC on hypergraphs. Since an MIS can be much smaller than a maximum IS (as in the case of a star graph), it is also of much interest to find ISs that have a guaranteed size. What is known in this context? Recall that given a hypergraph $H = (V, E)$, the *degree* of a vertex is the number of edges that it lies in; H is called k-uniform if all the edges have exactly k elements. Also, for any integer $\ell \geq 0$ and real r,

$$\binom{r}{\ell} \equiv \frac{r(r-1)\cdots(r-l+1)}{\ell!}. \tag{1}$$

Caro and Tuza showed in [5] that for any $k \geq 2$, any k-uniform hypergraph H contains an IS of size at least

$$\alpha_k(H) = \sum_{v \in V} \frac{1}{\binom{d(v) + 1/(k-1)}{d(v)}}; \tag{2}$$

here and from now on, $d(v)$ denotes the degree of $v \in V$. They showed that this bound is tight for a large class of hypergraphs and also gave a *sequential* algorithm that finds an IS of size $\alpha_k(H)$ in $O(km + n)$ steps. Note that when $k = 2$, the bound in (2) reduces to

$$\alpha_2(G) = \sum_{v \in V} \frac{1}{d(v) + 1}, \tag{3}$$

which is the classical Turán bound for graphs [28]. To give the reader a better feel for (2), we point out that

$$\binom{d(v) + 1/(k-1)}{d(v)} = \Theta((d(v))^{1/(k-1)}). \tag{4}$$

This can be shown as follows. We set in (1) $r = d(v) + 1/(k-1)$ and $\ell = d(v)$ to get

$$\binom{d(v) + 1/(k-1)}{d(v)} = \prod_{i=1}^{d(v)} \left(1 + \frac{1}{i(k-1)}\right). \tag{5}$$

By a standard expansion of $\ln(1 + x)$ for $x \geq 0$ (see, e.g., [24]),

$$\exp(x - x^2/2) \leq 1 + x \leq \exp(x). \tag{6}$$

(Here and in what follows, $\exp(x)$ denotes e^x.) Applying (6) to (5), we have (4).

Remark 1. To handle the case where $d(v) = 0$, the right-hand side of (4) should actually be $\Theta((d(v)+1)^{1/(k-1)})$. However, since vertices of degree zero are irrelevant and can always be added to our IS, we will always assume that vertex-degrees are nonzero.

Our main results are as follows; these are spelled out in more detail in section 1.2. Beame and Luby have presented a simple and elegant *RNC* algorithm for finding an IS in a given hypergraph H [3]; intriguingly, they also conjecture that repeatedly finding and removing such ISs, followed by obvious updates to H, finds an MIS in *RNC*. (The work of [3] also presents another, more involved, candidate *RNC* algorithm for finding an MIS in a given hypergraph. By the analysis of [16], this indeed yields an *RNC* algorithm for MIS in hypergraphs with constant-sized edges.) Thus, it is of interest to understand this algorithm better. We analyze the performance of this algorithm and show that it finds an IS of expected size $\alpha_k(H)$ for k-uniform hypergraphs, for any $k \geq 2$. We also analyze its performance for certain families of nonuniform hypergraphs. Finally, we derandomize the algorithm for certain special cases; in particular, in the case of graphs G, this yields an *NC* algorithm to find ISs of size essentially $\alpha_2(G)$.

1.1. Related work. The following parallel algorithms are known in the case of graphs. Spencer [26] gave an *RNC* algorithm that yields an IS of expected size $\alpha_2(G)$ in graphs G. Goldberg and Spencer [9] presented an *NC* algorithm that finds an IS of size at least $\lceil n^2/(2m+n) \rceil$ in any graph G, where n and m denote the number of vertices and edges in G, respectively. This bound equals $\alpha_2(G)$ when G is regular; in all other cases, $\alpha_2(G)$ is larger. (In fact, it is not hard to construct graph families for which $\alpha_2(G) = \Theta(n)$ and $n^2/(2m+n)$ is just $\Theta(1)$; see item (i) in section 1.2.) Alon, Babai, and Itai [1] studied the MIS problem on hypergraphs: they gave an NC algorithm that finds an IS of size $c(n^k/m)^{1/(k-1)}$, $0 < c < 1$, in k-uniform hypergraphs with $k \geq 2$ being any constant. The work of Karger and Koller [12] generalized this to arbitrary k.

1.2. Main results. The *RNC* algorithm of [3] that we shall analyze will be defined in section 2.1; we will refer to it as algorithm $\mathcal{A}$. We show that $\mathcal{A}$ finds an IS of expected size $\alpha_k(H)$ in k-uniform hypergraphs and also present related results and extensions. Our main contributions are as follows.

(i) The expected size of the IS produced by $\mathcal{A}$ is larger than the size of the IS found by other parallel algorithms for this problem. For instance, $\alpha_k(H)$ is always at least as large as the bound $c(n^k/m)^{1/(k-1)}$ of [1, 12]. Furthermore, one can construct families of k-uniform hypergraphs for which $\alpha_k(H)$ is $\Theta(n)$ while $(n^k/m)^{1/(k-1)}$ is $\Theta(1)$. For instance, in the case of constant k, consider hypergraphs H where:
   - the vertex set is partitioned into two equal-sized sets V_1 and V_2, and
   - H contains as (hyper)edges all k-sized subsets of V_1 and $n/(2k)$ many pairwise-disjoint k-sized subsets of V_2.

   It is easy to check that for such families of hypergraphs, $\alpha_k(H) = \Theta(n)$ and $(n^k/m)^{1/(k-1)} = \Theta(1)$.

(ii) We show how our analysis extends to certain families of nonuniform hypergraphs; we are not aware of any such bounds relating to large IS in this nonuniform case.

(iii) Regarding derandomized versions, we have the following. Let C be a suitably large positive constant. Given a graph or hypergraph, let $\epsilon = (n+m)^{-C}$. For graphs G, we show how to construct an IS of size at least $(1-\epsilon)\alpha_2(G)$, in

NC. For hypergraphs where all the vertex-degrees are at most $O(\lg(n+m))$, we present NC algorithms to construct ISs of size at least $(1-\epsilon)$ times the size guaranteed by our above-mentioned RNC algorithms.

(iv) All of our results extend without any change in the processor or time complexity to their weighted analogues. In the weighted analogues, we are given a nonnegative weight for each vertex and wish to find an IS of large total weight. The only way that we are aware of to extend arbitrary IS algorithms to their weighted analogues is by a suitable (polynomial) blowup in the size of the hypergraph, leading to a loss in efficiency.

One key facilitator of our results is a simple way of generating random permutations that provides sufficient stochastic independence to conduct our analysis (see Lemma 2.1).

2. The *RNC* algorithm.

2.1. Algorithms and tools. The algorithm $\mathcal{A}$ of [3] that we shall analyze is as follows.[2] Randomly permute the vertices; add a vertex $v \in V$ to the IS iff there is no edge $e \in E$ such that $v \in e$ and v is *last* among the vertices of e in the random order. It is easy to verify that we produce a valid IS in this fashion. Spencer independently considered this algorithm in the case of graphs and showed that the expected size of the IS found is $\alpha_2(G)$ [26]. (Actually, the algorithm of [3] adds v to the IS iff there is no edge $e \in E$ such that $v \in e$ and v is *first* among the vertices of e in the random order. It is easy to verify that this is equivalent to our description of $\mathcal{A}$ a few lines above; our description is given so that we get a direct generalization of the algorithm of [26].)

Our main analysis tool will be the Fortuin–Kasteleyn–Ginibre (FKG) inequality [7]; for our purposes, we now recall a special case of the inequality. The reader is referred to [2] for more about the inequality. Given a vector $\vec{Y} = (Y_1, Y_2, \ldots, Y_\ell)$ of *independent* random variables $Y_i \in \{0,1\}$ and an event F that is completely determined by the Y_i's, call F *increasing* iff the following holds: for any $\vec{a}$ such that F holds when $\vec{Y} = \vec{a}$, F also holds when $\vec{Y} = \vec{b}$ for any $\vec{b}$ that coordinatewise dominates $\vec{a}$ (i.e., $a_i \leq b_i$ for all i). Then, for any collection of *increasing* events $F_1, F_2, \ldots, F_t$, the FKG inequality shows that

$$\text{Prob}\left(\bigwedge_{i=1}^{t} F_i\right) \geq \prod_{i=1}^{t} \text{Prob}(F_i). \tag{7}$$

A specific way of implementing algorithm $\mathcal{A}$ is given in Figure 1. As will be seen in the proof of Lemma 2.1, the method of generating random permutations that we adopt provides sufficient independence to employ tools such as the FKG inequality.

It is readily seen that $\mathcal{A}$ can be implemented in RNC. For each edge, we first choose the vertex u in it of highest X_u value; removing duplicates from this multiset of chosen vertices (e.g., through sorting) yields the set of vertices that will *not* lie in our IS. This is easily done on a CREW PRAM in $O(\lg(m+n))$ steps, using $(\sum_{e_i \in E} |e_i|) \leq mn$ processors. Also, since this algorithm is simple and just uses some basic primitives, it should be easy to implement in other parallel/distributed settings.

The following notation will be used frequently in the rest of this paper: given a hypergraph H, denote by B_v the event that vertex v is in the final IS produced by

[2] The algorithm is called the *permutation algorithm* in [3].

Algorithm $\mathcal{A}$:
Independently for all $v \in V$ do:
  Sample $X_v \in [0,1)$ using the uniform distribution on $[0,1)$.
Define a permutation π of the vertices in which $\pi(v) < \pi(u)$ iff $X_v < X_u$.
$I := \emptyset$;
for all $v \in V$ do
  {
    j_v = true.
    For all $e \in E$ such that $v \in e$
      if $\pi(v) = \max_{u \in e} \pi(u)$, then j_v = false.
    If j_v, then $I := I \cup \{v\}$;
  }
return(I).

FIG. 1. *The algorithm $\mathcal{A}$.*

our algorithm.

Remark 2. In case generating random reals poses a problem, we can take up the following standard alternative for producing random permutations; this will also be useful in section 3. Our modified algorithm $\mathcal{A}'$ is as follows. Let $K > 2$ be an arbitrary constant. Instead of choosing the X_v from $[0,1)$, $\mathcal{A}'$ will independently select each X_v uniformly at random from the set $\{0, 1, \ldots, n^K - 1\}$; for any vertex v, we now add v to the IS iff there is no edge e such that $v \in e$ and such that $X_v \geq X_u$ for all $u \in e$. It is easy to verify that we will still produce an IS in this way. Furthermore, let $\mathcal{E}$ be the event that all the X_v are distinct. First, note that for any distinct v and w, $\text{Prob}(X_v = X_w) = 1/n^K$; so, by a union bound,

$$\text{Prob}(\overline{\mathcal{E}}) \leq n^2 \cdot 1/n^K = o(1).$$

Next, if we condition on $\mathcal{E}$, the permutation π of $\mathcal{A}'$ is distributed as a random permutation. In particular, let $\text{Prob}(B_v)$ and $\text{Prob}'(B_v)$ denote the probability of event B_v occurring, when we use algorithms $\mathcal{A}$ and $\mathcal{A}'$, respectively. Then,

$$\text{Prob}'(B_v) \geq \text{Prob}(\mathcal{E}) \cdot \text{Prob}'(B_v \mid \mathcal{E}) = \text{Prob}(\mathcal{E}) \cdot \text{Prob}(B_v) \geq (1 - n^{2-K}) \cdot \text{Prob}(B_v). \quad (8)$$

Thus, if $K > 2$, the expected size of the IS constructed by $\mathcal{A}'$ is essentially as large as that constructed by $\mathcal{A}$.

2.2. Analysis of the performance of $\mathcal{A}$. We now analyze the performance of algorithm $\mathcal{A}$. Our basic tool will be Lemma 2.1. Before presenting the lemma, we recall that a *linear hypergraph* is one in which every pair of distinct edges shares at most one vertex. Linear hypergraphs have been studied in the context of parallel construction of MISs, and NC algorithms are known for the MIS problem on linear hypergraphs [18, 27]. Hence, we also see how well algorithm $\mathcal{A}$ does on linear hypergraphs; Lemma 2.1 also helps provide an exact bound on our algorithm's performance in this case.

LEMMA 2.1. *Suppose a vertex v lies in edges $e_1, e_2, \ldots, e_t$, whose respective cardinalities are $k_1, k_2, \ldots, k_t$. Then,*

$$Prob(B_v) \geq \int_0^1 \left[\prod_{i=1}^{t}(1 - x^{k_i - 1})\right] \, dx.$$

Furthermore, this inequality becomes an equality in the case of linear hypergraphs.

Proof. Recall the random variables X_u from Figure 1. The main idea behind our proof is that the computations become tractable once we condition on the value of X_v. As we will see, the fact that the X_u's are independent will help us much; this way of introducing independence into a choice of permutations helps us use tools such as the FKG inequality. Let $x \in [0, 1)$ be arbitrary, and define, for all $u \neq v$, the random variable $Y_u = 1$ if $X_u > x$, and $Y_u = 0$ otherwise. For each edge e, define the random variable $C(e)$ to be 1 if $\max_{u:u\in e} X_u > x$, and $C(e)$ to be 0 otherwise. Then

$$\text{Prob}(B_v | X_v = x) = \text{Prob}\left(\left[\bigwedge_{e:v\in e} (C(e) = 1)\right] | X_v = x\right).$$

Now, even conditional on $X_v = x$, the random variables Y_u are independent with $\text{Prob}(Y_u = 1) = 1 - x$. Also, conditional on $X_v = x$, each $C(e)$ is determined by the values of the Y_u and is increasing as a function of the Y_u. Thus, by the FKG inequality,

$$\text{Prob}(B_v | X_v = x) \geq \prod_{i=1}^{t} \text{Prob}(C(e_i) = 1 | X_v = x) = \prod_{i=1}^{t}(1 - x^{k_i - 1}). \tag{9}$$

The first part of the lemma now follows from the fact that

$$\text{Prob}(B_v) = \int_0^1 \text{Prob}(B_v | X_v = x) \, dx.$$

It is also easy to check that the inequality in (9) becomes an equality in the case of linear hypergraphs. Hence, the inequality of this lemma is in fact an equality for linear hypergraphs. □

Applying the linearity of expectation, we get the following theorem on the expected quality of the IS produced by $\mathcal{A}$ for an arbitrary hypergraph.

THEOREM 2.2. *Suppose we are given an arbitrary hypergraph $H = (V, E)$ with a weight $w_v \geq 0$ for each vertex v. Suppose each vertex v lies in $d(v)$ edges, whose cardinalities are $k_{v,1}, k_{v,2}, \ldots, k_{v,d(v)}$. Then, the expected weight of the IS produced by $\mathcal{A}$ is at least*

$$\sum_v \left(w_v \cdot \int_0^1 \left[\prod_{i=1}^{d(v)}(1 - x^{k_{v,i} - 1})\right] \, dx \right);$$

in the case of linear hypergraphs, this lower bound is an exact bound on the expected weight.

The next theorem considers the performance of $\mathcal{A}$ for unweighted uniform hypergraphs and shows that the expected size of the IS produced is at least as large as $\alpha_k(H)$.

THEOREM 2.3. *For any $k \geq 2$ and any k-uniform hypergraph H, $\mathcal{A}$ finds an IS of expected size at least $\alpha_k(H)$.*

Proof. We will use the following identity from [10], which holds for any nonnegative integer d and any real x that does not lie in the set $\{-d, -d+1, \ldots, 0\}$:

$$\sum_{l=0}^{d} \binom{d}{l} \frac{(-1)^l}{x+l} = \frac{1}{x\binom{d+x}{d}}. \tag{10}$$

Specialized to k-uniform hypergraphs, Lemma 2.1 shows that

$$\begin{aligned}
\text{Prob}(B_v) &\geq \int_0^1 (1-x^{k-1})^{d(v)}\, dx \\
&= \sum_{l=0}^{d(v)} (-1)^l \binom{d(v)}{l} \int_0^1 x^{(k-1)l}\, dx \\
&= \sum_{l=0}^{d(v)} (-1)^l \binom{d(v)}{l} \frac{1}{1+(k-1)l} \\
&= \binom{d(v)+1/(k-1)}{d(v)}^{-1}
\end{aligned}$$

by (10).

Summing over all the vertices and applying the linearity of expectation completes the proof. □

Remark 3. If an edge is just a singleton $\{v\}$, then vertex v cannot lie in any IS. Hence, whenever we calculate $\text{Prob}(B_v)$ below, we assume without loss of generality that all edges that v lies in have size at least 2.

Any algorithm that works for k-uniform hypergraphs and whose output solution is a nonincreasing function of each vertex-degree (as is the function $\alpha_k(H)$) can be immediately extended to give the same guarantee for hypergraphs in which all edges have size *at least* k. We simply replace each edge by an arbitrary subset of it of size k to achieve this. Thus, our results such as Theorem 2.3 also hold for hypergraphs with at least k vertices in each edge. However, in various families of nonuniform hypergraphs we can do better than this simple approach, as we demonstrate in Theorem 2.4.

We need some notation. Suppose each vertex v lies in $D(j,v)$ edges of cardinality $k(j,v)$ for $j = 1, 2, \ldots, a(v)$. (In other words, vertex v lies in edges of $a(v)$ different sizes: $k(j,v)$ for $j = 1, 2, \ldots, a(v)$. If H is a uniform hypergraph, then $a(v) \equiv 1$.) As mentioned in Remark 3, we assume that $k(j,v) > 1$ for all j, v. Define

$$f(v) = \min_{j=1,2,\ldots,a(v)} [D(j,v)]^{-1/(k(j,v)-1)},$$

and let $b(v) = \min_j (k(j,v)-1)$. Then, Theorem 2.4 shows that given a weight $w(v)$ for each vertex, $\mathcal{A}$ produces an IS of expected total weight at least $\Omega(\sum_v (w(v)/a(v)^{1/b(v)}) \cdot f(v))$. We prove this by lower-bounding the quantity guaranteed by Theorem 2.2.

Our proof of Theorem 2.4 shows that the quantity guaranteed by Theorem 2.2 is $O(\sum_v w(v) f(v))$; so our "closed-form" bound $\Omega(\sum_v (w(v)/a(v)^{1/b(v)}) \cdot f(v))$ approximates the guarantee of Theorem 2.2 well when $a(v)$ is "small" or $b(v)$ is "large."

THEOREM 2.4. *In a hypergraph $H = (V, E)$, let the notation $a(v)$, $b(v)$, and $f(v)$ be as above. Then, given weights $w(v) \geq 0$ for the vertices, the expected weight of the IS produced by $\mathcal{A}$ is at least $\Omega(\sum_v (w(v)/a(v)^{1/b(v)}) \cdot f(v))$.*

Proof. Fix a vertex v. For notational simplicity, let $a = a(v)$, $b = b(v)$, $k(j) = k(j, v)$, $D(j) = D(j, v)$, and $f = f(v)$. By Theorem 2.2, it suffices to show that

$$\beta \doteq \int_0^1 \left[\prod_{j=1}^{a} (1 - x^{k(j)-1})^{D(j)} \right] dx \geq \Omega(f/a^{1/b}). \tag{11}$$

Let $s = s(v) = \operatorname{argmin}_{j=1,2,\ldots,a} [D(j)]^{-1/(k(j)-1)}$. The bound (4) and the proof of Theorem 2.3 help show that

$$\beta \leq \int_0^1 (1 - x^{k(s)-1})^{D(s)} \, dx = \Theta(f);$$

thus, (11) is tight to within a constant factor if, e.g., $a(v)$ is bounded by an absolute constant (in particular, (11) is a tight bound for hypergraphs of constant maximum degree), or if $b(v) = \Omega(\log n)$.

We now prove (11). Let $t = f/(2 \cdot a^{1/b})$, and note that $t \in [0, 1/2]$. Define $\gamma \doteq \prod_{j=1}^{a} (1 - t^{k(j)-1})^{D(j)}$. Note that $1 - z \geq \exp(-2z)$ for $0 \leq z \leq 1/2$. Thus we have

$$\begin{aligned}
\gamma &\geq \exp\left(-O\left(\sum_{j=1}^{a} D(j) t^{k(j)-1} \right) \right) \\
&\geq \exp\left(-O\left(\sum_{j=1}^{a} D(j) \frac{f^{k(j)-1}}{a} \right) \right) \quad \text{(by the definition of } b\text{)} \\
&= \exp\left(-O\left(\frac{1}{a} \cdot \sum_{j=1}^{a} D(j) f^{k(j)-1} \right) \right) \\
&\geq \exp\left(-O\left(\frac{1}{a} \cdot \sum_{j=1}^{a} 1 \right) \right) \quad \text{(by the definition of } f\text{)} \\
&= \Omega(1).
\end{aligned}$$

Thus,

$$\beta \geq \int_0^t \left[\prod_{j=1}^{a} (1 - x^{k(j)-1})^{D(j)} \right] dx \geq \int_0^t \gamma \, dx = t\gamma \geq \Omega(t),$$

establishing (11). ∎

3. NC algorithms. Recall algorithm $\mathcal{A}'$ of section 2.1, which is presented soon after defining algorithm $\mathcal{A}$. Note, in particular, from (8) that the expected size of the IS constructed by $\mathcal{A}'$ is essentially as large as that constructed by $\mathcal{A}$. We now take up the task of derandomizing algorithm $\mathcal{A}'$. We will employ an "automata-fooling" approach of [22, 23, 12, 20]; see [25] for a different perspective on this approach through Logspace-computable statistical tests. Specializing this approach for our purposes, we

have the following. Suppose we have h finite-state automata $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_h$ with respective state-sets $S_1, S_2, \ldots, S_h$, such that $S_i \cap S_j = \emptyset$ if $i \neq j$. Each S_i is partitioned into $n+1$ layers, numbered $0, 1, \ldots, n$. Layer 0 has a unique state s_i, which is also the start-state of $\mathcal{A}_i$. All transitions are only from one layer to the layer numbered one higher; there are no transitions from layer n. Outgoing arcs from a state are numbered by an integer in the range $\{0, 1, \ldots, 2^d - 1\}$ for some integer d. Given a word $\gamma_1\gamma_2\ldots\gamma_n$, where each γ_i is a d-bit string, each $\mathcal{A}_i$ moves from its start-state s_i to some state in layer n of S_i, in the obvious way. (In case some γ_i does not correspond to a valid transition, $\mathcal{A}_i$ moves to a unique "reject" state at which it remains from then on.) Now suppose we are given a parameter $\epsilon \in (0, 1)$. Let $R = r + 2^d + 1/\epsilon$, where $r = \sum_i |S_i|$. Then, there is an explicit deterministic parallel algorithm to construct a certain multiset T whose elements are from the set $\{0, 1, \ldots, 2^d - 1\}^n$ such that the total cardinality of T is at most poly(R) [22, 23, 12, 20]; this algorithm uses poly(R) processors and runs in polylog(R) time. The key property of T is as follows. Given a state t in layer n of S_i, let $p_1(i,t)$ be the probability of reaching state t if we choose $\gamma_1\gamma_2\ldots\gamma_n$ uniformly at random from $\{0, 1, \ldots, 2^d - 1\}^n$; let $p_2(i,t)$ be this probability if $\gamma_1\gamma_2\ldots\gamma_n$ is chosen uniformly at random from T. Then, T has the useful property that

$$\forall (i,t),\ |p_1(i,t) - p_2(i,t)| \leq \epsilon. \tag{12}$$

We will show how to adapt the above framework to derandomize some of our results. In our applications, automaton $\mathcal{A}_i$ will basically correspond to running $\mathcal{A}'$ and deciding if a vertex i is put in the IS or not by $\mathcal{A}'$; in particular, we will have $h = n$. Fix an automaton $\mathcal{A}_i$. The transitions from layer $j-1$ to j in it will correspond to the various choices of the value X_j. (Since X_j is chosen at random from the set $\{0, 1, \ldots, n^K - 1\}$, the value d will be $\Theta(\lg n)$.) Now suppose we can construct a layered automaton $\mathcal{A}_i$ of this type with poly$(n + m)$ states, such that the final state entered by $\mathcal{A}_i$ determines whether $\mathcal{A}'$ puts the vertex i in the IS or not. Then we can choose ϵ to be of the form $(n + m)^{-C}$, where C is a suitably large positive constant; by (12), we will have a polynomial-sized sample space T constructible in NC, such that sampling from T produces an IS of sufficiently large expected size. By trying out all points in T, we will achieve the desired NC-derandomization.

Thus, the question boils down to the following: for what families of hypergraphs do *polynomial-sized* layered automata of the above type exist? We show that this property holds for two families of hypergraphs:

(H1) graphs, and

(H2) hypergraphs where, for some constant c, all the vertex-degrees are bounded by $c \cdot \lg(n + m)$.

Fix a vertex i. By renumbering the vertices (just for the sake of this automaton), we will assume that $i = 1$. The automaton $\mathcal{A}_i$ for both these families of hypergraphs will have the following structure. Layer 0 has the unique start-state s_i. Layer 1 has n^K states numbered $s_{1,0}, s_{1,1}, \ldots, s_{1,n^K-1}$; $\mathcal{A}_i$ (i.e., $\mathcal{A}_1$ after our renumbering) will transit from s_i to state $s_{1,\ell}$ in layer 1 iff $X_i = \ell$; this way, the automaton can be made to remember the value of X_i. Layer n has two states "accept$_{i,n}$" and "reject$_{i,n}$": for a given choice of all the random variables X_v, $\mathcal{A}_i$ will enter these states if $\mathcal{A}'$, respectively, puts i or does not put i in the IS. Consider any other layer j of $\mathcal{A}_i$ (i.e., $2 \leq j \leq n - 1$). This layer has a state "reject$_{i,j}$" as well as a polynomial number of other states. All transitions from reject$_{i,j}$ are to the state reject$_{i,j+1}$ in the next layer. The invariant we wish to maintain is the following:

(P) For $j = 2, 3, \ldots, n$, $\mathcal{A}_i$ enters state $\text{reject}_{i,j}$ for a given sequence of values for the X_v iff, just by inspecting the value of X_1 (i.e., X_i) and those of $X_2, X_3, \ldots, X_j$, we have with probability one that $\mathcal{A}'$ will not put vertex i in the IS.

We may use the nonreject states to remember some useful information to assist us in maintaining the invariant (P).

The automata-construction is simple in the case of graphs: $\mathcal{A}_i$ simply has two states, $\text{ok-so-far}_{i,j}$ and $\text{reject}_{i,j}$, in layer j for $j = 2, 3, \ldots, n$, and the states $\text{ok-so-far}_{i,n}$ and $\text{accept}_{i,n}$ are identical. As above, all transitions from $\text{reject}_{i,j}$ go to $\text{reject}_{i,j+1}$. There is a transition from state $s_{1,\ell}$ in layer 1 to $\text{reject}_{i,2}$ iff vertex 2 is a neighbor of i, and if $X_2 \leq X_1 (= \ell)$; otherwise, we transit from $s_{1,\ell}$ to $\text{ok-so-far}_{i,2}$. In general, there is a transition from $\text{ok-so-far}_{i,j-1}$ to $\text{reject}_{i,j}$ iff vertex j is a neighbor of i, and if $X_j \leq X_i$; otherwise, we will transit from $\text{ok-so-far}_{i,j-1}$ to $\text{ok-so-far}_{i,j}$. It is easy to see that this construction satisfies (P), and so we are done in the case of graphs.

For the family of hypergraphs (H2), the situation is not much more difficult. Consider $\mathcal{A}_i$, and once again renumber the vertices so that $i = 1$. Let $e_1, e_2, \ldots, e_t$ be the sets in the given hypergraph that contain the vertex i, and define $e_{\ell,j} = e_\ell \cap \{1, 2, \ldots, j\}$. To maintain the invariant (P), it suffices to know which of the following three disjoint cases holds for each e_ℓ, after reading the values of $X_1, X_2, \ldots, X_j$:

(i) $e_{\ell,j} = e_\ell$, and $X_u \leq X_i$ for all $u \in e_{\ell,j}$;
(ii) $e_{\ell,j} \neq e_\ell$, and $X_u \leq X_i$ for all $u \in e_{\ell,j}$;
(iii) $X_u > X_i$ for some $u \in e_{\ell,j}$.

Note that $\mathcal{A}_i$ should enter state $\text{reject}_{i,j}$ iff case (i) holds for some ℓ. Otherwise, case (ii) or case (iii) holds for each ℓ. Thus, in addition to state $\text{reject}_{i,j}$, we need only $2^t \leq 2^{c \cdot \lg(n+m)} = (n+m)^c$ states to remember which case holds for each ℓ. Given these semantics for the states, it is also easy to see how the state-transitions should occur. Thus, we get a polynomial-sized bound for S_i for the family of hypergraphs (H2) also.

Constant-degree hypergraphs. When the maximum degree of any vertex is bounded by some constant $d > 1$, we propose an alternative approach for finding a random permutation of the vertices. The resulting algorithm yields an IS of weight at least $(1 - \epsilon)$ times the expected value presented in Theorem 2.2; ϵ here denotes an arbitrary positive constant.

Our approach uses algorithm $\mathcal{A}$, as given in Figure 1. Recall that B_v is the event that vertex v is in the final IS produced by $\mathcal{A}$. We rewrite $\text{Prob}(B_v)$ as follows. Suppose that the vertex v lies in edges $e_{v,1}, e_{v,2}, \ldots, e_{v,d(v)}$, where $d(v) \leq d$; as mentioned in Remark 3, we assume that all of these edges have size at least 2. Let $[k]$ denote the set $\{1, 2, \ldots, k\}$. Denote by $C_{v,u}$ the event "$X_v \geq X_u$"; then, by inclusion-exclusion,

$$\text{Prob}(B_v) = 1 + \sum_{S \subseteq [d(v)],\ |S| \geq 1} (-1)^{|S|} \cdot \text{Prob}\left(\bigwedge_{i \in S} [\forall u \in e_{v,i},\ C_{v,u}] \right) \tag{13}$$

$$= 1 + \sum_{S \subseteq [d(v)],\ |S| \geq 1} (-1)^{|S|} \cdot \text{Prob}\left(\forall u \in \left[\left(\bigcup_{i \in S} e_{v,i} \right) \setminus \{v\} \right],\ C_{v,u} \right) \tag{14}$$

$$= 1 + \sum_{S \subseteq [d(v)],\ |S| \geq 1} (-1)^{|S|} \cdot \left(1 + \left| \left(\bigcup_{i \in S} e_{v,i} \right) - \{v\} \right| \right)^{-1}; \tag{15}$$

equation (15) follows from the fact that each X_u, $u \in \bigcup_{i \in S} e_{v,i}$, is equally likely to be $\max\{X_w : w \in \bigcup_{i \in S} e_{v,i}\}$.

Denote by S_n the set of all permutations of $[n]$. A family of permutations $\mathcal{F} \subseteq S_n$ is defined in [4] to be *approximately minwise independent with relative error* $\delta > 0$ if, for all $X \subseteq [n]$ and $x \in X$, we have for a randomly chosen permutation $\pi \in \mathcal{F}$ that

$$\left| \text{Prob}(\min\{\pi(X)\} = \pi(x)) - \frac{1}{|X|} \right| \leq \frac{\delta}{|X|}.$$

We abbreviate the above property by (n, δ)-amw. We will use the property that for any n and $\delta > 0$, there is an explicitly constructible permutation family $F(n, \delta)$ that satisfies property (n, δ)-amw and has cardinality $n^{O(\lg(1/\delta))}$ [11]. Note that given such a family $F(n, \delta)$, we can generate a family of permutations of the same size[3] that satisfies

$$\frac{1-\delta}{|X|} \leq \text{Prob}(\max\{\pi(X)\} = \pi(x)) \leq \frac{1+\delta}{|X|}. \tag{16}$$

Thus, without loss of generality, when referring to $F(n, \delta)$, we assume that F satisfies (16). We show below that for an appropriate constant $\delta = \delta(\epsilon, d)$, the expected size of an IS produced by using a random element of $F(n, \delta)$ is at least $(1-\epsilon)$ times the value guaranteed by Theorem 2.2. Thus, it suffices to choose a random permutation from the explicit polynomial-sized set $F(n, \delta)$. We can then apply a parallel exhaustive search on the polynomial-sized $F(n, \delta)$ to find a "good" permutation in NC.

THEOREM 3.1. *Consider the family of hypergraphs with maximum degree at most d for any given constant $d > 0$. Then, given any constant $\epsilon > 0$, there is an NC algorithm to find in these hypergraphs an IS of total weight at least $(1 - \epsilon)$ times the expected weight guaranteed by Theorem* 2.2.

Proof. Let $\delta > 0$ be a constant (to be determined). We denote by $\text{Prob}(B_v | F(n, \delta))$ the probability that vertex v is in the final IS produced by $\mathcal{A}$, conditioned on the random choice of π from $F(n, \delta)$; then, it is sufficient to show that for any $v \in V$, $\text{Prob}(B_v | F(n, \delta))$ is at least $(1 - \epsilon)$ times the value guaranteed by Lemma 2.1. The statement of the theorem will then follow from the linearity of expectation. Denote by V_S the set of vertices that lie in the subset of edges $e_{v,i_1}, \ldots, e_{v,i_{|S|}}$. Then, by inclusion-exclusion,

$$\text{Prob}(B_v | F(n, \delta)) = \sum_{S \subseteq [d(v)]} (-1)^{|S|} \cdot \text{Prob}(v \text{ is last in } \pi \text{ among the vertices in } V_S \mid F(n, \delta)).$$

Using (15) and (16), we get that

$$\text{Prob}(B_v | F(n, \delta)) \geq 1 + \sum_{S \subseteq [d(v)]:\ |S| \geq 1} (-1)^{|S|} \cdot \left(1 + \left| \left(\bigcup_{i \in S} e_{v,i} \right) - \{v\} \right| \right)^{-1} - \delta \sum_{S \subseteq [d(v)]:\ |S| \geq 1} \left(1 + \left| \left(\bigcup_{i \in S} e_{v,i} \right) - \{v\} \right| \right)^{-1}$$

[3] This is done by taking, for any permutation π chosen from $\mathcal{F}$, the reverse permutation π', that is, $\pi'(i) = n + 1 - \pi(i)$ for all i.

$$\geq \text{Prob}(B_v) - \delta \sum_{S \subseteq [d(v)]:\ |S| \geq 1} (1/2)$$
$$= \text{Prob}(B_v) - \delta \left(2^{d-1} - \frac{1}{2}\right).$$

(The second inequality follows from our assumption that all the edges $e_{v,i}$ have size at least 2.) It follows from Lemma 2.1 that $\text{Prob}(B_v) \geq \int_0^1 (1-x)^{d(v)}\, dx = \frac{1}{d(v)+1} \geq \frac{1}{d+1}$. Hence, taking $\delta = \frac{2\epsilon}{(d+1)(2^d-1)}$ and summing over all $v \in V$, we get the statement of the theorem. □

4. Concluding remarks. A question that remains open is to obtain a full derandomization of our *RNC* algorithms. Any progress on the classical MIS problem on hypergraphs would also be most interesting.

Acknowledgments. We thank Noga Alon for valuable discussions and for pointing out to us the results of Caro and Tuza. Thanks to Andrei Broder and Mike Mitzenmacher for fruitful discussions. We also thank the SPAA 2001 program committee member(s) and referee(s), as well as the journal referee, for their helpful comments.

REFERENCES

[1] N. Alon, L. Babai, and A. Itai, *A fast and simple randomized parallel algorithm for the maximal independent set problem*, J. Algorithms, 7 (1986), pp. 567–583.
[2] N. Alon and J. H. Spencer, *The Probabilistic Method*, 2nd ed., Wiley-Interscience, New York, 2000.
[3] P. Beame and M. Luby, *Parallel search for maximal independence given minimal dependence*, in Proceedings of the ACM–SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 1990, pp. 212–218.
[4] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher, *Min-wise independent permutations*, J. Comput. System Sci., 60 (2000), pp. 630–659.
[5] Y. Caro and Z. Tuza, *Improved lower bounds on k-independence*, J. Graph Theory, 15 (1991), pp. 99–107.
[6] E. Dahlhaus, M. Karpinski, and P. Kelsen, *An efficient parallel algorithm for computing a maximal independent set in a hypergraph of dimension* 3, Inform. Process. Lett., 42 (1992), pp. 309–314.
[7] C. M. Fortuin, J. Ginibre, and P. N. Kasteleyn, *Correlational inequalities for partially ordered sets*, Comm. Math. Phys., 22 (1971), pp. 89–103.
[8] M. Goldberg and T. Spencer, *A new parallel algorithm for the maximal independent set problem*, SIAM J. Comput., 18 (1989), pp. 419–427.
[9] M. Goldberg and T. Spencer, *An efficient parallel algorithm that finds independent sets of guaranteed size*, SIAM J. Discrete Math., 6 (1993), pp. 443–459.
[10] M. Hofri, *Analysis of Algorithms*, Oxford University Press, Oxford, UK, 1995.
[11] P. Indyk, *A small approximately min-wise independent family of hash functions*, in Proceedings of the ACM–SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 1999, pp. 454–456.
[12] D. R. Karger and D. Koller, *(De)randomized construction of small sample spaces in NC*, J. Comput. System Sci., 55 (1997), pp. 402–413.
[13] R. M. Karp and V. Ramachandran, *Parallel algorithms for shared memory machines*, in Handbook of Theoretical Computer Science, Volume A, J. van Leeuwen, ed., Elsevier, New York, 1990, pp. 871–941.
[14] R. M. Karp, E. Upfal, and A. Wigderson, *The complexity of parallel search*, J. Comput. System Sci., 36 (1988), pp. 225–253.
[15] R. M. Karp and A. Wigderson, *A fast parallel algorithm for the maximal independent set problem*, JACM, 32 (1985), pp. 762–773.
[16] P. Kelsen, *On the parallel complexity of computing a maximal independent set in a hypergraph*, in Proceedings of the ACM Symposium on Theory of Computing, ACM, New York, 1992, pp. 339–350.

[17] M. Luby, *A simple parallel algorithm for the maximal independent set problem*, SIAM J. Comput., 15 (1986), pp. 1036–1053.
[18] T. Łuczak and E. Szymańska, *A parallel randomized algorithm for finding a maximal independent set in a linear hypergraph*, J. Algorithms, 25 (1997), pp. 311–320.
[19] R. J. McEliece and K. N. Sivarajan, *Performance limits for channelized cellular telephone systems*, IEEE Trans. Inform. Theory, 40 (1994), pp. 21–34.
[20] S. Mahajan, E. A. Ramos, and K. V. Subrahmanyam, *Solving some discrepancy problems in NC*, Algorithmica, 29 (2001), pp. 371–395.
[21] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, Cambridge, UK, 1995.
[22] N. Nisan, *Pseudorandom generators for space-bounded computation*, Combinatorica, 12 (1992), pp. 449–461.
[23] N. Nisan, $RL \subseteq SC$, in Proceedings of the ACM Symposium on Theory of Computing, ACM, New York, 1992, pp. 619–623.
[24] R. Sedgewick and P. Flajolet, *An Introduction to the Analysis of Algorithms*, Addison–Wesley, Reading, MA, 1996.
[25] D. Sivakumar, *Algorithmic derandomization via complexity theory*, in Proceedings of the ACM Symposium on Theory of Computing, ACM, New York, 2002, pp. 619–626.
[26] J. H. Spencer, *The probabilistic lens: Sperner, Turán and Brégman revisited*, in A Tribute to Paul Erdős, A. Baker, B. Bollobás, and A. Hajnal, eds., Cambridge University Press, Cambridge, UK, 1990, pp. 391–396.
[27] E. Szymańska, *Derandomization of a parallel MIS algorithm in a linear hypergraph*, in Proceedings of the International Colloquium on Automata, Languages and Programming, Satellite Workshops, Carleton Scientific, Waterloo, Canada, 2000, pp. 39–52.
[28] P. Turán, *On the theory of graphs*, Colloq. Math., 3 (1954), pp. 19–30.

SIAM J. DISCRETE MATH.
Vol. 18, No. 3, pp. 501–511

BIDIMENSIONAL PARAMETERS AND LOCAL TREEWIDTH*

ERIK D. DEMAINE†, FEDOR V. FOMIN‡, MOHAMMADTAGHI HAJIAGHAYI†, AND DIMITRIOS M. THILIKOS§

Abstract. For several graph-theoretic parameters such as vertex cover and dominating set, it is known that if their sizes are bounded by k, then the treewidth of the graph is bounded by some function of k. This fact is used as the main tool for the design of several fixed-parameter algorithms on minor-closed graph classes such as planar graphs, single-crossing-minor-free graphs, and graphs of bounded genus. In this paper we examine whether similar bounds can be obtained for larger minor-closed graph classes and for general families of graph parameters, including all those for which such behavior has been reported so far. Given a graph parameter P, we say that a graph family $\mathcal{F}$ has the *parameter-treewidth property for* P if there is an increasing function t such that every graph $G \in \mathcal{F}$ has treewidth at most $t(P(G))$. We prove as our main result that, for a large family of graph parameters called *contraction-bidimensional*, a minor-closed graph family $\mathcal{F}$ has the parameter-treewidth property if $\mathcal{F}$ has bounded local treewidth. We also show "if and only if" for some graph parameters, and thus, this result is in some sense tight. In addition we show that, for a slightly smaller family of graph parameters called *minor-bidimensional*, all minor-closed graph families $\mathcal{F}$, excluding some fixed graphs, have the parameter-treewidth property. The contraction-bidimensional parameters include many domination and covering graph parameters such as vertex cover, feedback vertex set, dominating set, edge-dominating set, and q-dominating set (for fixed q). We use our theorems to develop new fixed-parameter algorithms in these contexts.

Key words. treewidth, local treewidth, graph minors, dominating set

AMS subject classifications. 05C85, 68Q25, 68R10

DOI. 10.1137/S0895480103433410

1. Introduction.

The last ten years have witnessed the rapid development of a new branch of computational complexity called parameterized complexity; see the book of Downey and Fellows [19]. Roughly speaking, a parameterized problem with a parameter of nonnegative integer k is *fixed-parameter tractable* (FPT) if it admits an algorithm with running time $h(k)\,|I|^{O(1)}$. (Here h is a function depending *only* on k and $|I|$ is the size of the instance.)

A celebrated example of an FPT problem is the vertex cover, which asks whether an input graph has at most k vertices that are incident to all its edges. When parameterized by k, the k-vertex cover problem admits a solution as fast as $O(kn+1.285^k)$ [9]. Moreover, if we restrict k-vertex cover to planar graphs, then it is possible to design FPT algorithms where the contribution of k in the nonpolynomial part of their complexity is subexponential. The first algorithm of this type was given by Alber, Fernau, and Niedermeier [4]. Recently, Fomin and Thilikos reported an $O(k^4 + 2^{4.5\sqrt{k}} + kn)$

*Received by the editors August 19, 2003; accepted for publication (in revised form) May 6, 2004; published electronically December 30, 2004. The results of this paper appeared in the Proceedings of the 11th European Symposium on Algorithms (ESA 2003) [24] and in the Proceedings of the 6th Latin American Theoretical Informatics Symposium (LATIN 2004) [11].

http://www.siam.org/journals/sidma/18-3/43341.html

†MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar Street, Cambridge, MA 02139 (edemaine@mit.edu, hajiagha@mit.edu).

‡Department of Informatics, University of Bergen, N-5020 Bergen, Norway (fomin@ii.uib.no). Supported by Norges forskningsråd project 160778/V30.

§Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Campus Nord – Mòdul C5, c/Jordi Girona Salgado 1-3, E-08034, Barcelona, Spain (sedthilk@lsi.upc.es). Supported by the EU basic research project 001907 DELIS and by the Spanish CICYT project TIC-2002-04498-C05-03 (TRACER).

algorithm for planar k-vertex cover [25].

However, not all parameterized problems are FPT. A typical example of such a problem is the dominating set, asking whether an input graph has at most k vertices that are adjacent to the rest of the vertices. When parameterized by k, the k-dominating set problem is known to be $W[2]$-complete and thus it is not expected to be FPT [19]. Interestingly, the fixed-parameter complexity of the same problem can be distinct for special graph classes. During the last five years, there has been substantial work on fixed-parameter algorithms for solving the k-dominating set on planar graphs and different generalizations of planar graphs. For this class, the problem can be solved in $O(8^k n)$ time [2]. An algorithm with a sublinear exponent for the problem with running time $O(4^{6\sqrt{34k}}n)$ was given by Alber et al. [1]. Recently, Kanj and Perković [30] improved the running time to $O(2^{27\sqrt{k}}n)$ and Fomin and Thilikos to $O(2^{15.13\sqrt{k}}k + n^3 + k^4)$ [23, 25]. The fixed-parameter algorithms for extensions of planar graphs, like bounded-genus graphs and graphs excluding single-crossing graphs as minors, are introduced in [13, 15, 20].

In the majority of these results, the design of FPT algorithms for solving problems such as k-vertex cover or k-dominating set in a sparse graph class $\mathcal{F}$ is based on the following lemma: every graph G in $\mathcal{F}$, where the value of the graph parameter is at most k, has treewidth bounded by $t(k)$, where t is a strictly increasing function depending only on $\mathcal{F}$. With some work (sometimes very technical), a tree decomposition of width $t(k)$ is constructed and standard dynamic-programming techniques on graphs of bounded treewidth are implemented. Of course this method cannot be applied for any graph class $\mathcal{F}$. For instance, the n-vertex complete graph K_n has a dominating set of size one and treewidth equal to $n-1$. So the emerging question is: For which (larger) graph classes and for which graph parameters can the "bounding treewidth method" be applied?

In this paper we give a *complete* characterization of minor-closed graph families for which the aforementioned "bounding treewidth method" can be applied for a wide family of graph parameters. For a given graph parameter P, we say that a graph family $\mathcal{F}$ has the *parameter-treewidth property* for P if there is a strictly increasing function $t : \mathbb{N} \to \mathbb{N}$ such that every graph $G \in \mathcal{F}$ where $P(G) \leq k$ implies that G has treewidth at most $t(k)$. For example, it is known [1, 23, 30] that any planar graph with a dominating set of size at most k has treewidth $O(\sqrt{k})$. Therefore, the class of planar graphs has the parameter-treewidth property for the dominating-set parameter.

Our main result is that for a large family of graph parameters called *contraction-bidimensional*, a minor-closed graph family $\mathcal{F}$ has the parameter-treewidth property if $\mathcal{F}$ has bounded local treewidth. Moreover, we show that the inverse is also correct if some simple condition is satisfied by P. In addition we show that, for a slightly smaller family of graph parameters called *minor-bidimensional*, every minor-closed graph family $\mathcal{F}$ excluding some fixed graph has the parameter-treewidth property. The bidimensional-parameter family includes many domination and covering graph parameters such as vertex cover, feedback vertex set, dominating set, edge-dominating set, and q-dominating set (for fixed q) (see also [15, 12] for more examples). Another example of a contraction-bidimensional parameter is the length of a minimum traveling salesman tour, i.e., the smallest number of edges in a walk visiting all vertices of the graph.

The proof of the main result uses the characterization of Eppstein for minor-closed families of bounded local treewidth [21] and Diestel et al.'s modification of

the Robertson and Seymour excluded-grid-minor theorem [18]. In addition, the proof is constructive and can be used for constructing fixed-parameter algorithms to decide bidimensional graph parameters on minor-closed families of bounded local treewidth. These algorithms parallel the general fixed-parameter algorithm of Frick and Grohe [27] for properties definable in first-order logic in graph families of bounded local treewidth; our results apply, e.g., to minor-bidimensional parameters definable in monadic second-order logic in nontrivial minor-closed graph families. See section 5 for details.

This paper is organized as follows. Section 2 contains the formal definitions of the concepts used in the paper. Section 3 presents two combinatorial results which support the main result of the paper, proved in section 4. Finally, in section 5 we present the algorithmic consequences of our results and we conclude with some open problems.

2. Definitions and preliminary results. Let G be a graph with vertex set $V(G)$ and edge set $E(G)$. We let n denote the number of vertices of a graph when it is clear from context. For every nonempty $W \subseteq V(G)$, the subgraph of G induced by W is denoted by $G[W]$. We define the *q-neighborhood* of a vertex $v \in V(G)$, denoted by $N_G^q[v]$, to be the set of vertices of G at distance at most q from v. Notice that $v \in N_G^q[v]$. We put $N_G[v] = N_G^1[v]$. We also often say that a vertex v *dominates* subset $S \subseteq V(G)$ if $N_G[v] \supseteq S$.

Given an edge $e = \{x, y\}$ of a graph G, the graph G/e is obtained from G by contracting the edge e; that is, to get G/e we identify the vertices x and y and remove all loops and duplicate edges. A graph H obtained by a sequence of edge contractions is said to be a *contraction* of G. We use the notation $H \preceq_c G$ for H a contraction of G. Notice that the relation $H \preceq_c G$ partitions the edge set of G into edges that are also the edges of H and the contracted edges. We say that a vertex v of G is contracted to a vertex u of H if there is a path from u to v in G such that all edges in this path are contracted. A graph H is a *minor* of a graph G if H is the subgraph of a contraction of G. We use the notation $H \preceq G$ (respectively, $H \preceq_c G$) for H a minor (contraction) of G. A family (or class) of graphs $\mathcal{F}$ is *minor-closed* if $G \in \mathcal{F}$ implies that every minor of G is in $\mathcal{F}$. A minor-closed graph family $\mathcal{F}$ is *H-minor-free* if $H \notin \mathcal{F}$.

The $m \times m$ grid is the graph on $\{1, 2, \ldots, m^2\}$ vertices $\{(i, j) \colon 1 \le i, j \le m\}$ with the edge set

$$\{(i, j)(i', j') \colon |i - i'| + |j - j'| = 1\}.$$

For $i \in \{1, 2, \ldots, m\}$, the vertex set (i, j), $j \in \{1, 2, \ldots, m\}$, is referred to as the *ith row* and the vertex set (j, i), $j \in \{1, 2, \ldots, m\}$, is referred to as the *ith column* of the $m \times m$ grid. The vertices (i, j) of the $m \times m$ grid with $i \in \{1, m\}$ or $j \in \{1, m\}$ are called *boundary* vertices and the rest of the vertices are called *nonboundary* vertices.

The notion of treewidth was introduced by Robertson and Seymour [31]. A *tree decomposition* of a graph G is a pair $(\{X_i \mid i \in I\}, T = (I, F))$ with $\{X_i \mid i \in I\}$ a family of subsets of $V(G)$ and T a tree, such that

1. $\bigcup_{i \in I} X_i = V(G)$;
2. for all $\{v, w\} \in E(G)$, there is an $i \in I$ with $v, w \in X_i$; and
3. for all $i_0, i_1, i_2 \in I$, if i_1 is on the path from i_0 to i_2 in T, then $X_{i_0} \cap X_{i_2} \subseteq X_{i_1}$.

The *width* of the tree decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ is $\max_{i \in I} |X_i| - 1$. The treewidth $\mathbf{tw}(G)$ of a graph G is the minimum width of a tree decomposition of G.

We need the following facts about treewidth. The first fact is trivial.

- For any complete graph K_n on n vertices, $\mathbf{tw}(K_n) = n - 1$.

The second fact is well known but its proof is not trivial. (See, e.g., [17].)

- The treewidth of the $m \times m$ grid is m.

The next fact we need is the improved version of the Robertson and Seymour theorem on excluded grid minors [32] due to Diestel et al. [18]. (See also the textbook [17].)

THEOREM 2.1 (see [18]). *Let r, m be integers, and let G be a graph of treewidth at least $m^{4r^2(m+2)}$. Then G contains either K_r or the $m \times m$ grid as a minor.*

Formally, a *graph parameter* P is a function that maps graphs to nonnegative integers. The *parameterized problem associated with* P asks, for a fixed k, whether $P(G) \leq k$ for a given graph G. Given a graph parameter P, we say that a graph family $\mathcal{F}$ has the *parameter-treewidth property for* P if there is a strictly increasing function t such that every graph $G \in \mathcal{F}$ has treewidth at most $t(P(G))$.

DEFINITION 2.2. *Let $g : \mathbb{N} \to \mathbb{N}$ be a strictly increasing function. We say that a graph parameter P is* g-minor-bidimensional[1] *if the following apply:*

- *Contracting an edge, deleting an edge, or deleting a vertex in a graph G cannot increase $P(G)$.*
- *For the $r \times r$ grid R, $P(R) \geq g(r)$.*

Similarly, a graph parameter P is g-contraction-bidimensional *if the following apply:*

- *Contracting an edge in a graph G cannot increase $P(G)$.*
- *For any $r \times r$ augmented grid R of constant span, $P(R) \geq g(r)$.*

In the above definition, an *$r \times r$ augmented grid of span s* is an *$r \times r$ grid* with some extra edges such that each vertex is attached to at most s nonboundary vertices of the grid (see an example in Figure 2.1). Intuitively, bidimensional parameters are required to be "large" in two-dimensional grids.

We note that a g-minor-bidimensional parameter is also a g-contraction-bidimensional parameter. One can easily observe that many graph parameters, such as minimum sizes of a dominating set, q-dominating set (distance q-dominating set for a fixed q), vertex cover, feedback vertex set, and edge-dominating set (see exact definitions of the corresponding graph parameters in [15]), are $\Theta(r^2)$-minor-bidimensional or $\Theta(r^2)$-contraction-bidimensional parameters.

Here, we present a theorem for minor-bidimensional parameters on general minor-closed classes of graphs excluding some fixed graphs, which plays an important role in the main result of this paper.

THEOREM 2.3. *If a g-minor-bidimensional parameter P on an H-minor-free graph G has value at most k, then* $\mathbf{tw}(G) \leq 2^{4|V(H)|^2(g^{-1}(k)+2)\log(g^{-1}(k))} = 2^{O(g^{-1}(k)\log(g^{-1}(k)))}$.

Proof. Notice that $K_{|V(H)|}$ contains as a minor any graph on $|V(H)|$ vertices. Therefore we may assume that G is $K_{|V(H)|}$-minor-free. If the claimed upper bound for the treewidth of G is not correct, then Theorem 2.1 implies that G contains a $m \times m$ grid R as a minor for $m > g^{-1}(k)$. Because P is g-minor-bidimensional, $P(R) \geq g(m)$. The bidimensionality of P along with the fact that R is a minor of G yields $P(G) \geq g(m)$. Therefore, $k \geq g(m)$, a contradiction. □

Theorem 2.3 can be applied for minor-bidimensional parameters such as a vertex cover or feedback vertex set.

[1] Closely related notions of bidimensional parameters are introduced by the authors in [13].

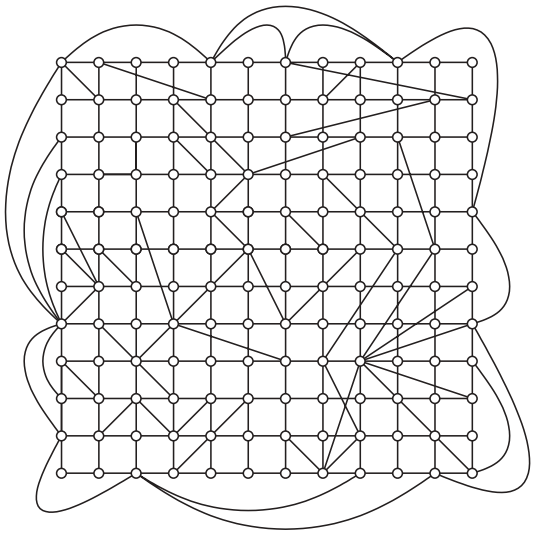


FIG. 2.1. *An augmented* 12×12 *grid with span* 8.

The notion of local treewidth was introduced by Eppstein [21] (see also [29]). The *local treewidth* of a graph G is

$$\mathbf{ltw}(G, r) = \max\{\mathbf{tw}(G[N_G^r[v]]) \colon v \in V(G)\}.$$

For a function $f \colon \mathbb{N} \to \mathbb{N}$ we define the minor-closed class of graphs of bounded local treewidth

$$\mathcal{L}(f) = \{G \colon \forall H \preceq G \ \forall r \geq 0,\ \mathbf{ltw}(H, r) \leq f(r)\}.$$

Also we say that a minor-closed class of graphs $\mathcal{C}$ has bounded local treewidth if $\mathcal{C} \subseteq \mathcal{L}(f)$ for some function f.

Well-known examples of minor-closed classes of graphs of bounded local treewidth are graphs of bounded treewidth, planar graphs, graphs of bounded genus, and single-crossing-minor-free graphs.

Many difficult graph problems can be solved efficiently when the input is restricted to graphs of bounded treewidth (see, e.g., Bodlaender's survey [7]). Eppstein [21] made a step forward by proving that some problems, like subgraph isomorphism and induced subgraph isomorphism, can be solved in linear time on minor-closed graphs of bounded local treewidth. Also, the classic Baker's technique [6] for obtaining approximation schemes on planar graphs for different NP-hard problems can be generalized to minor-closed families of bounded local treewidth. (See [21] for a generalization of these techniques.)

An *apex graph* is a graph G such that, for some vertex v (the *apex*), $G - v$ is planar. The following result is due to Eppstein [21].

THEOREM 2.4 (see [21]). *Let $\mathcal{F}$ be a minor-closed family of graphs. Then $\mathcal{F}$ is of bounded local treewidth if and only if $\mathcal{F}$ does not contain all apex graphs.*

3. Combinatorial lemmas. In this section we prove two combinatorial lemmas regarding grids and graphs of bounded local treewidth.

LEMMA 3.1. *Suppose we have an $m \times m$ grid H and a subset S of vertices in the central $(m - 2\ell) \times (m - 2\ell)$ subgrid H', where $s = |S|$ and $\ell = \lfloor \sqrt[4]{s} \rfloor$. Then H has as*

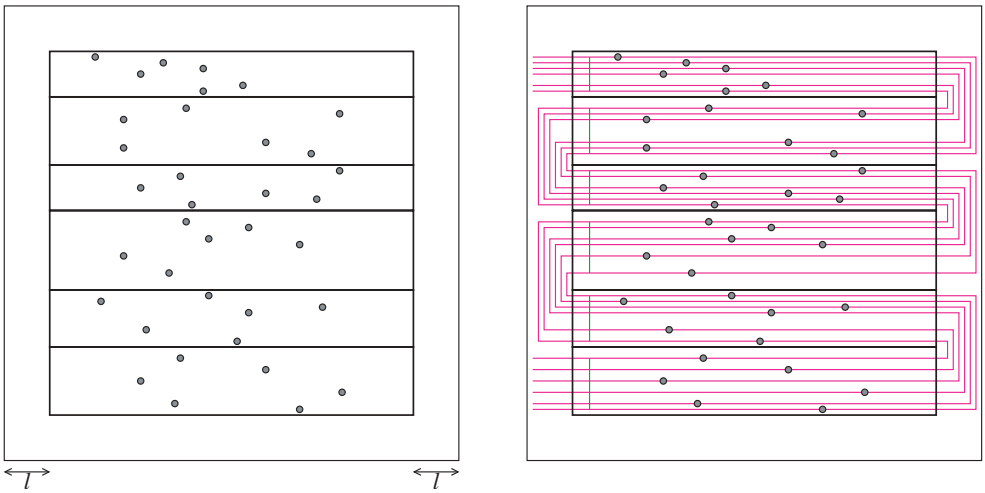


FIG. 3.1. *Left: The grid H, the points in S'', and their grouping. Here $\ell = 6$. Right: Construction of the minor $\ell \times \ell$ grid R passing through the points in S''.*

a minor the $\ell \times \ell$ grid R such that each vertex in R is a contraction of at least one vertex in S and other vertices in H.

Proof. Let s_x denote the number of distinct x coordinates of the vertices in S, and let s_y denote the number of distinct y coordinates of the vertices in S. Thus, $s \leq s_x \cdot s_y$. Assume by symmetry that $s_y \geq s_x$, and therefore $s_y \geq \sqrt{s}$.

We define the subset S' of S by removing all points but one that share a common y coordinate, for each y coordinate. Thus, all y coordinates of the vertices in S' are distinct, and $|S'| = s_y$. We discard all but ℓ^2 vertices in S' to form a slightly smaller set S'', because $|S'| = s_y \geq \sqrt{s} \geq (\lfloor \sqrt[4]{s} \rfloor)^2 = \ell^2$. We divide these ℓ^2 vertices into ℓ groups, each of exactly ℓ consecutive vertices according to the order of their y coordinates. Now we have the situation shown on the left of Figure 3.1.

We construct the minor grid R as shown on the right of Figure 3.1. Because each y coordinate is unique, we can draw long horizontal segments through every point. The ℓ columns on the left-hand and right-hand sides of H allow us to connect these horizontal segments together into ℓ vertex-disjoint paths, each passing through exactly ℓ vertices of S''. These paths can be connected by vertical segments within each group. By contracting each horizontal segment into a single vertex, and by some further contraction, we can obtain the desired $\ell \times \ell$ grid R as a minor. Each vertex of this grid R is a contraction of at least one vertex in S'' (and hence in S) and other vertices in H. □

LEMMA 3.2. *Let $G \in \mathcal{L}(f)$ be a graph containing the $m \times m$ grid H as a subgraph, $m > 2\ell$, where $\ell = f(2) + 1$. Then the central $(m - 2\ell) \times (m - 2\ell)$ subgrid H' has the property that every vertex $v \in V(G)$ is adjacent to less than ℓ^4 vertices in H'.*

Proof. Suppose for contradiction that there is a vertex $v \in V(G)$ such that $S = N_G[v] \cap V(H)$ has size $s = |S| \geq \ell^4$. By Lemma 3.1, H' has as a minor an $\ell \times \ell$ grid R such that each vertex in R is a contraction of at least one vertex in S and other vertices in H'. If we perform these contractions and deletions in G, then v is adjacent to all vertices in R. Define $R+v$ to be the grid R plus the vertex v (if v is not already in R) and the star of connections between v and all vertices in R. Then $R + v$ is a minor of G, but has diameter 2 and treewidth $\ell \geq f(2) + 1$, a contradiction. □

4. Main theorem. Now we are ready to present the main result of this paper.

THEOREM 4.1. *Let P be a contraction-bidimensional parameter. A minor-closed graph class $\mathcal{F}$ has the parameter-treewidth property for P if $\mathcal{F}$ is of bounded local treewidth. In particular, for any g-contraction-bidimensional parameter P, function $f\colon \mathbb{N} \to \mathbb{N}$, and any graph $G \in \mathcal{L}(f)$ on which P has value at most k, we have $\mathbf{tw}(G) \leq 2^{O(g^{-1}(k)\log g^{-1}(k))}$. (The constant in the O notation depends on $f(1)$ and $f(2)$.)*

Proof. Let $r = f(1) + 1$ and $\ell = f(2) + 1$. Let $G \in \mathcal{L}(f)$ be a graph on which the graph parameter P has value k. Let m be the largest integer such that $\mathbf{tw}(G) \geq m^{4r^2(m+2)}$. Without loss of generality, we assume G is connected, and $m > 2\ell$ (otherwise, $\mathbf{tw}(G)$ is a constant because both r and ℓ are constants). Then G has no complete graph K_r as a minor. By Theorem 2.1, G contains an $m \times m$ grid H as a minor. Thus there exists a sequence of edge contractions and edge/vertex deletions reducing G to H. We apply to G the edge contractions from this sequence, we ignore the edge deletions, and instead of deletion of a vertex v, we only contract v into one of its neighbors. Call the new graph G', which has the $m \times m$ grid H as a subgraph and, in addition, $V(G') = V(H)$. Because graph parameter P is contraction-bidimensional, its value on G' will not increase. By Lemma 3.2, we know that the central $(m - 2\ell) \times (m - 2\ell)$ subgrid H' of H has the property that every vertex $v \in V(G')$ is adjacent to less than ℓ^4 vertices in H'.

Now, suppose that in graph G', we further contract all 2ℓ boundary rows and 2ℓ boundary columns into two boundary rows and two boundary columns (one on each side) and call the new graph G''. Note that here, G'' and H' have the same set of vertices. The degree of each vertex of G'' to the vertices that are not on the boundary is at most $(\ell + 1)^2\ell^4$, which is a constant because ℓ is a constant. Here the factor $(\ell+1)^2$ is for the boundary vertices, each of which is obtained by contraction of at most $(\ell + 1)^2$ vertices. Again, because graph parameter P is contraction-bidimensional, its value on G'' does not increase and thus it is at most p. On the other hand, because the graph parameter is g-contraction-bidimensional, its value on graph G'' is at least $g(m-2\ell)$. Thus $g^{-1}(k) \geq m-2\ell$, so $m = O(g^{-1}(k))$. By Theorem 2.3, the treewidth of the original graph G is at most $2^{O(g^{-1}(k)\log g^{-1}(k))}$, as desired. ∎

The *apex graphs* A_i, $i = 1, 2, 3, \ldots$, are obtained from the $i \times i$ grid by adding a vertex v adjacent to all vertices of the grid. It is interesting to see that, for a wide range of graph parameters, the inverse of Theorem 4.1 also holds.

LEMMA 4.2. *Let P be any contraction-bidimensional parameter where $P(A_i) = O(1)$ for any $i \geq 1$. A minor-closed graph class $\mathcal{F}$ has the parameter-treewidth property for P only if $\mathcal{F}$ is of bounded local treewidth.*

Proof. The proof follows from Theorem 2.4. The apex graph A_i has diameter ≤ 2 and treewidth $\geq i$. So a minor-closed family of graphs with the parameter-treewidth property for P cannot contain all apex graphs and hence it is of bounded local treewidth. ∎

Typical examples of graph parameters satisfying Theorem 4.1 and Lemma 4.2 are the dominating set and its generalization q-dominating set for a fixed constant q (in which each vertex can dominate its q-neighborhood). These graph parameters are $\Theta(r^2)$-contraction-bidimensional and their value is 1 for any apex graph $A_i, i \geq 1$.

We can strengthen the "if and only if" result provided by Theorem 4.1 and Lemma 4.2 with the following lemma. We just need to use the fact that if the value of P is less than the value of P', then the parameter-treewidth property for P implies the parameter-treewidth property for P' as well.

LEMMA 4.3. *Let P be a graph parameter whose value is lower bounded by some contraction-bidimensional parameter and let $P(A_i) = O(1)$ for any $i \geq 1$. Then a minor-closed graph class $\mathcal{F}$ has the parameter-treewidth property for P if and only if $\mathcal{F}$ is of bounded local treewidth.*

Proof. The "only if" direction is the same as in Lemma 4.2. Suppose now that P' is a contraction-bidimensional parameter where, for any graph G, $P'(G) \leq P(G)$. Applying Theorem 4.1 to P' we obtain that, if $\mathcal{F}$ is of bounded local treewidth, then $\mathcal{F}$ has the parameter-treewidth property for P', which means that there exists a strictly increasing function t such that, for any graph $G \in \mathcal{F}$, $\mathbf{tw}(G) \leq t(P'(G))$. As $P'(G) \leq P(G)$, we have that $\mathbf{tw}(G) \leq t(P(G))$, and thus $\mathcal{F}$ has the parameter-treewidth property for P. □

Lemma 4.3 can be used not only for contraction-bidimensional graph parameters. As an example, we mention the *clique-transversal number* of a graph, i.e., the minimum number of vertices meeting all the maximal cliques of a graph. (The clique-transversal number is not contraction-bidimensional because an edge contraction may create a new maximal clique and the value of the clique-transversal number may increase.) It is easy to see that this graph parameter always exceeds the domination number (the size of a minimum dominating set) and that any graph in A_i has a clique-transversal set of size 1.

Another application is the Π-*domination number*, i.e., the minimum cardinality of a vertex set that is a dominating set of G and satisfies some property Π in G. If this property is satisfied for any one-element subset of $V(G)$, then we call it *regular*. Examples of known variants of the parameterized dominating-set problem corresponding to the Π-domination number for some regular property Π are the following parameterized problems: the independent dominating set problem, the total dominating set problem, the perfect dominating set problem, and the perfect independent dominating set problem (see the exact definitions in [1]).

We summarize the previous observations with the following.

COROLLARY 4.4. *Let P be any of the following graph parameters: the minimum cardinality of a dominating set, the minimum cardinality of a q-dominating set (for any fixed q), the minimum cardinality of a clique-transversal set, or the minimum cardinality of a dominating set with some regular property Π. A minor-closed family of graphs $\mathcal{F}$ has the parameter-treewidth property for P if and only if $\mathcal{F}$ is of bounded local treewidth. The function $t(k)$ in the parameter-treewidth property is $2^{O(\sqrt{k}\log k)}$.*

5. Algorithmic consequences and concluding remarks. Courcelle [10] proved a metatheorem on graphs of bounded treewidth; he showed that, if ϕ is a property of graphs that is definable in monadic second-order logic, then ϕ can be decided in linear time on graphs of bounded treewidth. Frick and Grohe [27] partially extended this result to graphs of bounded local treewidth; they showed that, for any property ϕ that is definable in first-order logic and for any class of graphs of bounded local treewidth, there is an $O(n^{1+\varepsilon})$-time algorithm deciding whether a given graph has property ϕ for any $\varepsilon > 0$. The constant in the O notation depends on $1/\varepsilon$, ϕ, and the local treewidth bound. However, the running time of Frick and Grohe's algorithm remains unanalyzed in terms of ϕ: their algorithm transforms ϕ into so-called "Gaifman normal form" [28] and the complexity of this transformation is unknown.

Using Theorems 2.3 and 4.1, we obtain a result along lines similar to Frick and Grohe. Specifically, consider any property that is solvable in polynomial time on graphs of bounded treewidth, e.g., properties definable in monadic second-order logic. If the property is minor-bidimensional, we obtain a fixed-parameter algorithm on

general minor-closed graph families excluding some fixed graphs; if the property is contraction-bidimensional, we obtain a fixed-parameter algorithm on minor-closed graph families of bounded local treewidth. The differences between our result and Frick and Grohe's result are that our properties must be bidimensional but need not be definable in first-order logic, and our graph families must be minor-closed but need not have bounded local treewidth (for minor-bidimensional properties). Also, in contrast to the work of Frick and Grohe, the running time of our algorithm has an explicit bound.

THEOREM 5.1. *Let P be a graph parameter such that, given a tree decomposition of width at most w for a graph G, the graph parameter can be computed in $h(w)n^{O(1)}$ time. Now, if P is a g-minor-bidimensional parameter and G belongs to a minor-closed graph family excluding some fixed graphs, or P is a g-contraction-bidimensional parameter and G belongs to a minor-closed family of graphs of bounded local treewidth, then we can compute P on G in $h(2^{O(g^{-1}(k)\log g^{-1}(k))})n^{O(1)} + 2^{2^{O(g^{-1}(k)\log g^{-1}(k))}}n^{3+\varepsilon}$ time for any $\varepsilon > 0$.*

Proof. The algorithm is as follows. We check whether $\mathbf{tw}(G)$ is in $2^{O(g^{-1}(k)\log g^{-1}(k))}$. By Theorems 2.3 and 4.1, if it is not, graph parameter P has value more than k on graph G. This step can be performed by Amir's algorithm [5], which, for a given graph G and integer ω, either reports that the treewidth of G is at least ω or produces a tree decomposition of width at most $(3 + \frac{2}{3})\omega$ in time $O(2^{3.698\omega}n^3\omega^3\log^4 n)$. Thus, by using Amir's algorithm we can either compute a tree decomposition of G of size $2^{O(g^{-1}(k)\log g^{-1}(k))}$ in time $2^{2^{O(g^{-1}(k)\log g^{-1}(k))}}n^{3+\varepsilon}$ or conclude that the treewidth of G is not in $2^{O(g^{-1}(k)\log g^{-1}(k))}$.

Now if we find a tree decomposition of the aforementioned width, we can compute P on G in $h(2^{O(g^{-1}(k)\log g^{-1}(k))})n^{O(1)}$ time. The running time of this algorithm is the one mentioned in the statement of the theorem. □

For example, let G be a graph from a minor-closed family $\mathcal{F}$ of bounded local treewidth. Because the dominating set of a graph with a given tree decomposition of width at most ω can be computed in time $O(2^{2\omega}n)$ [1], Theorem 5.1 gives an algorithm which either computes a dominating set of size at most k or concludes that there is no such dominating set in $2^{2^{O(\sqrt{k}\log k)}}n^{O(1)}$ time. The same result holds also for computing the minimum size of a q-dominating set. Indeed, Theorem 5.1 can be applied because the q-dominating set of a graph with a given tree decomposition of width at most ω can be computed in time $O(q^{O(\omega)}n)$ [12]. Also, algorithms on graphs of bounded treewidth for the clique-transversal set and Π-domination set appeared in [8] and [1], respectively. Using these algorithms, and the fact that all these graph parameters are lower bounded by the domination number, the methodology of the proof of Theorem 5.1 can give algorithmic results for the clique-transversal set and Π-domination set with the same running times as in the case of the dominating set (i.e., $2^{2^{O(\sqrt{k}\log k)}}n^{O(1)}$).

Clearly, the algorithmic results of this paper are mainly of theoretical importance. Toward more practical algorithms, we mention some open problems. It is known that, for any planar graph G with a dominating set of size at most k, we have $\mathbf{tw}(G) = O(\sqrt{k})$. The same holds for many other graph parameters [1]. The same bound has also been proved for more general graph classes like graphs of bounded genus [13, 26, 16] and minor-closed graph families of bounded local treewidth [14]. It is natural to ask whether such a smaller bound holds in the case of any bidimensional parameter. This would provide subexponential fixed-parameter algorithms on minor-closed graph families of bounded local treewidth for any such graph parameter.

It is known that the dominating set problem admits a linear size kernel on planar graphs [3]. Recently, this result was extended to graphs of bounded genus [26]. It is tempting to ask whether such a kernel exists for any minor-closed graph class of bounded local treewidth, i.e., any minor-closed graph class with the parameter-treewidth property for a dominating set. The same question can be asked for other bidimensional parameters. In particular, we wonder whether there is any link between linear kernels and bidimensionality.

Acknowledgment. Thilikos is grateful to Maria Satratzemi for technically supporting his research at the Department of Applied Informatics of the University of Macedonia, Thessaloniki, Greece.

REFERENCES

[1] J. Alber, H. L. Bodlaender, H. Fernau, T. Kloks, and R. Niedermeier, *Fixed parameter algorithms for dominating set and related problems on planar graphs*, Algorithmica, 33 (2002), pp. 461–493.

[2] J. Alber, H. Fan, M. R. Fellows, H. Fernau, R. Niedermeier, F. A. Rosamond, and U. Stege, *Refined search tree technique for dominating set on planar graphs*, in Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science (MFCS 2001), Lecture Notes in Comput. Sci. 2136, Springer, Berlin, 2001, pp. 111–122.

[3] J. Alber, M. R. Fellows, and R. Niedermeier, *Polynomial-time data reduction for dominating set*, J. ACM, 51 (2004), pp. 363–384.

[4] J. Alber, H. Fernau, and R. Niedermeier, *Parameterized complexity: Exponential speed-up for planar graph problems*, J. Algorithms, 52 (2004), pp. 26–56.

[5] E. Amir, *Efficient approximation for triangulation of minimum treewidth*, in Uncertainty in Artificial Intelligence: Proceedings of the 17th Conference (UAI-2001), Morgan Kaufmann Publishers, 2001, pp. 7–15.

[6] B. S. Baker, *Approximation algorithms for NP-complete problems on planar graphs*, J. ACM, 41 (1994), pp. 153–180.

[7] H. L. Bodlaender, *A tourist guide through treewidth*, Acta Cybernet., 11 (1993), pp. 1–23.

[8] M. S. Chang, T. Kloks, and C. M. Lee, *Maximum clique transversals*, in Proceedings of the 27th International Workshop on Graph-Theoretic Concepts in Computer Science, Lecture Notes in Comput. Sci. 2204, Springer, Berlin, 2001, pp. 300–310.

[9] J. Chen, I. A. Kanj, and W. Jia, *Vertex cover: Further observations and further improvements*, J. Algorithms, 41 (2001), pp. 280–301.

[10] B. Courcelle, *Graph rewriting: An algebraic and logic approach*, in Handbook of Theoretical Computer Science, Vol. B, Elsevier, Amsterdam, 1990, pp. 193–242.

[11] E. D. Demaine, F. V. Fomin, M. T. Hajiaghayi, and D. M. Thilikos, *Bidimensional parameters and local treewidth*, in Proceedings of the 6th Latin American Theoretical Informatics Symposium (LATIN 2004), Lecture Notes in Comput. Sci. 2976, Springer, Berlin, 2004, pp. 109–118.

[12] E. D. Demaine, F. V. Fomin, M. T. Hajiaghayi, and D. M. Thilikos, *Fixed-parameter algorithms for the (k, r)-center in planar graphs and map graphs*, in Proceedings of the 30th International Colloquium on Automata, Languages, and Programming (ICALP 2003), Lecture Notes in Comput. Sci. 2719, Springer, Berlin, 2003, pp. 829–844.

[13] E. D. Demaine, F. V. Fomin, M. T. Hajiaghayi, and D. M. Thilikos, *Subexponential parameterized algorithms on graphs of bounded genus and H-minor-free graphs*, in Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms (SODA 2004), New Orleans, LA, 2004, pp. 823–832.

[14] E. D. Demaine and M. T. Hajiaghayi, *Equivalence of local treewidth and linear local treewidth and its algorithmic applications*, in Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms (SODA 2004), New Orleans, LA, 2004, pp. 833–842.

[15] E. D. Demaine, M. T. Hajiaghayi, and D. M. Thilikos, *Exponential speedup of fixed parameter algorithms on $K_{3,3}$-minor-free or K_5-minor-free graphs*, in Proceedings of the 13th International Symposium on Algorithms and Computation (ISAAC 2002), Lecture Notes in Comput. Sci. 2518, Springer, Berlin, 2002, pp. 262–273.

[16] E. D. Demaine, M. T. Hajiaghayi, and D. M. Thilikos, *The bidimensional theory of bounded-genus graphs*, in Proceedings of the 29th International Symposium on Mathematical Foundations of Computer Science (MFCS 2004), Lecture Notes in Comput. Sci. 3153, Springer, Berlin, 2004, pp. 191–203.

[17] R. Diestel, *Graph Theory*, Springer-Verlag, New York, 2000.

[18] R. Diestel, T. R. Jensen, K. Y. Gorbunov, and C. Thomassen, *Highly connected sets and the excluded grid theorem*, J. Combin. Theory Ser. B, 75 (1999), pp. 61–73.

[19] R. G. Downey and M. R. Fellows, *Parameterized Complexity*, Springer-Verlag, New York, 1999.

[20] J. Ellis, H. Fan, and M. Fellows, *The dominating set problem is fixed parameter tractable for graphs of bounded genus*, in Proceedings of the 8th Scandinavian Workshop on Algorithm Theory (SWAT 2002), Lecture Notes in Comput. Sci. 2368, Springer, Berlin, 2002, pp. 180–189.

[21] D. Eppstein, *Diameter and treewidth in minor-closed graph families*, Algorithmica, 27 (2000), pp. 275–291.

[22] J. Flum and M. Grohe, *Fixed-parameter tractability, definability, and model-checking*, SIAM J. Comput. 31 (2001), pp. 113–145.

[23] F. V. Fomin and D. M. Thilikos, *Dominating sets in planar graphs: Branch-width and exponential speed-up*, in Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms (SODA 2003), Baltimore, MD, 2003, pp. 168–177.

[24] F. V. Fomin and D. M. Thilikos, *Dominating sets and local treewidth*, in Proceedings of the 11th European Symposium on Algorithms (ESA 2003), Lecture Notes in Comput. Sci. 2832, Springer, Berlin, 2003, pp. 221–229.

[25] F. V. Fomin and D. M. Thilikos, *A simple and fast approach for solving problems on planar graphs*, in Proceedings of the 21st International Symposium on Theoretical Aspects of Computer Science (STACS 2004), Lecture Notes in Comput. Sci. 2996, Springer, Berlin, 2004, pp. 56–67.

[26] F. V. Fomin and D. M. Thilikos, *Fast parameterized algorithms for graphs on surfaces: Linear kernel and exponential speed-up*, in Proceedings of the 31th International Colloquium on Automata, Languages, and Programming (ICALP 2004), Lecture Notes in Comput. Sci. 3142, Springer, Berlin, pp. 581–592.

[27] M. Frick and M. Grohe, *Deciding first-order properties of locally tree-decomposable graphs*, J. ACM, 48 (2001), pp. 1184–1206.

[28] H. Gaifman, *On local and nonlocal properties*, in Proceedings of the Herbrand Symposium (Marseilles, 1981), North-Holland, Amsterdam, 1982, pp. 105–135.

[29] M. Grohe, *Local tree-width, excluded minors, and approximation algorithms*, Combinatorica, 23 (2003), pp. 613–632.

[30] I. Kanj and L. Perković, *Improved parameterized algorithms for planar dominating set*, in Proceedings of the 27th International Symposium on Mathematical Foundations of Computer Science (MFCS 2002), Lecture Notes in Comput. Sci. 2420, Springer, Berlin, 2002, pp. 399–410.

[31] N. Robertson and P. D. Seymour, *Graph minors. II. Algorithmic aspects of tree-width*, J. Algorithms, 7 (1986), pp. 309–322.

[32] N. Robertson and P. D. Seymour, *Graph minors. V. Excluding a planar graph*, J. Combin. Theory Ser. B, 41 (1986), pp. 92–114.

SIAM J. MATRIX ANAL. APPL.
Vol. 18, No. 3, pp. 512–524

PARTITION-OPTIMIZATION WITH SCHUR CONVEX SUM OBJECTIVE FUNCTIONS*

FRANK K. HWANG† AND URIEL G. ROTHBLUM‡

Abstract. We study optimization problems over partitions of the finite set $N = \{1, \dots, n\}$, where each element i in the partitioned set N is associated with a real number θ^i and the objective associated with a partition $\pi = (\pi_1, \dots, \pi_p)$ has the form $F(\pi) = f(\theta_\pi)$, where $\theta_\pi = (\sum_{i\in\pi_1} \theta^i, \dots, \sum_{i\in\pi_p} \theta^i)$. When F is to be either maximized or minimized, we obtain conditions that allow for simple constructions of partitions that are uniformly optimal for all Schur convex functions f.

Key words. partitions, optimization, Schur-convexity

AMS subject classifications. 90CZ7, 26B25, 90C25

DOI. 10.1137/S0895480198347167

1. Introduction.

We consider partitions of the finite set $N = \{1, \dots, n\}$ into nonempty parts. When a corresponding partition π has p parts, we refer to it as a *p-partition* and denote it by $\pi = (\pi_1, \dots, \pi_p)$; also, we refer to the vector $(|\pi_1|, \dots, |\pi_p|)$ as the *shape* of the partition π.

Throughout, we assume that each element i in the partitioned set N is associated with a real number θ^i and, by possibly permuting the elements of N, we may assume that $\theta^1 \le \theta^2 \le \cdots \le \theta^n$. A partition is called *consecutive* if (after the possible permutation of N) the elements in each part are consecutive integers.

We consider optimization problems (maximization and minimization) over families of partitions where the objective value $F(\pi)$ associated with a partition π is given through a real-valued function f that is defined on R^p and $F(\pi) = f\big(\sum_{i\in\pi_1} \theta^i, \dots, \sum_{i\in\pi_\pi} \theta^i\big)$; such partitioning problems are called *sum partitioning problems*. Of particular interest are *constrained shape*, *bounded-shape*, and *single-shape* problems, where the underlying sets of partitions are defined, respectively, by restrictions, bounds, and specification on the shape of partitions. For many applications of partitioning problems see, for example, [1, 2, 3, 4].

An important tool for studying optimization problems is the identification of properties that are satisfied by optimal solutions. In particular, determining the existence of optimal solutions with a particular property allows one to restrict the search for an optimal solution to a smaller class of feasible solutions, namely, those that satisfy the property. For partitioning problems, consecutiveness is a particularly valuable property, as the number of p-partitions with prescribed shape is exponential in n, while the number of consecutive p-partitions is $p!$. Conditions on the function f that suffice for the optimality of consecutive partitions have been studied extensively in the literature. Hwang and Rothblum [3] introduced a class of functions called *asymmetric Schur convex functions*, unifying classical (quasi) convexity and Schur

*Received by the editors November 12, 1998; accepted for publication (in revised form) December 1, 2003; published electronically February 25, 2005.
http://www.siam.org/journals/sidma/18-3/34716.html

†Department of Applied Mathematics, Chiaotung University, Hsinchu, 30045 Taiwan, Republic of China (fhwang@math.nctu.edu.tw).

‡Faculty of Industrial Engineering and Management, Technion—Israel Institute of Technology, Haifa 32000, Israel (rothblum@ie.technion.ac.il). The research of this author was supported by a grant from the Israel Science Foundation and by the E. and J. Bishop Research Fund at Technion.

convexity; asymmetric Schur convexity was shown in Gao, Hwang, Li, and Rothblum [1] to be sufficient for optimality of consecutive partitions, generalizing many earlier results.

The goal of the current paper is to study bounded-shape partitioning problems where the function f is Schur convex and the objective is to either maximize F or to minimize it. We identify conditions that allow for explicit solution of such problems without the need to scan through all consecutive partitions. Under these conditions, optimality turns out to be invariant of the particular (Schur convex) function f. It follows that, depending on whether the objective function is to be maximized or minimized, the vector associated with an invariant optimal partition must majorize or be majorized by the vectors associated with all other feasible partitions (see section 2 for formal definitions). For bounded-shape maximization problems, we explicitly construct an invariant consecutive optimal partition when the ranking of the coordinates of the lower bounds on the part-sizes is consistent with that of the upper bounds and, in addition, the θ^i's have the uniform sign; further, we demonstrate that if either of these two conditions is dropped, an invariant optimal partition need not exist. For bounded-shape minimization problems, we explicitly construct an invariant solution when all the θ^i's are 1, that is, when the vector associated with a partition is the shape of the partition; further, we show via an example that this restriction cannot be relaxed. Our proof for minimization problems first identifies a vector which is majorized by all vectors that satisfy prescribed lower and upper bounds and have a prescribed coordinate-sum. We then show that when the bounds and the prescribed coordinate-sum are integers, the majorized vector can be rounded up/down to an integer vector that is majorized by all corresponding integer vectors. Results of Veinott [7] concern the construction of majorized vectors in a more general context of network flows, and his proofs depend on yet unpublished results in [8]. The proofs we derive herein are self-contained and simpler.

2. Preliminaries. Throughout, we let n be a positive integer and $N \equiv \{1, \dots, n\}$. A *partition* (of N) is an ordered collection of sets $\pi = (\pi_1, \dots, \pi_p)$, where $\pi_1, \dots, \pi_p$ are disjoint *nonempty* subsets of N whose union is N. In this case we refer to p as the *size* of π and to the sets $\pi_1, \dots, \pi_p$ as the *parts of* π. Also, if the number of elements in the parts of the partition $\pi = (\pi_1, \dots, \pi_p)$ are $n_1, \dots, n_p$, respectively, we refer to $(n_1, \dots, n_p)$ as the *shape* of π; of course, in this case $\sum_{j=1}^{p} n_j = |N| = n$. We sometimes refer to *p-partitions* or to $(n_1, \dots, n_p)$*-partitions* as partitions of size p or of shape $(n_1, \dots, n_p)$, respectively. A partition is called *consecutive* if its parts consist of consecutive integers, that is, if there is an enumeration of its parts, say, $\pi_{j_1}, \dots, \pi_{j_p}$, such that for $t = 1, \dots, p$ and corresponding positive integers $n_{j_1}, \dots, n_{j_p}$, $\pi_{j_t} = \left\{\sum_{s=1}^{t-1} n_{j_s} + 1, \dots, \sum_{s=1}^{t} n_{j_s}\right\}$.

We assume that each element i in the given partitioned set N is associated with a real number θ^i and, without loss of generality,

$$\theta^1 \le \theta^2 \le \cdots \le \theta^n. \tag{2.1}$$

We denote by θ the vector $(\theta^1, \dots, \theta^n) \in R^n$. Also, for a subset $S \subseteq \{1, \dots, n\}$ we define the *S-summation scalar* θ_S by $\theta_S \equiv \sum_{i \in S} \theta^i$. For a p-partition $\pi = (\pi_1, \dots, \pi_p)$ we define the *π-summation-vector* θ_π by $\theta_\pi \equiv (\theta_{\pi_1}, \dots, \theta_{\pi_p}) \in R^p$.

Throughout this paper we let p be a fixed positive integer. Given a real-valued function F over a set Π of p-partitions, we consider the problem of maximizing F over

Π. The problem is called *sum-partitioning* if there is a function $f{:}R^p \to R$ such that

$$F(\pi) = f(\theta_\pi) \quad \text{for each } p\text{-partition } \pi. \tag{2.2}$$

We refer to *single-shape*, *bounded-shape* and *constrained-shape* problems as partitioning problems with Π as the set of partitions with a prescribed shape, with a shape that satisfies the prescribed lower and upper bound and with a shape in a prescribed set, respectively. For constrained-shape problems the set of partitions is defined through a set Γ of positive integer p-vectors with the coordinate-sum n. For bounded-shape problems, Γ is defined by two positive integer p-vectors L and U satisfying $L \le U$ and $\sum_{j=1}^p L_j \le |N| \le \sum_{j=1}^p U_j$; we then write $\Gamma^{(L,U)}$ for Γ and $\Pi^{(L,U)}$ for the corresponding set of partitions. Finally, for single-shape problems, Γ is defined by a single positive integer p-vector $(n_1, \dots, n_p)$ satisfying $\sum_{j=1}^p n_j = |N|$; we then write $\Gamma^{(n_1,\dots,n_p)}$ for Γ and $\Pi^{(n_1,\dots,n_p)}$ for the corresponding set of partitions.

For a vector $x \in R^n$ and $k = 1, \dots, n$, let $x_{[k]}$ be the kth largest coordinate of x. We say that a vector $a \in R^p$ *majorizes* a vector $b \in R^p$, written $a \gg b$, if

$$\sum_{i=1}^k a_{[i]} \ge \sum_{i=1}^k b_{[i]} \quad \text{for all} \quad k = 1, \dots, p \tag{2.3}$$

and

$$\sum_{i=1}^p a_{[i]} = \sum_{i=1}^p b_{[i]}; \tag{2.4}$$

we note that (2.3) and (2.4) are, respectively, equivalent to

$$\max_{|I|=k} \sum_{i \in I} a_i \ge \max_{|I|=k} \sum_{i \in I} b_i \quad \text{for all} \quad k = 1, \dots, p \tag{2.3$'$}$$

and

$$\sum_{i=1}^p a_i = \sum_{i=1}^p b_i. \tag{2.4$'$}$$

We say that *a strictly majorizes b* if a majorizes b but does not majorize a.

A real-valued function f on a subset B of R^p is called *Schur convex* if $f(a) \ge f(b)$ for all $a, b \in B$ satisfying $a \gg b$, that is, if f is order-preserving with respect to the partial order majorization. The function f is called *strictly Schur convex* if it is Schur convex and $f(a) > f(b)$ for all $a, b \in B$ for which a strictly majorizes b. For example, a real-valued function f on R^p with $f(x) = \sum_{j=1}^p g(x_j)$, where g is a (strictly) convex real-valued function on R, is known to be (strictly) Schur convex (see [6]); such functions are called *separable (strictly) Schur convex*. We say that f is (strictly) Schur concave if $-f$ is (strictly) Schur convex.

We say that a p-vector z is a *majorizing* vector in a finite set $\Lambda \subseteq R^p$ if $z \in \Lambda$ and z majorizes every vector in Λ; we say that z is a *minorizing* vector in Λ if $z \in \Lambda$ and z is majorized by every vector in Λ. Since majorization is a partial order that does not provide comparisons for all pairs of vectors, majorizing and minorizing vectors need not exist.

For $j = 1, \dots, p-1$, let $f^{(j)}$ be the real-valued function on R^p with $f^{(j)}(x) = \max_{\{I \subseteq \{1,\dots,p\}:|I|=j\}} \sum_{u \in I} x_u$ for each $x \in R^p$ (these functions are convex as the

maximum of linear functions). The characterization of majorization through (2.3′)–(2.4′) shows that a finite set $\Lambda \subseteq R^p$ contains a majorizing/minorizing vector if and only if the functions $f^{(1)}, \dots, f^{(p-1)}$ are simultaneously maximized/minimized over Λ and, in addition, all vectors in Λ have a common coordinate-sum.

3. Maximization problems with f Schur convex. In this section we focus on maximization problems where the function f is Schur convex.

Let Π be a set of partitions. We say that a partition π^* is *shape-majorizing in* Π if $\pi^* \in \Pi$ and the shape of π^* majorizes the shape of every other partition in Π; when Π is defined as the set of partitions with its shape in a prescribed set Γ, π^* is shape-majorizing if and only if its shape is a majorizing vector in Γ. The next result shows that if Γ has a majorizing vector, a shape-majorizing partition exists.

PROPOSITION 3.1. *Suppose Γ is a set of positive integer p-vectors with coordinate-sum n and Π is the set of partitions with its shape in Γ. If $(n_1, \dots, n_p)$ is a majorizing vector in Γ, then there exists a consecutive shape-majorizing partition in Π.*

Proof. The conclusion of the lemma follows from the existence of consecutive partitions with any prescribed shape (in fact, the consecutive partitions with prescribed shape are in one-to-one correspondence with the permutations over $\{1, \dots, p\}$). □

We say that θ is *sign-uniform* if it is either nonpositive or nonnegative. The next result shows that this condition together with the assumptions of Proposition 3.1 facilitate a uniform solution for sum-partitioning problems under all Schur convex functions f. This is accomplished by first determining a majorizing shape and then assigning the elements to parts greedily (where greedily has different meanings for the case where $\theta \le 0$ and for the case where $\theta \ge 0$).

THEOREM 3.2. *Suppose f is Schur convex, Γ is a set of positive integer p-vectors with the coordinate-sum n, $(n_1, \dots, n_p)$ is a majorizing vector in Γ with $n_1 \le \cdots \le n_p$, and Π is the (constrained-shape) set of partitions with its shape in Γ.*

(i) *If $\theta \le 0$, then the (consecutive) p-partition π^- with $\pi_j^- = \{n - \sum_{u=1}^{j} n_u + 1, \dots, n - \sum_{u=1}^{j-1} n_u\}$ for $j = 1, \dots, p$ is in Π and maximizes $F(.)$ over Π.*

(ii) *If $\theta \ge 0$, then the (consecutive) p-partition π^+ with $\pi_j^+ = \{\sum_{u=1}^{j-1} n_u + 1, \dots, \sum_{u=1}^{j} n_u\}$ for $j = 1, \dots, p$ is in Π and maximizes $F(.)$ over Π.*

Further, if f is strictly Schur convex, the inequalities of (2.1) *hold strictly, and the θ^i's are nonzero, then π^- and π^+ are, respectively, the only optimal partitions.*

Proof. We first consider the case where $\theta \ge 0$. Since the shape of π^+ is $(n_1, \dots, n_p) \in \Gamma$, then π^+ is shape-majorizing in Π. Also, from $n_1 \le \cdots \le n_p$ we have that $|\pi_1^+| \le \cdots \le |\pi_p^+|$. These properties of π^+ ensure that for each $\pi \in \Pi$, $j \in \{1, \dots, p\}$ and enumeration $u_1, \dots, u_p$ of the elements $1, \dots, p$,

$$\begin{aligned} \sum_{s=1}^{j} |\pi_{u_s}| &\le \max_{\{I \subseteq \{1,\dots,p\}:|I|=j\}} \sum_{u \in I} |\pi_u| \le \max_{\{I \subseteq \{1,\dots,p\}:|I|=j\}} \sum_{u \in I} |\pi_u^+| \\ &= \sum_{u=p-j+1}^{p} |\pi_u^+| = \sum_{u=p-j+1}^{p} n_u. \end{aligned} \tag{3.1}$$

We conclude from (3.1), (2.1), the nonnegativity of the θ^i's, and the definition of π^+ that

$$\sum_{s=1}^{j} (\theta_\pi)_{u_s} = \sum_{i \in \pi_{u_1} \cup \cdots \cup \pi_{u_j}} \theta^i \le \sum_{i=n_1+\cdots+n_{p-j}+1}^{n} \theta^i = \sum_{u=p-j+1}^{p} (\theta_{\pi^+})_u, \tag{3.2}$$

with equality holding when $j = p$. Since π^+ is in Π, it also satisfies (3.2). Applying (3.2) to π^+ and to π, we conclude that

$$\max_{\{I\subseteq\{1,\dots,p\}:|I|=j\}} \sum_{u\in I} (\theta_\pi)_u \le \sum_{i=n_1+\cdots+n_{p-j}+1}^{n} \theta^i = \max_{\{I\subseteq\{1,\dots,p\}:|I|=j\}} \sum_{u\in I} (\theta_{\pi^+})_u \tag{3.3}$$

with equality holding when $j = p$. Thus, θ_{π^+} majorizes θ_π and, therefore, the Schur convexity of f implies that $F(\pi^+) = f(\theta_{\pi^+}) \ge f(\theta_\pi) = F(\pi)$.

Next, assume that $\theta \le 0$. Since the shape of π^- is $(n_1, \dots, n_p) \in \Gamma$, π^- is also shape-majorizing in Π. Also, from $n_1 \le \cdots \le n_p$ we have that $|\pi_1^-| \le \cdots \le |\pi_p^-|$. These properties of π^- ensure that for each $\pi \in \Pi$, $j \in \{1, \dots, p\}$ and enumeration $u_1, \dots, u_p$ of the elements $1, \dots, p$,

$$\begin{aligned} \sum_{s=j+1}^{p} \left|\pi_{u_s}\right| &\le \max_{\{I\subseteq\{1,\dots,p\}:|I|=p-j\}} \sum_{u\in I} \left|\pi_u\right| \le \max_{\{I\subseteq\{1,\dots,p\}:|I|=p-j\}} \sum_{u\in I} \left|\pi_u^-\right| \\ &= \sum_{u=j+1}^{p} \left|\pi_u^-\right| = \sum_{u=j+1}^{p} n_u, \end{aligned} \tag{3.4}$$

and, therefore,

$$\sum_{s=1}^{j} \left|\pi_{u_s}\right| = n - \sum_{s=j+1}^{p} \left|\pi_{u_s}\right| \ge n - \sum_{u=j+1}^{p} \left|\pi_u^-\right| = \sum_{u=1}^{j} n_u. \tag{3.5}$$

From (2.1), (3.5), the nonpositivity of the θ^i's, and the definition of π^-, we see that

$$\sum_{s=1}^{j} (\theta_\pi)_{u_s} = \sum_{i\in\pi_{u_1}\cup\cdots\cup\pi_{u_j}} \theta^i \le \sum_{i=n-(n_1+\cdots+n_j)+1}^{n} \theta^i = \sum_{u=1}^{j} (\theta_{\pi^-})_u, \tag{3.6}$$

with equality holding when $j = p$. Since π^- is in $\Pi^{(n_1,\dots,n_p)}$, it also satisfies (3.6). Applying (3.6) to π^- and to π, we conclude that

$$\max_{\{I\subseteq\{1,\dots,p\}:|I|=j\}} \sum_{u\in I} (\theta_\pi)_u \le \sum_{i=n-(n_1+\cdots+n_j)+1}^{n} \theta^i = \max_{\{I\subseteq\{1,\dots,p\}:|I|=j\}} \sum_{u\in I} (\theta_{\pi^-})_u \tag{3.7}$$

with equality holding when $j = p$. Thus, θ_{π^-} majorizes θ_π and, therefore, the Schur convexity of f implies that $F(\pi^-) = f(\theta_{\pi^-}) \ge f(\theta_\pi) = F(\pi)$.

Finally, if the inequalities of (2.1) hold strictly and the θ^i's are nonzero, then for each $\pi \ne \pi^+$, (3.4) implies that (3.5) holds as a strict inequality for at least one j; thus, θ_{π^+} strictly majorizes θ_π. Consequently, if f is strictly Schur convex, we have that $F(\pi^+) = f(\theta_{\pi^+}) > f(\theta_\pi) = F(\pi)$. A similar argument shows that if the inequalities of (2.1) hold strictly, the θ^i's are nonzero, and f is strictly Schur convex, then $F(\pi^-) = f(\theta_{\pi^-}) > f(\theta_\pi) = F(\pi)$. □

Solution of constrained-shape partitioning problems with f Schur convex, sign-uniform θ, and given majorizing shape. Let Γ be a set of positive integer p-vectors with coordinate-sum n and let $(n_1, \dots, n_p)$ be a majorizing vector in Γ with $n_1 \le \cdots \le n_p$. Also, assume the $\theta^1, \dots, \theta^n$ are given and satisfy (2.1). Of course, if either the θ^i's and/or the n_u's are not ranked a priori, one can sort them

and renumber indices in time $O[n(lg\, n)]$ and/or $O[p(lg\, p)]$, respectively. Once the indices are renumbered, Theorem 3.2 provides an explicit solution of the partitioning problem when either $\theta \geq 0$ or $\theta \leq 0$; only the partial sums of the n_j's are needed, and these can be determined with p additions and the associated vector can be determined with, at most, n additions.

Next we explain how the "expensive" sorting of the θ^i's can be reduced. Suppose a sorting of $n_1, \ldots, n_p$ is executed if needed (requiring time $O[p(lg\, p)]$ comparisons), and an index-enumeration $j_1, \ldots, j_p$ satisfying $n_{j_1} \leq n_{j_2} \leq \cdots \leq n_{j_p}$ becomes available. It is then not necessary to fully sort $\theta^1, \ldots, \theta^n$ in order to determine the optimal partition; all that is needed is to determine the set of n_{j_1}-smallest coordinates of θ, the next n_{j_2}-smallest coordinates, and so on. This block-sorting can be executed with $O(pn)$ comparisons [5], yielding an improved complexity bound of $O(pn)$. If the data is given with (2.1) in force, Theorem 3.2 provides an explicit solution of the partitioning problem requiring only the sorting of $n_1, \ldots, n_p$; so, in this case the problem is solvable in time $O[p(lg\, p)]$.

Theorem 3.2 yields an explicit solution to partitioning problems when a majorizing shape within the set of allowable shapes Γ is available. Such a shape is trivially available when Γ contains a single shape, e.g., when either $\sum_{j=1}^{p} L_j = n$ or $\sum_{j=1}^{p} U_j = n$. Next we obtain a sufficient condition for the existence of a majorizing shape in nondegenerate bounded-shape problems; further, under this condition the majorizing shape is easily computable.

LEMMA 3.3. *Let L and U be positive integer p-vectors satisfying $L \leq U$ and $\sum_{j=1}^{p} L_j < n < \sum_{j=1}^{p} U_j$. Then there exists an index $j \in \{1, \ldots, p\}$ with $\sum_{u=1}^{j} L_u + \sum_{u=j+1}^{p} U_u = \sum_{u=1}^{p} U_u - \sum_{u=1}^{j} (U_u - L_u) \leq n$; further, if j^* is the first such index and $\mu^* \equiv n - \sum_{u=1}^{j^*-1} L_u - \sum_{u=j^*+1}^{p} U_u$, then $(n_1^*, \ldots, n_p^*) \equiv (L_1, \ldots, L_{j^*-1}, \mu^*, U_{j^*+1}, \ldots, U_p) \in \Gamma^{(L,U)}$, and*

$$\sum_{u=1}^{k} n_u^* = \max \left\{ \sum_{u=1}^{k} L_u, n - \sum_{u=k+1}^{p} U_u \right\} \quad \text{for } k = 1, \ldots, p. \tag{3.8}$$

Moreover, if

$$L_1 \leq L_2 \leq \cdots \leq L_p \tag{3.9}$$

and

$$U_1 \leq U_2 \leq \cdots \leq U_p, \tag{3.10}$$

then $n_1^* \leq \cdots \leq n_p^*$ and $(n_1^*, \ldots, n_p^*)$ majorizes every vector in $\Gamma^{(L,U)}$.

Proof. The existence of an index $j \in \{1, \ldots, p\}$ with $\sum_{u=1}^{j} L_u + \sum_{u=j+1}^{p} U_u = \sum_{u=1}^{p} U_u - \sum_{u=1}^{j} (U_u - L_u) \leq n$ is immediate from the fact that $\sum_{u=1}^{p} U_u > n$ and $\sum_{u=1}^{p} U_u - \sum_{u=1}^{p} (U_u - L_u) = \sum_{u=1}^{p} L_u < n$. With j^* as the first such index and with the definition of μ^* and $(n_1^*, \ldots, n_p^*)$ as in the statement of the lemma, we clearly have that $L_{j^*} \leq \mu^* < U_{j^*}$ and $(n_1^*, \ldots, n_p^*) \in \Gamma^{(L,U)}$. Also, from the definition of j^* and n_j^*'s we have that

$$\sum_{u=1}^{k} n_u^* = \begin{cases} \sum_{u=1}^{k} L_u > n - \sum_{u=k+1}^{p} U_u & \text{if } k < j^*, \\ n - \sum_{u=k+1}^{p} u_u^* \geq \sum_{u=1}^{k} L_u & \text{if } k \geq j^*. \end{cases}$$

When either $k < j^*$ or $k \geq j^*$, we have that (3.8) holds.

Next, assume that (3.9) and (3.10) hold. To verify that the coordinates of $(n_1^*, \ldots, n_p^*)$ are nondecreasing, observe that if $t < j^*$, we have $n_t^* = L_t \leq L_{t+1} \leq n_{t+1}^*$, and if $t \geq j^*$, we have $n_t^* \leq U_t \leq U_{t+1} = n_{t+1}^*$. Next, let I be a subset of $\{1, \ldots, p\}$ and let $(n_1, \ldots, n_p)$ be a vector in $\Gamma^{(L,U)}$. The complement of I within $\{1, \ldots, p\}$ will be denoted I^c. Since $\sum_{u \in I} n_u \leq \sum_{u \in I} U_u$ and $n - \sum_{u \in I} n_u = \sum_{u \in I^c} n_u \geq \sum_{u \in I^c} L_u$, we have that

$$\sum_{u \in I} n_u \leq \min \left\{ n - \sum_{u \in I^c} L_u, \sum_{u \in I} U_u \right\} \leq \min \left\{ n - \sum_{u=1}^{p-|I|} L_u, \sum_{u=p-|I|+1}^{p} U_u \right\}, \tag{3.11}$$

where (3.9)–(3.10) are used for the second inequality in (3.11). Also, for each $j = 1, \ldots, p-1$, we get from (3.8) (with $k = p - j$) that

$$\begin{aligned} \sum_{u=p-j+1}^{p} n_u^* &= n - \sum_{u=1}^{p-j} n_u^* = n - \max \left\{ \sum_{u=1}^{p-j} L_u, n - \sum_{u=p-j+1}^{p} U_u \right\} \\ &= \min \left\{ n - \sum_{u=1}^{p-j} L_u, \sum_{u=p-j+1}^{p} U_u \right\}. \end{aligned} \tag{3.12}$$

Since $(n_1^*, \ldots, n_p^*) \in \Gamma^{(L,U)}$, (3.11) applies to $(n_1^*, \ldots, n_p^*)$. It follows from (3.11) applied to $(n_1, \ldots, n_p)$ and to $(n_1^*, \ldots, n_p^*)$ and from (3.12) that, for $j = 1, \ldots, p-1$,

$$\begin{aligned} \max_{\{I \subseteq \{1, \ldots, p\} : |I| = j\}} \sum_{u \in I} n_u &\leq \min \left\{ n - \sum_{u=1}^{p-j} L_u, \sum_{u=p-j+1}^{p} U_u \right\} = \sum_{u=p-j+1}^{p} n_u^* \\ &= \max_{\{I \subseteq \{1, \ldots, p\} : |I| = j\}} \sum_{u \in I} n_u^*, \end{aligned}$$

verifying that $(n_1^*, \ldots, n_p^*)$ majorizes $(n_1, \ldots, n_p)$. □

Next we state two immediate conclusions from Theorem 3.2 and Lemma 3.3.

THEOREM 3.4. *Suppose f is Schur convex and L and U are positive integer p-vectors satisfying $L \leq U$, $\sum_{j=1}^{p} L_j < n < \sum_{j=1}^{p} U_j$,* (3.9)*, and* (3.10)*. Let $(n_1^*, \ldots, n_p^*)$ be as in Lemma* 3.3*.*

(i) *If $\theta \leq 0$, then the (consecutive) p-partition π^- with $\pi_j^- = \left\{ n - \sum_{u=1}^{j} n_u^* + 1, \ldots, n - \sum_{u=1}^{j-1} n_u^* \right\}$ for $j = 1, \ldots, p$ is in $\Pi^{(L,U)}$ and maximizes $F(.)$ over $\Pi^{(L,U)}$.*

(ii) *If $\theta \geq 0$, then the (consecutive) p-partition π^+ with $\pi_j^+ = \left\{ \sum_{u=1}^{j-1} n_u^* + 1, \ldots, \sum_{u=1}^{j} n_u^* \right\}$ for $j = 1, \ldots, p$ is in $\Pi^{(L,U)}$ and maximizes $F(.)$ over $\Pi^{(L,U)}$.*

Further, if f is strictly Schur convex, the inequalities of (2.1) *hold strictly, and the θ^i's are nonzero, then π^- and π^+ are, respectively, the only optimal partitions.*

Under the assumptions of Theorem 3.4, the solution method discussed following Theorem 3.2 applies; further, Lemma 3.3 shows that the computation of the majorizing-shape vector $(n_1^*, \ldots, n_p^*)$ is available with $O(p)$ arithmetic operations.

We say that two vectors, L and U, in R^p are *consistent* if there exists a permutation $(\{u_1\}, \ldots, \{u_p\})$ such that the vectors $\left(L_{u_1}, \ldots, L_{u_p}\right)$ and $\left(U_{u_1}, \ldots, U_{u_p}\right)$ satisfy (3.9)–(3.10). Corollary 3.4 implies that when f is Schur convex, L and U are consistent positive integer p-vectors satisfying $L \leq U$ and $\sum_{j=1}^{p} L_j < n < \sum_{j=1}^{p} U_j$,

and θ is sign-uniform, there exists a majorizing vector in $\Gamma^{(L,U)}$ and a (consecutive, shape-majorizing) partition in $\Pi^{(L,U)}$ which is optimal uniformly under all Schur convex functions f. Further, such a partition is easily computable by first (jointly) sorting the L_u's and U_u's and then selecting either of the two partitions constructed in Theorem 3.2.

Two important cases for which the assumptions of Lemma 3.3 and Theorem 3.4 apply are as follows:

(i) single-shape problem, where the coordinates of a single prescribed shape, say, $(n_1, \ldots, n_p)$, can be ranked and permuted to satisfy the monotonicity assumption (3.9)–(3.10) with $L = U = (n_1, \ldots, n_p)$, and

(ii) uniform bounded shape problem, where L_u's and U_u's are, respectively, independent of u.

The next two examples demonstrate, respectively, that neither the consistency of L and U nor the sign-uniformity of θ can be removed from Corollary 3.5.

EXAMPLE I. *Suppose* $p = 3$, $n = 9$, $L_1 = 1$, $L_2 = L_3 = 2$, $U_1 = 5$, $U_2 = U_3 = 4$, *and* $\theta^i = 1$ *for* $i = 1, \ldots, 9$. *With* $\Pi \equiv \Pi^{(L,U)}$, $\max_{\pi\in\Pi} \max_u (\theta_\pi)_u = 5$, *and the maximum is realized by exactly the partitions with shape* $(5, 2, 2)$. *However,* $\max_{\pi\in\Pi} \max_{u,v}[(\theta_\pi)_u + (\theta_\pi)_v] = 8$, *and the maximum is realized by exactly the partitions with shape* $(1, 4, 4)$. *Thus, there is no shape-majorizing partition in* $\Pi^{(L,U)}$. *It is easily noted that* $\Gamma^{(L,U)}$ *does not have a vector which majorizes all other vectors in the set.*

To see that no partition is optimal uniformly under all (separable) Schur convex functions f, *let* f_1 *and* f_2 *be the (separable, strictly Schur convex) functions with* $f_1(x) = \sum_{u=1}^3 |x_u|^3$ *and* $f_2(x) = \sum_{u=1}^3 |x_u - 4|^3$. *The shapes in* $\Gamma^{(L,U)}$ *are* $(5, 2, 2)$, $(4, 3, 2)$, $(4, 2, 3)$, $(3, 4, 2)$, $(3, 3, 3)$, $(3, 2, 4)$, $(2, 4, 3)$, $(2, 3, 4)$, *and* $(1, 4, 4)$; *the values of these vectors under* (f_1, f_2) *are, respectively,* $(141, 17)$, $(99, 9)$, $(99, 9)$, $(99, 9)$, $(81, 3)$, $(99, 9)$, $(99, 9)$, $(99, 9)$, *and* $(129, 27)$. *So, the optimal partitions with the objective defined by* f_1 *and* f_2 *are, respectively, those with shape* $(5, 2, 2)$ *and those with shape* $(1, 4, 4)$.

EXAMPLE II. *Suppose* $p = 3$, $n = 6$, $n_j = j$ *for* $j = 1, 2, 3$, $\theta^i = -1$ *for* $i = 1, 2, 3$, *and* $\theta^i = 1$ *for* $i = 4, 5, 6$. *With* $\Pi \equiv \Pi^{(1,2,3)}$, $\max_{\pi\in\Pi} \max_u (\theta_\pi)_u = 3$, *and the maximum is realized by the partitions with* $\pi_3 = \{4, 5, 6\}$ *and only by those. However,* $\max_{\pi\in\Pi} \max_{u,v}[(\theta_\pi)_u + (\theta_\pi)_v] = 3$, *and the maximum is realized by the partition with* $\pi_3 = \{1, 2, 3\}$ *and only by them. Thus, there is no partition* π' *in* Π *with* $\theta_{\pi'}$ *majorizing each of the vectors associated with a partition* π *in* Π. *To see that no partition is optimal uniformly under all Schur convex functions* f, *let* f_1 *and* f_2 *be the (separable, strictly Schur convex) functions with* $f_1(x) = \sum_{u=1}^3 |x_u + 3|^3$ *and* $f_2(x) = \sum_{u=1}^3 |x_u - 3|^3$; *the optimal partitions with* f_1 *and* f_2 *are, respectively, precisely the partitions* π *with* $\pi_3 = \{4, 5, 6\}$ *and those with* $\pi_3 = \{1, 2, 3\}$.

4. Minimization problems with *f* Schur convex. In this section we focus on minimization problems where the function f is Schur convex. The main result of this section can be derived from more general results of Veinott [6, Theorem 2, p. 554] which depend on (yet unpublished) results of [8]; the proofs provided herein are self-contained and more elementary.

Let Π be a set of partitions. We say that a partition π^* is *shape-minorizing in* Π if $\pi^* \in \Pi$ and the shape of π^* is majorized by the shape of every other partition in Π; when Π is defined as the set of partitions with its shape in a prescribed set Γ, π^* is shape-minorizing if and only if its shape is a minorizing vector in Γ. The next result shows that if Γ has a minorizing vector, a shape-minorizing partition exists.

PROPOSITION 4.1. *Suppose* Γ *is a set of positive integer* p*-vectors with a coordinate-sum* n *and* Π *is the set of partitions with its shape in* Γ*. If* $(n_1, \ldots, n_p)$ *is a minorizing vector in* Γ*, then there exists a consecutive shape-minorizing partition in* Π*.*

Proof. As for Proposition 3.1, the conclusion follows from the existence of consecutive partitions with any prescribed shape. □

The next result is in the spirit of Theorem 3.2 with minimization replacing maximization—it provides conditions for the existence of a uniform solution to constrained-shape partitioning problems under the assumptions of Proposition 4.1. But here, more restrictive conditions than sign-uniformity of θ are required.

THEOREM 4.2. *Suppose that* $\theta^i = 1$ *for each* i *(that is, the objective function is a function of the shape of a partition). Then any shape-minorizing partition is optimal (minimizing) uniformly under all Schur convex functions* f*.*

Proof. The assumptions of the theorem imply that for each partition π, θ_π is the shape of π, and the conclusion of the theorem follows from the definition of Schur convexity. □

The next example demonstrates that sign-uniformity of θ is not sufficient for the set of vectors associated with partitions having a prescribed shape to contain a minorizing vector, nor is it sufficient for the existence of a uniformly minimizing partition under all Schur convex functions. So, in general, the conclusions of Theorem 3.2 do not generalize when minorization replaces majorization. It is noted that the example concerns a single-shape problem.

EXAMPLE III. *Let* $n = 11$*,* $p = 3$*,* $n_1 = 2$*,* $n_2 = 4$*,* $n_3 = 5$*,* $\theta^i = 1$ *for* $i = 1, 2, 3, 4$*,* $\theta^i = 2$ *for* $i = 5, 6, 7, 8$*, and* $\theta^i = 6$ *for* $i = 9, 10, 11$*. Let* X *be the set of positive integer* 3*-vectors with coordinate-sum* 30*. All vectors associated with feasible partitions are in* X*. Now,* $x^1 \equiv (10, 10, 10)$ *is majorized by all vectors in* X *and* $x^2 \equiv (11, 10, 9)$ *is majorized by all vectors in* X *except for* x^1*. But neither* x^1 *nor* x^2 *is realizable by a feasible partition because neither* 9 *nor* 10 *nor* 11 *is the sum of two elements among* $\{1, 2, 6\}$*. Next we observe that* $x^3 = (11, 11, 8)$ *and* $x^4 = (12, 9, 9)$ *are majorized by all vectors in* $X \backslash \{x^1, x^2, x^3, x^4\}$*, but neither majorizes the other. Representing parts of partitions by the multiset of the* θ^i*'s, we observe that* $(11, 11, 8)$ *is realizable by the partition* $\pi^3 = (\{5, 9\}, \{1, 6, 7, 10\}, \{2, 3, 4, 8, 11\})$ *and* $(12, 9, 9)$ *is realizable by the partition* $\pi^4 = (\{10, 11\}, \{1, 2, 3, 9\}, \{4, 5, 6, 7, 8\})$*.*

For $t > 0$*, let* $f_t : R^3 \to R$ *be given by* $f_t(x) = \sum_{j=1}^3 |x_j - 10 - t|^3$ *for each* $x \in R^3$*. These functions are separable and strictly Schur convex; further, for all sufficiently small positive* t*,* $f_t(x^3) > f_t(x^4)$*, and the reverse inequality holds for all sufficiently large negative* t*. Since every vector in* $X \backslash \{x^1, x^2, x^3, x^4\}$ *majorizes either* x^3 *or* x^4*, the Schur convexity of the* f_t*'s implies that* π^4 *is optimal for all sufficiently small positive* t*, and* π^3 *is optimal for all sufficiently large negative* t*.*

We next show that every set of bounded shapes contains a minorizing shape, without the restriction concerning the consistency of the lower bound and the upper bound. Of course, Example III demonstrates that shape-minorization does not yield uniform optimality as does shape-majorization with sign-uniform θ. Our first step considers noninteger vectors.

THEOREM 4.3. *Let* L *and* U *be* p*-vectors satisfying* $L \le U$ *and* $\sum_{j=1}^p L_j < n < \sum_{j=1}^p U_j$*, respectively. For every real* $\beta > 0$ *define* $x(\beta)$ *as the* p*-vector with*

$$x(\beta)_j \equiv \begin{cases} L_j & \text{if } \beta \le L_j, \\ \beta & \text{if } L_j < \beta < U_j, \\ U_j & \text{if } \beta \ge U_j. \end{cases} \tag{4.1}$$

Then $x(.)$ is nondecreasing and continuous, and $\{x(\beta) \in R^p : \sum_{j=1}^p x(\beta)_j = n\}$ contains a single vector, say, x^*, which is majorized by every vector in $\{x \in R^p : L \leq x \leq U$ and $\sum_{j=1}^p x_j = n\}$.

Proof. The fact that $x(.)$ is nondecreasing and continuous is immediate from (4.1). Further, since $\sum_{j=1}^p x(\beta)_j = \sum_{j=1}^p L_j < n$ for $\beta \leq \min_j L_j$, and $\sum_{j=1}^p x(\beta)_j = \sum_{j=1}^p U_j > n$ for $\beta \geq \max_j U_j$, continuity arguments assure that $\sum_{j=1}^p x(\beta)_j = n$ for some $\min_j L_j < \beta < \max_j U_j$. Since $x(.)$ is nondecreasing, $\sum_{j=1}^p x(\beta)_j = \sum_{j=1}^p x(\beta')_j$ if and only if $x(\beta) = x(\beta')$. So, $\{x(\beta) \in R^p : \sum_{j=1}^p x(\beta)_j = n\}$ contains a single element, say, x^*. We note that $\{\beta \in R : x(\beta) = x^*\}$ is a nonempty closed interval which is nondegenerate when $\{j = 1, \ldots, p : L_j < x_j^* < U_j\} = \varnothing\}$.

Let $N_- \equiv \{j = 1, \ldots, p : x_j^* = L_j\}$, $N_0 \equiv \{j = 1, \ldots, p : L_j < x_j^* < U_j\}$, $N_+ = \{j = 1, \ldots, p : x_j^* = U_j > L_j\}$, $v_- \equiv |N_-|$, $v_0 \equiv |N_0|$, and $v_+ \equiv |N_+|$. Of course, $v_- + v_0 + v_+ = p$. Select β^* such that $x(\beta^*) = x^*(\beta^*$ is unique when $N_0 \neq \varnothing)$. We then have that $x_j^* = L_j \geq \beta^*$ for $j \in N_-$, $x_j^* = \beta^*$ for $j \in N_0$, and $x_j^* = U_j \leq \beta^*$ for $j \in N_+$. It follows that by possibly permuting indices, we can assume that x^*'s coordinates are nonincreasing, all elements in N_- precede all elements in N_0, and all elements in N_0 precede all elements in N_+; in particular, $N_- = \{1, \ldots, v_-\}$, $N_0 = \{v_- + 1, \ldots, v_- + v_0\}$, and $N_+ = \{v_- + v_0 + 1, \ldots, p\}$.

Let $X \equiv \{x \in R^p : L \leq x \leq U$ and $\sum_{j=1}^p x_j = n\}$. Also, for $k = 1, \ldots, p$, let $W^k \equiv \{w \in R^p : 0 \leq w \leq 1$ and $\sum_{j=1}^p w_j = k\}$ (with 1 representing the vector $(1, \ldots, 1)^T$ in R^p), and let $h_k : X \to R$ with $h_k(x)$ for x in X being the sum of the k largest coordinates of x. We observe that the functions h_k have representations

$$h_k(x) = \sum_{u=1}^k x_{[u]} = \max_{[I]=k} \sum_{u \in I} x_u = \max_{w \in W^k} \sum_{u=1}^k w_u x_u = \max_{w \in W^k} w^T x. \tag{4.2}$$

The claim that $x^* \in X$ is majorized by all vectors x in X means that x^* minimizes each h_k over X. We consider three ranges for k.

$1 \leq k \leq v_-$: In this case for each $x \in X$,

$$h_k(x^*) = \sum_{u=1}^k x_{[u]}^* = \sum_{u=1}^k x_u^* = \sum_{u=1}^k L_u \leq \sum_{u=1}^k x_u \leq \sum_{u=1}^k x_{[u]} = h_k(x). \tag{4.3}$$

$p - v_+ \leq k \leq p$: In this case for each $x \in X$,

$$\begin{aligned} h_k(x^*) &= \sum_{u=1}^k x_{[u]}^* = \sum_{u=1}^k x_u^* = n - \sum_{u=k+1}^p x_u^* = n - \sum_{u=k+1}^p U_u \\ &\leq \sum_{u=1}^p x_u - \sum_{u=k+1}^p x_u \leq \sum_{u=1}^k x_u \leq \sum_{u=1}^k x_{[u]} = h_k(x). \end{aligned} \tag{4.4}$$

$v_- k < p - v_+$: We will construct a vector w^* in W^k that satisfies

$$w^T x^* \leq (w^*)^T x^* \leq (w^*)^T x \quad \text{for each} \quad x \in X \quad \text{and} \quad w \in W^k. \tag{4.5}$$

It will then follow from (4.2) that for every $x \in X$, $h_k(x^*) = \max_{w \in W^k} (w)^T x^* = (w^*)^T x^* \leq (w^*)^T x \leq h_k(x)$. (In fact, a variant of the classic minmax theorem of game theory ensures that the existence of such a vector w^* is necessary and sufficient

for x^* to minimize h_k over X.) Specifically, let $\omega \equiv (k - v_-)/v_0$, and let w^* be the p-vector with

$$w_u^* \equiv \begin{cases} 1 & \text{for } u = 1, \dots, v_-, \\ \omega & \text{for } u = v_- + 1, \dots, v_- + v_0, \\ 0 & \text{for } u = v_- + v_0 + 1, \dots, p. \end{cases} \tag{4.6}$$

Since $v_- < k < p - v_+ = v_- + v_0$, we have that $v_0 = p - v_- - v_+ > 0$ and $0 < \omega < 1$; in particular, $w^* \in W^k$.

For $z \in R^p$ and $j = 0, 1, \dots, p$, let $\bar{z}_j = \sum_{u=1}^{j} z_u$; in particular, $\bar{x}_p = n$ and $\bar{w}_p = k$ for each $x \in X$ and $w \in W^k$. Further,

(4.7)
$$w^{*T}x = \sum_{u=1}^{p} w_u^* x_u = \sum_{u=1}^{p} w_u^*(\bar{x}_u - \bar{x}_{u-1}) = \sum_{u=1}^{p-1}(w_u^* - w_{u+1}^*)\bar{x}_u + w_p^* n \quad \text{for each} \quad x \in X$$

and

(4.8)
$$w^T x^* = \sum_{u=1}^{p} w_u x_u^* = \sum_{u=1}^{p}(\bar{w}_u - \bar{w}_{u-1})x_u^* = \sum_{u=1}^{p-1} \bar{w}_u(x_u^* - x_{u+1}^*) + k x_u^* \quad \text{for each} \quad w \in W.$$

Applying (4.7) to x^* and to arbitrary $x \in X$, we observe that

$$\begin{aligned}(w^*)^T x^* - (w^*)^T x &= \sum_{u=1}^{p-1}(w_u^* - w_{u+1}^*)(\bar{x}_u^* - \bar{x}_u) \\ &= (1 - \omega)(\bar{x}_{v_-}^* - \bar{x}_{v_-}) + \omega(\bar{x}_{v_- + v_0}^* - \bar{x}_{v_- + v_0})\end{aligned} \tag{4.9}$$

(the cases where $v_- = 0$ and/or $v_+ = 0$ require special attention). From (4.3) with $k = v_-$, we have that $\bar{x}_{v_-}^* \le \bar{x}_{v_-}$, and from (4.4) with $k = v_- + v_0 = p - v_+$, we have that $\bar{x}_{v_- + v_0}^* \le \bar{x}_{v_- + v_0}$; since $0 \le \omega \le 1$, we conclude from (4.9) that $(w^*)^T x^* \le w^{*T} x$, establishing the right-hand side inequalities of (4.5). Next, by applying (4.8) to w^* and to arbitrary $w \in W^k$, we observe that

$$\begin{aligned}w^{*T}x^* - w^T x^* &= \sum_{u=1}^{p-1}(\bar{w}_u^* - \bar{w}_u)(x_u^* - x_{u+1}^*) \\ &= \sum_{u=1}^{v_-}(u - \bar{w}_u)(x_u^* - x_{u+1}^*) + \sum_{u=v_-+1}^{v_- + v_- - 1}(k - \bar{w}_u)(\beta^* - \beta^*) \\ &\quad + \sum_{u=v_- + v_0}^{p}(k - \bar{w}_u)(x_u^* - x_{u+1}^*)\end{aligned} \tag{4.10}$$

(here again, the cases where $v_- = 0$ and/or $v_+ = 0$ require special attention). Since $\bar{w}_u \le u$ and $\bar{w}_u \le k$ for each $w \in W^k$ and $u = 1, \dots, p$ and since $x_1^* \ge x_2^* \ge \cdots \ge x_p^*$, we conclude from (4.10) that $(w^*)^T x^* \ge w^T x^*$ for every $w \in W^k$, completing the proof of (4.5). □

In the next result, we use the notation $\| \ \|_\infty$ for the 1_∞ norm in R^p defined for $x \in R^p$ by $\|x\|_\infty = \max_{u \in \{1, \dots, p\}} x_u$.

THEOREM 4.4. *Let L and U be positive integer p-vectors satisfying $L \le U$ and $\sum_{j=1}^{p} L_j < n < \sum_{j=1}^{p} U_j$, and let x^* be as in Theorem 4.3. Then there exists an integer p-vector z^* with $\|z^* - x^*\|_\infty < 1$, and each such vector is majorized by every integer vector in $\{x \in R^p : L \le x \le U$ and $\sum_{j=1}^{p} x_j = n\}$.*

Proof. The conclusion of this theorem is trivial when x^* is integral, so assume that this is not the case. Let N_-, N_0, N_+, v_-, v_0, and v_+ be as in the proof of Theorem 4.3, and as in that proof assume that x^*'s coordinates are nonincreasing, all elements in N_- precede all elements in N_0, and all elements in N_0 precede all elements in N_+; in particular, $N_- = \{1, \ldots, v_-\}$, $N_0 = \{v_- + 1, \ldots, v_- + v_0\}$, and $N_+ = \{v_- + v_0 + 1, \ldots, p\}$. The assertion that x^* is not integral means that $N_0 \ne \varnothing$ and the unique β^* with $x(\beta^*) = x^*$ is not integral.

Let $X \equiv \{x \in R^p : L \le x \le U$ and $\sum_{j=1}^{p} x_j = n\}$, let $\lfloor \beta^* \rfloor$ be the largest integer less than β^*, and let $\lceil \beta^* \rceil \equiv \lfloor \beta^* \rfloor + 1$. The integrality of L and U ensures that $L_u \le \lfloor \beta^* \rfloor < \beta^* < \lceil \beta^* \rceil \le U_u$ for $u \in N_0$. Further, we observe that $v_0\beta^* = n - \sum_{u \in N_-} L_u - \sum_{u \in N_+} U_u$ is an integer and $v_0\lfloor \beta^* \rfloor < v_0\beta^* < v_0\lceil \beta^* \rceil$, implying that $\mu \equiv v_0\beta^* - v_0\lfloor \beta^* \rfloor$ is an integer satisfying $1 \le \mu < v_0$ and $\mu\lceil \beta^* \rceil + (v_0 - \mu)\lfloor \beta^* \rfloor = v_0\lfloor \beta^* \rfloor + \mu(\lceil \beta^* \rceil - \lfloor \beta^* \rfloor) = v_0\lfloor \beta^* \rfloor + \mu = v_0\beta^*$. It follows that the p-vector z^* with z^*_u for $u = 1, \ldots, p$ given by

$$z^*_u \equiv \begin{cases} x^*_u & \text{if } u \in N_- \cup N_0, \\ \lceil \beta^* \rceil & \text{if } u = v_- + 1, \ldots, v_- + \mu, \\ \lfloor \beta^* \rfloor & \text{if } u = v_- + \mu + 1, \ldots, v_- + v_0 \end{cases} \tag{4.11}$$

is integral, is in X, and satisfies $\|z^* - x^*\|_\infty < 1$. We will show that z^* is majorized by any integer vector z in X by showing that $h_k(z) \ge h_k(z^*)$ for $k = 1, \ldots, p$, where $h_k(.)$ is the function assigning to each p-vector the sum of its k largest coordinates (see the proof of Theorem 4.3).

Let z be an integer vector in X. For $u \in N_-, L_u \ge \beta^*$, and the integrality of L_u implies that $L_u \ge \lceil \beta^* \rceil$. Similarly, for $u \in N_+$, $U_u \le \beta^*$, and the integrality of U_u implies that $U_u \le \lfloor \beta^* \rfloor$. Consequently, z^*'s coordinates are nonincreasing and, therefore, $h_k(z^*) = \sum_{j=1}^{k} z^*_{[j]} = \sum_{j=1}^{k} z^*_j$ for $k = 1, \ldots, p$. From Theorem 4.3, $h_k(z) \ge h_k(x^*) = h_k(z^*)$ for $1 \le k \le v_-$ and for $v_0 + v_+ \le k \le p$. Further, as Theorem 4.3 ensures that $h_{v_-+1}(z) \ge h_{v_-+1}(x^*) = h_{v_-}(x^*) + \beta^*$, the integrality of $h_{v_-+1}(z)$ and $h_{v_-}(x^*)$ implies that $h_{v_-+1}(z) \ge h_{v_-}(x^*) + \lceil \beta^* \rceil = h_{v_-+1}(z^*)$. To prepare for an inductive argument, assume that $h_k(z) \ge h_k(z^*)$ and $h_{k+1}(z) < h_{k+1}(z^*)$ for some $v_- + 1 \le k < v_0 + v_+ - 1$. Then $h_k(z^*) + z^*_{k+1} = h_{k+1}(z^*) > h_{k+1}(z) = h_k(z) + z_{[k+1]}$, implying that $z_{[k+1]} < h_k(z^*) + z^*_{k+1} - h_k(z) \le z^*_{k+1} \le \lceil \beta^* \rceil$. Since $z_{[k+1]}$ and $\lceil \beta^* \rceil$ are integral, we conclude that $z_{[k+1]} \le \lceil \beta^* \rceil - 1 = \lfloor \beta^* \rfloor$ and, therefore, $z_{[j]} \le \lfloor \beta^* \rfloor$ for $j = k+2, \ldots, v_- + v_o$ (recall that the coordinates of z^* are nonincreasing). It follows that

$$\begin{aligned} h_{v_-+v_0}(z) &= h_{k+1}(z) + \sum_{u=k+2}^{v_-+v_0} z_{[u]} < h_{k+1}(z^*) + (v_- + v_0 - k - 1)\lfloor \beta^* \rfloor \\ &= \sum_{u=1}^{k+1} z^*_u + (v_- + v_0 - k - 1)\lfloor \beta^* \rfloor \le \sum_{u=1}^{v_-+v_0} z^*_u = h_{v_-+v_0}(x^*). \end{aligned}$$

This inequality contradicts the conclusion of Theorem 4.3, asserting that x^* is majorized by z, and thereby completes an inductive proof that $h_k(z) \ge h_k(z^*)$ for $k \in \{v_- + 1, \ldots, v_0 + v_+\}$.

We finally observe that an integer vector z is in X and satisfies $\|z - x^*\|_\infty < 1$ if and only if $z_u = x^*_u$ for $u \in N_- \cup N_+$ (as each such x^*_u is integral), it has exactly μ of the v_0 coordinates z_u indexed by $u \in N_0$ equal $\lceil \beta^* \rceil$, and it has the remaining $v_0 - \mu$ coordinates indexed by $u \in N_0$ equal $\lfloor \beta^* \rfloor$. It follows that for each such z, a coordinate permutation of z^* exists, implying that $h_k(z) = h_k(z^*)$ for each $k = 1, \ldots, p$; in particular, such z, like z^*, is majorized by all integer vectors in X. □

REFERENCES

[1] B. Gao, F. K. Hwang, W. W-C. Li, and U. G. Rothblum, *Partition-polytopes over 1-dimensional points*, Math. Program., 85 (1999), pp. 335–362.

[2] F. K. Hwang, S. Onn, and U. G. Rothblum, *A polynomial time algorithm for shaped partition problems*, SIAM J. Optim., 10 (1999), pp. 70–81.

[3] F. K. Hwang, and U. G. Rothblum, *Directional-quasi-convexity, asymmetric Schur-convexity and optimality of consecutive partitions*, Math. Oper. Res., 21 (1996), pp. 540–554.

[4] F. K. Hwang, and U. G. Rothblum, *Partitions: Optimality and Clustering*, World Scientific, to appear.

[5] D. Knuth, *The Art of Computer Programming*, 2nd ed., Addison-Wesley, Reading, MA, 1981.

[6] A. W. Marshall, and I. Olkin, *Inequalities, Theory of Majorization and Its Applications*, Academic Press, New York, 1979.

[7] A. F. Veinott, Jr., *Least d-majorized network flows with inventory and statistical applications*, Management Sci., 17 (1971), pp. 547–567.

[8] A. F. Veinott, Jr., *On d-majorization and d-Schur convexity*, to appear.

SIAM J. DISCRETE MATH.
Vol. 18, No. 3, pp. 525–541

MATROIDS INDUCED BY PACKING SUBGRAPHS*

MAREK JANATA†

Abstract. This paper is concerned with the classification of families of graphs $\mathcal{T}$ with the following property: For any graph G, the subsets of vertices of G that can be saturated by packing copies of graphs from $\mathcal{T}$ form a collection of independent sets of a matroid. From this point of view, we present a characterization of so-called EHP-families of graphs (i.e., families consisting of K_2, hypomatchable graphs, and propellers). The main result is the following: For a matroid-inducing EHP-family $\mathcal{T}$, we characterize connected graphs H such that the family $\mathcal{T} \cup \{H\}$ is also matroid-inducing.

Key words. graph packing, matroid

AMS subject classifications. 05C70, 05B35

DOI. 10.1137/S0895480102379830

1. Introduction.

Let $G = (V(G), E(G))$ be a graph. A matching of G can be viewed as a set of vertex disjoint subgraphs of G, each isomorphic to K_2. A natural generalization is a set of vertex disjoint subgraphs of G, each a member of a family $\mathcal{F}$ of subgraphs of G. This generalization is called an $[\mathcal{F}]$-*packing* of G. (We use brackets in this notation to avoid confusion with a more special generalization, which will be defined further.)

Let us introduce the following terminology, which derives from matching theory: An $[\mathcal{F}]$-packing $\mathcal{Q}$ *covers* a vertex $v \in V(G)$ if one of the subgraphs included in $\mathcal{Q}$ contains v. Otherwise $\mathcal{Q}$ *skips* v. An $[\mathcal{F}]$-packing *saturates* a set of vertices $X \subseteq V(G)$ if it covers every member of X. For an $[\mathcal{F}]$-packing $\mathcal{Q}$, $V(\mathcal{Q})$ denotes the set of all vertices covered by $\mathcal{Q}$. An $[\mathcal{F}]$-packing $\mathcal{Q}$ of G is *maximal* if there is no $[\mathcal{F}]$-packing $\mathcal{Q}'$ of G with $V(\mathcal{Q}') \supsetneq V(\mathcal{Q})$ and is *perfect* if it covers all vertices of G. A graph G is $[\mathcal{F}]$-*saturable* if it admits a perfect $[\mathcal{F}]$-packing and is *non*-$[\mathcal{F}]$-*saturable* if it has no perfect $[\mathcal{F}]$-packing. The $[\mathcal{F}]$-*packing problem* in G consists of finding an $[\mathcal{F}]$-packing of G saturating a set of maximum cardinality.

A special case of $[\mathcal{F}]$-packing appears when $\mathcal{F}$ consists of all subgraphs of G isomorphic to members of a fixed family $\mathcal{T}$ of graphs. In this special case, "$\mathcal{T}$" will be used instead of "$[\mathcal{F}]$" throughout the notation described above.

The $[\mathcal{F}]$-packing problem has been extensively studied from many points of view. The most important are the cases when the $[\mathcal{F}]$-packing problem can be solved in polynomial time (see section 2 for examples). A common feature proved for many of the polynomially solvable cases is that the sets of vertices saturated by some $[\mathcal{F}]$-packing form a collection of independent sets of a matroid.

Let X be a set and let $\mathcal{M}$ be a nonempty hereditary system of subsets of X (i.e., if $A \in \mathcal{M}$ and $A' \subseteq A$, then $A' \in \mathcal{M}$). The maximal sets of $\mathcal{M}$ (under set inclusion) are called *bases*. The pair $(X, \mathcal{M})$ is called a *matroid* if the set $\mathcal{B}$ of its bases satisfies the exchange axiom:

(EA) $\forall B, B' \in \mathcal{B}; \forall x \in B \setminus B'; \exists y \in B' \setminus B : (B' \setminus \{y\}) \cup \{x\} \in \mathcal{B}$.

*Received by the editors May 2, 2002; accepted for publication (in revised form) May 17, 2004; published electronically February 25, 2005.

http://www.siam.org/journals/sidma/18-3/37983.html

†Department of Applied Mathematics and Institute of Theoretical Computer Science (ITI), Charles University, Malostranské n. 25, 118 00 Praha 1, Czech Republic (janata@kam.mff.cuni.cz).

If $M = (X, \mathcal{M})$ is a matroid, then X is called the *ground set* of M, and the subsets of X contained in $\mathcal{M}$ are called the *independent sets* of M. The exchange axiom implies the fact that all bases of a matroid have the same cardinality.

If G is a graph and $\mathcal{F}$ a family of its subgraphs, then we denote by $M(G, [\mathcal{F}])$ the family of all subsets of $V(G)$ that can be saturated by some $[\mathcal{F}]$-packing. If $\mathcal{F}$ consists of all subgraphs of G isomorphic to members of a family $\mathcal{T}$ of graphs, then $M(G, \mathcal{T})$ stands for $M(G, [\mathcal{F}])$. This paper is concerned with the classification of families $\mathcal{T}$ of connected graphs such that $M(G, \mathcal{T})$ is a collection of independent sets of a matroid in every graph G. A family $\mathcal{T}$ with this property is called *matroid-inducing*.

1.1. Notation and basic notions. For two graphs H, H', $H' \subseteq H$ denotes that H' is a subgraph of H. If $x \in V(H)$, then $H \setminus x$ is the graph obtained from H by deleting the vertex x. If $D \subseteq V(H)$, then $H \setminus D$ is the graph obtained from H by deleting every vertex $x \in D$. For a graph H, the number of vertices of H will be denoted by $|H|$.

Let H be a connected graph and let $u, v \in V(H)$. The *distance from u to v*, denoted by $dist(u, v)$, is the minimum number of edges on a path from u to v. The distance between two subgraphs of H and between a vertex and a subgraph are defined analogously.

A graph H is *hypomatchable* if it has no perfect matching but for every $x \in V(H)$, $H \setminus x$ admits a perfect matching. A single vertex is considered hypomatchable. A fact that we will use is that hypomatchable graphs do not contain vertices of degree one.

A *k-star* S_k is a complete bipartite graph $K_{1,k}$, i.e., the graph with $k+1$ vertices $c, v_1, \ldots, v_k$ and k edges $cv_1, \ldots, cv_k$. The vertex c is called the *center* and k is the *index* of the star.

2. History and results. The basic polynomially solvable cases of the $[\mathcal{F}]$-packing problem in which $M(G, [\mathcal{F}])$ is a matroid are the following.

(E) *Matching (edge-packing)* [3, 4]. $\mathcal{F}$ consists of all edges of G.

(EH) *Packing by edges and a set of hypomatchable graphs* [1, 2, 5]. $\mathcal{F}$ consists of all edges of G and some hypomatchable subgraphs of G.

(S) *Packing by sequential sets of stars* [7]. For some integer r, $\mathcal{F}$ consists of all subgraphs isomorphic to a star $S_i, 1 \leq i \leq r$.

In [8, 9], Loebl and Poljak studied the following case of the $[\mathcal{F}]$-packing problem.

(EHP) *Packing by edges, hypomatchable graphs, and propellers.* $\mathcal{F}$ consists of all edges of G, some hypomatchable subgraphs of G, and a family $\mathcal{R} \subsetneq \mathcal{F}$ of subgraphs of G named *propellers*. Let us now introduce the notion of propeller.

A connected graph P is a *k-propeller* ($k \geq 1$ is the *index* of P) if it has a vertex c, called the *center*, such that $P \setminus c$ consists of $k+1$ components $D_0, \ldots, D_k$, where $|D_0| = 1$ and every D_i is hypomatchable. Note that every $(k+1)$-star is a k-propeller.

A propeller is called *rooted* if it has one chosen vertex r of degree one, called the *root*. We denote a rooted propeller P with root r by (P, r). If P has more vertices of degree one (like a star), then there are more possibilities of choosing the root, and the corresponding rooted propellers are considered to be distinct. Since hypomatchable graphs have no vertices of degree one, the root r must be a neighbor of the center c. The edge rc is called the *stick* of (P, r). Let $D_0, \ldots, D_k$ be the components of $P \setminus c$. Without loss of generality we may suppose that $D_0 = \{r\}$. The remaining components $D_1, \ldots, D_k$ are called the *blades* of (P, r). We denote the set of all blades of (P, r) by $D(P, r)$.

A family $\mathcal{R}$ of rooted propellers that are subgraphs of a graph G is called *G-closed* if it satisfies the following three axioms.

Heredity. If $(H,r) \in \mathcal{R}$ and (H',r) are rooted propellers with the same stick and $D(H',r) \subseteq D(H,r)$, then $(H',r) \in \mathcal{R}$.

Stick exchange. If $(H,r) \in \mathcal{R}$ is a rooted propeller with stick cr and r' is a vertex of $G \setminus H$ adjacent (in G) to c, then $(H \setminus r) \cup cr'$ rooted in r' belongs to $\mathcal{R}$.

Blade exchange. Let $(H,r),(H',r) \in \mathcal{R}$ be rooted propellers with the same stick rc and $D(H',r) \not\subseteq D(H,r)$. Then for any blade D of (H,r), disjoint to all blades of (H',r), there is some blade $D' \in D(H',r) \setminus D(H,r)$ such that the rooted propeller (H'',r) with stick rc and blades $(D(H',r) \setminus D') \cup D$ belongs to $\mathcal{R}$.

Note that the Blade exchange axiom does not specify which edges connect the centers of the propeller with their blades. Hence there may be more than one rooted propeller with stick rc and blades $(D(H',r') \setminus D') \cup D$ that belong to $\mathcal{R}$. However, each such propeller is in $\mathcal{R}$ if $\mathcal{R}$ is G-closed (by Heredity).

A family of graphs containing only edges, hypomatchable graphs, and propellers is called an *EHP-family*. The result of Loebl and Poljak concerning the (EHP) problem is summarized in the following theorem.

THEOREM 2.1. *Let G be a graph and let $\mathcal{F} = E(G) \cup \mathcal{H} \cup \mathcal{R}$ be an EHP-family of its subgraphs, where $\mathcal{H}$ is a family of hypomatchable graphs and $\mathcal{R}$ is a G-closed family of rooted propellers. Then the $[\mathcal{F}]$-packing problem can be solved in polynomial time and $M(G,[\mathcal{F}])$ is a matroid.*

Let us observe that a sequential family of stars induces a closed family of propellers in every graph G. Hence the (EHP) case contains all of (E), (EH), and (S).

Note that (E) and (S) also concern the more special $\mathcal{T}$-packing problem: In (E), $\mathcal{T} = \{K_2\}$, and in (S), $\mathcal{T} = \{S_1,\dots,S_r\}$ for some integer r. In both of these cases the $\mathcal{T}$-packing problem can be solved in polynomial time in any graph G, and $\mathcal{T}$ is a matroid-inducing family.

In [8, 10], Loebl and Poljak studied the following case of the $\mathcal{T}$-packing problem.

(E+1) *Packing by edges and copies of a single graph.* $\mathcal{T}$ consists of K_2 and a fixed graph H. The complexity of this case was fully characterized in [10] and the matroid-inducing property of $\mathcal{T}$ in [8]. The two results show that in this case the $\mathcal{T}$-packing problem is polynomially solvable if and only if $\mathcal{T}$ is a matroid-inducing family (otherwise it is NP-complete). The characterization of the matroid-inducing property of $\mathcal{T}$ in this case is given by the following theorem.

THEOREM 2.2. *Let H be a connected graph. Then $\{K_2,H\}$ is a matroid-inducing family if and only if H is perfectly matchable, hypomatchable, or a 1-propeller.*

The above result may also be viewed as follows: It is a characterization of graphs H that can be added to the matroid-inducing EHP-family $\{K_2\}$ such that $\{K_2\} \cup \{H\}$ is also a matroid-inducing family. The interesting property is that the resulting family is always an EHP-family. In the author's Master's thesis [6], this result was extended to the following.

(S+1) *Packing by a sequential set of stars and copies of a single graph.* $\mathcal{T}$ consists of a sequential family $\mathcal{S} = \{S_1,S_2,\dots,S_n\}$ of stars and of a fixed graph H. The characterization of the matroid-inducing property of $\mathcal{T}$ in this case is given by the following theorem.

THEOREM 2.3. *Let $\mathcal{S} = \{S_1,\dots,S_k\}, k > 1$, be a family of stars and let H be a connected graph. Then $\mathcal{T} = \mathcal{S} \cup \{H\}$ is a matroid-inducing family if and only if H is $\mathcal{S}$-saturable, hypomatchable, or the star S_{k+1}.*

This paper is concerned with matroid-inducing EHP-families $\mathcal{T}$. At first, we will

reformulate Loebl's and Poljak's necessary condition on EHP-families $\mathcal{F}$ of subgraphs of G, guaranteeing that $M(G,[\mathcal{F}])$ forms a matroid to a necessary condition guaranteeing that a general EHP-family $\mathcal{T}$ of graphs is matroid-inducing. Moreover, we will prove that this reformulated condition is sufficient, which is a new result: a full characterization of the (EHP) case for $\mathcal{T}$-packing. This work will be done in section 3.

The main purpose of this paper is to give a full characterization of individual graphs H that can be added to a general matroid-inducing EHP-family $\mathcal{T}$, such that $\mathcal{T}\cup\{H\}$ is also a matroid-inducing family. We will show that a graph H has this property if and only if H is $\mathcal{T}$-saturable or $\mathcal{T}\cup\{H\}$ is a matroid-inducing EHP-family, which shows that the property proved by Loebl and Poljak for the case $\mathcal{T}=\{K_2\}$ holds throughout the whole class of matroid-inducing EHP-families $\mathcal{T}$. The proof of this fact will be given in section 4.

The results given in this paper are in fact generalizations of the results of Loebl and Poljak presented in [8]. In many places, we will be able to use proofs and arguments similar to those used in [8]. We will call the reader's attention to this correspondence when it occurs.

3. Matroid-inducing EHP-families. In this section, a full characterization of EHP-families w.r.t. the matroid-inducing property will be presented. Let us start with a few more notions concerning propellers and families of propellers.

A propeller $P'\supseteq K_2$ is a *subpropeller* of a propeller P with center c if P' arises from P by deleting one or more components of $P\setminus c$. Suppose P_1,P_2 are two disjoint graphs, such that each P_i is either a propeller with center c_i or an edge with one end-vertex c_i. We denote by P_1+P_2 the graph that arises from P_1,P_2 by glueing vertices c_1,c_2 into one vertex c, by glueing arbitrary neighbors r_1,r_2 of c_1,c_2 with degree one into one vertex r, and by deleting the multiple edge.

The following claim will be used.

CLAIM 3.1. *The center c of a k-propeller P is the only vertex of P, such that $P\setminus c$ has more than one component with no perfect matching.*

Proof. $P\setminus c$ really has $k+1\geq 2$ such components $D_0,\dots,D_k$ (every D_i is hypomatchable). If there exists a vertex $x\neq c$, such that $P\setminus x$ has at least two components with no perfect matching, then $x\in D_j$ for some j. Let B be a component of $P\setminus x$ that has no perfect matching and does not contain c. Then B is also a component of $D_j\setminus x$, which is a contradiction, because D_j is hypomatchable. □

Let L be a graph and let $\mathcal{T}$ be a family of graphs. By "$[L]\in\mathcal{T}$" we mean "there is some $L'\in\mathcal{T}$ isomorphic to L." The following definition mimics the definition of G-closed family of propellers that are subgraphs of G.

A family $\mathcal{R}$ of propellers is called *closed* if it satisfies the following two axioms.

Heredity. If $P\in\mathcal{R}$ and P' is a subpropeller of P, then $[P']\in\mathcal{R}$.

Blade exchange. If $P,P'\in\mathcal{R}$ are propellers and P' is isomorphic to P_1+P_2, where $P_1\in\mathcal{R}$ is a 1-propeller and P_2 is a subpropeller of P or an edge connecting the center of P to a vertex of degree one, then there exists a component B of $P\setminus P_2$ such that $[(P\setminus B)+P_1]\in\mathcal{R}$.

Let $\mathcal{T}'$ be a family of graphs. Note that in a $\mathcal{T}'$-packing, we may avoid the use of any member of $\mathcal{T}'$ that has a perfect packing by the remaining members. We say that $\mathcal{T}\subseteq\mathcal{T}'$ is a *sufficient subfamily* of $\mathcal{T}'$ if all graphs from $\mathcal{T}'\setminus\mathcal{T}$ have perfect $\mathcal{T}$-packing. The following theorem characterizes matroid-inducing EHP-families.

THEOREM 3.1. *An EHP-family $\mathcal{T}$ of graphs is matroid-inducing if and only if $\mathcal{T}$ is a sufficient subfamily of some EHP-family $\mathcal{T}'=\{K_2\}\cup\mathcal{H}\cup\mathcal{R}$, where $\mathcal{H}$ is a family*

of hypomatchable graphs and $\mathcal{R}$ is a closed family of propellers.

Proof. The "if" part of the proof is a simple application of Theorem 2.1. Let $\mathcal{T}$ be a sufficient subfamily of a family $\mathcal{T}'$ consisting of K_2, hypomatchable graphs, and a closed family of propellers. Let G be an arbitrary graph. We will construct a family $\mathcal{F}$ of subgraphs of G in two steps: In the first step we will insert in $\mathcal{F}$ all subgraphs of G isomorphic to members of $\mathcal{T}'$ with all possible selections of roots for every propeller. In the second step we will enlarge $\mathcal{F}$ to include every rooted propeller that is a subgraph of G and has a common center, root, and blades with any of the rooted propellers added in the first step (the only difference may be in the edges connecting the blades and the center of the propeller). Obviously $M(G,[\mathcal{F}]) = M(G,\mathcal{T})$. It can be easily verified that $\mathcal{F}$ satisfies the supposition of Theorem 2.1. Thus $M(G,[\mathcal{F}]) = M(G,\mathcal{T})$ is a matroid and since G was arbitrary, $\mathcal{T}$ is a matroid-inducing family.

We will prove the "only if" part by constructing counterexamples. We want to prove that if a family $\mathcal{T}$ cannot be enlarged by adding $\mathcal{T}$-saturable graphs to a family $\mathcal{T}'$ consisting of K_2, hypomatchable graphs, and a closed family of propellers, then there exists a counterexample G such that $M(G,\mathcal{T})$ is not a matroid.

Let $\mathcal{T}$ be a family that cannot be enlarged this way. There are two cases.

(i) *$\mathcal{T}$ violates Heredity.* There exists a propeller $P \in \mathcal{T}$ such that one of its subpropellers P' is not $\mathcal{T}$-saturable (thus $[P'] \notin \mathcal{T}$, and we cannot add P' into $\mathcal{T}$).

(ii) *$\mathcal{T}$ violates Blade exchange.* There are two propellers $P, P' \in \mathcal{T}$ such that for some 1-propeller $P_1 \in \mathcal{T}$, P' is isomorphic to $P_1 + P_2$, where P_2 is a subpropeller of P or an edge connecting the center of P to a vertex of degree one, and for no component B of $P \setminus P_2$, the propeller $(P \setminus B) + P_1$ is $\mathcal{T}$-saturable.

In Lemmas 3.2 and 3.3 we will show counterexamples to both (i) and (ii). For each case we will construct a counterexample G, such that $M(G,\mathcal{T})$ is not a matroid (we will find two bases of different cardinality). That will be the proof of the "only if" part of Theorem 3.1. □

LEMMA 3.2 (counterexample for an EHP-family violating Heredity (i)). *Let $\mathcal{T}$ be an EHP-family. If there exists a $\mathcal{T}$-saturable propeller P and a non-$\mathcal{T}$-saturable subpropeller P' of P, then $\mathcal{T}$ is not matroid-inducing.*

Proof. Let P be a $\mathcal{T}$-saturable propeller with center c and P' its non-$\mathcal{T}$-saturable subpropeller. Without loss of generality we may suppose that P is a propeller with minimum $|P|$ and that P' is a k-propeller with maximum k.

With this supposition, for one component D of $P \setminus c$, $P' = P \setminus D$. Let $r \in V(P')$ be an arbitrary neighbor of c with degree one (such r exists since P' is a propeller). Let us denote by b the number of non-$\mathcal{T}$-saturable components of $P' \setminus cr$. We know that $b \geq 1$; otherwise the whole P' would be $\mathcal{T}$-saturable. Let us observe the following claim.

CLAIM 3.2. *There is no $\mathcal{T} \setminus \{P\}$-packing of P saturating the whole $P' = P \setminus D$.*

Proof of Claim 3.2. Let $\mathcal{Q}$ be a $\mathcal{T} \setminus \{P\}$-packing of P saturating P' and let B be a non-$\mathcal{T}$-saturable component of $P' \setminus cr$. Let $W \in \mathcal{Q}$ be the graph containing the center c of P'. It follows that $r \in W$ and that $W \cap B$ is not empty and non-$\mathcal{T}$-saturable.

$W \setminus c$ is a graph with at least two non-$\mathcal{T}$-saturable components. Using Claim 3.1, we conclude that W is a propeller with center c. Let $\mathcal{A}$ be the family of all components of $W \setminus cr$ intersecting D and let W' be a propeller that arises from W by deleting all members of $\mathcal{A}$. W' is a subpropeller of W and, due to the minimality of P, W' has a perfect $\mathcal{T}$-packing Q' (if $[W'] \in \mathcal{T}$, then trivially $\mathcal{Q}' = \{W'\}$). Let $\mathcal{Q}_1$ be the part of $\mathcal{Q}$ intersecting $D \setminus W$. We may construct a perfect $\mathcal{T}$-packing $((\mathcal{Q} \setminus \{W\}) \cup \mathcal{Q}') \setminus \mathcal{Q}_1$ of P', which is a contradiction. □

Let us construct a counterexample G and prove that $M(G,\mathcal{T})$ is not a matroid by introducing two bases of different cardinality. G arises from two copies P_1, P_2 of P (vertices of each P_i will be denoted by the index i) by glueing the center c_2 of P_2 to an arbitrary vertex x_1 from the component D_1 of $P_1 \setminus c_1$ (see Figure 2(a)).

Consider a base B_2 of $M(G,\mathcal{T})$ containing $V(P_2)$. We may construct a $\mathcal{T}$-packing $\mathcal{Q}$ of G saturating $V(P_2)$ as follows: $\mathcal{Q}$ uses a copy P_2 of P, a perfect matching of $D_1 \setminus x_1$, a copy c_1r_1 of K_2, and maximum $\mathcal{T}$-packings of all components of $P_1 \setminus (D_1 \cup \{r_1, c_1\})$. Hence $|B_2| \geq |G| - b$, where b is the number of non-$\mathcal{T}$-saturable components of $P' \setminus cr$ (we know that $b \geq 1$).

Let B_1 be a base containing $V(P_1)$. If $\mathcal{N}$ is a $\mathcal{T}$-packing corresponding to B_1, then $\mathcal{N}$ uses a graph W covering the center c_1 of P_1. Obviously $W \setminus c_1$ has at least $b+1$ non-$\mathcal{T}$-saturable components, and so, according to Claim 3.1, W is a propeller and c_1 its center.

If $c_2 \notin W$, then let L be the graph covering c_2 (there exists such a graph since $V(P_1) \subseteq B_1$). We know that c_1 is contained in the graph W and thus $L \cap P_1$ contains only vertices from D_1. Let us delete L from $\mathcal{N}$ and consider the induced $\mathcal{T}$-packing $\mathcal{N}'$ of P_1. $\mathcal{N}'$ saturates the whole $P_1 \setminus D_1$ and by a cardinality argument $\mathcal{N}'$ does not use any copy of P, which is a contradiction by Claim 3.2.

If $c_2 \in W$, then every component of $W \setminus c_2$ intersecting P_2 is $\mathcal{T}$-saturable (W is a propeller with center c_1). None of the $b+1$ non-$\mathcal{T}$-saturable components of $P_2 \setminus c_2$ can be saturated by $\mathcal{N}$ and thus $|B_1| \leq |G| - (b+1) < |B_2|$. We have found two bases of different cardinality and so G is a counterexample showing that $\mathcal{T}$ is not a matroid-inducing family. □

We have shown a counterexample for an EHP-family violating Heredity. In the next lemma we will complete the proof of Theorem 3.1 by introducing a counterexample for an EHP-family violating Blade exchange.

LEMMA 3.3 (Counterexample for an EHP-family violating Blade exchange (ii)). *Let $\mathcal{T}$ be an EHP-family. Let $\mathcal{T}$ fulfill Heredity; i.e., if a propeller $R \in \mathcal{T}$, then let all subpropellers of R be $\mathcal{T}$-saturable.*

If there are two propellers $P, P' \in \mathcal{T}$ such that for some 1-propeller $P_1 \in \mathcal{T}$, P' is isomorphic to $P_1 + P_2$, where P_2 is a subpropeller of P or an edge connecting the center of P to a vertex of degree one, and for no component B of $P \setminus P_2$, the propeller $(P \setminus B) + P_1$ is $\mathcal{T}$-saturable, then $\mathcal{T}$ is not matroid-inducing.

Proof. Let P, P', P_1, P_2 be the propellers described above. Let us consider the graph $G = P + P_1$. Note that G is a propeller: let us denote by c its center and by r the glued-together neighbor of c of degree one (see Figure 2(b)). Without loss of generality assume that $r \in V(P_2)$.

Let B_1 be a base of $M(G,\mathcal{T})$ containing $V(P)$. Then $|B_1| \geq |G| - 1$, since we may construct a $\mathcal{T}$-packing of G saturating $V(P)$ using a copy of P and a maximum matching of $P_1 \setminus cr$.

Let B_2 be a base containing $V(P_1) \cup V(P_2)$. If $|B_2| = |G| - 1$, then the $\mathcal{T}$-packing $\mathcal{Q}$ corresponding to B_2 skips exactly one vertex in one component W of $P \setminus P_2$. If $|B_2| = |G|$, then let W be an arbitrary component of $P \setminus P_2$. We will prove that $G \setminus W$ is $\mathcal{T}$-saturable.

Consider the graph $L \in \mathcal{Q}$ covering the center c of G. L covers the whole edge cr and so L is a copy of K_2 or a propeller with center c.

If L is a copy cr of K_2, then every component of $G \setminus (W \cup \{c, r\})$ is $\mathcal{T}$-saturable. Thus the whole $G \setminus W$ is $\mathcal{T}$-saturable.

If L is a propeller, then let $\mathcal{A}$ be the set of all components of $L \setminus c$ intersecting

W. Let L' be the graph that arises from L by deleting all components from $\mathcal{A}$. L' is a copy of K_2 or a propeller and due to Heredity, L' has a perfect $\mathcal{T}$-packing $\mathcal{Q}'$. Denote by $\mathcal{Q}_1$ the part of $\mathcal{Q}$ intersecting $W \setminus L$. We may construct a perfect $\mathcal{T}$-packing $((\mathcal{Q} \setminus \{L\}) \cup \mathcal{Q}') \setminus \mathcal{Q}_1$ of $G \setminus W$.

We have proved that $G \setminus W$ is $\mathcal{T}$-saturable. It gives us a contradiction, because W is a component of $P \setminus P_2$ and $(G \setminus W) = (P \setminus W) + P_1$, which has no perfect $\mathcal{T}$-packing by the assumption. Thus $|B_2| < |G| - 1 \leq |B_1|$, which proves that G is a counterexample showing that $\mathcal{T}$ is not a matroid-inducing family. □

4. (EHP+1)-packing. In this section we will introduce a characterization of the graphs H that can be added to a matroid-inducing EHP-family $\mathcal{T}$ such that $\mathcal{T} \cup \{H\}$ is a matroid-inducing family. We will prove the following extension of Theorem 2.2.

THEOREM 4.1. *Let $\mathcal{T}$ be a matroid-inducing EHP-family and let H be a graph. Then the family $\mathcal{T} \cup \{H\}$ is matroid-inducing if and only if H is $\mathcal{T}$-saturable or $\mathcal{T} \cup \{H\}$ is a matroid-inducing EHP-family.*

The "if" part of the theorem is trivial. Moreover, the characterization of matroid-inducing EHP-families (Theorem 3.1) gives us a partial negative result for adding a propeller H such that $\mathcal{T} \cup \{H\}$ is not matroid-inducing.

For proving the "only if" part of the Theorem 4.1, it remains to show that if a matroid-inducing EHP-family $\mathcal{T}$ is enlarged by a graph H that is not hypomatchable, $\mathcal{T}$-saturable, or a propeller, then $\mathcal{T} \cup \{H\}$ is not matroid-inducing. We will show it at the end of this section by discussing the structure of H and by showing counterexamples.

Before proving Theorem 4.1, some auxiliary notions and lemmas concerning the structure of packings will be introduced. This technical work will be done in sections 4.1 and 4.2. The proof of Theorem 4.1 will be given in section 4.3.

4.1. Economical packings. Economical packing is a notion first defined by Loebl and Poljak in [8] for packing by edges and hypomatchable subgraphs. Economical packings try to cover as many vertices by copies of K_2 as possible. We will define a similar notion for $\mathcal{T}$-packing by matroid-inducing EHP-families $\mathcal{T}$ and prove a short lemma needed in the proof of Theorem 4.1.

Let $\mathcal{T}$ be a matroid-inducing EHP-family. Consider a graph G and a $\mathcal{T}$-packing $\mathcal{Q}$ of G. We denote by $V(\mathcal{Q})$ and $E(\mathcal{Q})$ the set of all vertices and edges of G belonging to some graph of $\mathcal{Q}$. Thus, $G_{\mathcal{Q}} = (V(\mathcal{Q}), E(\mathcal{Q}))$ is a subgraph of G whose components are copies of graphs from $\mathcal{T}$, and conversely, this graph uniquely determines $\mathcal{Q}$.

A vertex $v \in V(G)$ is called *fixed* w.r.t. $\mathcal{Q}$ if it is covered by a copy of K_2 or by a center of a propeller in $\mathcal{Q}$. For $\mathcal{Q}$, we define a packing $\mathcal{Q}_h$ by the following: $G_{\mathcal{Q}_h}$ is the subgraph of $G_{\mathcal{Q}}$ induced by nonfixed vertices. Obviously $\mathcal{Q}_h$ consists only of hypomatchable graphs (all hypomatchable graphs in $\mathcal{Q}$ and, for each propeller $P \in \mathcal{Q}$ with center c, all components of $P \setminus c$).

DEFINITION 4.2. *We say that a $\mathcal{T}$-packing $\mathcal{Q}$ of G is* economical *if there is no $\mathcal{T}$-packing $\mathcal{Q}'$ of G such that $\forall v \in V(G)$, if $deg_{\mathcal{Q}}(v) = 1$, then $deg_{\mathcal{Q}'}(v) = 1$, and*

(i) *$V(\mathcal{Q}') \supsetneq V(\mathcal{Q})$ and $\mathcal{Q}'_h \subseteq \mathcal{Q}_h$ or*

(ii) *$V(\mathcal{Q}') = V(\mathcal{Q})$ and $\mathcal{Q}'_h \subsetneq \mathcal{Q}_h$.*

Note that if a set $X \subseteq V(G)$ is saturable by some $\mathcal{T}$-packing, then there exists an economical $\mathcal{T}$-packing saturating X. Let $\mathcal{Q}$ be a $\mathcal{T}$-packing of G, such that each propeller used in $\mathcal{Q}$ is assumed to be rooted in an arbitrarily selected root ($\mathcal{Q}$ is an *arbitrarily rooted $\mathcal{T}$-packing*). For $\mathcal{Q}$, we define two packings $\mathcal{Q}_e, \mathcal{Q}_b$ by the following: $G_{\mathcal{Q}_e}$ is the subgraph of $G_{\mathcal{Q}}$ induced by fixed vertices and roots of propellers and $G_{\mathcal{Q}_b}$

is the subgraph of $G_{\mathcal{Q}}$ induced by vertices covered by hypomatchable graphs and blades of rooted propellers. Obviously $\mathcal{Q}_e$ contains only edges (copies of K_2 involved in $\mathcal{Q}$ and sticks of propellers from $\mathcal{Q}$) and $\mathcal{Q}_b \subseteq \mathcal{Q}_h$.

Given two rooted packings $\mathcal{Q}, \mathcal{Q}'$ we say that C is a *component* of $\mathcal{Q} \cup \mathcal{Q}'$ if it is a component of the graph $G_{\mathcal{Q}_e} \cup G_{\mathcal{Q}_b} \cup G_{\mathcal{Q}'_e} \cup G_{\mathcal{Q}'_b}$. Note that the components of $\mathcal{Q} \cup \mathcal{Q}'$ are not necessarily components of the graph $G_{\mathcal{Q}} \cup G_{\mathcal{Q}'}$, since they do not contain the edges connecting the centers and blades of propellers from $\mathcal{Q}, \mathcal{Q}'$ (in [8], where only edges and hypomatchable graphs are used in the packings, these two types of components are the same). The following lemma (in fact an extension of Theorem 3 from [8] proved by a similar technique) describes the components of $\mathcal{Q} \cup \mathcal{Q}'$ for two rooted economical $\mathcal{T}$-packings $\mathcal{Q}, \mathcal{Q}'$.

LEMMA 4.3. *Let $\mathcal{T}$ be a matroid-inducing EHP-family and let G be a graph. Let $\mathcal{Q}$ and $\mathcal{Q}'$ be two rooted economical $\mathcal{T}$-packings of G and let C be a component of $\mathcal{Q} \cup \mathcal{Q}'$. Then*

(i) *C contains at most one vertex that is uncovered by $\mathcal{Q}$;*

(ii) *C contains at most one graph from $\mathcal{Q}_b$;*

(iii) *if C contains exactly one graph from $\mathcal{Q}_b$, then $\mathcal{Q}$ saturates C.*

Proof. Let us call a path P *alternating* w.r.t. $\mathcal{Q}$ if it alternately contains edges of $\mathcal{Q}_e$ and $E(G) \setminus E(Q)$. An alternating path P with end vertices u and v is called *augmenting* if $\{u, v\} \cap V(\mathcal{Q}_e) = \varnothing$ and $\{u, v\} \subseteq H$ for no $H \in \mathcal{Q}_b$ (i.e., the end vertices of P do not belong to the same hypomatchable graph of $\mathcal{Q}_b$). It is easy to see that an economical packing does not admit an augmenting path.

For a contradiction, let C contain two graphs H_1, H_2, such that each H_i is either a single vertex uncovered by $\mathcal{Q}$ or a member of $\mathcal{Q}_b$. Assume that the distance between H_1, H_2 is a minimum. Then C contains a path P, such that P has end vertices in H_1 and H_2, and P does not contain any edge of a graph from $\mathcal{Q}_b$.

Let us denote by $j(P)$ the maximum number of consecutive edges of a graph from $\mathcal{Q}'_b$, which do not belong to any graph from $\mathcal{Q}$, involved in P. Let us assume that P is a path with minimum $j(P)$.

If $j(P) \leq 1$, then P is an augmenting path w.r.t. $\mathcal{Q}$, which is a contradiction. If P contains edges of at least two distinct graphs from $\mathcal{Q}'_b$, then P contains an augmenting path w.r.t. $\mathcal{Q}'$, which is a contradiction. Thus P contains edges of exactly one graph $H' \in \mathcal{Q}'_b$.

Let x be the first vertex (in the direction from H_1 to H_2) adjacent to two consecutive edges of H' in P that do not belong to $\mathcal{Q}$. Note that $x \in V(\mathcal{Q}_e)$; otherwise the beginning of P would be an augmenting path. Let $\mathcal{Q}''$ be the packing that arises from $\mathcal{Q}'$ by substituting H' with a perfect matching of $H' \setminus x$. Let P' be the (unique) path starting in x by the (unique) edge from $\mathcal{Q}_e$ and alternating w.r.t. both $\mathcal{Q}$ and $\mathcal{Q}''$. P' starts in x; denote by z the other end vertex. If $z \in H_1$, then $j(P)$ is not a minimum; otherwise we may find an augmenting path w.r.t. $\mathcal{Q}$ or $\mathcal{Q}'$, which is a contradiction. □

4.2. Structural lemma. Let H be a graph. For a vertex $x \in V(H)$, we will denote by H_x the graph that arises from H by adding a new vertex x_a and a new edge xx_a.

For a matroid-inducing EHP-family $\mathcal{T}$, $k(\mathcal{T})$ denotes the maximum index of a star included in $\mathcal{T}$. Note that if $k(\mathcal{T}) \geq 2$ ($\mathcal{T}$ contains the star S_2), then all hypomatchable graphs are $\mathcal{T}$-saturable. Hence all hypomatchable graphs and propellers different from stars may be avoided in every $\mathcal{T}$-packing.

If $\mathcal{T}$ is a family of graphs and H is a graph, then we will denote by $\mu_{\mathcal{T}}(H)$

the maximum number of vertices of H saturated by some $\mathcal{T}$-packing of H. If $\mathcal{T}$ is a matroid-inducing family, then it follows that every maximal $\mathcal{T}$-packing of H saturates exactly $\mu_{\mathcal{T}}(H)$ vertices.

LEMMA 4.4. *Let $\mathcal{T}$ be a matroid-inducing EHP-family. Let H be a graph that is neither $\mathcal{T}$-saturable nor hypomatchable. Then there exists a vertex $x \in V(H)$, such that $\mu_{\mathcal{T}}(H_x) \leq |H_x| - 2$.*

Proof. We will prove this lemma in two steps: In the first step we will find a vertex $x \in V(H)$ and an economical $\mathcal{T}$-packing $\mathcal{Q}$ of H_x with $|V(\mathcal{Q})| \leq |H_x| - 2$. In the second step we will prove that $\mathcal{Q}$ has to be maximal w.r.t. H_x. Because $\mathcal{T}$ is matroid-inducing, we will have $\mu_{\mathcal{T}}(H_x) = |V(\mathcal{Q})| \leq |H_x| - 2$.

(i) Let $\mathcal{N}$ be an economical maximal $\mathcal{T}$-packing of H. Because H is not $\mathcal{T}$-saturable, there exists a vertex $w \in V(H) \setminus V(\mathcal{N})$. We will color w red. Let v be a neighbor of w. Obviously $\mathcal{N}$ covers v by a copy of K_2 or by a center of a propeller.

Consider the graph H_v. If $\mathcal{N}$ is economical w.r.t. H_v, then we set $x = v$ and $\mathcal{Q} = \mathcal{N}$ and we are finished with the first part of the proof.

If $\mathcal{N}$ is not economical w.r.t. H_v, then there exists a $\mathcal{T}$-packing $\mathcal{N}'$ of H_v with

(o) $\forall v \in V(H)$: if $deg_{\mathcal{N}}(v) = 1$, then $deg_{\mathcal{N}'}(v) = 1$, and

(i) $V(\mathcal{N}') \supsetneq V(\mathcal{N})$ and $\mathcal{N}'_h \subseteq \mathcal{N}_h$ or

(ii) $V(\mathcal{N}') = V(\mathcal{N})$ and $\mathcal{N}'_h \subsetneq \mathcal{N}_h$.

If $v_a \notin V(\mathcal{N}')$, then $\mathcal{N}'$ is a $\mathcal{T}$-packing of H, proving that $\mathcal{N}$ is not economical w.r.t. H, which is a contradiction. Therefore $v_a \in V(\mathcal{N}')$.

If the star $S_2 \notin \mathcal{T}$, then every propeller from $\mathcal{T}$ has a unique vertex of degree one. Hence $\mathcal{N}, \mathcal{N}'$ may be unambiguously viewed as rooted $\mathcal{T}$-packings. Note that due to (o), $V(\mathcal{N}_e) \subseteq V(\mathcal{N}'_e)$. Let J be the (unique) maximal alternating path starting in v_a by the edge $v_a v$ and alternately containing edges of $\mathcal{N}'_e$ and $\mathcal{N}_e$. Since $V(\mathcal{N}_e) \subseteq V(\mathcal{N}'_e)$, J has an odd number of edges. Let z be the last vertex of J. We know that z is uncovered by $\mathcal{N}$ or covered by $\mathcal{N}_b$. If $z \neq w$, then by substituting the first edge $v_a v$ in J by the edge wv we get an augmenting path w.r.t. $\mathcal{N}$ in H, which is a contradiction. Therefore, the last vertex of J is w.

On the other hand, if $S_2 \in \mathcal{T}$, then we may assume that $\mathcal{N}, \mathcal{N}'$ consist of stars only. In this case, consider a sequence of edges J of maximal possible length starting in v_a by the edge $v_a v$ and alternately containing edges of $\mathcal{N}' \setminus \mathcal{N}$ and edges of $\mathcal{N} \setminus \mathcal{N}'$ leading to vertices of degree one. Note that the edges of $\mathcal{N}' \setminus \mathcal{N}$ are uniquely determined by the property (o). Let z be the last vertex of J. Due to (o), J cannot have an even number of edges. Thus J has an odd number of edges and z is either uncovered by $\mathcal{N}$ or a member of $\mathcal{N}_b$ or a center of a star $S_j, 1 \leq j < k(\mathcal{T})$, in $\mathcal{N}$ with all edges covered by $\mathcal{N}'$ or already in J. If the last possibility appears, then we may construct a new $\mathcal{T}$-packing of H contradicting the maximality of $\mathcal{N}$ by swapping edges and nonedges of $\mathcal{N}$ along $(J \setminus \{v_a v\}) \cup \{wv\}$. Otherwise, similarly as above, we conclude that $z = w$. Without loss of generality, assume that J is a path (cycles in J may be skipped).

In both cases we have found an odd cycle C in H. C consists of the edge wv and the path $J \setminus v_a$. We can conclude that none of the edges of C belongs to a propeller from $\mathcal{N}$ ($\mathcal{N}$ would not be economical w.r.t. H). Thus C contains only copies of K_2 from $\mathcal{N}$ and edges uncovered by $\mathcal{N}$. Let us color all vertices of C red. Note that for every red vertex y there exists an economical maximal $\mathcal{T}$-packing $\mathcal{N}^y$ of H skipping y and covering all other red vertices by copies of K_2.

Let us take a red vertex w' that has a nonred neighbor v'. Let $\mathcal{N}^{w'}$ be an economical maximal $\mathcal{T}$-packing of H skipping w' and covering all red vertices by copies of K_2. As above, either we will find that $\mathcal{N}^{w'}$ is economical also w.r.t. $H_{v'}$, or

we will find an odd cycle C' containing v' and w' and consisting of copies of K_2 from $\mathcal{N}^{w'}$ and edges uncovered by $\mathcal{N}^{w'}$. In the latter case let us again color all vertices of C' red and observe that (still) for every red vertex y there is an economical maximum $\mathcal{T}$-packing $\mathcal{N}^y$ of H skipping y and covering all other red vertices by copies of K_2.

Continuing analogously, we cannot finish with all vertices colored red. Then H would be hypomatchable (for every [red] vertex there would be a $\mathcal{T}$-packing skipping it and covering all other [red] vertices by copies of K_2). Hence, in the ith step we will find a red vertex $w^{(i)}$, its nonred neighbor $v^{(i)}$, and a $\mathcal{T}$-packing $\mathcal{N}^{w^{(i)}}$ of H, which is economical w.r.t. $H_{v^{(i)}}$ and skips $w^{(i)}$ and the newly added vertex $v_a^{(i)}$. Let us set $x = v^{(i)}$ and $\mathcal{Q} = \mathcal{N}^{w^{(i)}}$. The first step of the proof is finished.

(ii) We have found a vertex $x \in V(H)$ and an economical maximal $\mathcal{T}$-packing $\mathcal{Q}$ of H, which is economical also w.r.t. H_x and skips at least the newly added vertex x_a and one more neighbor u of x. If $\mathcal{Q}$ is not maximal w.r.t. H_x, then there is a $\mathcal{T}$-packing $\mathcal{L}'$ of H_x with $V(\mathcal{L}') \supsetneq V(\mathcal{Q})$. Let $\mathcal{L}$ be an economical $\mathcal{T}$-packing of H_x covering the same set of vertices as $\mathcal{L}'$. If $x_a \notin V(\mathcal{L})$, then $\mathcal{L}$ proves that $\mathcal{Q}$ is not maximal w.r.t. H, which is a contradiction. Thus $\mathcal{L}$ covers x_a. Assume $\mathcal{L}$ and $\mathcal{Q}$ are arbitrarily rooted. Let B be the component of $\mathcal{L} \cup \mathcal{Q}$ containing x_a. According to Lemma 4.3, B does not contain any other vertex that is uncovered by $\mathcal{Q}$. In particular, $u \notin V(B)$. Let D be the component containing u. We know that $D \setminus u$ is saturated by $\mathcal{Q}$ and that $G_{\mathcal{Q}} \cap D$ does not contain any graph from $\mathcal{Q}_b$. Let $\mathcal{Q}'$ be a $\mathcal{T}$-packing that arises from $\mathcal{L}$ by replacing the edge xx_a by xu, by replacing all graphs of $\mathcal{L}$ intersecting D with graphs (edges) of $\mathcal{Q}_e$ intersecting D, and by substituting the newly constructed graphs with their perfect $\mathcal{T}$-packings where necessary. It may be simply observed that $\mathcal{Q}'$ is a correctly defined $\mathcal{T}$-packing of H with $V(\mathcal{Q}') \supseteq V(\mathcal{Q})$ and so $\mathcal{Q}$ is not maximal w.r.t. H, which is a contradiction.

Thus $\mathcal{Q}$ has to be maximal w.r.t. H_x. Because $\mathcal{Q}$ skips at least two vertices of H_x, Lemma 4.4 is proved. □

4.3. Proof of the negative part of Theorem 4.1. Let $\mathcal{T}$ be a matroid-inducing EHP-family and let H be a graph that is neither $\mathcal{T}$-saturable nor hypomatchable nor a propeller. To prove Theorem 4.1 we need to show that $\mathcal{T} \cup \{H\}$ is not a matroid-inducing family. We will proceed by discussing the structure of H and introducing counterexamples. For each type of a bad graph H we will find a counterexample G, such that $M(G, \mathcal{T} \cup \{H\})$ is not a matroid. Moreover, we will always find two bases of different cardinality. Let us introduce an auxiliary claim about bases, analogous to the claim introduced on page 344 in [8].

CLAIM 4.1. *Let H be a graph and let $x \in V(H)$ be a vertex with less than $k(\mathcal{T})$ neighbors of degree one. Let $G = H_x$ and let B_2^x be a base of $M(G, \mathcal{T} \cup \{H\})$ such that $x, x_a \in B_2^x$, and for each $w \in V(H_x)$ with $j_w > 0$ neighbors of degree one, there are $n = \min(j_w, k(\mathcal{T}))$ vertices $y_1, \ldots, y_n \in V(H_x)$ of degree one, such that $wy_i \in E(H)$ and $y_i \in B_2^x$ for each i.*

If $\mathcal{Q}$ is a $\mathcal{T} \cup \{H\}$-packing of G corresponding to B_2^x, then $\mathcal{Q}$ uses no copy of H.

Proof of Claim 4.1. Assume $\mathcal{Q}$ uses a copy H' of H. As $|H'| \leq |B_2^x| \leq |H|+1$, we have $B_2^x = V(H')$. If the vertex x has h neighbors of degree one in H $(0 < h < k(\mathcal{T}))$, then we get a contradiction, because H' would have more vertices with $h+1$ neighbors of degree one than H.

If x has no neighbors of degree one, then let us denote by $c(H)$ the set of all vertices of H that have neighbors of degree one. Let $dst(H) = \sum_{u,v \in c(H)} (1 + dist(u, v))$, where $dist(u, v)$ is the distance between u and v. We get $dst(H') > dst(H)$, which is again

a contradiction. □

Let us start by discussing the structure of H: According to Lemma 4.4 we know that if H is neither $\mathcal{T}$-saturable nor hypomatchable, then there exists a vertex $x \in V(H)$, such that $\mu_{\mathcal{T}}(H_x) \leq |H_x| - 2$. Let us denote by $A(H)$ the set of all such vertices of H. There are several cases.

Case 1 (analogous to Case 1 of [8]). There exists a vertex $a \in A(H)$ with fewer than $k(\mathcal{T})$ neighbors of degree one.

Let $G = H_a$ (see Figure 2(c)). Let B_1 be a base of $M(G, \mathcal{T} \cup \{H\})$ containing $V(H)$. We know that $|B_1| \geq |G| - 1$ using a copy of H. Let B_2^a be the base defined in Claim 4.1. By this claim, B_2^a uses no copy of H and thus every $(\mathcal{T} \cup \{H\})$-packing corresponding to B_2^a is in fact a $\mathcal{T}$-packing. Because $a \in A(H)$, we have $|B_2^a| \leq |H_a| - 2 < |B_1|$, and so G is a counterexample with two bases of different cardinality.

If Case 1 does not occur, then every vertex from $A(H)$ has at least $k(\mathcal{T})$ neighbors of degree one. Let us denote the set of all such neighbors by $B(H)$. On the other hand, the following claim also holds.

Claim 4.2. *If $a \in V(H)$ has $k \geq k(\mathcal{T})$ neighbors of degree one, then $a \in A(H)$.*

Proof of Claim 4.2. Let B be the set of $k \geq k(\mathcal{T})$ neighbors of a of degree one in H. If $a \notin A(H)$, then $\mu_{\mathcal{T}}(H_a) \geq |H_a| - 1$. Let $\mathcal{Q}$ be a maximal $\mathcal{T}$-packing of H_a covering all vertices from B. We know that $\mathcal{Q}$ skips a_a; otherwise $S_{k+1} \in \mathcal{T}$. Hence $\mathcal{Q}$ saturates H, which is a contradiction. □

Let us follow the discussion and introduce other cases.

Case 2 (analogous to Case 2 of [8]). $\mu_{\mathcal{T}}(H \setminus b) \leq |H| - 3$ for some $b \in B(H)$. Denote by a the unique neighbor of b in H. Consider the graph H_b (vertex b_a and edge bb_a were added to H). Let $\mathcal{Q}$ be a maximal $\mathcal{T}$-packing of H_b covering b_a.

If $\mathcal{Q}$ covers the edge bb_a by a copy of K_2, then $|V(\mathcal{Q})| = |H_b| - 2$, and so $b \in A(H)$. Since b has no neighbors of degree one, we may use Case 1.

If $S_2 \in \mathcal{T}$ and $\mathcal{Q}$ covers b_a by a copy abb_a of S_2, then $\mathcal{Q}$ skips all of the $k(\mathcal{T}) - 1 \geq 1$ neighbors of a with degree one different from b. Moreover, we may observe that the number of vertices skipped by $\mathcal{Q}$ is strictly greater than 1; otherwise H is $\mathcal{T}$-saturable. Hence $|V(\mathcal{Q})| \leq |H_b| - 2$ and so $b \in A(H)$, which leads to Case 1.

It remains to inspect maximal $\mathcal{T}$-packings of H_b that cover bb_a by a 1-propeller different from S_2. Without loss of generality we may suppose that $S_2 \notin \mathcal{T}$ (otherwise the use of propellers different from stars could be eliminated). Let $\mathcal{Q}$ be a maximal $\mathcal{T}$-packing of H_b covering bb_a by a 1-propeller $P \neq S_2$. We know that $|V(\mathcal{Q})| \geq |H_b| - 1$; otherwise we can use Case 1. Let $\mathcal{N}$ be a $\mathcal{T}$-packing of H that arises from $\mathcal{Q}$ by substituting P with a perfect matching of $P \setminus b_a$. Obviously $|V(\mathcal{N})| \geq |H| - 1$, and because H is not $\mathcal{T}$-saturable, $\mathcal{N}$ is maximal w.r.t. H and $|V(\mathcal{N})| = |H| - 1$.

Let $d \neq b$ be an arbitrary vertex adjacent in P to a. We may observe that d has no neighbor of degree one in H. Such a vertex d' could not be in P, because hypomatchable graphs do not contain vertices of degree one, but we could easily enlarge $V(\mathcal{Q})$ by d', which is a contradiction with the maximality of $\mathcal{Q}$. Consider the graph H_d. $\mathcal{N}$ is a $\mathcal{T}$-packing of H_d skipping two vertices. We will show that $\mathcal{N}$ is maximal w.r.t. H_d and so $d \in A(H)$.

If $\mathcal{N}$ is not maximal w.r.t. H_d, then there exists a $\mathcal{T}$-packing $\mathcal{L}$ of H_d with $V(\mathcal{L}) \supsetneq V(\mathcal{N})$. If $d_a \notin V(\mathcal{L})$, then $\mathcal{L}$ proves that $\mathcal{N}$ is not maximal w.r.t. H, which is a contradiction. So $d, d_a \in V(\mathcal{L})$, and because $|V(\mathcal{N})| = |H| - 1$, we get $|V(\mathcal{L})| \geq |H_d| - 1$. Let us pay attention to the edge ab: $\mathcal{L}$ must cover the edge ab by a copy of K_2 or by a propeller R with $d \notin V(R)$. We will construct a new

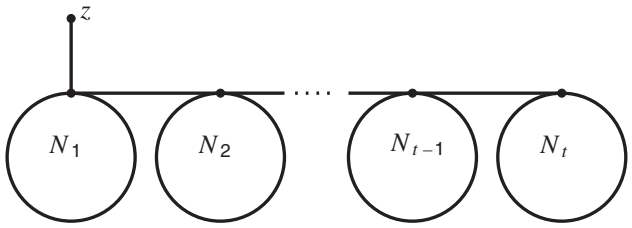


FIG. 1. *Case* 3*: Decomposition of* H.

$\mathcal{T}$-packing $\mathcal{L}'$: $\mathcal{L}'$ arises from $\mathcal{L}$ by replacing the edges $d_a d$ and ab with the edge ad and by substituting the new graph covering a with its perfect $\mathcal{T}$-packing if necessary. It can be simply verified that $\mathcal{L}'$ is a correctly defined $\mathcal{T}$-packing skipping b with $|V(\mathcal{L}')| \geq |H| - 2$, which contradicts the assumption of Case 2.

We have shown that $\mathcal{N}$ is maximal w.r.t. H_d. Since $|V(\mathcal{N})| = |H_d| - 2$ and $\mathcal{T}$ is a matroid-inducing family, we get $d \in A(H)$. Because d has no neighbor of degree one in H, we may use Case 1. This completes Case 2.

Case 3 (analogous to Case 3 of [8]). $\mu_{\mathcal{T}}(H \setminus b) = |H| - 2$ for every $b \in B(H)$. Let t be the maximum integer such that H can be decomposed as in Figure 1, where $z \in B(H)$ and $\bigcup_{i=s}^{t}(N_i)$ is not $\mathcal{T}$-saturable for any $s = 1, \ldots, t$. Let us construct a counterexample G such that $M(G, \mathcal{T} \cup \{H\})$ is not a matroid: G arises from two copies H_1, H_2 of H (vertices of each H_i are indexed by i) by glueing the vertex z_2 of H_2 to the unique neighbor c_1 of z_1 in H_1. (The construction from Case 3 of [8] was used; see Figure 2(d).)

Let B_1 be a base of $M(G, \mathcal{T} \cup \{H\})$ containing $V(H_1)$. Then $|B_1| \geq |G| - 1$ using a copy H_1 of H and a maximum $\mathcal{T}$-packing of $H_2 \setminus z_2$. Let B_2 be a base containing $V(H_2)$. Assume that $|B_2| = |B_1|$. Because neither H_2 nor $H_2 \setminus z_2$ is $\mathcal{T}$-saturable, we have to use a copy H' of H with $H' \cap H_2 \neq \varnothing$. Denote $T_2 = H' \cap H_2$. Let $(z_2, N_1, \ldots, N_t)$ be the decomposition of H_2 as in Figure 1. Denote $N_i' = N_i \cap T_2$. Because $H_2 \setminus T_2$ must be $\mathcal{T}$-saturable, T_2 has to intersect every N_i, and so $N_i' \neq \varnothing$, $i = 1, \ldots, t$. Moreover, $\bigcup_{i=s}^{t}(N_i')$ is not $\mathcal{T}$-saturable for any $s = 1, \ldots, t$ (in particular, $T_2 = \bigcup_{i=1}^{t}(N_i')$ is not $\mathcal{T}$-saturable).

By a cardinality argument H' contains the vertex $x = z_2 = c_1$. Denote $T_1 = H' \cap H_1$. If x has $k(\mathcal{T})$ neighbors $r_1, \ldots, r_k$ of degree one in T_1, then $x \in A(H')$ and $(r_1, T_1 \setminus r_1, N_1', \ldots, N_t')$ is a decomposition of H' into $t+1$ parts. Because $r_1 \in B(H')$, by the assumption of Case 3, $H' \setminus r_1 = (T_1 \setminus r_1) \cup \bigcup_{1}^{t}(N_i')$ is not $\mathcal{T}$-saturable and we get a contradiction with the maximality of t.

If x has less than $k(\mathcal{T})$ neighbors of degree one in T_1, then without loss of generality assume that H' does not contain z_1. Thus $(H_1 \setminus z_1) \setminus H'$ has to be $\mathcal{T}$-saturable and so $H' \cap (H_1 \setminus \{z_1, x\})$ is non-$\mathcal{T}$-saturable. Hence $\mu_{\mathcal{T}}(H' \setminus x) = |H'| - 3$. The graph $T_1 = H' \cap H_1$ is in this case also non-$\mathcal{T}$-saturable since $\mu(H_1 \setminus z_1) = |H_1| - 2$. We will prove that $x \in A(H')$ and then, because x has less then $k(\mathcal{T})$ neighbors of degree one in H', we will use Case 1.

For a contradiction, let $x \notin A(H')$. Then there exists a maximal $\mathcal{T}$-packing $\mathcal{Q}$ of H_x' with $|V(\mathcal{Q})| \geq |H_x'| - 1$. The added vertex $x_a \in V(\mathcal{Q})$; otherwise H' is $\mathcal{T}$-saturable. If $\mathcal{Q}$ covers the new edge xx_a by a copy of K_2, then either T_1 or T_2 is $\mathcal{T}$-saturable, which is a contradiction. If $\mathcal{Q}$ covers the new edge xx_a by a propeller P, then by Heredity of $\mathcal{T}$, one of the graphs T_1, T_2, $T_1 \setminus x$, $T_2 \setminus x$ is $\mathcal{T}$-saturable, which is a contradiction. This completes Case 3.

Case 4. $\mu_{\mathcal{T}}(H) = |H| - 1$, and there exists a vertex $a \in A(H)$ and a component D of $H \setminus a$ that is saturated by every maximal $\mathcal{T}$-packing of H. Note that in this case $H \setminus D$ cannot be saturated by any $\mathcal{T}$-packing of H; otherwise a base containing $V(H \setminus D)$ would lead to a contradiction. Because all vertices from $A(H)$ have neighbors of degree one, there must exist a vertex $d \in D \setminus A(H)$. Thus $\mu_{\mathcal{T}}(H_d) \geq |H_d| - 1$. Assume that the distance between a, d is maximum possible. Let us construct a graph G such that $M(G, \mathcal{T} \cup \{H\})$ is not a matroid: G arises from two copies H_1, H_2 of H (vertices of each H_i are indexed by i) and one new vertex r_0 by adding the edge $r_0 d_1$ and edges $a_2 x_1$ for every $x_1 \in V(H_1)$ such that $x_1 d_1 \in E(H_1)$ (see Figure 2(e)). Let us find two bases of $M(G, \mathcal{T} \cup \{H\})$ with different cardinality.

Let B_2 be a base containing $V(H_2)$. Then $|B_2| \geq |G| - 1$ using two copies H_1, H_2 of H (vertex r_0 will remain uncovered). Let B_1 be a base containing $V(H_1) \cup \{r_0\}$. Since $H_1 \setminus D_1$ is not saturated by any $\mathcal{T}$-packing of H_1, B_1 has to use a copy H' of H intersecting $H_1 \setminus D_1$ in a nonempty and non-$\mathcal{T}$-saturable subgraph. Obviously $a_1 \in V(H')$, and so all $k(\mathcal{T})$ neighbors of a_1 with degree one from H_1 are in $V(H')$. According to Claim 4.2, $a_1 \in A(H')$. Denote $T_1 = H' \cap (H_1 \setminus D_1)$. We know that every maximal $\mathcal{T}$-packing of H' skips exactly one vertex. The skipped vertex is always in T_1; otherwise there exists a $\mathcal{T}$-packing of H_1 saturating $H_1 \setminus D_1$, which is a contradiction. So $H' \setminus T_1$ is saturated by every maximum $\mathcal{T}$-packing of H'.

If H' does not intersect H_2, then by a cardinality argument $d_1 \in V(H')$. Thus $r_0 \in V(H')$ and there has to be a vertex $y \in V(H_1) \setminus V(H')$. Note that the degree of y in H_1 must be 1; otherwise, similarly as in Claim 4.1, $dst(H') > dst(H)$. Vertex y has to be covered by a graph containing the edge ya_2, and so y has to be a neighbor of d_1. If $\mu_{\mathcal{T}}(H_{1,y}) \geq |H_{1,y}| - 1$, then we have a contradiction with the supposed maximality of the distance between a, d. Thus $\mu_{\mathcal{T}}(H_{1,y}) \leq |H_{1,y}| - 2$ and so $y \in A(H_1)$. Because y has no neighbors of degree one in H_1, we may use Case 1.

If H' intersects H_2, then $a_2 \in V(H')$. If $|B_1| = |B_2| \geq |G| - 1$, then $H' \setminus a_2$ must have a non-$\mathcal{T}$-saturable component $B \subseteq H_2 \setminus a_2$. Let $\mathcal{Q}$ be a maximal $\mathcal{T}$-packing of H'. We know that $\mathcal{Q}$ saturates all the vertices of $H' \setminus T_1$. Hence $\mathcal{Q}$ saturates B and so there is a graph $L \in \mathcal{Q}$ covering a_2 and intersecting B in a nonempty subgraph. If $|L \cap B| > 1$, then $L \cap B$ is a part of a hypomatchable subgraph or a propeller and there exists a vertex $w \in L \cap B$ such that $\mu_{\mathcal{T}}(H' \setminus w) \geq |H' \setminus w| - 1$. Therefore, $\mu_{\mathcal{T}}(H'_w) \geq |H'_w| - 1$, which contradicts the supposed maximality of the distance between a and d. Thus $L \cap B = \{v\}$ (a single vertex). The vertex v has no neighbors of degree one in H' (these could not be covered by $\mathcal{Q}$). If $\mu_{\mathcal{T}}(H'_v) \geq |H'_v| - 1$, then v contradicts the supposed maximality of the distance between a and d. Otherwise $v \in A(H')$, and because v has no neighbors of degree one, we may use Case 1. We have proved that $|B_1| \neq |B_2|$, which proves that $\mathcal{T} \cup \{H\}$ is not a matroid-inducing family. This is the end of Case 4.

Case 5. $\mu_{\mathcal{T}}(H) = |H| - 1$, and there are vertices $b \in B(H)$ and $x \in V(H)$, such that $\mu_{\mathcal{T}}(H \setminus b) = |H| - 1$ and $\mu_{\mathcal{T}}(H \setminus x) = |H| - 2$.

Let $a \in A(H)$ be the (unique) neighbor of b. We know that $x \neq a$; otherwise H is $\mathcal{T}$-saturable. The counterexample graph G is constructed as follows: G arises from two copies H_1, H_2 of H (vertices of each H_i are denoted by the index i) by glueing the vertex a_2 to x_1 (see Figure 2(f)). Let B_2 be a base containing $V(H_2)$. $|B_2| \geq |G| - 1$ using a copy of H and a maximal $\mathcal{T}$-packing of $H_1 \setminus x_1$. Let B_1 be a base containing $V(H_1)$ and let $\mathcal{N}$ be a $\mathcal{T} \cup \{H\}$-packing associated with B_1. Because neither H_1 nor $H_1 \setminus a_2$ is $\mathcal{T}$-saturable, $\mathcal{N}$ has to use a copy H' of H intersecting $H_1 \setminus a_1$ in a nonempty and non-$\mathcal{T}$-saturable subgraph. By a cardinality argument,

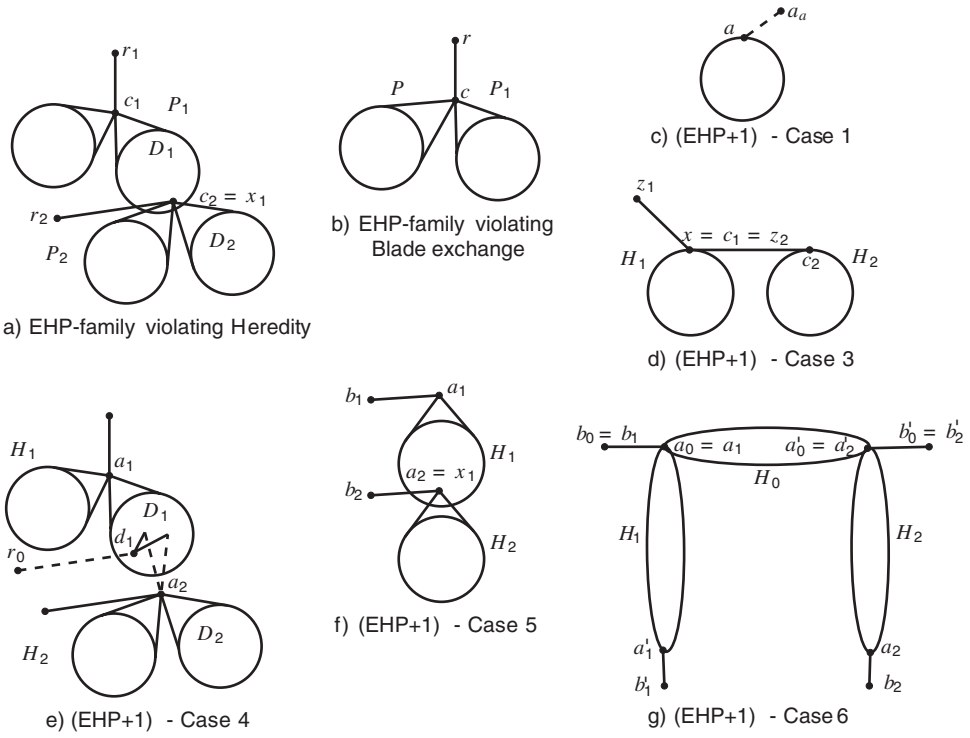


FIG. 2. *Summary of counterexamples.*

$a_2 \in V(H')$. If H' does not contain all the neighbors of a_2 from $B(H_2)$, then without loss of generality $\mathcal{N}$ skips b_2. If $|B_1| \geq |G| - 1$, then $V(\mathcal{N}) = V(G \setminus b_2)$. Let $\mathcal{Q}$ be a maximal $\mathcal{T}$-packing of H'. We know that $\mathcal{Q}$ skips exactly one vertex of H', and so either $H' \cap H_1 \setminus a_2$ or $H' \cap H_2 \setminus a_2$ is saturated by $\mathcal{Q}$. By Heredity, one of the graphs $H_1, H_2, H_1 \setminus a_2, H_2 \setminus a_2 b_2$ is $\mathcal{T}$-saturable, which is a contradiction.

If H' contains all the neighbors of $a_2 (= x_1)$ from $B(H_2)$, then $a_2 \in A(H')$. Note that every $\mathcal{T}$-packing of H' skips exactly one vertex in $H' \cap (H_1 \setminus x_1)$; otherwise H_1 or $H_1 \setminus x_1$ is $\mathcal{T}$-saturable by Heredity. Thus $H' \cap (H_2 \setminus a_2) \neq \varnothing$ is saturated by every maximal $\mathcal{T}$-packing of H' and we may use Case 4. This completes Case 5.

Case 6 (analogous to Case 4 of [8]). There are two vertices $a, a' \in A(H)$, such that if $b \in B(H)$ is a neighbor of a or a', then $H \setminus b$ is $\mathcal{T}$-saturable. If L is a graph and $\mathcal{Q}$ is its $\mathcal{T}$-packing, then we call the size of $|V(L) \setminus V(\mathcal{Q})|$ the *defect of* $\mathcal{Q}$. The *defect of* L is the defect of a maximum $\mathcal{T}$-packing of L. Let $b, b' \in B(H)$ be neighbors of a, a', respectively. Let us denote by m the defect of $H \setminus \{a, b, a', b'\}$, and by n, n' the defects of $H \setminus \{a, b\}$ and $H \setminus \{a', b'\}$, respectively.

At first assume that $m < n + n'$, and let us construct a counterexample graph G: G arises from three copies H_0, H_1, H_2 of H (vertices of each H_i are indexed by i) by glueing vertices a_0 to a_1, b_0 to b_1, a'_0 to a'_2, and b'_0 to b'_2 (and deleting the multiple edges). For each $i \in \{1, 2\}$ we will denote $I_i = H_i \setminus H_0$. (The construction from Case 4 of [8] was used; see Figure 2(g).)

Let B_1 be a base containing $V(H_1) \cup V(H_2)$. Then $|B_1| \geq |G| - m$. Let B_2 be a base containing $V(H_0)$. Because H_0 is not $\mathcal{T}$-saturable, B_2 has to use a copy H' of H intersecting H_0. If $H' = H_0$, then $|B_2| = |G| - (n + n') < |G| - m \leq |B_1|$, which

proves that $\mathcal{T}$ is not matroid-inducing.

If $H' \neq H_0$, then $|H' \cap I_1| \geq 1$ or $|H' \cap I_2| \geq 1$. Without loss of generality let $|H' \cap I_1| \geq 1$. Note that H' has to cover all neighbors of a_0 of degree one in H_0, and so according to Claim 4.2, $a_0 \in A(H')$. If there does not exist a $\mathcal{T}$-packing of H' saturating $H' \cap H_0$, then $H' \cap I_1$ is saturated by every maximum $\mathcal{T}$-packing of H'. Because $H' \cap I_1$ contains one or more components of $H' \setminus a_0$ and $a_0 \in A(H')$, we can use Case 4. If there exists a $\mathcal{T}$-packing of H' saturating $H' \cap H_0$, then by Heredity of $\mathcal{T}$, $H' \cap H_0$ is $\mathcal{T}$-saturable. Hence $H_0 \setminus H'$ is not $\mathcal{T}$-saturable and B_2 has to use another copy H'' of H with $|H'' \cap H_2| > 0$, $|H'' \cap H_0| > 0$, and $(H'' \cap H_0)$ not $\mathcal{T}$-saturable. Similarly as above, $a_0' \in A(H'')$. It follows that $H'' \cap H_2$ is saturated by every maximum $\mathcal{T}$-packing of H'', and so we may use Case 4.

It remains to prove that $m \geq n + n'$ does not occur: If Case 5 does not hold, then for every $x \in V(H)$, the defect of $H \setminus x$ is either 0 or at least 2. Note that if the defect is at least 2, then x is covered by an edge or by a center of a propeller in each maximal $\mathcal{T}$-packing of H. Let $\mathcal{Q}$ be an economical maximal $\mathcal{T}$-packing covering both ab and $a'b'$. Assuming that K_2 is a "0-propeller," the vertices a, a' have to be covered by centers of propellers in $\mathcal{Q}$. Let us denote the propellers covering a, a' by P, P', respectively, and assume that P, P' are rooted in b, b', respectively, and that $\mathcal{Q}$ is a $\mathcal{T}$-packing with minimum sum of indexes of P, P'. Let $\mathcal{Q}'$ be a $\mathcal{T}$-packing of H constructed from $\mathcal{Q}$ by substituting all blades of the two aforementioned rooted propellers by their maximum matchings (one vertex in every blade will remain uncovered). It is simple to prove that $\mathcal{Q}'$ is an economical $\mathcal{T}$-packing of H (generally, every $\mathcal{T}$-packing that arises from an economical $\mathcal{T}$-packing by substituting some hypomatchable graphs and blades of propellers by their maximum matchings is economical). Note that there is one more vertex $y \notin V(P) \cup V(P')$ skipped by $\mathcal{Q}'$: y is the vertex skipped by the original $\mathcal{T}$-packing $\mathcal{Q}$. If the sum of the indexes of P, P' was a minimum, then the defect of $\mathcal{Q}'$ is exactly m, since in this case $\mathcal{Q}'$ induces a maximal $\mathcal{T}$-packing in $H \setminus \{a, b, a', b'\}$.

Similarly, we will construct two other $\mathcal{T}$-packings $\mathcal{N}_1, \mathcal{N}_1'$ of $H \setminus \{ab\}$ and $H \setminus \{a'b'\}$, respectively. $\mathcal{N}_1$ and $\mathcal{N}_1'$ arise from $\mathcal{Q}$ by replacing blades of only one rooted propeller P or P', respectively, by their maximum matchings and by skipping the edge $ab, a'b'$, respectively. If both $\mathcal{N}_1, \mathcal{N}_1'$ are maximal, then obviously $m < n + n'$.

Let us assume that without loss of generality $\mathcal{N}_1$ is not maximal. Let $\mathcal{N}_2$ be an economical maximal $\mathcal{T}$-packing of $H \setminus ab$ with $V(\mathcal{N}_2) \supsetneq V(\mathcal{N}_1)$. The $\mathcal{T}$-packing $\mathcal{N} = \mathcal{N}_2 \cup \{ab\}$ is an economical $\mathcal{T}$-packing of the whole H. Assume that $\mathcal{Q}$ and $\mathcal{N}$ are arbitrarily rooted.

Subcase 1. If $y \in \mathcal{N}$, then every vertex skipped by $\mathcal{N}$ lies in a component of $\mathcal{Q} \cup \mathcal{N}$ intersected by P in $\mathcal{Q}$. Let $\mathcal{C} = \{C_1, \ldots, C_t\}$ be a collection of all such components and let $V(\mathcal{C})$ denote the set of all vertices in all C_i. We may construct a perfect $\mathcal{T}$-packing of H by exchanging the graphs of $\mathcal{N}$ intersecting $V(\mathcal{C}) \cup \{a, b\}$ with graphs of $\mathcal{Q}$ intersecting $V(\mathcal{C}) \cup \{a, b\}$ (we will use a subpropeller of P) and by replacing the newly constructed graphs by their perfect $\mathcal{T}$-packings where necessary, which is a contradiction.

Subcase 2. If $y \notin \mathcal{N}$, then $\mathcal{N}$ covers a vertex z covered by a blade of P in $\mathcal{Q}$. Since Case 5 does not occur, $H \setminus z$ is $\mathcal{T}$-saturable. Let $\mathcal{D}$ be an economical maximal rooted $\mathcal{T}$-packing of H saturating $V(H) \setminus z$. We will subsequently modify $\mathcal{N}$ and $\mathcal{Q}$ to obtain a contradiction. Throughout the sequence of modifications, we will maintain the following invariant.

(I) At most one vertex skipped by $\mathcal{N}$ (denoted by y) is not covered by a propeller with center a in $\mathcal{Q}$. If y exists, then z is covered by a propeller with center a in $\mathcal{Q}$.

In the beginning, (I) is satisfied. There are several situations.

Situation A. If y, z are in the same component C of $\mathcal{D} \cup \mathcal{N}$, then by replacing all graphs (edges) of $\mathcal{N}$ in C with the graphs (edges) of $\mathcal{D}$ in C, we get Subcase 1.

Situation B. If the component C of $\mathcal{D} \cup \mathcal{N}$ containing y is intersected by a hypomatchable graph in $\mathcal{D}$, then we get a contradiction with the maximality of $\mathcal{N}_2$.

Situation C. If the component C of $\mathcal{D} \cup \mathcal{N}$ containing y is intersected by a blade B of a rooted propeller (W, r) with center $c \neq a$ in $\mathcal{D}$, then let us focus on the vertex c. If c is uncovered by $\mathcal{N}$ or covered by a hypomatchable graph, by a root or a blade of a rooted propeller, or by a copy of K_2 in $\mathcal{N}$, then we get a contradiction with the maximality of $\mathcal{N}_2$. Thus c is a center of a rooted propeller (W', r') in $\mathcal{N}$. If $D(W', r') \subseteq D(W, r)$, then we may enlarge $\mathcal{N}_2$ to saturate $V(C)$, which is a contradiction. Otherwise, by Blade exchange there exists a blade $B' \in D(W', r') \setminus D(W, r)$ such that the graph induced by $V(W' \setminus B') \cup V(B)$ is $\mathcal{T}$-saturable. Let us modify $\mathcal{N}$ by replacing W' with a perfect $\mathcal{T}$-packing of $V(W' \setminus B') \cup V(B)$, by replacing the graphs of $\mathcal{N}$ in C with the graphs of $\mathcal{D}$ in C, and by replacing B' with a maximum matching of $B' \setminus x_1$, where $x_1 \in V(B')$ is arbitrary. After this modification, let us newly consider $y = x_1$, observe that (I) is satisfied, and continue our sequence of modifications according to the current situation.

Situation D. The last situation occurs when the component C of $\mathcal{D} \cup \mathcal{N}$ containing y is intersected by a blade B of a rooted propeller (W, r) with center a in $\mathcal{D}$. Without loss of generality we may suppose that $y \in V(B)$ and $C = B$, since $\mathcal{N}$ may be simply modified to satisfy this condition.

If there is a blade Y of P with $V(Y) \cap V(B) \neq \varnothing$, then let $x_2 \in V(Y) \cap V(B)$ and let us modify $\mathcal{N}$ by replacing the graphs of $\mathcal{N}$ in B with a perfect matching of $B \setminus x_2$. After this modification we are in Subcase 1.

If B does not intersect any blade of P, then according to Blade exchange, there is a blade D of P such that the graph induced by $V(P \setminus D) \cup V(B)$ is $\mathcal{T}$-saturable. Let us modify $\mathcal{Q}$ by replacing the graphs of $\mathcal{Q}$ with the graphs of $\mathcal{D}$ in the component of $\mathcal{Q} \cup \mathcal{D}$ containing B and by replacing P with a perfect $\mathcal{T}$-packing of the graph induced by $V(P \setminus D) \cup V(B)$ and with a perfect matching of $D \setminus x_3$, where $x_3 \in V(D)$ is arbitrarily selected if $\mathcal{N}$ saturates D and is the unique vertex of D skipped by $\mathcal{N}$ otherwise.

If $\mathcal{N}$ saturates D (this occurs, e.g., when $z \in V(D)$), then this modification leads to Subcase 1. If $\mathcal{N}$ does not saturate $V(D)$, then let us newly consider $y = x_3$ and observe that (I) is satisfied. Hence we may continue our sequence of modifications according to the current situation.

For a rooted $\mathcal{T}$-packing $\mathcal{L}$, we denote by $D(\mathcal{L})$ the set of all graphs induced by a blade and a center of any propeller included in $\mathcal{L}$. The above sequence of modifications is finite, since in each step the size of one of $D(\mathcal{D}) \setminus D(\mathcal{Q})$, $D(\mathcal{D}) \setminus D(\mathcal{N})$ decreases. We end our sequence in Subcase 1 or by a contradiction with the maximality of $\mathcal{N}_2$. This concludes the proof of Case 6.

It remains to prove that the list of cases is complete. If H is a connected graph which is neither $\mathcal{T}$-saturable nor hypomatchable, then according to Lemma 4.4, $A(H) \neq \varnothing$. If none of Cases 1, 2, and 3 holds, then there is a vertex $b \in B(H)$ such that $H \setminus b$ is $\mathcal{T}$-saturable. Because H is not $\mathcal{T}$-saturable, we have $\mu_{\mathcal{T}}(H) = |H| - 1$. Let $a \in A(H)$ be the (unique) neighbor of b in H. If any of the components of $H \setminus a$ is saturated by every maximum $\mathcal{T}$-packing of H, then we can use Case 4. So for every component of $H \setminus a$ there exists a maximum $\mathcal{T}$-packing of H skipping one of its vertices. Let us iterate through all the components of $H \setminus a$. For every component D

we will take a maximum $\mathcal{T}$-packing of H skipping one of the vertices $x \in V(D)$. We will color x red and follow the coloring algorithm described informally in the proof of Lemma 4.4 until all vertices of D are red or until we find a vertex $a' \in V(D) \cap A(H)$. If we finish with all vertices in all components colored red, then every component of $H \setminus a$ is hypomatchable and H is a propeller, which was completely addressed in section 3. Note that this way we can find not only a vertex $a' \in A(H), a' \neq a$, but also a maximal $\mathcal{T}$-packing $\mathcal{Q}$ of H skipping exactly one of the neighbors of a'. Let us observe that for every neighbor $b' \in B(H)$ of a', $\mathcal{Q}$ can be easily changed to a $\mathcal{T}$-packing skipping exactly b'. We have found two vertices $a, a' \in A(H)$, such that if $b \in B(H)$ is a neighbor of a or a', then $H \setminus b$ is $\mathcal{T}$-saturable, which can be solved by Case 6. This concludes the proof of Theorem 4.1.

5. Conclusion. We have introduced a full characterization of EHP-families of graphs $\mathcal{T}$ such that $M(G, \mathcal{T})$ is a matroid for every graph G. Moreover, we have fully characterized the enlargements of matroid-inducing EHP-families by one graph H by proving that $\mathcal{T} \cup \{H\}$ is a matroid-inducing family if and only if H is $\mathcal{T}$-saturable or $\mathcal{T} \cup \{H\}$ is a matroid-inducing EHP-family.

The paper studies the matroidal aspects of the $\mathcal{T}$-packing problem. Many other results have been recently extended from matching to packing by EHP-families. The most important results are those concerning complexity. These were introduced by Loebl and Poljak in [9] and [10].

Acknowledgment. The author thanks the referees for many helpful comments and Martin Loebl for helpful consultations.

REFERENCES

[1] G. Cornuejols and D. Hartvigsen, *An extension of matching theory*, J. Combin. Theory Ser. B, 40 (1986), pp. 285–296.
[2] G. Cornuejols, D. Hartvigsen, and W. R. Pulleybank, *Packing subgraphs in a graph*, Oper. Res. Lett., 1 (1982), pp. 449–467.
[3] J. Edmonds, *Paths, trees and flowers*, Canad. J. Math, 17 (1965), pp. 449–467.
[4] J. Edmonds and D. R. Fulkerson, *Transversals and matroid partition*, J. Res. Nat. Bur. Standards Sect. B, 69 (1965), pp. 147–153.
[5] P. Hell and D. Kirkpatrick, *Packing by cliques and by finite families of graphs*, Discrete Math., 49 (1984), pp. 118–133.
[6] M. Janata, *Packing Matroids*, Master's thesis, Charles University, Prague, 2000.
[7] M. Las Vergnas, *An extension of Tutte's* 1*-factor problem*, Discrete Math, 23 (1978), pp. 241–255.
[8] M. Loebl and S. Poljak, *On matroids induced by packing subgraphs*, J. Combin. Theory Ser. B, 44 (1988), pp. 338–355.
[9] M. Loebl and S. Poljak, *Good family packing*, in Fourth Czechoslovak Symposium on Combinatorics, Graphs and Complexity, J. Nešetřil and M. Fiedler, eds., Elsevier, 1992, pp. 181–186.
[10] M. Loebl and S. Poljak, *Efficient subgraph packing*, J. Combin. Theory Ser. B, 59 (1993), pp. 106–121.

SIAM J. DISCRETE MATH.
Vol. 18, No. 3, pp. 542–553

PRECOLORING EXTENSIONS OF BROOKS' THEOREM*

MICHAEL O. ALBERTSON†, ALEXANDR V. KOSTOCHKA‡, AND DOUGLAS B. WEST§

Abstract. Let G be a connected graph with maximum degree k (other than a complete graph or odd cycle), let W be a precolored set of vertices in G inducing a subgraph F, and let D be the minimum distance in G between components of F. If the components of F are complete graphs and $D \geq 8$ (for $k \geq 4$) or $D \geq 10$ (for $k = 3$), then every proper k-coloring of F extends to a proper k-coloring of G. If the components of F are single vertices and $D \geq 8$, and the vertices outside W are assigned color lists of size k, then every k-coloring of F extends to a proper coloring of G with the color on each vertex chosen from its list. These results are sharp.

Key words. coloring extension, list coloring, Brooks' Theorem

AMS subject classification. 05C15

DOI. 10.1137/S0895480103425942

1. Introduction. For $k \geq 3$, the famous theorem of Brooks [6] states that a graph with maximum degree k is k-colorable if it does not have K_{k+1} as a component. Our general aim in this paper is to strengthen this result by allowing some vertices to have arbitrarily specified colors.

Albertson [1] proved that if a set W of vertices in an r-colorable graph is separated pairwise by distance at least 4, then every coloring of W from a set of $r + 1$ colors extends to a proper $(r+1)$-coloring of W. The result was generalized by letting $G[W]$ be a disjoint union of complete graphs with at most j vertices, where $G[W]$ denotes the subgraph of G induced by W. If the components of $G[W]$ are far enough apart, then every proper $(r + 1)$-coloring of $G[W]$ extends to a proper $(r + 1)$-coloring of G. Albertson showed that distance $6j - 2$ is enough, Kostochka (see [2]) lowered the threshold to $4j$, and Albertson and Moore [2] showed that distance $3j$ suffices when $j = r$.

These results require an extra color; generally a partial coloring of an r-chromatic graph may not extend to a proper r-coloring, regardless of the distance between the precolored vertices. Can the extension conclusions be strengthened when more colors are allowed? With $\Delta(G) + 1$ colors allowed, every partial proper coloring extends, even in a list coloring sense. That is, if each uncolored vertex has a list of $\Delta(G) + 1$ available colors, then we can extend a partial coloring in an arbitrary vertex order; when we reach a vertex, there is always a color in its list that has not already been used on any neighbor. What happens when only $\Delta(G)$ colors are allowed?

THEOREM 1.1. *Let W be a set of vertices in a graph G with $\Delta(G) \geq 4$ and $K_{\Delta(G)+1} \not\subseteq G$. If the components of $G[W]$ are complete graphs and the distance between any two such components is at least 8, then every proper $\Delta(G)$-coloring of*

*Received by the editors April 14, 2003; accepted for publication (in revised from) February 2, 2004; published electronically February 25, 2005. The work of the second and third authors was supported in part by the NSF under award DMS-0099608 and by the NSA under award MDA904-03-1-0037, respectively.

http://www.siam.org/journals/sidma/18-3/42594.html

†Department of Mathematics, Smith College, Northampton, MA 01063 (albertson@math.smith.edu).

‡Department of Mathematics, University of Illinois, Urbana, IL 61801, and Institute of Mathematics, Novosibirsk, Russia (kostochk@math.uiuc.edu).

§Department of Mathematics, University of Illinois, Urbana, IL 61801 (west@math.uiuc.edu).

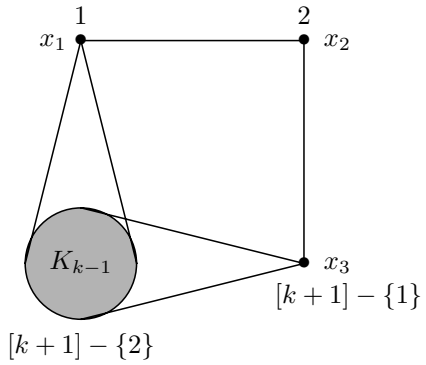


FIG. 1. *Failure of list coloring extension.*

$G[W]$ extends to a proper $\Delta(G)$-coloring of G. The same statement holds for $\Delta(G) = 3$ with 10 in place of 8.

We will shortly present examples showing that Theorem 1.1 is sharp, except that distance 8 suffices when $\Delta(G) = 3$ if the components of $G[W]$ are isolated vertices (as guaranteed by Theorem 1.2 below).

In the special case when G has chromatic number $\Delta(G)$, Theorem 1.1 provides an extension theorem using no "extra" colors. As discussed in [3], such results are rare. Also, in comparison to the earlier result of Kostochka, Theorem 1.1 shows that with $\Delta(G)$ colors, the sizes of the components of $G[W]$ are irrelevant, and there is a constant distance that suffices.

Our second result, proved together with the first, is a list version of the theorem when W is an independent set. This result was also proved independently by Axenovich [4].

THEOREM 1.2. *Let W be a set of vertices in a graph G with $\Delta(G) \geq 3$. Let L be a function that assigns to each vertex a list of $\Delta(G)$ available colors. If the distance between any two vertices of W is at least 8, then every coloring of W chosen from the lists extends to a proper coloring f of G such that $f(v) \in L(v)$ for all $v \in V(G)$.*

Using the word "list" for the set of colors available for a vertex is standard in this setting. A function L assigning a list to each vertex is a *list assignment* for a graph G, and a proper coloring f such that $f(v) \in L(v)$ for all $v \in V(G)$ is an *L-coloring*. Since L can assign the same list of $\Delta(G)$ colors at each vertex, Theorem 1.2 strengthens the special case of Theorem 1.1 where the components of $G[W]$ are single vertices. Since the claim is made for each choice of colors on W, we may view the precoloring on W as lists of size 1. In discussing lists, it is helpful to use the notation $[k]$ for the set $\{1, \ldots, k\}$.

When $G[W]$ is not an independent set, no list extension theorem is possible. For $k \geq 2$, consider the graph shown in Figure 1. It consists of a path with vertices x_1, x_2, x_3 in order and a copy of K_{k-1} whose vertices are adjacent to x_1 and x_3. All vertices have degree k, except that x_2 has degree 2. The colors on x_1 and x_2 are specified as 1 and 2, respectively. Let $L(x_3) = [k+1] - \{1\}$, and let $L(v) = [k+1] - \{2\}$ for each v outside $\{x_1, x_2, x_3\}$. In a proper extension of the coloring on $\{x_1, x_2\}$, some color j outside $\{1, 2\}$ must be used on x_3. Since the remaining vertices have list $[k+1]-\{2\}$, no color in $\{1, 2, j\}$ can be used on these vertices, which leaves only $k-2$ available colors in $[k+1]$ for the copy of K_{k-1}.

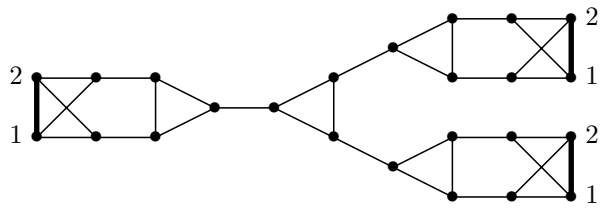


FIG. 2. *Failure of* Δ*-extension when* $\Delta(G) = 3$ *and distance is* 9.

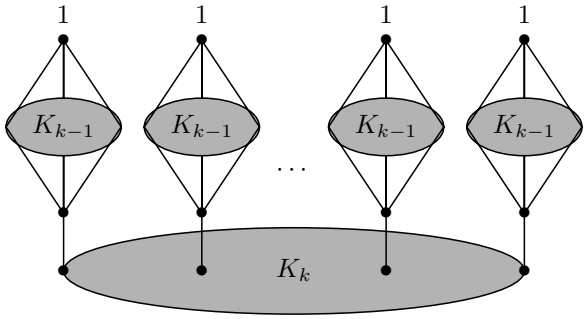


FIG. 3. *Failure of extension for distance* 7.

For $\Delta(G) = 2$, there is no extension theorem with $\Delta(G)$ colors, since extendibility of a coloring of two points on a long path depends on the parity of the distance between them.

For $\Delta(G) = 3$, the graph in Figure 2 shows that the distance threshold of 10 for the extension theorem with nontrivial cliques is sharp. Here the precolored set W consists of the three peripheral 2-cliques shown in bold. A proper 3-coloring that extends this must have the third color on each vertex neighboring the central triangle, but then the coloring cannot be extended to the center. The distance between two precolored cliques is 9.

For $\Delta(G) \geq 4$ in Theorem 1.1 and $\Delta(G) \geq 3$ in Theorem 1.2, the graph in Figure 3 shows that the distance threshold of 8 is sharp. In the list case, we use the same list $[k]$ on all vertices of $G - W$. To construct G, first let H consist of K_{k+1} with one edge deleted and instead a pendant edge attached to one of the deficient vertices. Let G consist of k disjoint copies of H plus edges making the pendant vertices in the copies of H into a clique. Let W consist of the vertices of degree $k-1$, all given the same color. Although G has maximum degree k, and the distance between any two vertices of W is 7, this coloring of W does not extend to a proper k-coloring of G.

The proofs of the upper bounds use many common ideas, so we develop them together. The general approach is to derive contradictory properties for a minimal counterexample.

2. Background and preliminaries. A precoloring extension problem can be modeled as a list coloring problem. The colors on the precolored vertices are removed from the lists of colors available at their neighbors. If the number of colors available at a vertex is at least its degree, then this remains true after the precolored vertices are deleted, because at most one color is lost for each neighbor deleted.

We use $d_G(v)$ or simply $d(v)$ to denote the degree of a vertex v in a graph G (all our graphs are simple). Also $d_G(u,v)$ or $d(u,v)$ denotes the distance between vertices u and v in G. We extend this notation to vertex subsets: $d_G(A,B)$ is the minimum of the distances in G between a vertex of A and a vertex of B.

A graph G is *degree-choosable* if it has an L-coloring whenever L is a list assignment with $|L(v)| \geq |d(v)|$ for all $v \in V(G)$. We say that such a list assignment is *supervalent*.

Given a supervalent list assignment L for G, Vizing [9] showed that if the size of some list exceeds the degree of the vertex, or if G is 2-connected and the lists are not identical, then G has an L-coloring. This and its consequence that a connected graph having a degree-choosable induced subgraph is also degree-choosable are easy to prove.

These observations lead to the characterization of degree-choosable graphs by Erdős, Rubin, and Taylor [7]: A connected graph fails to be degree-choosable if and only if it is a *Gallai tree*, which is a connected graph in which every block is a complete graph or an odd cycle. Furthermore, the lists in a supervalent list assignment not permitting a proper coloring have a restricted form.

THEOREM 2.1 (Borodin [5], Erdős–Rubin–Taylor [7]). *If L is a supervalent list assignment for a connected graph G and there is no L-coloring of G, then*

(a) *$|L(v)| = d(v)$ for every $v \in V(G)$.*
(b) *G is a Gallai tree.*
(c) *$L(v) = \cup_{B \in \mathcal{B}(v)} L_B$ for all $v \in V(G)$, where $\mathcal{B}(v)$ is the set of blocks containing v, and for each block B, L_B is a set of $\chi(B) - 1$ colors.*

A short proof of Theorem 2.1 appears in [8]. Note that each block B is an $|L_B|$-regular graph, and all vertices of a single block that are not cut-vertices of G have the same list.

Henceforth let $k = \Delta(G)$, and let f denote a precoloring of W. In the setting of Theorem 1.1, which we call the *clique case*, f is a proper k-coloring of $G[W]$ using the set $[k]$ as colors. In the setting of Theorem 1.2, which we call the *list case*, the coloring f is any proper coloring of $G[W]$. We discuss both cases together as a list coloring problem by defining $L(v) = [k]$ for all $v \in V(G) - W$ in the clique case. Hence the "theorem statement" refers to both theorems together. Let $N_G(v)$ denote the neighborhood of a vertex v in a graph G.

CLAIM 1. *Let G be a graph with maximum degree k, precoloring f, lists L, and precolored set W. If G is a smallest counterexample to the theorem statement, then the following hold for the graph H defined by $H = G - W$ and the list assignment L_f on $V(H)$ defined by $L_f(v) = L(v) - f(N_G(v) \cap W)$.*

(a) *H is connected.*
(b) *Every component of H is a Gallai tree, and in every block the lists L_f on the non-cut-vertices are the same and have size equal to vertex degree.*
(c) *If $v \in V(H)$, then $d_G(v) = k$.*

Proof. (a) When W is a separating set in G, extension of the coloring to the various components of $G - W$ is independent, and deleting one does not violate the hypotheses of the theorem. By the minimality of G, we may therefore assume that H is connected.

(b) An L_f-coloring for H would permit the extension of the coloring for G, so there is no L_f-coloring for H. Since $|L(v)| = k$ and we lose at most one color for each lost neighbor, $|L_f(v)| \geq d_H(v)$ for all $v \in V(H)$. Hence L_f is supervalent, and H has no L_f-coloring, so Theorem 2.1 applies to H and immediately yields the claim.

(c) For $v \in V(H)$,

$$d_H(v) = |L_f(v)| = |L(v) - f(N_G(v) \cap W)| \geq |L(v)| - |N_G(v) \cap W|.$$

Since $d_G(v) = d_H(v) + |N_G(v) \cap W|$, we obtain $d_G(v) \geq |L(v)| = k$. Since $\Delta(G) = k$, equality holds. □

Henceforth we maintain the notation (k, f, L, W, H, L_f) and assumptions (G is a smallest counterexample) of Claim 1. By the *distance requirement*, we mean the hypothesis that the distance between components of $G[W]$ is at least 8 in general and is at least 10 when $k = 3$ and we are in the clique case.

Remarks. The computation in the proof of Claim 1(c) implies that the colors used on neighbors of v in W are distinct and appear in $L(v)$. By the distance requirement, $N_G(v) \cap W$ lies in a single component of $G[W]$. In the clique case, $|N_G(v) \cap W| \leq k-1$, since $K_{k+1} \not\subseteq G$. In the list case, $\delta(H) \geq k-1$, since W consists of isolated vertices.

We next consider the edges joining $V(H)$ and W. A *leaf block* in a graph H is a block of H containing at most one cut-vertex of H. For a block B in H, we henceforth let B' denote the set of vertices in B that are not cut-vertices of H.

Claim 2. *Let B be a leaf block of H in a smallest counterexample G, and let $m = |V(B)|$.*

(a) *The neighbors in W of vertices in B lie in the same component of $G[W]$; call it $Q(B)$.*

(b) *Every vertex in $Q(B)$ is adjacent to all or none of the set B' of non-cut-vertices in B.*

(c) *B is a complete graph, and $Q(B)$ has exactly $k-m+1$ vertices with neighbors in B and at most one vertex with no neighbors in B.*

(d) *H has more than one block.*

Proof. (a) This follows immediately from the distance requirement, since all vertices of B except possibly one have neighbors in W (by Claim 1(c)), and the distance between neighbors of adjacent vertices of B is at most 3.

(b) By Claim 1(b), the lists under L_f are the same for all $v \in B'$; let S be this common list. By Theorem 2.1(a), $|S| = d_H(v)$. By part (a), $N_G(v) \cap W$ lies in a single component of $G[W]$, so the colors assigned to its vertices by f are distinct. In the clique case, $L(v) = [k]$, so arriving at S requires each vertex of B' to lose the same colors from its list.

In the list case, $|L(v)| = k$ for $v \in B'$, and $Q(B)$ consists of only one vertex. Also $K_{k+1} \not\subseteq G$ implies $d_H(v) < k$, so each $v \in B'$ loses one color from its list. Thus the one vertex of $Q(B)$ is adjacent to all of B'.

(c) If B is a cycle of length at least 5, then by Claim 1(c) and part (a), each vertex of B' has two neighbors in B and $k-2$ neighbors in $Q(B)$. By part (b), these neighbors in $Q(B)$ have degree at least $k-3+4$, since $|B'| \geq 4$. This degree would exceed $\Delta(G)$.

Hence B is a complete graph, by Claim 1(b). The vertices of B' have $m-1$ neighbors in H. By Claim 1(c), they have $k-m+1$ neighbors in $Q(B)$. By part (b), these are always the same $k-m+1$ vertices.

These $k-m+1$ vertices in $Q(B)$ have $k-m$ neighbors among themselves and at least $m-1$ neighbors in B', so they have at most one more neighbor in $Q(B)$.

(d) If H has only one block, then the first statement in part (c) makes it a complete graph, but the second then yields $K_{k+1} \subseteq G$, which is forbidden. □

3. Leaf blocks. We begin with a tool for studying the structure of leaf blocks of H. As before, B' is the set of vertices in B that are not cut-vertices of H.

CLAIM 3. *There is no partial extension of f to a partial coloring f' that gives the same color to two neighbors of an uncolored vertex of H.*

Proof. Let f' be such an extension, and let U be the set of vertices outside W to which f' assigns colors. Note that $f'(u) \in L_f(u)$ is required for all $u \in U$. Let $G' = G - W - U$; note that G' is an induced subgraph of H.

For $v \in V(G')$, let $L_{f'}(v) = L(v) - f(N_G(v) \cap (W \cup B'))$. By the same argument as for L_f, we have $|L_{f'}(v)| \geq d_{G'}(v)$ for all $v \in V(G')$. Also, if $x \in V(G')$ has neighbors with the same color under f', then $|L_{f'}(x)| > d_{G'}(x)$. By Theorem 2.1, G' then has an $L_{f'}$-coloring, which yields an L-coloring of G. Since G has no L-coloring, there is no such f'. □

When B is a leaf block of H, we let x_B denote the cut-vertex of G contained in B. If B has m vertices, then Claim 2(c) yields $|Q(B)| \in \{k-m+1, k-m+2\}$. Define B to have *Type j* when $|Q(B)| = k-m+j$. In the list case, always $|Q(B)| = 1$, which requires that $m = k$ and that $B \cong K_k$ and B has Type 1.

Let $Q'(B)$ denote the set of vertices in $Q(B)$ having neighbors in $V(B)$. If B has Type 1, then $Q'(B) = Q(B)$, and each vertex of $Q(B)$ may have one neighbor that is not in $B' \cup Q(B)$. If B has Type 2, then vertices of $Q'(B)$ have no such additional neighbors, and we let w_B denote the vertex of $Q(B) - Q'(B)$.

CLAIM 4. *If B is a leaf block of Type 2, then x_B has no neighbor in $Q(B)$.*

Proof. By Claim 2, $B' \cup Q'(B)$ is a clique of size k. Since also $Q'(B) \subseteq N_G(w_B)$, x_B has no neighbor in $Q'(B)$. Finally, since leaf blocks of Type 2 occur only in the clique case, every extension of f to B' uses on B' all the colors not used on $Q'(B)$, including $f(w_B)$. If x_B is adjacent to w_B, then we have formed a partial extension of f that is forbidden by Claim 3. □

CLAIM 5. *In the clique case, if B is a leaf block of Type 1 and $y \in N_H(x_B) - B'$, then y has no neighbor in $Q(B)$.*

Proof. Suppose that $wy \in E(G)$, where $w \in Q(B)$. Since w also has $k-1$ neighbors in $B' \cup Q(B)$, we conclude that $x_B w \notin E(G)$. Since $B' \cup Q(B)$ is a clique, we can extend f to B by using on B' the colors of $[k] - f(Q(B))$, and then we can use color $f(w)$ on x_B. This partial extension gives the same color to two neighbors of y, which violates Claim 3. Hence no such edge wy exists. □

CLAIM 6. *Let B_1 and B_2 be distinct leaf blocks in H such that $Q(B_1) = Q(B_2)$.*

(a) *If $Q'(B_1) \cap Q'(B_2) = \varnothing$, then B_1 and B_2 have Type 2 with k vertices, and $|Q(B_1)| = 2$.*

(b) *If $Q'(B_1) \cap Q'(B_2) \neq \varnothing$, then B_1 and B_2 have Type 1 with two vertices, and $|Q(B_1)| = k-1$ (also $Q'(B_1) = Q'(B_2) = Q(B_1)$).*

(c) *The condition in the hypothesis arises only in the clique case.*

(d) *There is no third leaf block B_3 with $Q(B_3) = Q(B_1)$.*

(e) $x_{B_1} \neq x_{B_2}$.

Proof. (a) By Claim 2(c), each $Q(B_i)$ has at most one vertex not in $Q'(B_i)$, and it is the only candidate for $Q'(B_{3-i})$. Since $|B_i' \cup Q'(B_i)| = k$, the sizes are as claimed.

(b) Consider $u \in Q'(B_1) \cap Q'(B_2)$. For $i \in \{1,2\}$, since $u \in Q'(B_i)$, there is at most one neighbor of u outside $B_i' \cup Q'(B-i)$, and such a neighbor exists in B_{3-i}'. Hence B_i has Type 1, and $Q'(B_i) = Q(B_i)$. Also $|B_{3-i}'| = 1$, so B_{3-i} has two vertices and $Q(B_{3-i}) = k-1$.

(c) The conclusions above yield $|Q(B_i)| > 1$, which occurs only in the clique case. The two possibilities are shown in Figure 4.

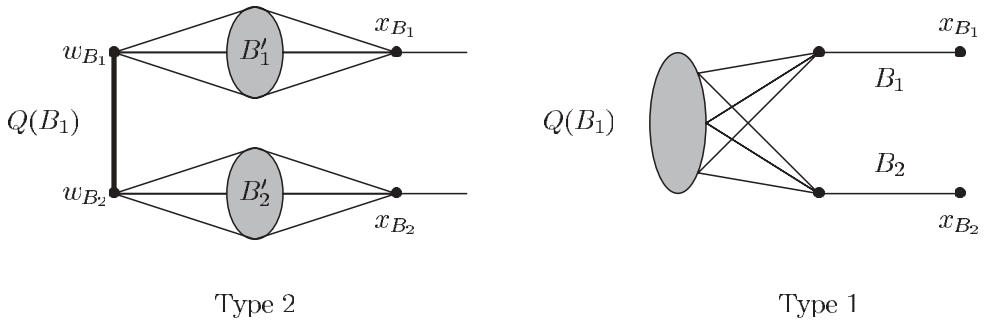


FIG. 4. *Leaf blocks B_1 and B_2 with $Q(B_1) = Q(B_2)$.*

(d) If such a B_3 exists and B_1 has Type 2, then applying part (a) to the pairs $\{B_1, B_2\}$ and $\{B_1, B_3\}$ gives $2k-1$ neighbors to w_{B_1}. If B_1 has Type 1, then applying part (b) to these pairs gives $k+1$ neighbors to each vertex of $Q(B_1)$.

(e) Suppose that $x_{B_1} = x_{B_2}$. If these blocks have Type 2, then $d_H(x_B) \geq 2k-2$, which exceeds k when $k > 2$. If they have Type 1, then let b_i be the unique vertex of B'_i. Both b_1 and b_2 have neighborhood $Q(B) \cup \{x_B\}$. Since we are in the clique case, there is only one choice of color when extending f to b_1 and b_2. Since these both neighbor x_B, Claim 3 implies that no such partial extension exists. □

4. Remote blocks. The *block-cutpoint graph* of a graph H has a vertex for each block in H and a vertex for each cut-vertex of H, and a cut-vertex v is adjacent to a block B if $v \in V(B)$. The block-cutpoint graph of a connected graph H is a tree, and its leaves correspond to blocks in H.

We continue to discuss a smallest counterexample G, with notation as defined in the preceding section. Let T be the block-cutpoint tree of H. We define a *remote block* in H to be a block corresponding to a vertex of maximum eccentricity in T. Our strategy will be to work our way in from a remote block, restricting the structure of H as we go.

CLAIM 7. *A remote block in H intersects only one other block in H.*

Proof. Let B be a remote block in H. If x_B lies in two non-remote blocks, then B is not remote, so at most one block containing x_B is non-remote. If at least two blocks other than B contain x_B, then at least one is a remote block C. Since neighbors of x_B in B and C have neighbors in W, the distance requirement yields $Q(C) = Q(B)$. Now $x_C = x_B$ contradicts Claim 6(e). □

When B is a remote block in H, we let $F(B)$ denote the other block sharing x_B. At this point $F(B)$ may be a complete graph or an odd cycle.

CLAIM 8. *Let B be a remote block in H. If C is a leaf block in H, and $d_H(x_B, x_C)$ is 1 when B has Type 1 and is at most 3 when B has Type 2, then $F(B) \cong K_2$, unless $k = 3$ and $F(B) \cong K_3$, as on the left in Figure 5.*

Proof. By the distance requirement, $Q(B) = Q(C)$. Now Claim 6 implies that B and C have the same Type and that this is the clique case. If B and C have Type 2, then Claim 6(a) yields $|V(B)| = k$. Hence x_B has $k-1$ neighbors in B and only one in $F(B)$, as desired.

Hence we may assume that B and C have Type 1. By Claim 6(b), $|V(B)| = |V(C)| = 2$. Since $x_B x_C \in E(H)$, Claim 5 implies that x_C has no neighbor in $Q(B)$. Thus x_C has $k-1$ neighbors in $F(B)$, so $|V(F(B))| \geq k$.

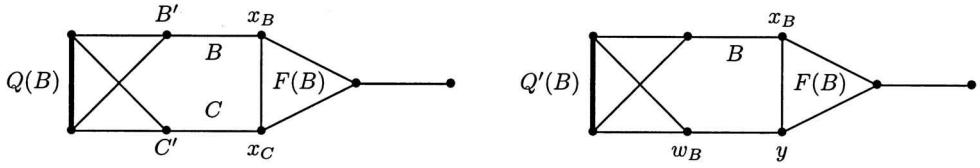


FIG. 5. *Exceptions in Claims* 8 *and* 9 *when* $k = 3$.

Claim 6(b) also yields $|Q(B)| = k - 1$. In addition to $k - 2$ neighbors within $Q(B)$, each vertex of $Q(B)$ has neighbors in B' and C'. Indeed, one neighbor is added for each neighbor of x_B or x_C in $F(B)$ that belongs to a leaf block. Hence there are no such vertices other than x_B and x_C. Also, since B is a remote block, $F(B)$ has at most one vertex belonging to a non-remote block other than $F(B)$. Since $\{x_B, x_C\}$ has at least two additional neighbors when $|V(F(B))| \geq 4$, the requirement of $|V(F(B))| \geq k$ yields $k = 3$. In that remaining case, the configuration is as on the left in Figure 5. □

CLAIM 9. *If* B *is a remote block in* H*, then* $F(B) \cong K_2$*, except that* $F(B)$ *may have three vertices as in Figure* 5 *in the clique case with* $k = 3$.

Proof. In the list case, $|Q(B)| = 1$, which forces $|V(B)| = k$. By Claim 2(c), $B \cong K_k$. Hence x_B has only one neighbor outside B. This argument applies to all leaf blocks in the list case, not just the remote ones.

Now consider the clique case. Let y be a vertex of $F(B)$ other than x_B. If B has Type 1, then Claim 5 and the distance requirement imply that y has no neighbors in W. Since $d_G(y) = k$ (by Claim 1(c)), y is a cut-vertex of H. Since B is remote, at most one vertex of $F(B)$ belongs to a non-remote block other than $F(B)$. Hence if x_B has more than one neighbor in $F(B)$, then Claim 8 implies that in fact it has only one such neighbor, unless $k = 3$ and the configuration is as on the left in Figure 5.

Otherwise, B has Type 2. Since $d_G(y) = k$ (by Claim 1(c)), and y has at most one neighbor in $Q(B)$ (namely, w_B), we conclude that y is a cut-vertex of H unless it has $k - 1$ neighbors in $F(B)$ and is adjacent to w_B.

This requires that $F(B) \cong K_k$ or that $k = 3$ and $F(B)$ is an odd cycle. In either case, x_B has $k - 1$ neighbors in $F(B)$ and only one in B, so $|V(B)| = 2$ and $|Q'(B)| = k - 1$ (by Claim 2(c)). Hence w_B has at most one neighbor outside $Q'(B)$, so at most one vertex of $F(B)$ fails to be a cut-vertex. Also, at most one vertex of $F(B)$ belongs to a non-remote block in H other than $F(B)$. Hence some vertex of $F(B)$ within distance 2 of x_B belongs to a remote block and Claim 8 finishes the proof, unless $k = 3$ and $F(B) \cong K_3$. In that remaining case, we may again have $F(B)$ with three vertices, as on the right in Figure 5. □

The two exceptional configurations in Figure 5 are essentially the same. In the clique case with $k = 3$, only the colors 1, 2, 3 can be used. When $Q'(B)$ is precolored, the common neighbors of these two vertices must have the third color. Thus it does not matter whether w_B in Figure 5 is precolored or not; either way, every extension uses $f(Q'(B))$ on $\{x_B, y\}$, and the third color is forced on the remaining vertex of $F(B)$. However, in transforming the problem we must avoid decreasing the distance between components of $G[W]$; hence we may assume that the exceptional case occurs only in Type 1, as on the left in Figure 5.

This exceptional case is in fact the building block and argument used in the example of Figure 2, showing that distance 9 is not enough for the extension theorem when $k = 3$.

5. Nearly remote blocks. Working in from a remote block B in H, we now consider the less remote vertex in $F(B)$. Based on Claim 9, we say that $F(B) \cong K_2$ is the *usual case*, while $F(B) \cong K_3$ with $k = 3$ is the *exceptional case*, which we may assume occurs only when B has Type 1.

In both the usual and exceptional cases, let y_B denote the unique vertex of $F(B)$ that is farthest from $Q'(B)$. Also define a *branching path* to be a path in H whose edges lie in distinct blocks.

CLAIM 10. *If B is a remote block of H, then H cannot have two leaf blocks reached from $F(B)$ along branching paths in H that exit y_B on different edges and have length at least 3. The same conclusion holds when the edges leaving y_B are in different blocks and the paths have length at least 2.*

Proof. Suppose that the claim fails, and C_1 and C_2 are two such leaf blocks. If two blocks of H are joined by a branching path in H of length l, then the distance between them in the block-cutpoint tree T is $2l$. Depending on whether the paths from $F(B)$ to C_1 and C_2 depart from y_B using edges of the same block B^* (solid edges) or different blocks B_1 and B_2 (dashed edges), the subgraph of T consisting of the paths among B, C_1, and C_2 is as shown in Figure 6. For any block of H, the distance to one of $\{C_1, C_2\}$ in T will exceed the distance to B. This contradicts the remoteness of B, so there is no such pair $\{C_1, C_2\}$. The same argument holds in the dashed case with paths shorter by one block and cut-vertex. □

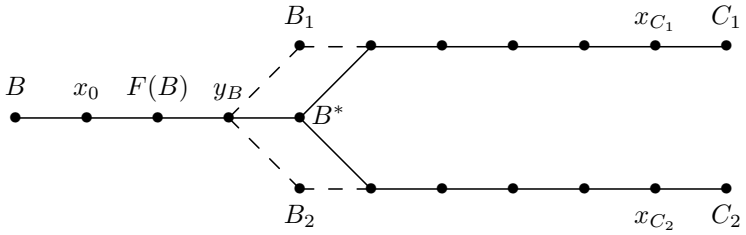


FIG. 6. *Portion of T involving distant blocks from B.*

CLAIM 11. *If B is a remote block of H in the usual case, and C is a leaf block of H, then $d_H(B, C) \geq 4$.*

Proof. Suppose that $d_H(B, C) \leq 3$. Since the cut-vertex contained in a leaf block of H has distance at most 2 from W, the distance requirement yields $Q(B) = Q(C)$. By Claim 6, this occurs only in the clique case, with B and C occurring as B_1 and B_2 in Figure 4. Claim 6(e) implies that $d_H(B, C) \geq 1$.

If B is Type 1, then Claim 6(b) implies that x_B has one neighbor in B and none in $Q(B)$, and Claim 9 implies that x_B has one neighbor in $F(B)$. Hence $d_G(x_B) = 2 < k$, which contradicts Claim 2. We conclude that B and C have Type 2 as on the left in Figure 4, and the block sharing x_C with C is a single edge. (In particular, Type 1 for such blocks B and C occurs only in the exceptional case.)

Since $d_G(v) = k$ for all $v \in V(H)$, every vertex in H has a neighbor in W or is a cut-vertex of H. If we follow a branching path from y_B starting in a block incident to y_B, we eventually reach a vertex of a leaf block. Claim 6(d) and the distance requirement imply that a branching path reaching a leaf block other than B or C takes at least three steps from y_B. By Claim 10, there is at most one such leaf block. Hence y_B has at most one neighbor not in $F(B)$ or along its path to C.

By Claim 9, $F(B) \cong K_2$. Since $d_H(B, C) \geq 1$, no leaf block is incident to y_B.

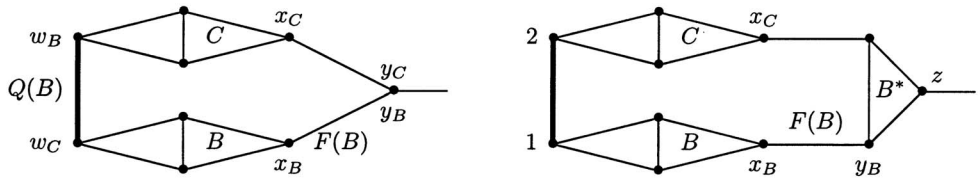


FIG. 7. *Exclusion of leaf blocks near a remote block B.*

Hence y_B has $k-1$ neighbors in H other than x_B. If $k \geq 4$, then we can pick two of them not along the path to C, which we have just shown cannot occur. Hence we may assume that $k = 3$.

If $d_H(B, C) = 1$, then $y_B = x_C$. Now $H = B \cup F(B) \cup C$ and $Q(B) = W$ and the precoloring extends. Hence $d_H(B, C) > 1$, and y_B lies in no leaf block.

If $d_H(B, C) = 2$, then $y_B \in N_H(x_B) \cap N_H(x_C)$. Since y_B has exactly 1 neighbor in the blocks it shares with each of x_B and x_C, it has one other neighbor in H, as shown on the left in Figure 6. In this case, we replace the configuration with the exceptional case on the left in Figure 5. That is, we delete one vertex from each of C' and B', make the remaining vertex of each adjacent to all of $Q(B)$, and add the edge $x_B x_C$. In both configurations, the distance from $Q(B)$ to other vertices of W is the same, and in each case every proper extension of f must give y_B the only color not in $f(Q(B))$. Hence G is a counterexample if and only if the smaller graph is a counterexample. By the minimality of G, we may thus exclude the configuration on the left in Figure 7.

Finally, suppose that $d_H(B, C) = 3$. Now x_C is not a neighbor of y_B but has distance 2 from it. If y_B lies in two blocks other than $F(B)$, then the one not leading to C begins a long enough branching path to contradict Claim 10 with C. Hence y_B lies in only one block other than $F(B)$; call it B^*. Since $d_H(y_B) = 3$, B^* is a triangle, and the vertex z in B^* that is not on the path to C is a cut-vertex of H, as shown on the right in Figure 7. In this situation, the coloring can be extended from $Q(B)$ to put any of the three colors on z. Hence we may delete the vertices in this figure other than z and its neighbor outside B^*, extend the coloring from $W - Q(B)$ to the rest of G, and then extend the coloring from $Q(B)$ to agree with it. This excludes this configuration. □

CLAIM 12. *If B is a remote block of H for a minimal counterexample G, then $k = 3$ and y_B belongs to exactly one block of H other than $F(B)$.*

Proof. Let B be a remote block of H for a minimal counterexample G. Together, Claims 11 and 10 imply in the usual case that y_B has at most one neighbor in H outside $F(B)$ that is a cut-vertex of H. By Claim 9, y_B has only one neighbor in $F(B)$.

In the clique case, if B has Type 1, then Claim 9 implies that x_B has $k-1$ neighbors in the k-clique $B' \cup Q(B)$. Hence only one vertex of $Q(B)$ can have a neighbor outside B, and it has only one such neighbor. This also holds in the list case or in the clique case when B has Type 2.

The neighbors of y_B outside $F(B)$ that are not cut-vertices must have a neighbor in $Q(B)$. Hence there is at most one vertex that is of that type or equals y_B. We have shown that y_B together with its neighbors outside $F(B)$ includes only two vertices and that y_B belongs to only one block of H other than $F(B)$.

These remarks yield the conclusion that $k = 3$ in the usual case, but also $k = 3$

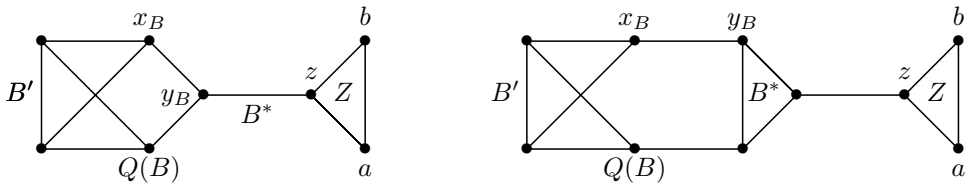


FIG. 8. *List case and clique Type* 1 *usual case when* $k = 3$.

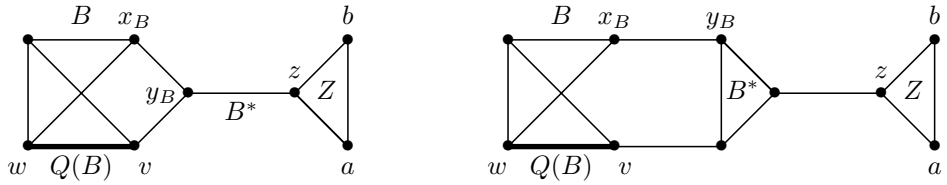


FIG. 9. *Clique Type* 2 *usual case when* $k = 3$.

in the exceptional case. □

When B is a remote block of H in the remaining case ($k = 3$), we let B^* denote the block of H other than $F(B)$ that contains y_B.

CLAIM 13. *There is no minimal counterexample* G.

Proof. Otherwise, Claim 12 yields $k = 3$. Let B be a remote block of H.

For the list case and for the usual clique case with B having Type 1, there are two remaining configurations, depending on whether the one vertex of $Q(B)$ is adjacent to y_B or to a vertex of B^* other than y_B. These configurations appear in Figure 8, where the additional vertices z, a, and b forming block Z are defined.

By the distance requirement, a and b have no neighbors in W; hence they are cut-vertices of H. Since $d_G(a, Q(B)) \le 4$, the distance requirement for $k = 3$ implies that every leaf block reached from a via a branching path along the block other than Z has distance at least 4 from a in G (it may be two steps more to W). The same is true of leaf blocks reached from b. These leaf blocks have distance at least 9 from Z in T, and the path P joining them in T passes through Z. On the other hand, $d_T(B, Z) \le 8$ via a path reaching P at Z. This contradicts the remoteness of B, so these cases do not occur. (This argument is not valid when the distance threshold is only 8.)

The usual clique case with B having Type 2 is very similar to that above. We merely relabel the picture as in Figure 9. We have $|Q(B)| = 2$. Let w be the neighbor of x_B in $Q(B)$, and let v be the nonneighbor of x. We have v adjacent to y_B or to a neighbor of y_B in B^*. Since again $d_G(a, Q(B)) \le 4$, the previous argument still works.

We have reduced the problem to the exceptional clique case with B having Type 1. Expanding the picture on the left in Figure 5 yields the configuration shown in Figure 10; it is a relabeling of those on the right in Figures 8 and 9. The argument mirrors those in the earlier cases. Note that $Q(B)$ is now one step farther from a and b, so the constraint from the distance requirement is weaker. However, B is now one step closer to a and b, so the remoteness argument is strengthened by the same amount that it is weakened. Again we contradict the remoteness of B. □

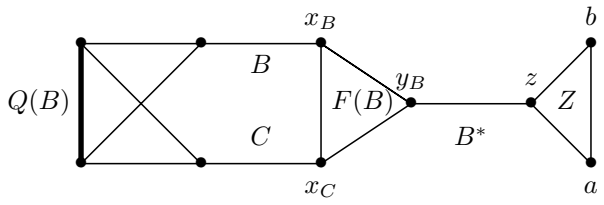


FIG. 10. *The exceptional case.*

REFERENCES

[1] M. O. ALBERTSON, *You can't paint yourself into a corner*, J. Combin. Theory Ser. B, 73 (1998), pp. 189–194.

[2] M. O. ALBERTSON AND E. H. MOORE, *Extending graph colorings*, J. Combin. Theory Ser. B, 77 (1999), pp. 83–95.

[3] M. O. ALBERTSON AND E. H. MOORE, *Extending graph colorings using no extra colors*, Discrete Math., 234 (2001), pp. 125–132.

[4] M. AXENOVICH, *A note on graph coloring extensions and list-colorings*, Electron. J. Combin., 10 (2003).

[5] O. V. BORODIN, *Criterion of chromaticity of a degree prescription*, in Abstracts of IV All-Union Conference on Theoretical Cybernetics (Novosibirsk), 1977, pp. 127–128 (in Russian).

[6] R. L. BROOKS, *On colouring the nodes of a network*, Proc. Cambridge Philos. Soc., 37 (1941), pp. 194–197.

[7] P. ERDŐS, A. L. RUBIN, AND H. TAYLOR, *Choosability in graphs*, in Proceedings of the West Coast Conference on Combinatorics, Graph Theory and Computing, Utilitas Math., Winnipeg, MB, Canada, 1980, pp. 125–157.

[8] A. V. KOSTOCHKA, M. STIEBITZ, AND B. WIRTH, *The colour theorems of Brooks and Gallai extended*, Discrete Math., 162 (1996), pp. 299–303.

[9] V. G. VIZING, *Coloring the vertices of a graph in prescribed colors*, Diskret. Analiz, 29 (1976), pp. 3–10, 101 (in Russian).

SIAM J. DISCRETE MATH.
Vol. 18, No. 3, pp. 554–570

CERTIFYING LexBFS RECOGNITION ALGORITHMS FOR PROPER INTERVAL GRAPHS AND PROPER INTERVAL BIGRAPHS*

PAVOL HELL† AND JING HUANG‡

Abstract. Recently, D. Corneil found a simple 3-sweep lexicographic breadth first search (LexBFS) algorithm for the recognition of proper interval graphs. We point out how to modify Corneil's algorithm to make it a certifying algorithm, and then describe a similar certifying 3-sweep LexBFS algorithm for the recognition of proper interval bigraphs. It follows from an earlier paper that the class of proper interval bigraphs is equal to the better known class of bipartite permutation graphs, and so we have a certifying algorithm for that class as well. All our algorithms run in time $O(m+n)$, including the certification phase. The certificates of representability (the intervals) can be authenticated in time $O(m+n)$. The certificates of nonrepresentability (the forbidden subgraphs) can be authenticated in time $O(n)$.

Key words. proper interval graphs, proper interval bigraphs, bipartite permutation graphs, bipartite trapezoid graphs, proper circular arc graphs, lexicographic breadth first search, recognition algorithms, certifying algorithms, forbidden subgraph characterizations

AMS subject classifications. 05C62, 05C85, 68R10

DOI. 10.1137/S0895480103430259

1. Background.

A graph H is an *interval graph* if there is a family $\mathcal{F}$ of intervals I_v, $v \in V(H)$, on the real line such that u, v are adjacent if and only if I_u, I_v intersect. The family $\mathcal{F}$ is called an *interval representation* of H. If $\mathcal{F}$ can be chosen so that no interval contains another, then H is called a *proper interval graph* and $\mathcal{F}$ is called a *proper interval representation* of H. Similarly, a graph is a *circular arc graph* if there is a family $\mathcal{F}$ of arcs A_v, $v \in V(H)$, on a circle so that u, v are adjacent if and only if A_u, A_v intersect. In this case $\mathcal{F}$ is called a *circular arc representation* of H. If $\mathcal{F}$ can be chosen so that no arc contains another, then H is called a *proper circular arc graph* and $\mathcal{F}$ is called a *proper circular arc representation* of H. Clearly, every interval graph is a circular arc graph and every proper interval graph is a proper circular arc graph. A well-known theorem of Wegner [35] asserts that a graph G is a proper interval graph if and only if it does not contain, as an induced subgraph, a cycle of length at least four, or *a claw*, *a net*, or *a tent*, depicted in Figure 1.1. Since the algorithm we are about to present will identify for each graph G either a proper interval representation or an induced claw, net, tent, or cycle of length at least four, our results in this paper imply Wegner's characterization.

Let H be a bipartite graph with bipartition (X, Y). Then H is called an *interval bigraph* if there is a family $\mathcal{F}$ of intervals I_v, $v \in (X \cup Y)$, such that, for all $x \in X$ and $y \in Y$, x, y are adjacent in H if and only if I_x, I_y intersect. The family $\mathcal{F}$ is called an *interval representation* for the bipartite graph H. Note that two intervals I_u, I_v with both $u, v \in X$ or with both $u, v \in Y$ may or may not intersect. As above, H is called a *proper interval bigraph* if $\mathcal{F}$ can be chosen so that no interval contains

*Received by the editors June 21, 2003; accepted for publication (in revised form) March 23, 2004; published electronically February 25, 2005.
http://www.siam.org/journals/sidma/18-3/43025.html
†School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada V5A 1S6 (pavol@cs.sfu.ca).
‡Department of Mathematics and Statistics, University of Victoria, P.O. Box 3045, Victoria, B.C., Canada V8W 3P4 (jing@math.uvic.ca).

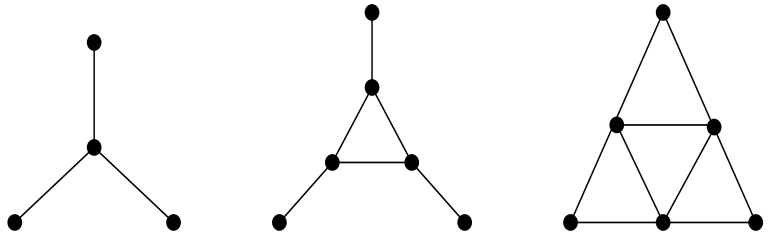


FIG. 1.1. *The claw, net, and tent.*

another. *Circular arc bigraphs* and *proper circular arc bigraph* are defined analogously. It is again easy to see that every interval bigraph is a circular arc bigraph and every proper interval bigraph is a proper circular arc bigraph. We have shown in [17] that a bipartite graph G is a proper interval bigraph if and only if it does not contain, as an induced subgraph, a cycle of length at least six, or a *bipartite claw*, *net*, or *tent*, depicted in Figure 1.2. Since the algorithm we will present identifies for each bipartite graph G either a proper interval bigraph representation or an induced bipartite claw, net, or cycle of length at least six, we will obtain another proof of our characterization.

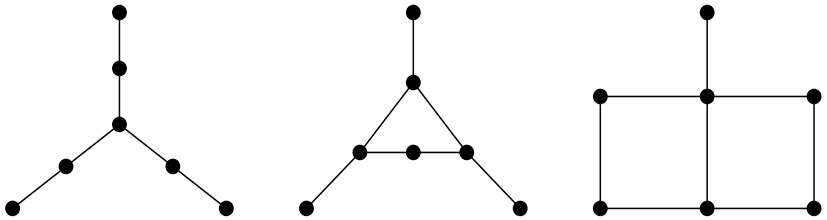


FIG. 1.2. *The bipartite claw, net, and tent.*

There are surprising connections between interval bigraphs and circular arc graphs, and between proper interval bigraphs and proper circular arc graphs. We have shown in [17] that interval bigraphs are precisely those bipartite graphs whose complements are circular arc graphs with a circular arc representation in which no two arcs cover the whole circle. The situation is even simpler for graphs that admit inclusion-free representations: We have shown in [17] that the class of proper interval bigraphs is precisely the class of bipartite graphs whose complements are proper circular arc graphs.

In fact we have shown in [17] that the class of proper interval bigraphs is also the (better known) class of bipartite permutation graphs. This result relates the class of proper interval bigraphs to all the other classes of graphs known to be equal to the class of bipartite permutation graphs, including bipartite asteroidal-triple free graphs, bipartite trapezoid graphs, and bipartite cocomparability graphs [3, 4, 17, 19, 27, 29, 36]. Thus proper interval bigraphs appear to be an important class of graphs.

We also remark here that proper interval bigraphs are the same class as unit interval bigraphs; i.e., they always admit representation in which all intervals have length one. This easily follows from the analogous fact about proper interval graphs [25]. Note that proper circular arc graphs are not necessarily representable by unit circular arcs [29].

Recently, Corneil [7] found a very simple linear time algorithm for recognizing proper interval graphs. (Earlier algorithms for recognizing this class of graphs include [8, 12, 14, 16, 20].) Corneil's algorithm uses three sweeps of lexicographic breadth first search (LexBFS) to produce a linear ordering $<$ of the vertices of the input graph $H = (V, E)$ and to check whether or not the ordering satisfies the following 3-*vertex*

condition:

$$\text{For all } x < y < z,\ xz \in E \text{ implies } xy \in E \text{ and } yz \in E.$$

If the ordering satisfies the 3-vertex condition, then the input graph is a proper interval graph (cf. below); otherwise it is shown in [7] that H is not a proper interval graph. (This also follows from our results below.)

The 3-sweep algorithm in [7] uses the following LexBFS algorithm.

Procedure LexBFS(H, u)

Input: a connected graph $H = (V, E)$, and a distinguished vertex u
Output: an ordering of the vertices of H, given by the numbers $\sigma(v)$

Begin
label$(u) \leftarrow |V|$
for each vertex v in $V - \{u\}$ **do**
label$(v) \leftarrow \Lambda$ (the empty string)
for $i \leftarrow |V|$ **downto** 1 **do**
begin
pick an unnumbered vertex v with the lexicographically largest label $(\star)$
$\sigma(v) \leftarrow |V| + 1 - i$ (number v by $|V| + 1 - i$)
for each unnumbered vertex w in $N(v)$ **do**
append i to label(w)
end
end

The following *property P* holds for any LexBFS ordering $v_1, v_2, \ldots, v_n$: If $j < k < l$ and v_j is adjacent to v_l but not v_k, then there exists an $i < j$ such that v_i is adjacent to v_k and not v_l.

The LexBFS algorithm has been used for the recognition of chordal graphs [26] and of several other families of graphs defined by having certain vertex orderings [6, 11, 13, 15]; cf. [1].

We now describe the multisweep LexBFS algorithm. This kind of algorithm has been pioneered in [2, 5, 7, 9, 22, 28]. A multisweep LexBFS algorithm calls LexBFS several times; each time is called a *sweep*. Every sweep, except the first, uses the ordering(s) produced by the preceding sweep (or sweeps) to break ties occurring in step $(\star)$.

Specifically, having in hand one LexBFS ordering, we proceed as follows.

Procedure LexBFS+(H, τ)
Suppose τ is a LexBFS ordering obtained by a previous sweep. In the current LexBFS sweep, at step $(\star)$, let S be the set of vertices with the lexicographically largest label. Now v is picked to be the vertex in S that appears *last* in τ.

In addition to the property P enjoyed by all LexBFS orderings, an ordering τ^+ produced by the LexBFS+ algorithm, has the following additional *property R*: If v_k precedes v_l both in τ and in τ^+, then there exists a v_i that precedes v_k in τ^+, such that v_i is adjacent to v_k and not v_l.

Both algorithms we discuss are variants of the following *three-sweep* technique.

The 3-Sweep Algorithm

Input: a connected graph H

1. Perform an arbitrary LexBFS, yielding σ.
2. Perform LexBFS+(H, σ), yielding σ^+.
3. Perform LexBFS+(H, σ^+), yielding σ^{++}.

Corneil [7] showed that if H is a proper interval graph, then σ^{++} satisfies the 3-vertex condition. The 3-sweep algorithm can be implemented to run in linear time, i.e., time $O(m+n)$ (where m and n are, respectively, the number of edges and vertices of the input graph). It is also easy to test in linear time whether or not σ^{++} satisfies the 3-vertex condition. When the algorithm finds an ordering satisfying the 3-vertex condition, a representation by an inclusion-free family of intervals can be found in time $O(m+n)$ (cf. Proposition 2.1) and authenticated also in time $O(m+n)$ (cf. [18]).

We shall make an addition to the algorithm so that if the input graph does not satisfy the 3-vertex condition, the algorithm actually finds either an induced cycle of length at least four or an induced claw, net, or tent, from Figure 1.1. The entire algorithm will run in time $O(m+n)$. Authenticating the certificates of nonrepresentability, i.e., the induced cycle, claw, net, or tent, can be accomplished in time $O(n)$, as in [18].

Algorithms which provide a certificate with each of their answers have been of interest since LexBFS was first used, in [26]. Recently, there has been renewed interest in finding certifying algorithms [18, 23, 34]. Since the first version of this paper, we have learned [10] that another $O(m+n)$ LexBFS-based certifying algorithm for recognizing proper interval graphs has independently been found in [24].

A graph G is *chordal* if it does not contain an induced cycle of length at least four, or, equivalently [26], if it admits an ordering without configuration C in Figure 2.1; such an ordering is called a *perfect elimination ordering*. The algorithm from [26, 32, 33] produces, in time $O(m+n)$, either a perfect elimination ordering, certifying the graph is chordal, or an induced cycle of length at least four, certifying the graph is not chordal.

We also adapt Corneil's approach to recognizing proper interval bigraphs. (Earlier algorithms for recognizing this class of graphs include [5, 30, 31].) Our algorithm will again produce a certificate for each of its answers. Specifically, we show that the vertex ordering of a bipartite graph produced by the 3-sweep algorithm either satisfies the "weak 3-vertex condition" (defined at the beginning of section 3), or the input graph is not a proper interval bigraph. The algorithm certifies the first possibility by presenting a proper interval bigraph representation, which can be authenticated in time $O(m+n)$, and certifies the second possibility by presenting an induced cycle of length at least six or an induced forbidden subgraph from Figure 1.2, which can be authenticated in time $O(n)$. The entire algorithm can be implemented to run in time $O(m+n)$.

Our algorithms assume that the input graph H is connected, and in the case of proper interval bigraphs we also assume that H is bipartite. If necessary, we may perform a check, before the algorithms are invoked, that the input graph H satisfies these assumptions. The test can also be done in time $O(m+n)$ and may, in fact, be part of the first LexBFS sweep.

2. Proper interval graphs. If a vertex ordering $v_1, v_2, \ldots, v_n$ of a graph H satisfies the 3-vertex condition, then we can obtain a proper interval representation of

H as follows [12].

For each $i = 1, 2, \ldots, n$, let $U(i)$ be the greatest subscript such that $v_{U(i)}$ is either adjacent or equal to v_i. To each vertex v_i, we associate the interval $I_i = [i, U(i)+1-\frac{1}{i}]$.

PROPOSITION 2.1 (see [12]). *If the ordering $v_1, v_2, \ldots, v_n$ of the vertices of a graph H satisfies the 3-vertex condition, then the family $I_i, i = 1, 2, \ldots, n$, is a proper interval representation of H.* □

Figure 2.1 demonstrates the three different ways the 3-vertex condition can be violated.

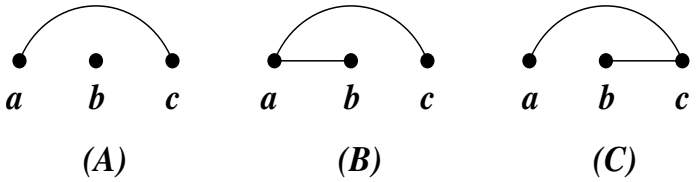


FIG. 2.1. *The violations of the 3-vertex condition.*

We define the *level* $\ell_\sigma(x)$ of a vertex x in an ordering σ to be the length of a shortest path from x to the first vertex of σ. For simplicity, we shall write $\ell(x)$ for $\ell_\sigma(x)$, $\ell^+(x)$ for $\ell_{\sigma^+}(x)$, and $\ell^{++}(x)$ for $\ell_{\sigma^{++}}(x)$. If x precedes y in σ (respectively, σ^+ and σ^{++}), we write $x < y$ (respectively, $x <_+ y$ and $x <_{++} y$). Since all our orderings are obtained by breadth first search, the levels are nondecreasing; moreover, the levels can be computed during the LexBFS procedures. Thus, if A or B occurs, either the levels of a, b, c are all the same, or the levels of a and c differ by one, and the level of b is equal to one of them. Let us call B_1 the case of B in which the level of b is different from the level of c, and B_2 the case of B in which the levels of b and c are the same. (In B_2 the level of a can be the same as b or one smaller.) Note that in A and in B_2 there are always two nonadjacent vertices of the same level. Also note that an ordering has C if and only if it is not a perfect elimination ordering, and a LexBFS ordering is a perfect elimination ordering if and only if the graph is chordal [26]. Thus if one LexBFS ordering does not have C, then none do.

Here is our certifying 3-sweep algorithm.

The 3-Sweep Certifying Algorithm for the Recognition of Proper Interval Graphs

Input: a connected graph H

1. Perform an arbitrary LexBFS, yielding an ordering σ.
2. Test σ for occurrence of C. If C occurs, output an induced cycle of length at least four, obtained using the algorithm of [33].
3. Perform LexBFS+(H, σ), yielding an ordering σ^+.
4. Test σ^+ for occurrence of A or B_2. If A or B_2 occurs, output a forbidden subgraph from Figure 1.1, obtained using Proposition 2.3.
5. Perform LexBFS+(H, σ^+), yielding an ordering σ^{++}.
6. Test σ^{++} for occurrence of B. If B_1 occurs, output a claw obtained using Proposition 2.6. If B_2 occurs, output a forbidden subgraph obtained using Proposition 2.3.
7. Output a proper interval representation of H, obtained from σ^{++} using Proposition 2.1.

When H is not chordal, we will find in the first sweep an occurrence of C, and halt with a certificate which is an induced cycle of length at least four. Otherwise, the graph H is chordal, and hence we shall not find an occurrence of C in any LexBFS ordering.

We shall use the following observation.

LEMMA 2.2. *Let τ be a breadth first ordering of the vertices of H. Then A or B_2 occurs in τ if and only if there exist two nonadjacent vertices of the same level.*

Proof. If $a <_\tau b <_\tau c$ form a B_2, then b, c are two nonadjacent vertices of the same level. If $a <_\tau b <_\tau c$ form an A, then either a and b are nonadjacent vertices of the same level or b and c are nonadjacent vertices of the same level.

Conversely, suppose that x, y with $x <_\tau y$ are two nonadjacent vertices of level k. Then $k \geq 1$ and any neighbor of y of level $k-1$ together with x, y forms either an A or a B_2 in τ. □

When an occurrence of A or B_2 is detected, the following proposition explains how to find a forbidden claw, net, or tent from Figure 1.1.

PROPOSITION 2.3. *Suppose that H is a chordal graph. If A or B_2 occurs in σ^+ or σ^{++}, then H contains a claw, net, or tent.*

Proof. It suffices to prove the statement for σ^+ (as $\sigma^{++} = (\sigma^+)^+$). Suppose that A or B_2 occurs in σ^+. Then by Lemma 2.2 we obtain two nonadjacent vertices x and y of the same level. Let $\ell^+(x) = \ell^+(y) = k$. Since H is a chordal graph, no C occurs in σ. This implies $k \geq 2$. Indeed, if $k = 1$, then the vertex z of level zero, which is first in σ^+, was last in σ, and is adjacent to both x and y. Thus σ contains a C formed by $x < y < z$ or $y < x < z$, depending whether $x < y$ or $y < x$, a contradiction.

Suppose first that x and y have a common neighbor x' of level $k-1$. Then x, y, x', and any neighbor of x' of level $k-2$ induce a claw. Otherwise, let x', y' be vertices of level $k-1$, which are neighbors of x, y, respectively. Since x and y have no common neighbor of level $k-1$, we must have $x' \neq y'$, and $x'y, xy'$ must not be edges. If x' and y' are nonadjacent, then we consider x' and y' in place of x and y. So we assume that x' and y' are adjacent and proceed as follows.

If x' and y' do not have a common neighbor of level $k-2$, then x, x', y' together with any neighbor of x' of level $k-2$ induce a claw. Suppose that x' and y' have a common neighbor z of level $k-2$. If $k \geq 3$, then x, y, x', y', z together with any neighbor of z of level $k-3$ induce a net. Thus assume that $k = 2$.

Consider the ordering σ. Since H is a chordal graph, σ does not contain C from Figure 2.1. Note that z is the first vertex of σ^+ and therefore the last vertex of σ. Suppose (without loss of generality) that y' precedes x' in σ. Recall that we write $<$ for the order of vertices in σ; thus we have $y' < x' < z$. Since C does not occur in σ, we must have $x' < x$ (or else x, x', y' would yield a C). Now z is adjacent to y', which is not adjacent to x, and yet $x < z$ in σ. Since σ is a LexBFS ordering, there exists a vertex x'' with $x'' < y'$, which is adjacent to x but not to z. The absence of C implies that x'' is adjacent to both x' and y'. If x'' is not adjacent to y, then $x'', y, y,', z$ induce a claw. On the other hand, if x'' is adjacent to y, then x'', x, x', z, y, y' induce a tent. □

It remains to explain how to handle occurrences of B_1. We shall use the following lemma.

LEMMA 2.4. *Suppose that σ^+ contains neither A nor B_2. If $\ell^+(x) < \ell^+(y)$, then $y <_{++} x$.*

Proof. Suppose to the contrary that $x <_{++} y$. Let $k = \ell^+(y)$; we may assume that, among all vertices of level smaller than k in σ^+, x is the first vertex in the

ordering σ^{++}. Clearly x is not the last vertex in σ^+, and hence not the first vertex in σ^{++}. Since $x <_+ y$ as well as $x <_{++} y$, there exists, by the property R, a vertex x' with $x' <_{++} x$, which is adjacent to x but not to y. The choice of x implies that $\ell^+(x') \geq k > \ell^+(x)$. The fact that x' is adjacent to x implies that $\ell^+(x') = k$. Thus we have two nonadjacent vertices x' and y of level k in σ^+. By Lemma 2.2, A or B_2 occurs in σ^+, in contradiction to the hypothesis. □

PROPOSITION 2.5. *Suppose that H is chordal and σ^+ contains neither A nor B_2. Then σ^{++} does not contain A.*

Proof. Suppose that σ^{++} contains A with $a <_{++} b <_{++} c$. Since σ^+ does not contain A or B_2, vertices of the same level in σ^+ are pairwise adjacent, according to Lemma 2.2. Thus $\ell^+(a) \neq \ell^+(b)$ and $\ell^+(b) \neq \ell^+(c)$. By Lemma 2.4, $\ell^+(a) > \ell^+(b) > \ell^+(c)$. So levels of a and c differ by at least two, which is impossible as a and c are adjacent. □

The crucial property of the three-sweep algorithm is captured in the next proposition.

PROPOSITION 2.6. *Suppose that H is chordal and that σ^+ contains neither A nor B_2. If B_1 occurs in σ^{++}, then H contains a claw.*

Proof. Assume that σ^{++} has B_1 with $a <_{++} b <_{++} c$ and $\ell^{++}(a) = \ell^{++}(b) = \ell^{++}(c) - 1$. Then b and c are not adjacent. Since A or B_2 do not occur in σ^+, $\ell^+(b) \neq \ell^+(c)$ by Lemma 2.2. According to Lemma 2.4 and the fact that $b <_{++} c$, we have $\ell^+(b) > \ell^+(c)$ and in particular $c <_+ b$. Since H is chordal, $a <_+ b$, as otherwise $c <_+ b <_+ a$ would form a C. Since $a <_+ b$ as well as $a <_{++} b$, property R implies that there exists a vertex c' with $c' <_{++} a$ which is adjacent to a but not to b. The vertex c' cannot be adjacent to c either, as otherwise $c' <_{++} b <_{++} c$ form an A, contradicting Proposition 2.5. Therefore a, b, c', c induce a claw. □

The following two examples show that the test for B in the ordering σ^{++} is necessary: for the graph in Figure 2.2, σ^{++} contains a B_2 with a, c, e (and a, b, c, e induce a claw), and for the graph in Figure 2.3, σ^{++} contains a B_1 with f, e, c (and f, e, c, h induce a claw).

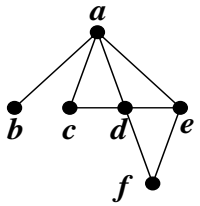


FIG. 2.2. $\sigma : a\ b\ c\ d\ e\ f$; $\sigma^+ : f\ e\ d\ a\ c\ b$; $\sigma^{++} : b\ a\ c\ d\ e\ f$.

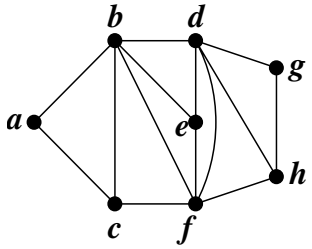


FIG. 2.3. $\sigma : d\ h\ g\ f\ e\ b\ c\ a$; $\sigma^+ : a\ c\ b\ f\ e\ d\ h\ g$; $\sigma^{++} : g\ h\ d\ f\ e\ b\ c\ a$.

THEOREM 2.7. *The 3-sweep certifying algorithm for the recognition of proper interval graphs is correct and can be implemented in time $O(m+n)$ (including the certification step).*

Proof. We test for C in σ; hence neither σ^+ nor σ^{++} will have C. We test for A or B_2 in σ^+; hence σ^{++} cannot have A either by Proposition 2.5. Thus testing for B in step 6 is sufficient. This proves that σ^{++} in step 7 satisfies the 3-vertex condition, and hence the algorithm is correct.

We now address the implementation and the complexity of the algorithm. There are three tests the algorithm performs: a search for C in step 2, a search for A or B_2 in step 4, and a search for B in step 6.

The search for C is given in [32], and the algorithm which uses C to produce an induced cycle of length at least four, given in [33], is proved to be $O(m+n)$ in that paper.

The search for A or B_2 in step 4 can be performed using the characterization in Lemma 2.2. In fact, we are searching for two nonadjacent vertices on the same level. It is a simple exercise to accomplish this in time $O(m+n)$. If such vertices are found, we need to identify a forbidden induced subgraph: We have written the proof of Proposition 2.3 in such a way that it can be implemented as an $O(n)$ algorithm to actually construct an induced claw, net, or tent.

The search for B in step 6 can be done by searching for A or B, since Proposition 2.5 shows that A cannot occur at this point. Recall that for any ordering τ of vertices $v_1, v_2, \ldots, v_n$, we have defined $U(i)$ to be the greatest subscript j such that v_i is adjacent or equal to v_j. We now similarly define $L(i)$ to be the smallest subscript k such that v_i is adjacent or equal to v_k. Finally, let $d_L(i)$ denote the number of neighbors v_k of v_i with $k < i$. Given τ, these parameters can easily be computed in time $O(m+n)$; in fact, if τ is a LexBFS ordering they can be computed during the LexBFS procedure. To detect an A or B, we need to check only whether or not $d_L(i) = i - L(i)$ for every i. To actually identify a B, once an i with $d_L(i) < i - L(i)$ is detected, it suffices to test at most $d_L(i)$ vertices with subscripts between i and $L(i) - 1$ before a subscript j is found, such that $v_j v_i$ is not an edge, and hence $v_i, v_j, v_{L(i)}$ form B. (Thus we can do all this in time $O(n)$.) If we have found a B_2, we proceed as above using Proposition 2.3. If a B_1 is found, Proposition 2.6 implies that a claw must exist and suggests how to find it in time $O(n)$. If no claw, net, or tent have been found, then the 3-vertex condition is satisfied, and the representation explained at the beginning section is computed in time $O(m+n)$.

Thus the entire algorithm, including the certification stage, can be implemented in time $O(m+n)$. □

By computing an additional parameter, $d_U(i)$, the number of neighbors v_j of v_i with $j > i$, we can actually also search for A or B_2 in time $O(n)$, rather than using our simpler search of order $O(m+n)$ explained above. (The details are similar to the search for A or B discussed in the proof.) Hence, after the three LexBFS searches have been performed and the parameters $U(i), L(i), d_U(i), d_L(i)$ are computed, and after the testing for C has concluded and the possible nonchordality of H is certified by finding an induced cycle of length at least four (these two tasks are not needed if the given graph is known to be chordal), we have to test only for A, B and find either a forbidden subgraph or a proper interval representation; all these remaining tasks can be performed in time $O(n)$.

As noted earlier, a certificate of representability (an inclusion-free family of intervals whose intersection graph is the given graph) can be authenticated in time $O(m+n)$,

as in [18]. A certificate of nonrepresentability (an induced cycle of length greater than three, claw, net, or tent) can be authenticated in time $O(n)$; cf. [18].

3. Proper interval bigraphs. In this section we consider only connected bipartite graphs $H = (V, E)$. The 3-sweep certifying algorithm in section 2 may be modified to recognize proper interval bigraphs. We shall use the same notion of *level* as before. Note that, as H is bipartite, each level must be an independent set.

It follows from [17] that a bipartite graph H is a proper interval bigraph if and only if the vertices can be ordered satisfying the following *weak 3-vertex condition*:

$$\text{For all } x < y < z,\ xz \in E \text{ implies either } xy \in E \text{ or } yz \in E.$$

(Note that x and z are in different color classes of the bipartite graph H, and hence y can only be adjacent to one of them.)

When H is not a proper interval bigraph, our algorithm will find a violation (of the weak 3-vertex condition) in the vertex ordering produced by the second or the third sweep. Using the violation, an induced cycle of length at least six, or an induced bipartite claw, net, or tent from Figure 1.2, will be found in H.

When H is a proper interval bigraph, the vertex ordering of H produced by the third sweep will be shown to satisfy the weak 3-vertex condition, and a proper interval bigraph representation of H will be obtained.

Suppose that τ is a breadth first ordering $v_1, v_2, \ldots, v_n$ of the vertices of H. This means, in particular, that the vertices of any one level appear consecutively in τ. As noted above, they form an independent set, and they have only neighbors on the previous level and on the next level. These simple facts are useful to keep in mind when reading the proofs in this section.

For each $i = 1, 2, \ldots, n$, let $U(i), u'(i), R(i)$ be defined as follows:

- $U(i)$ is the greatest subscript such that $v_{U(i)}$ is either adjacent or equal to v_i.
- $u'(i)$ is the greatest subscript such that $v_{u'(i)}$ is adjacent to v_i (note that $u'(i)$ can be smaller than i).
- $R(i) = U(i)$, except when $U(i) = i$ and $U(u'(i)) > i$, in which case $R(i) = U(u'(i))$.

PROPOSITION 3.1. *Suppose that the vertex ordering $v_1, v_2, \ldots, v_n$ of H satisfies the weak 3-vertex condition. Let $I_i = [i, R(i) + 1 - \frac{1}{i}]$ be the interval associated with v_i for each $i = 1, 2, \ldots, n$. Then the family $I_i, i = 1, 2, \ldots, n$, is a proper interval bigraph representation of H.*

Proof. It follows immediately from the definitions that $i \le U(i) \le R(i)$ for each $i = 1, 2, \ldots, n$. Suppose that v_i, v_k with $i < k$ are adjacent. Then $k \le U(i) \le R(i)$ and I_i and I_k intersect as they both contain the point k. Conversely, suppose that I_i and I_k intersect and that v_i and v_k are not in the same color class of H. Assume without loss of generality that $i < k$. Then we must have $k \le R(i)$. If $R(i) = U(i)$, then, by the definition of $R(i)$ and the fact that $i < k \le R(i)$, v_i is adjacent to $v_{R(i)}$. (Thus v_k and $v_{R(i)}$ are in the same color class of H.) The weak 3-vertex condition implies that v_i is adjacent to v_k. If $R(i) \ne U(i)$, then by the definitions of $U(i), u'(i), R(i)$ we must have $R(i) = U(u'(i)) > i = U(i) > u'(i)$. Since $v_{u'(i)}$ is adjacent to both v_i and $v_{U(u'(i))} = v_{R(i)}$, v_i and $v_{R(i)}$ are of the same level. Thus v_i should be of the same level as v_k because $i < k < R(i)$. But this contradicts the assumption that v_i and v_k are not in the same color class of H.

It remains to show that no interval in the family contains another. In view of the definition of the intervals, it suffices to show that $R(i) \le R(k)$ for all $1 \le i < k \le n$.

If $R(i) \leq k$, then $R(i) \leq k \leq R(k)$ and it is true. So assume that $k < R(i)$. Suppose that $R(i) = U(i)$. Then v_i is adjacent to $v_{U(i)} = v_{R(i)}$. If v_k is in the same color class as v_i, then the weak 3-vertex condition implies that v_k is adjacent to $v_{U(i)}$. Hence $R(k) \geq U(k) \geq U(i) = R(i)$. If v_k is not in the same color class as v_i, then v_k is adjacent to v_i by the weak 3-vertex condition (as $i < k < R(i) = U(i)$ and v_i is adjacent to $v_{U(i)} = v_{R(i)}$). Either $U(k) = k$, in which case we must have $R(k) = U(u'(k)) \geq U(i) = R(i)$, or $U(k) \geq U(i) = R(i)$, in which case $R(k) = U(k) \geq U(i) = R(i)$. Suppose that $R(i) \neq U(i)$. This can only happen when $U(i) = i > u'(i)$. By definition $R(i) = U(u'(i))$. Since $u'(i) < i < k < R(i) = U(u'(i))$ and $v_{u'(i)}$ is adjacent to both v_i and $v_{R(i)}$, the three vertices $v_i, v_k, v_{R(i)}$ are all of the same level, and hence are in the same color class. If $U(k) \geq R(i)$, then $R(k) = U(k) \geq R(i)$. Otherwise $U(k) = k > u'(i) = u'(k)$ and we have $R(k) = U(u'(k)) = U(u'(i)) = R(i)$. This completes the proof. □

Since the input graph is bipartite, there are essentially only two ways the weak 3-vertex condition can be violated. We illustrate these violations in Figure 3.1. In the first violation the colors of b and c are the same, while the color of a is different. In the second violation, the colors of a and b are the same, and the color of c is different. (The figures show the colors as particular choices of black and white, but they could be the opposite colors as well; in other words, D can have a white and b and c black, and similarly for E.)

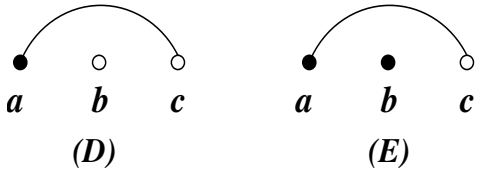


FIG. 3.1. *The two possible violations of the weak 3-vertex condition.*

Note that in D the vertices b, c have the same level (and a has level one lower), while in E the vertices a, b have the same level (and c has level one higher).

In Figure 3.2, we shall distinguish two important cases of D and E, which are parts of two 4-vertex configurations.

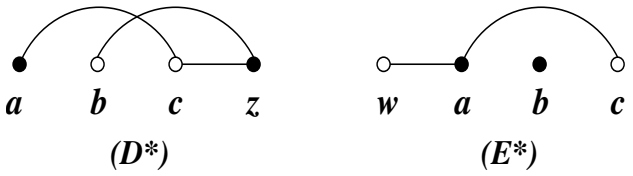


FIG. 3.2.

A triple $D = (a, b, c)$ will be called D_1 if there exists a vertex z such that (a, b, c, z) is a quadruple D^* as illustrated in Figure 3.2, and will be called D_2 otherwise. A triple $E = (a, b, c)$ will be called E_1 if there exists a vertex w such that (w, a, b, c) is a quadruple E^*, as illustrated in Figure 3.2, and will be called E_2 otherwise.

The 3-Sweep Certifying Algorithm for the Recognition of Proper Interval Bigraphs

Input: a connected bipartite graph H

1. Perform an arbitrary LexBFS, yielding σ.
2. Perform LexBFS+(H, σ), yielding σ^+.
3. Test σ^+ for occurrence of D or E_1. If D or E_1 occurs, obtain an induced cycle of length at least six or an induced forbidden subgraph from Figure 1.2, using Lemma 3.3 and Proposition 3.4.
4. Perform LexBFS+(H, σ^+), yielding σ^{++}.
5. Test σ^{++} for occurrence of D or E_1. If D or E_1 occurs, obtain a forbidden subgraph from Figure 1.2, using Proposition 3.4.
6. Output a proper interval bigraph representation of H, obtained from σ^{++} using Proposition 3.1.

Let x and y be of level k in a LexBFS ordering τ of H. Recall that any neighbor of x or of y is of level $k-1$ or $k+1$ in τ. We use $N_{k-1}(x)$ (respectively, $N_{k+1}(x)$) to denote the set of all neighbors of x of level $k-1$ (respectively, $k+1$), all with respect to τ. We say that x and y are *consistent*, with respect to τ, if all the following properties are satisfied:

- either $N_{k-1}(x) \subseteq N_{k-1}(y)$ or $N_{k-1}(x) \supseteq N_{k-1}(y)$;
- either $N_{k+1}(x) \subseteq N_{k+1}(y)$ or $N_{k+1}(x) \supseteq N_{k+1}(y)$;
- if $N_{k-1}(x) \subset N_{k-1}(y)$, then $N_{k+1}(x) \supseteq N_{k+1}(y)$;
- if $N_{k-1}(x) \supset N_{k-1}(y)$, then $N_{k+1}(x) \subseteq N_{k+1}(y)$.

Otherwise, we say that x and y are *inconsistent*.

LEMMA 3.2. *Let τ be any LexBFS ordering of the vertices of H. Then D or E_1 occurs in τ if and only if τ has two inconsistent vertices.*

Proof. A LexBFS ordering τ containing D or E_1 must have two inconsistent vertices. Indeed, if E_1 occurs, then $N_{k-1}(a) \not\subseteq N_{k-1}(b)$ and $N_{k+1}(a) \not\subseteq N_{k+1}(b)$, so a and b are inconsistent. On the other hand, if D occurs, then a is adjacent to c but not to b—hence the property P of the LexBFS ordering guarantees that there is a vertex $x <_\tau a$ adjacent to b but not to c. Since the levels of both a and x are one less than the level of b and c, the vertices b and c are inconsistent.

Conversely, suppose that x and y of level k (for some k) are inconsistent. There are, up to symmetry, two reasons this can happen: First, x and y can have neighbors x' and y', respectively, on level $k-1$, such that x is not adjacent to y' and y is not adjacent to x'. In this case either $x' <_\tau y <_\tau x$ or $y' <_\tau x <_\tau y$ form a D. Second, x has neighbors x' of level $k-1$ and x'' of level $k+1$, which are not adjacent to y. In this case, either $x' <_\tau y <_\tau x$ form a D, or $x' <_\tau x <_\tau y <_\tau x''$ form an E_1. □

LEMMA 3.3. *Suppose that τ is a LexBFS ordering of H. If τ contains D^* with $a < b < c < z$, then H contains an induced cycle of length at least six.*

Proof. Set $\ell(a) = k$. Then $\ell(b) = \ell(c) = k+1$ and $\ell(z) = k+2$. Since $b < c$ and $a \in N_k(c) - N_k(b)$, there is a vertex $x \in N_k(b) - N_k(c)$. If there is a vertex $v \in N_{k-1}(a) \cap N_{k-1}(x)$, then $vaczbxv$ is an induced cycle of length six. Otherwise there exist vertices $e \in N_{k-1}(a) - N_{k-1}(x)$ and $f \in N_{k-1}(x) - N_{k-1}(a)$. If e and f

have a common neighbor of level $k-2$, then we obtain an induced cycle of length eight; otherwise each of e and f has a neighbor of level $k-2$ which is not a neighbor of the other. Continuing this way, we will obtain an induced cycle of length at least six. □

It follows from [21] that, conversely, if H contains an induced cycle of length at least six, then any LexBFS ordering of H contains an occurrence of D^*, and hence of D_1. In other words, H contains no induced cycles of length at least six (i.e., is *chordal bipartite*) if and only if D_1 does not occur in any LexBFS of H. Thus if one LexBFS ordering does not have D_1, then none do.

PROPOSITION 3.4. *If σ^+ or σ^{++} contains D or E_1, then H contains an induced cycle of length at least six or an induced bipartite claw, net, or tent from Figure* 1.2.

Proof. It will again suffice to prove the proposition for σ^+. Thus assume that σ^+ contains two inconsistent vertices x, y; note that inconsistent vertices have, by definition, the same level. Thus, let $\ell^+(x) = \ell^+(y) = k$. (The neighborhoods $N_j(v)$ in this proof are all taken with respect to σ^+.)

In view of Lemma 3.3, it will suffice to show that either D^* occurs in σ or in σ^+ or H contains an induced bipartite claw, net, or tent, from Figure 1.2. We will frequently appeal to the following lemma.

LEMMA 3.5. *Let τ be a LexBFS ordering of H. Suppose that a and b are distinct vertices of level k (for some k) with a common neighbor c of level $k+1$ in τ. Then a and b also have a common neighbor of level $k-1$, or there is a D^* in τ.*

Proof. Assume without loss of generality $a <_\tau b$. Let d be a neighbor of b of level $k-1$ in τ. Either d is a neighbor of a, and hence a common neighbor of a and b, or $d <_\tau a <_\tau b <_\tau c$ form a D^* in τ. □

We now continue with the proof of Proposition 3.4. The two essentially different reasons for σ^+ to have inconsistent vertices x and y are taken up in the two following cases.

Case 1. Suppose that $N_{k+1}(x) \not\subseteq N_{k+1}(y)$ and $N_{k+1}(x) \not\supseteq N_{k+1}(y)$.

If $N_{k-1}(x) \not\subseteq N_{k-1}(y)$ and $N_{k-1}(x) \not\supseteq N_{k-1}(y)$, then any pair of vertices from $N_{k-1}(x) - N_{k-1}(y)$ and $N_{k-1}(y) - N_{k-1}(x)$, respectively, are inconsistent and we consider them in place of x and y. So assume that either $N_{k-1}(x) \subseteq N_{k-1}(y)$ or $N_{k-1}(x) \supseteq N_{k-1}(y)$. This implies that $N_{k-1}(x) \cap N_{k-1}(y) \neq \emptyset$, since $N_{k-1}(x) \neq \emptyset$ and $N_{k-1}(y) \neq \emptyset$.

Let $z \in N_{k-1}(x) \cap N_{k-1}(y)$, $x' \in N_{k+1}(x) - N_{k+1}(y)$, and $y' \in N_{k+1}(y) - N_{k+1}(x)$. If $k \geq 3$, let $w \in N_{k-2}(z)$ and $v \in N_{k-3}(w)$. Then we obtain a bipartite claw in H induced by v, w, z, x, y, x', y'. So assume $k \leq 2$.

If $k = 1$, then consider the ordering σ. Since z is the first vertex in σ^+, it is the last vertex of σ. Assume without loss of generality that $x <_+ y$. Then we must have $y < x$. Clearly, x and y are of the same level in σ. If $x' < y$, then $x' < y < x < z$ form D^*. Therefore $x' > y$, and hence x' is of the same level in σ as z.

Since $y < x' < z$ and y is adjacent to z but not to x', there exists by the property P a vertex $x'' < y$ adjacent to x' but not to z. Furthermore, x'' is not adjacent to y' or else we would have a six-cycle induced by x, x', x'', y', y, z. Note that the level of x'' in σ is the same as the level of x (and y). By Lemma 3.5, either D^* occurs in σ, or x and y have a common neighbor u of level $\ell(x) - 1$. If u is not adjacent to x'', then $u < x'' < x < x'$ form D^*. So assume that u is adjacent to x''. If u is not the first vertex of σ, then we obtain a bipartite tent induced by z, x, y, x', x'', u and any neighbor of u of level smaller than u in σ. So assume that u is the first vertex of σ. This implies that y' can only be of the same level as z in σ and thus $x < y' < z$. Since x is adjacent to z but not to y', there exists, again by the property P, a vertex $y'' < x$

adjacent to y' but not to z. We must have $y'' \neq x''$, since x'' is not adjacent to y' (as noted above). Thus we obtain the bipartite tent in H induced by z, x, y, u, y', y'', x''.

Now assume $k = 2$. Let w be the first vertex in σ^+. Then w is the last vertex in σ and we again consider the ordering σ. Suppose that x and y are both of the same level $\ell(z) - 1$. By Lemma 3.5 either D^* occurs in σ, or x and y have a common neighbor z' of level $\ell(x) - 1$ and we obtain a bipartite net induced by w, z, x, y, x', y', z'.

So x and y are either of the same level as w or of different levels in σ. First assume that x and y are of the same level as w in σ. Then x', y' are of the same level as z in σ, smaller than the level of x, y, w. Let z' be any neighbor of z, of level smaller than z in σ. If z' is adjacent to both x' and y', then we obtain a bipartite tent induced by z', x', y', z, x, y, w. Otherwise, z' is not adjacent to x' or to y'; say it is not adjacent to x'. If $x' < z$, then $z' < x' < z < x$ form D^*. So assume $z < x'$. By Lemma 3.5, z and x' must have a common neighbor z'' of level $\ell(z) - 1 = \ell(z')$. Note that z' and z'' are distinct and have a common neighbor z. Applying the lemma once more, we obtain common neighbor z''' of z' and z'' and therefore a bipartite tent induced by $w, x, x', z, z', z'', z'''$.

Assume now that x and y are of different levels in σ. Since both x, y are adjacent to z, one is of level $\ell(z) - 1$ and the other is of level $\ell(z) + 1$. Assume without loss of generality that $\ell(x) - 1 = \ell(z) = \ell(y) + 1$. Thus x' and z are of the same level and have a common neighbor x in σ. By the lemma, either D^* occurs or x' and z have a common neighbor x'' of level $\ell(z) - 1$. Now y and x'' are of the same level and have a common neighbor z. Hence, by the lemma, either D^* occurs or y, or x'' have a common neighbor x''' of level $\ell(y) - 1$ and we obtain a bipartite tent induced by $w, x, x', x''x''', y, z$.

Case 2. Suppose that $N_{k-1}(x) \supset N_{k-1}(y)$ and $N_{k+1}(x) \supset N_{k+1}(y)$.

Let $x' \in N_{k-1}(x) - N_{k-1}(y)$ and $x'' \in N_{k+1}(x) - N_{k+1}(y)$. Since $N_{k-1}(x) \supset N_{k-1}(y)$, we must have $x <_+ y$, by property P. Let $z \in N_{k-1}(y)$. Then $z \in N_{k-1}(x)$ as $N_{k-1}(x) \supset N_{k-1}(y)$. Thus x' and z are of the same level in σ^+ and have a common neighbor x of level $\ell^+(x') + 1$. By the lemma, either D^* occurs or x' and z have a common neighbor w of level $k - 2$ in σ^+. Thus we must have $k \geq 2$. If $k \geq 3$, then we obtain a bipartite net in H induced by w, x', z, x, y, x'' and any neighbor of w of level $k - 3$. So assume $k = 2$.

Consider again the ordering σ. Since w is the first vertex in σ^+, it is the last vertex in σ. Thus z, x' are of the same level in σ.

Suppose that x is of level smaller than z in σ. If y is of the same level as w, then there must be a vertex $u < x'$ adjacent to y but not to w (as $y < w$), by the property P. If u is adjacent to x, then we obtain the bipartite tent induced by u, x, x', x'', z, y, w. So assume that u is not adjacent to x. If $u < z$, then $x < u < z < y$ form D^* in σ. Hence $z < u < x'$. By the lemma, either D^* occurs in σ, or z and u have a common neighbor v of level $\ell(z) - 1$. Thus v and x are of the same level and have a common neighbor z. Again by the lemma, either D^* occurs in σ, or v and x have a common neighbor v' of level $\ell(v) - 1$. We obtain a bipartite tent induced by v', v, x, z, u, y, w. Thus let y be not of the same level as w, and hence of the same level as x in σ. By the lemma, either D^* occurs in σ, or x and y have a common neighbor u' of level $\ell(x) - 1$ and we obtain a bipartite tent induced by u', x, x', x'', z, y, w.

Therefore assume that x is of level greater than z in σ, i.e., of the same level as w. Thus x'' and z both adjacent to x are of the same level. By the lemma, either D^* occurs in σ, or x'' and z have a common neighbor z' of level $\ell(z) - 1$. If z' is not adjacent to x', then we obtain a bipartite tent induced by y, z', x'', z, x', x, w. So assume that z'

is adjacent to x'.

Suppose that y is of the same level as w in σ. Since $y < w$ and x' is adjacent to w but not to y, there exists, by the property P, a vertex $y' < x'$ which is adjacent to y but not to w. If y' is adjacent to z', then we obtain a bipartite tent induced by y', z', y, z, x'', x, w. So y' is not adjacent to z'. This implies in particular that $y' > z$, or else $z' < y' < z < y$ form a D^* in σ. By the lemma, either D^* occurs in σ, or z and y' have a common neighbor y'' of level $\ell(z) - 1$. Note that y'' and z' are distinct (as z' is not adjacent to y'), of the same level, and both adjacent to z. Once more, by the lemma, either D^* occurs in σ, or z' and y'' have a common neighbor y''' of level smaller than $\ell(z') - 1$, and we obtain a bipartite tent induced by $y''', z', y'', z, y', y, w$.

Finally, if y is not of the same level as w in σ, and thus of the same level as z', then, again by the lemma, either D^* occurs in σ, or y and z' must have a common neighbor z'' of level $\ell(y) - 1$, and we obtain a bipartite tent induced by z'', z', y, z, x'', x, w. □

We now make the following observation which will be useful in the next proposition.

LEMMA 3.6. *Suppose that σ^+ contains neither D nor E_1. If $\ell^+(x) \leq \ell^+(y) - 2$, or if $\ell^+(x) = \ell^+(y)$ and x has a neighbor of level $\ell^+(x) - 1$ nonadjacent to y, then $y <_{++} x$.*

Proof. We first note that under both assumptions we must have $x <_+ y$. Indeed, this is obvious if $\ell^+(x) \leq \ell^+(y) - 2$. On the other hand, if $\ell^+(x) = \ell^+(y)$, then $y <_+ x$ would only be possible, by property P, if another vertex of level $\ell^+(x) - 1$ was a neighbor of y and nonadjacent to x. This implies that the vertices x and y are inconsistent and contradicts the fact that σ^+ contains neither D nor E_1.

Now suppose to the contrary that $x <_{++} y$. We may assume that, among vertices v such that $\ell^+(v) \leq \ell^+(y) - 2$, or $\ell^+(v) = \ell^+(y)$ and v has a neighbor of level $\ell^+(v) - 1$ nonadjacent to y, the vertex x comes first in the ordering σ^{++}.

Since $x <_+ y$ and $x <_{++} y$, property R implies that there must be a vertex $x' <_{++} x$ which is adjacent to x but not to y. We claim that x' cannot be the first vertex of σ^{++}, i.e., that x' cannot be the last vertex of σ^+. In fact, we must have $\ell^+(x') < \ell^+(y)$: This is clear if $\ell^+(x) \leq \ell^+(y) - 2$, since x' is adjacent to x. If $\ell^+(x) = \ell^+(y)$, and x'' is a neighbor of x of level $\ell^+(x) - 1$ nonadjacent to y, then we cannot have $\ell^+(y) < \ell^+(x')$, since this would mean that $x'' <_+ x <_+ y <_+ x'$ form an E^*. (Note that x' and y have different colors, and hence cannot be of the same level.)

Consider a shortest path Q from x' to the first vertex of σ^{++} (which is the last vertex of σ^+). Thus Q contains only one vertex from each level of σ^{++}, up to the level of x', which is smaller than the level of x and of y. In particular, y is not adjacent to any vertex of the path Q. On the other hand, since $\ell^+(x') < \ell^+(y)$, the path Q must contain a pair of adjacent vertices w and u with $\ell^+(w) = \ell^+(y) = \ell^+(u) + 1$. This contradicts the choice of x, since w comes before x in σ^{++}, $\ell^+(w) = \ell^+(y)$, and w has a neighbor u of level $\ell^+(w) - 1$ which is not adjacent to y. □

The crucial property of the third sweep is captured in the following result.

PROPOSITION 3.7. *Suppose that σ^+ does not contain D or E_1. Then σ^{++} does not contain E_2.*

Proof. Suppose to the contrary that $x <_{++} y <_{++} z$ form E_2. Since x is adjacent to z, we must have $\ell^{++}(x) = \ell^{++}(y)$. Let w be any neighbor of x of level $\ell^{++}(x) - 1$. By the definition of E_2, w is also a neighbor of y. Note that w and z have the same color, as they are both adjacent to x. Hence either $|\ell^+(w) - \ell^+(z)| \geq 2$ or $\ell^+(w) = \ell^+(z)$. Since $w <_{++} z$, by Lemma 3.6, either $\ell^+(z) \leq \ell^+(w) - 2$, or $\ell^+(w) = \ell^+(z)$ and w does not have a neighbor of level $\ell^+(w) - 1$ nonadjacent to z.

Suppose first that $\ell^+(z) \leq \ell^+(w) - 2$. Since x is a common neighbor of w and z, we must have $\ell^+(w) - 1 = \ell^+(x) = \ell^+(z) + 1$. Since y is adjacent to w, the level of y in σ^+ is $\ell^+(w) + 1 \geq \ell^+(x) + 2$ or $\ell^+(w) - 1 = \ell^+(x)$. In either case, Lemma 3.6 implies that $y <_{++} x$, which contradicts our assumption that $x <_{++} y$.

Thus suppose that $\ell^+(w) = \ell^+(z)$ and w does not have a neighbor of level $\ell^+(w)-1$ nonadjacent to z. Since y is a neighbor of w but not of z, the level of y in σ^+ can only be $\ell^+(w) + 1$. Now consider the vertex x which is a common neighbor of w and z. The level of x in σ^+ is $\ell^+(w) - 1 \leq \ell^+(y) - 2$ or $\ell^+(w) + 1 = \ell^+(y)$. In either case we again have, by Lemma 3.6, that $y <_{++} x$, which again contradicts our assumption that $x <_{++} y$. □

THEOREM 3.8. *The 3-sweep certifying algorithm for the recognition of proper interval bigraphs is correct and can be implemented to run in time $O(m+n)$ (including the certification step).*

Proof. The correctness of the algorithm follows from Lemma 3.3 and Propositions 3.4 and 3.7.

To test for the occurrences of D or E_1 in σ^+ and σ^{++} we shall compute the following parameters for each vertex v_i and each of the LexBFS orderings σ^+ and σ^{++}:

- $u(i)$, the greatest subscript j such that v_j is of the same level as v_i;
- $l(i)$, the smallest subscript k such that v_k is of the same level as v_i;
- $U(i)$, the greatest subscript j such that v_j is adjacent to or of the same level as v_i;
- $L(i)$, the smallest subscript k such that v_k is adjacent to or of the same level as v_i;
- $d_U(i)$, the number of neighbors v_j of v_i with $j > i$;
- $d_L(i)$, the number of neighbors v_k of v_i with $k < i$;
- $x(i)$, the greatest subscript j such that v_j is of the same level as v_i and $d_L(j) = d_L(i)$;
- $d_x(i)$, the number of neighbors v_k with $x(L(i)) < k \leq u(L(i))$.

It is easy to compute all these parameters in time $O(m + n)$.

It is easy to see that $d_U(i) \leq U(i) - u(i)$ for all $i = 1, 2, \ldots, n$ and that a D exists if and only if $d_U(i) < U(i) - u(i)$ for some i. (Such a vertex v_i is not adjacent to some v_j with $u(i) < j < U(i)$ and $v_i, v_j, v_{U(i)}$ form a D.) Thus to test for occurrence of D, we check whether $d_U(i) = U(i) - u(i)$ for each $i = 1, 2, \ldots, n$. If $d_U(i) < U(i) - u(i)$ for some i, then we detect D by checking at most $d_U(i)$ vertices with subscripts between $u(i) + 1$ and $U(i) - 1$. Clearly this can be done in time $O(n)$.

Suppose that no D is detected; that is, $d_U(i) = U(i) - u(i)$ for each $i = 1, 2, \ldots, n$. Then we proceed to test for occurrence of E_1. Observe that the absence of D implies that if v_p and v_q with $p < q$ are two vertices of the same level, then $d_L(p) \geq d_L(q)$. Hence there exists an E_1 with third vertex v_i if and only if some vertex v_k with $x(L(i)) < k \leq u(L(i))$ is nonadjacent to v_i. This can be checked by testing whether or not $d_x(i) = u(L(i)) - x(L(i))$. If the equality does not hold for some i, we can find, in time $O(n)$, a vertex v_k and a vertex w, which, together with $v_{L(i)}$ and v_i, form E_1.

When D or E_1 is detected, Lemma 3.3 and Proposition 3.4 show how to find an induced cycle of length at least six or an induced forbidden subgraph from Figure 1.2. It is easy to see that the proofs of Proposition 3.4 represent algorithms that can be implemented to run in time $O(n)$ and that the proof of Lemma 3.3 represents an algorithm that can be implemented to run in time $O(m + n)$. Therefore the overall complexity of the algorithm is $O(m + n)$.

We again remark that except for the lexicographic searches, the computation of

the various parameters itemized above, and the computation implicit in the proof of Lemma 3.3 (not needed if the graph is known to be chordal bipartite), the remaining tasks can be performed in time $O(n)$. □

We note in passing that we have avoided checking for D_1 alone, and at present there is no $O(m+n)$ algorithm known for finding (or deciding the existence of) an ordering without D_1 (i.e., without D^*).

When the algorithm finds an ordering satisfying the weak 3-vertex condition, a representation by an inclusion-free family of intervals can be found in time $O(m+n)$ (Proposition 3.1) and authenticated also in time $O(m+n)$ (cf. [18]). When it finds an induced cycle of length at least six or induced bipartite claw, net, or tent, these can be authenticated in time $O(n)$, as in [18].

We thank the referees for their insightful suggestions.

REFERENCES

[1] A. Brandstädt, V. B. Le, and J. P. Spinrad, *Graph Classes: A Survey*, SIAM Monogr. Discrete Math. Appl. 3, SIAM, Philadelphia, 1999.

[2] A. Bretscher, D. G. Corneil, M. Habib, and C. Paul, *A Simple LexBFS Based Cograph Recognition Algorithm*, manuscript.

[3] D. E. Brown, J. R. Lundgren, and S. C. Flink, *Characterizations of interval bigraphs and unit interval bigraphs*, in Congressus Num., 157 (2002), pp. 79–93.

[4] D. E. Brown and J. R. Lundgren, *Several Characterizations of Unit Interval Bigraphs*, manuscript, 2003.

[5] J.-M. Chang, C.-W. Ho, and M.-T. Ko, *LexBFS-ordering in asteroidal triple-free graphs*, in Algorithms and Computation, Lecture Notes in Comput. Sci. 1741, Springer-Verlag, Berlin, 1999, pp. 163–172.

[6] V. Chepoi, *On distance-preserving and domination elimination orderings*, SIAM J. Discrete Math., 11 (1998), pp. 414–436.

[7] D. G. Corneil, *A simple 3-sweep LBFS algorithm for the recognition of unit interval graphs*, Discrete Appl. Math., 138 (2004), pp. 371–379.

[8] D. G. Corneil, H. Kim, S. Natarajan, S. Olariu, and A. P. Sprague, *Simple linear time recognition of unit interval graphs*, Inform. Process. Lett., 55 (1995), pp. 99–104.

[9] D. G. Corneil, S. Olariu, and L. Stewart, *A Multisweep LBFS Algorithm for the Recognition of Interval Graphs*, manuscript.

[10] D. G. Corneil, *private communication*, 2003.

[11] E. Dahlhaus, P. L. Hammer, F. Maffray, and S. Olariu, *On domination elimination orderings and domination graphs*, in Proceedings of the 20th International Workshop in Graph-Theoretic Concepts in Computer Science, Lecture Notes in Comput. Sci. 903, Springer-Verlag, Berlin, 1994, pp. 81–92.

[12] X. Deng, P. Hell, and J. Huang, *Linear-time representation algorithms for proper circular-arc graphs and proper interval graphs*, SIAM J. Comput., 25 (1996), pp. 390–403.

[13] F. F. Dragan and F. Nicolai, *LexBFS-Orderings of Distance-Hereditary Graphs*, Math. Tech. Report, SM-DU-322, University of Duisburg, Germany, 1996.

[14] C. M. H. de Figueiredo, J. Meidanis, and C. P. de Mello, *A linear time algorithm for proper interval graph recognition*, Inform. Process. Lett., 56 (1995), pp. 179–184.

[15] M. Habib, R. McConnell, C. Paul, and L. Viennot, *Lex-bfs and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing*, Theoret. Comput. Sci., 234 (2000), pp. 59–84.

[16] P. Hell and J. Huang, *Lexicographic orientation and representation algorithms for comparability graphs, proper circular arc graphs, and proper interval graphs*, J. Graph Theory, 20 (1995), pp. 361–374.

[17] P. Hell and J. Huang, *Interval bigraphs and circular arc graphs*, J. Graph Theory, 46 (2004), pp. 313–327.

[18] D. Kratsch, R. M. McConnell, K. Mehlhorn, and J. P. Spinrad, *Certifying algorithms for recognizing interval graphs and permutation graphs*, in Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Baltimore, MD, 2003, pp. 158–167.

[19] T. H. Lai and S. S. Wei, *Bipartite permutation graphs with application to the minimum buffer size problem*, Discrete Appl. Math., 74 (1997), pp. 33–55.

[20] P. J. Looges and S. Olariu, *Optimal greedy algorithm for indifference graphs*, Comput. Math. Appl., 25 (1993), pp. 15–25.
[21] A. Lubiw, *Doubly lexical orderings of matrices*, SIAM J. Comput., 16 (1987), pp. 854–879.
[22] T. Ma, unpublished manuscript related to problems with [28].
[23] R. M. McConnell, *A certifying algorithm for the consecutive-ones property*, in Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, 2004, pp. 761–770.
[24] D. Meister, *Recognition and computation of minimal triangulations for AT-free, claw-free and co-comparability graphs*, Discrete Appl. Math., to appear.
[25] F. S. Roberts, *Indifference graphs*, in Proof Techniques in Graph Theory, F. Harary, ed., Academic Press, New York, 1969, pp. 301–310.
[26] D. J. Rose, R. E. Tarjan, and G. S. Lueker, *Algorithmic aspects of vertex elimination on graphs*, SIAM J. Comput., 5 (1976), pp. 266–283.
[27] M. Sen and B. K. Sanyal, *Indifference digraphs: A generalization of indifference graphs and semiorders*, SIAM J. Discrete Math., 7 (1994), pp. 157–165.
[28] K. Simon, *A new simple linear algorithm to recognize interval graphs*, in International Workshop on Computational Geometry CG'91, Bern, Switzerland, 1991, H. Bieri and H. Noltemeyer, eds., Lecture Notes in Comput. Sci. 553, Springer-Verlag, Berlin, 1992, pp. 289–308.
[29] J. Spinrad, *Efficient Graph Representations*, AMS, Providence, RI, 2003.
[30] J. Spinrad, A. Brandstädt, and L. Stewart, *Bipartite permutation graphs*, Discrete Appl. Math., 18 (1987), pp. 279–292.
[31] A. P. Sprague, *Recognition of bipartite permutation graphs*, Congr. Numer., 112 (1995), pp. 151–161.
[32] 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.
[33] R. E. Tarjan and M. Yannakakis, *Addendum: Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs*, SIAM J. Comput., 14 (1985), pp. 254–255.
[34] H. Wasserman and M. Blum, *Software reliability via run-time result-checking*, J. ACM, 44 (1997), pp. 826–849.
[35] G. Wegner, *Eigenschaften der nerven homologische-einfactor familien in R^n*, Ph.D. thesis, Universität Gottingen, Germany, 1967.
[36] D. B. West, *Short proofs for interval digraphs*, Discrete Math., 178 (1998), pp. 287–292.

SIAM J. DISCRETE MATH.
Vol. 18, No. 3, pp. 571–576

ON THE CO-P_3-STRUCTURE OF PERFECT GRAPHS*

CHÍNH T. HOÀNG† AND BRUCE REED‡

Abstract. Let $\mathcal{F}$ be a family of graphs. Two graphs $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ are said to have the same $\mathcal{F}$-structure if there is a bijection $f : V_1 \to V_2$ such that a subset S induces a graph belonging to $\mathcal{F}$ in G_1 if and only if its image $f(S)$ induces a graph belonging to $\mathcal{F}$ in G_2. We characterize those graphs which have the same $\{P_3, \overline{P}_3\}$-structure, or the same $\{K_3, \overline{K}_3\}$-structure. This characterization shows that graph H is perfect if and only if it has the $\{P_3, \overline{P}_3\}$-structure of some perfect graph G. In proving the main result, we need and prove the following result, which is of independent interest: If a graph J is claw-free and co-claw-free, then either (i) J has at most nine vertices, or (ii) every component of J is a path or a hole, or (iii) every component of $\overline{J}$ is a path or a hole.

Key words. graph coloring, perfect graph, claw-free graph

AMS subject classifications. 05C15, 05C17

DOI. 10.1137/S0895480100370797

1. Introduction.

Let $\mathcal{F}$ be a family of graphs. Two graphs $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ are said to have the same $\mathcal{F}$-structure if there is a bijection $f : V_1 \to V_2$ such that a subset S induces a graph belonging to $\mathcal{F}$ in G_1 if and only if its image $f(S)$ induces a graph belonging to $\mathcal{F}$ in G_2. In [4], Chvátal discussed the $\mathcal{F}$-structure when $\mathcal{F} = \{P_4\}$ in the context of perfect graph theory. A graph G is *perfect* if for each induced subgraph H of G, the chromatic number of H equals the number of vertices in a largest clique of H. A conjecture of Berge, which was proved by Lovász [10], states that a graph is perfect if and only if its complement is. This result is nowadays known as the Perfect Graph Theorem (PGT). Berge [1] also made a stronger conjecture stating that a graph is perfect if and only if it does not contain as an induced subgraph the odd chordless cycle on at least five vertices or its complement. This conjecture was known as the Strong Perfect Graph Conjecture (SPGC). Graphs satisfying the hypothesis of the SPGC are called *Berge graphs*. Recently, Chudnovsky et al. [3] announced a proof of the SPGC. This result is now known as the Strong Perfect Graph Theorem (SPGT).

Chvátal [4] conjectured that a graph H is perfect if and only if it has the P_4-structure of some perfect graph G. This was first proved by Reed [12], and is now a consequence of [3]. Further research into $\mathcal{F}$-structures also centered around its relationship to the SPGC [7], [8], [9].

In this paper, we prove two theorems of this nature. Let K_t (respectively, P_t) denote the clique (respectively, induced path) on t vertices.

THEOREM 1. *A graph H is perfect if and only if it has the $\{K_3, \overline{K}_3\}$-structure of a perfect graph G.* □

THEOREM 2. *A graph H is perfect if and only if it has the $\{P_3, \overline{P}_3\}$-structure of a perfect graph G.*

*Received by the editors March 29, 2000; accepted for publication (in revised form) June 6, 2004; published electronically February 25, 2005. This research was supported by NSERC.
http://www.siam.org/journals/sidma/18-3/37079.html

†Department of Physics and Computer Science, Wilfrid Laurier University, Waterloo, Ontario, Canada N2L 3C5 (choang@wlu.ca).

‡School of Computer Science, McGill University, Montreal, Quebec, Canada.

It is easy to see that two graphs have the same $\{K_3, \overline{K}_3\}$-structure if and only if they have the same $\{P_3, \overline{P}_3\}$-structure. Thus, Theorems 1 and 2 are equivalent. Furthermore, it is a tedious but routine matter to show that graphs with the $\{P_3, \overline{P}_3\}$-structure of a Berge graph are Berge. Thus, our two theorems are implied by the SPGT. These two theorems were proved before the proof of the SPGC was announced and the original motivation was that they might be a step towards a proof of the SPGC. We feel they are still interesting because (a) they might help in finding a shorter proof of the SPGC, and (b) we prove the following result of independent interest. A *triangle* is the clique on three vertices. A *claw* is the graph with vertices a, b, c, d and edges ab, ac, ad.

THEOREM 3. *If a graph J and its complement $\overline{J}$ are claw-free, then either* (i) *J has at most nine vertices, or* (ii) *J and $\overline{J}$ are triangle-free.*

In section 2, we will discuss background results needed to prove Theorems 2 and 3, the proofs of which will be given in section 3.

2. Definitions and background. Before proving Theorem 2, we need to introduce some definitions and background results. Let G be a graph. Let S be a set of vertices of G. $G[S]$ will denote the subgraph of G induced by S (for simplicity, we will write $G[a, b, \ldots]$ for $G[\{a, b, \ldots\}]$). Let x be a vertex of G. Then $N_G(x)$ denotes the set of vertices adjacent to x in G. If $xy \in E(G)$, then we say that x *sees* y in G, otherwise we say that x *misses* y in G. We use $v_1v_2 \ldots v_k$ to denote the chordless path with vertices $v_1, v_2, \ldots, v_k$ and edges v_iv_{i+1} for $i = 1, 2, \ldots, k-1$. A *hole* is a chordless cycle with at least four vertices. Two vertices x, y are *antitwins* of G if every vertex z, different from x and y, sees x or y but not both. Note that x, y are antitwins in G if and only if they are antitwins in $\overline{G}$. Two nonempty sets A, B of vertices of G are a *homogeneous pair* if (i) A or B has at least two vertices, (ii) there are at least two vertices outside $A \cup B$, and (iii) for any vertex x outside $A \cup B$, if x sees some vertex in A (respectively, B) then it sees all vertices of A (respectively, B). If A, B are a homogeneous pair in G, then they are a homogeneous pair in $\overline{G}$.

A graph is *minimal imperfect* if it is not perfect but each of its proper induced subgraphs is. The PGT implies that a graph is minimal imperfect if and only if its complement is. We will rely on the following two results on minimal imperfect graphs.

Olariu [11] proved that

(1) a minimal imperfect graph cannot contain antitwins.

Chvátal and Sbihi [5] proved that

(2) a minimal imperfect graph cannot contain a homogeneous pair.

If two graphs have the same $\{P_3, \overline{P}_3\}$-structure, then we will say that they have the same *co-P_3-structure*. If F is a graph, then *co-F* denotes the complement of F. For example, the cotriangle is the complement of a triangle. The *pyramid* is the graph obtained from taking a triangle with vertices a, b, c and a cotriangle with vertices x, y, z and adding edges xa, xb, yb, yc, za, zb; such a pyramid will be referred to as $P(a, b, c, x, y, z)$. A graph is *elementary* [6] if its edges can be colored by two colors in such a way that each edge receives a color and there is no monochromatic P_3. Pyramids and copyramids, claws, odd holes, and odd antiholes are not elementary.

3. The proofs. We are going to prove a theorem stronger than Theorem 2.

THEOREM 4. *If two graphs G and H have the same co-P_3-structure and H is not isomorphic to G or $\overline{G}$, then either* (i) *both G and H have antitwins or a homogeneous pair, or* (ii) *G and H have at most nine vertices.*

The key to Theorem 4 is Theorem 3.

Proof of Theorem 3. Let J be a claw-free, co-claw-free graph. Suppose that both (i) and (ii) fail. Thus, J contains a triangle K, and a cotriangle S. Let $K = \{a, b, c\}$ and $S = \{x, y, z\}$.

Note that

(3) every vertex u of $J - K$ must see at least one vertex of K,

for otherwise there is a coclaw induced by K and u.

Suppose that $K \cap S = \emptyset$. We claim that

there cannot be a vertex of S that sees at least two vertices of K and

(4) another vertex of S that misses at least two vertices of K.

Suppose x sees vertices a, b of K, and without loss of generality, y misses at least two vertices of K. If y misses a, b then $\{a, b, x, y\}$ is a coclaw in J. So, without loss of generality we may assume y misses a and c. Now, y sees b (by (3)), and x misses c (for otherwise $\{a, c, x, y\}$ is a coclaw). Now, $\{b, x, y, c\}$ is a claw. So (4) holds. Next, we prove that

(5) there cannot be a vertex of S that sees three vertices of K.

Suppose x sees all three vertices of K. Then (4) implies that y sees (at least) two vertices, say a, b, of K. Both a and b must miss z, for otherwise there is a claw with vertices x, y, z and a or b. Now, there is a coclaw with z, a, b, x. So (5) holds. Next, we prove that

(6) if some vertex of S sees two vertices of K, then $K \cup S$ induces a pyramid.

If some vertex of S sees two vertices of K, then, by (3), (4), and (5), every vertex of S sees exactly two vertices of K. If two vertices, say x, y, of S see the same two vertices, say a, b, of K, then a and b miss z (for otherwise, there is a claw with S and a or b), a contradiction to (4). It is now easy to see that $K \cup S$ induces a pyramid. Next, we prove that

(7) if $K \cup S$ induces a pyramid, then J contains a claw or coclaw.

By assumption, J has a vertex t outside the pyramid $P(a, b, c, x, y, z)$. Vertex t cannot be adjacent to all vertices in S, for otherwise $\{t, x, y, z\}$ induces a claw. We may assume that t misses x.

Suppose t sees a. Then t sees z (for otherwise $\{a, x, t, z\}$ is a claw) and c (for otherwise $\{a, x, t, c\}$ is a claw). But now $\{x, t, z, c\}$ is a coclaw.

So t misses a, and by symmetry t misses b. But then $\{t, x, a, b\}$ induces a coclaw. We have proved (7).

By (6) and (7), we may assume that no vertex of S sees two vertices of K. It follows from (3) that every vertex of S sees exactly one vertex of K. If some two vertices, say x, y, of S see the same vertex, say a, in K, then there is a claw (induced by $\{a, x, y, b\}$). Now, it is easy to see that $K \cup S$ induces a copyramid. By (7), J contains a claw or coclaw. Thus,

(8) J cannot contain a triangle that is disjoint from a cotriangle.

Consider a vertex a that is the intersection of a triangle $K = \{a, b, c\}$ and a cotriangle S. Let N be the set of neighbors of a, and M be the set of nonneighbors of a (different from a). By (8), M has no cotriangle. Also, M has no triangle, for otherwise this triangle and a form a coclaw. Thus M has at most five vertices (this is a well-known case of Ramsey's theorem).

Suppose M has five vertices. Then M is the C_5. Enumerate the vertices of the C_5 as $v_1, v_2, \ldots, v_5$ in the cyclic order. By (3), each v_i sees b, or c, or both. We may suppose v_1 sees b. Then b has to miss v_3 and v_4, for otherwise $\{b, a, v_1, v_3\}$ or $\{b, a, v_1, v_4\}$ induces a claw. Now (3) implies that c sees v_3 and v_4, and therefore, c misses v_1, for otherwise $\{c, a, v_3, v_1\}$ induces a claw. But then $\{v_1, c, v_3, v_4\}$ induces a coclaw. (This argument actually implies that M is P_4-free.)

So, M has at most four vertices. A similar argument applied to $\overline{J}$ shows that N has at most four vertices. □

COROLLARY 5. *If a graph J is claw-free and co-claw-free, then either* (i) *J has at most nine vertices, or* (ii) *every component of J is a path or a hole, or* (iii) *every component of $\overline{J}$ is a path or a hole.*

Proof of Corollary 5. By Theorem 3, we may assume that J is triangle-free. Consider a component C of J. If C contains a hole F of length at least 4, then $F = C$, for otherwise some vertex in $C - F$ has a neighbor in F and it follows that C contains a claw. Thus C must be a tree and therefore a path since J is claw-free. □

Proof of Theorem 4. Let G and H be two graphs with the same co-P_3-structure. We may assume that G and H are defined on the same vertex-set in such a way that for any set X of vertices, X induces a P_3 or co-P_3 in G if and only if X does so in H. We may suppose H is not isomorphic to G or $\overline{G}$. A pair (x, y) of vertices will be called *variant* if x sees y in H but misses it in G, or vice versa. We may assume that

(9) H contains a variant pair (x, y),

for otherwise H is isomorphic to G or to $\overline{G}$, a contradiction. A pair of vertices is *invariant* if it is not variant. By the *variant graph* J, we mean a graph whose vertices are those of G and in which a pair of vertices is joined by an edge if and only if they are a variant pair. We say that J is the variant graph for G and H. We can two-color the edges of J so that xy is colored 1 if xy is an edge of G, and 2 if xy is an edge of H. We say that a set is *bad* if it induces a P_3 or co-P_3 in H but does not do so in G, or vice versa. The fact that G and H have the same co-P_3-structure implies that J contains no monochromatic P_3 (such a P_3 would be bad); thus

(10) J is elementary.

It is easy to see that

(11) an elementary graph is bipartite if and only if it is triangle-free.

Since $\overline{J}$ is the variant graph for G and $\overline{H}$, $\overline{J}$ is elementary. Thus J is claw-free and co-claw-free. By Corollary 5, we may assume that each component of J is a path or a hole. Since J is elementary,

(12) each component of J is a path or an even cycle.

OBSERVATION 1. *Let ab be an edge of J and let x be a vertex with $xa, xb \notin E(J)$. Then, in H and G, x sees exactly one vertex of $\{a, b\}$.*

Proof of Observation 1. In G and in H, x cannot see, or miss, both a and b, for otherwise the set $\{a, b, x\}$ is bad. So, in G and in H, x sees exactly one vertex in $\{a, b\}$. □

We know there is a component with at least two vertices. We will prove that

if J has a component, different from a C_4,

(13) with at least two vertices, then H has antitwins.

Consider a component K of J with at least two vertices. Since K is a path or a cycle, we can enumerate the vertices of K as $v_1, v_2, \ldots, v_t$ such that $v_i v_{i+1} \in E(J)$ for $i = 1, 2, \ldots t-1$, and if K is a cycle, then $v_1 v_t \in E(J)$. We may assume that $v_1 v_2 \in E(H)$ (for otherwise, we can replace H by $\overline{H}$ and G by $\overline{G}$ in the following argument. Note that antitwins of H are antitwins of $\overline{H}$). It follows that $v_i v_{i+1} \in E(H)$ if and only if i is odd. We are going to show that if K is not a C_4, then either $\{v_1, v_2\}$ or $\{v_2, v_3\}$ is a pair of antitwins of H.

Suppose K is not a C_4 but $\{v_1, v_2\}$ is not a pair of antitwins of H. Then, in H there is a vertex u seeing, or missing, both vertices v_1 and v_2. Observation 1 implies $u = v_3$ or $v_t v_1 \in E(J)$ and $u = v_t$. Without loss of generality, we may assume $u = v_3$. In H, since v_3 misses v_2, v_3 misses v_1. Now, in H, v_4 sees v_2, for otherwise, $\{v_2, v_3\}$ is a pair of antitwins by Observation 1. The same observation, with $a = v_1, b = v_2, x = v_4$ implies $v_4 v_1 \notin E(H)$. But then the observation is contradicted with $a = v_3, b = v_4, x = v_1$. We have established (13).

Suppose some component K of J is a C_4. Then it is easy to see that H has a homogeneous pair (with $A = \{v_1, v_3\}, B = \{v_2, v_4\}$) whenever J has at least six vertices, and if J has five vertices, then (ii) holds. Theorem 4 is proved. □

Now, we prove Theorem 2.

Proof of Theorem 2. Let G and H be two graphs with the same co-P_3-structure. Suppose G is perfect but H is not. Every imperfect graph contains a minimal imperfect graph. So we may assume H is minimal imperfect. By Theorem 4, (1), (2), and the perfect graph theorem, H has at most nine vertices.

It is a tedious but routine matter to verify that all graphs on at most nine vertices with the co-P_3-structure of a perfect graph are Berge (indeed, one can show easily that graphs with the co-P_3-structure of Berge graphs are Berge but this is much more than is needed here). It is well known and easy to prove that Berge graphs with at most nine vertices are perfect. □

Note added in proof. After this paper was written, we learned that a slightly weaker version of Corollary 5 was previously proved by A. Brandstädt and S. Mahfud [2]. They proved that if G is claw-free and co-claw-free, then G or $\overline{G}$ is a chordless path or cycle, or G has a homogeneous set, or G has at most nine vertices.

REFERENCES

[1] C. Berge, *Les problèmes de coloration en théorie des graphes*, Publ. Inst. Stat. Univ. Paris, 9 (1960), pp. 123–160.

[2] A. Brandstädt and S. Mahfud, *Maximum weight stable set on graphs without claw and co-claw (and similar graph classes) can be solved in linear time*, Inform. Process. Lett., 84 (2002), pp. 251–259.

[3] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas, *The Strong Perfect Graph Theorem*, manuscript, 2002.

[4] V. Chvátal, *A semi-strong perfect graph conjecture*, in Topics on Perfect Graphs, C. Berge and V. Chvátal, eds., North-Holland, Amsterdam, 1984, pp. 279–280.

[5] V. Chvátal and N. Sbihi, *Bull-free perfect graphs*, Graphs Combin., 3 (1987), pp. 127–140.
[6] V. Chvátal and N. Sbihi, *Recognizing claw-free berge graphs*, J. Combin. Theory Ser. B, 44 (1988), pp. 154–176.
[7] C. Hoàng, *On the disc-structure of perfect graphs* I. *The paw-structure*, Discrete Appl. Math., 94 (1999), pp. 247–262.
[8] C. Hoàng, *On the disc-structure of perfect graphs II. The co-C_4-structure*, Discrete Math., 252 (2002), pp. 141–159.
[9] S. Hougardy, *On the P_4-Structure of Perfect Graphs*, Shaker-Verlag, Aachen, 1995.
[10] L. Lovász, *Normal hypergraphs and the perfect graph conjecture*, Discrete Math., 2 (1972), pp. 253–267.
[11] S. Olariu, *No antitwins in minimal imperfect graphs*, J. Combin. Theory Ser. B, 45 (1988), pp. 255–257.
[12] B. Reed, *A semi-strong perfect graph theorem*, J. Combin. Theory Ser. B, 43 (1987), pp. 223–240.

SIAM J. DISCRETE MATH.
Vol. 18, No. 3, pp. 577–580

EDGE-DISJOINT ISOMORPHIC MULTICOLORED TREES AND CYCLES IN COMPLETE GRAPHS*

GREGORY M. CONSTANTINE†

Abstract. It is shown that a complete graph with a prime number $p\,(>2)$ of vertices can be properly edge-colored with p colors in such a way that the edges can be partitioned into edge-disjoint multicolored Hamitonian cycles. When the number of vertices is $n\,(\geq 8)$, with n a power of two or five times a power of two, a proper edge-coloring of the complete graph exists such that its edges can be partitioned into isomorphic multicolored spanning trees. A subgraph is multicolored if each of its edges carries a different color.

Key words. proper coloring, orthogonal Latin squares, multicolored spanning paths

AMS subject classifications. 05C15, 05C05

DOI. 10.1137/S0895480101397402

1. Multicolored edge partitions in complete graphs. Basic terminology and notation on graph theory is found in [4]. A coloring of edges of a graph is *proper* if, whenever two edges have one vertex in common, they carry different colors. A graph with colored edges is called *multicolored* if no two of its edges have the same color. Two subgraphs are *edge disjoint* if they do not share common edges. Two graphs with colored edges are *isomorphic* if there exists a bijection σ between the sets of vertices and a bijection η between the sets of colors such that (i,j) is an edge of color c if and only if $(\sigma(i),\sigma(j))$ is an edge of color $\eta(c)$. Denote by K_s the complete graph on s vertices. A connected graph with m vertices and m edges is called a *unicycle*; such a graph necessarily consists of a spanning tree plus another edge. An example of such a graph is a (Hamiltonian) cycle. We investigate the possibility of producing a proper edge-coloring of K_s such that its edges can be partitioned into either edge-disjoint isomorphic multicolored unicycles (this requires s odd) or isomorphic multicolored spanning trees (s even). When this is possible, we obtain what we call a *multicolored cycle (or tree) parallelism* of K_s. Such a partition of the edges of K_s can be viewed as a parallelism as defined in [8] with an additional restriction due to color.

When no coloring is involved, it is well known, and a classical result of Euler, that the edges of K_{2n} can be partitioned into isomorphic spanning trees (paths, for example). Each of these spanning trees can easily be made multicolored, but the resulting edge coloring usually fails to be proper. Indeed, it is easy to verify that the sole proper coloring of K_6 does not admit a partition into multicolored spanning (trees which are) paths. In addition, there exists a proper coloring of K_8 that does not admit even a single multicolored spanning path; see [7]. Euler also decomposed K_{2n+1} into n edge-disjoint Hamiltonian cycles. In this paper, these results are extended to properly colored complete graphs by showing that edge-disjoint partitions into isomorphic multicolored spanning trees (or Hamiltonian cycles) exist for infinite families of complete graphs. The generating function of the multicolored spanning trees in

*Received by the editors November 5, 2001; accepted for publication (in revised form) March 9, 2004; published electronically February 25, 2005. This work was funded in part by a Scaife Family Foundation grant.

http://www.siam.org/journals/sidma/18-3/39740.html

†Department of Mathematics, University of Pittsburgh, Pittsburgh, PA 15260 (gmc@euler.math.pitt.edu).

any edge colored graph can be expressed as a sum of formal determinants; cf. [2] and [3]. These results have been used in constructing edge partitions into multicolored spanning tress for complete graphs on a small number of vertices. Algorithms for finding multicolored spanning trees are also discussed in [5]. An application of parallelisms of complete designs to population genetics data is found in [1]. Parallelisms are also useful in partitioning consecutive positive integers into sets of equal size with equal power sums; cf. [12]. Discussions of colored matchings and design parallelisms to parallel computing appear in [11].

2. Construction of multicolored edge partitions. Our main result is the following.

THEOREM.

(a) *For $p(>2)$ prime there exists a proper edge coloring of K_p that admits a partition of edges into multicolored Hamiltonian cycles.*

(b) *For $n = 6$, or $n = 2^m$, $m \geq 3$, or $n = 5 \cdot 2^m$, $m \geq 1$, there exists a proper edge coloring of K_n that admits a partition of edges into isomorphic multicolored spanning trees.*

Proof. Part (a). Label the vertices of K_p by $1, 2, \ldots, p$ and treat them as elements of the finite field with p elements in any subsequent arithmetic. Label p is the identity element for addition. Produce a $p \times \frac{(p-1)}{2}$ array E, the (i,j)th entry of which is the edge $(i, i+j)$ of K_p. Using the fact that p is a prime, it is easy to check that the columns of E represent edge-disjoint Hamiltonian cycles that partition the edges of K_p. Denote also colors by $1, 2, \ldots, p$ and use the same arithmetic as above. Define an array C whose (i,j)th entry is color $i + (j-1)(\frac{p+1}{2})$. Color the edge in the (i,j)th entry of E with the color found in the (i,j)th entry of C. Fixing a column index it is easy to see that the Hamiltonian cycle in that column is multicolored. We thus conclude that all Hamiltonian cycles that appear as columns in E are multicolored. It remains to verify that the edge-coloring of K_p so obtained is proper. By the cyclic nature of the construction, it suffices to check that vertex 1 is adjacent with edges carrying distinct colors. Indeed, vertex 1 is adjacent with edges of all colors except color $\frac{p+1}{2}$.

Part (b). Starting with a multicolored tree parallelism for K_{2m}, it suffices to prove that we can obtain a multicolored tree parallelism for K_{4m}. To complete the proof we simply iterate the process. Take a copy of the multicolored tree parallelism for K_{2m} and call it L. Take another copy of the multicolored tree parallelism for K_{2m}, on a disjoint set of vertices from those of L but using the same set of colors, and call it R. The graph having $L \cup R$ as vertices, with edges connecting any vertex of L with any vertex of R, is called B. It is apparent that we have thus constructed a graph K_{4m} on the vertex set $L \cup R$. Edges of B are still to be colored. Color the edges of B in accordance with a pair of orthogonal Latin squares. For a definition and basic properties of orthogonal Latin squares the reader is referred to [9, p. 366]. It is well known that a pair of orthogonal Latin squares on n symbols exists for all $n \neq 2, 6$; see [6]. The rows of the Latin squares are indexed by the vertices of L, and the columns by the vertices of R. Colors used are disjoint from those used on the edges of L and R. Entries in the first Latin square represent the assignment of colors to the edges. We have thus completed an edge coloring of K_{4m}. It is a proper coloring, since it is proper within L and R by assumption, and the distribution of colors in accordance with the entries of the first Latin square ensures that edges emanating from each vertex carry all possible colors. We now describe the spanning tree decomposition that produces a multicolored tree parallelism for K_{4m}. In general, denote by $s(M)$

the set of multicolored spanning trees present in the multicolored tree parallelism of complete graph M. Let $B(i)$ be the set of edges of B associated with positions in which symbol i occurs in the second of the orthogonal Latin squares; $1 \leq i \leq 2m$. Consider any bijection α between the set of symbols in the second Latin square and the set $s(L) \cup s(R)$. The set $s(L \cup R)$ is now described as follows:

$$s(L \cup R) = \{B(i) \cup T_{\alpha(i)} : 1 \leq i \leq 2m\}.$$

Elements of $s(L \cup R)$ are spanning trees of $L \cup R$. Any one of them consists of a spanning tree of L (or R) appended with a set of pendant edges $B(i)$ for some i. They are therefore isomorphic as uncolored trees. By construction it is evident that they are multicolored. It follows that they are isomorphic multicolored spanning trees. Moreover, they are edge disjoint. The only possible overlap may occur among the edges in B. But the orthogonality of the Latin squares ensures that an edge occurs in precisely one such spanning tree.

To complete the proof we exhibit multicolored tree parallelisms K_6, K_8, and K_{10}. Rows represent colors and columns represent isomorphic multicolored spanning trees. In the case of K_6 we have

$$\begin{array}{ccc} 12 & 35 & 46 \\ 23 & 16 & 45 \\ 34 & 15 & 26 \\ 56 & 24 & 13 \\ 36 & 25 & 14 \end{array}.$$

A partition for K_8 is

$$\begin{array}{cccc} 18 & 34 & 56 & 27 \\ 17 & 36 & 28 & 45 \\ 16 & 38 & 47 & 25 \\ 26 & 15 & 48 & 37 \\ 23 & 14 & 58 & 67 \\ 46 & 78 & 35 & 12 \\ 57 & 24 & 13 & 68 \end{array}.$$

A proper coloring of K_{10} appears below:

$$\begin{array}{ccccc} 12 & 34 & 90 & 56 & 78 \\ 24 & 13 & 69 & 57 & 80 \\ 60 & 58 & 14 & 79 & 23 \\ 37 & 89 & 15 & 40 & 26 \\ 49 & 25 & 70 & 38 & 16 \\ 50 & 46 & 28 & 17 & 39 \\ 67 & 30 & 18 & 29 & 45 \\ 68 & 19 & 47 & 20 & 35 \\ 59 & 27 & 36 & 48 & 10 \end{array}.$$

3. Study of small complete graphs and conjectures. For complete graphs with a small number of vertices parallelisms of isomorphic multicolored spanning trees (or unicyclic subgraphs) exist for *any* proper edge-coloring. An outcome of this study, interesting to us, is that there is a sole isomorphism type of spanning tree that affords such a partition in the case of $n = 6$, all six nonequivalent proper colorings for

$n = 8$, and many nonequivalent proper colorings on 10 vertices. For 6 and 8 vertices it reduces to the Dynkin diagram associated with the special root systems E_6 and E_8; see [10, p. 326]. In general, the spanning tree in question is obtained from the Dynkin diagram of E_6 by appending a path of an appropriate length to exactly one vertex of degree one that is joined to a vertex of degree two. When m vertices are involved, we call this type of spanning tree E_m. We believe that an edge-partition of K_{2n} into multicolored spanning trees of type E_{2n} is possible for any proper coloring of K_{2n} $(n > 2)$. For an odd number of vertices it seems likely that a partition into multicolored Hamiltonian cycles is possible for any proper coloring of such complete graphs. Considering the very large number of nonequivalent colorings of K_n for large n, we prefer to conjecture as follows.

CONJECTURE. (a) *Any proper coloring of the edges of a complete graph on an odd number of vertices allows a partition of the edges into multicolored isomorphic unicyclic subgraphs.*

(b) *Any proper coloring of the edges of a complete graph on an even number of (more than four) vertices allows a partition of the edges into multicolored isomorphic spanning trees.*

REFERENCES

[1] D. BANKS, G. CONSTANTINE, A. MERRIWETHER, AND R. LAFRANCE, *Nonparametric inference on mtDNA mismatches*, J. Nonparametre. Statist., 11 (1999), pp. 215–232.

[2] R. B. BAPAT, *Mixed discriminants and spanning trees*, Sankhyā, Ser. A, 54 (1992), pp. 49–55.

[3] R. B. BAPAT AND G. M. CONSTANTINE, *An enumerating function for spanning forests with color restrictions*, Linear Algebra Appl., 173 (1992), pp. 231–237.

[4] C. BERGE, *Principles of Combinatorics*, Academic Press, New York, 1971.

[5] H. BROERSMA AND X. LI, *Spanning trees with many of few colors in edge-colored graphs*, Discuss., Math., Graph Theory, 17 (1997), pp. 259–269.

[6] R. BRUALDI, *Introductory Combinatorics*, Prentice Hall, Upper Saddle River, NJ, 1999.

[7] M. BULIGA, *Colored Spanning Trees*, Doctoral dissertation, Department of Mathematics, University of Pittsburgh, Pittsburgh, PA, 2002.

[8] P. J. CAMERON, *Parallelisms of Complete Designs*, London Math. Soc. Lecture Notes Ser. 23, Cambridge University Press, Cambridge, UK, 1976.

[9] G. M. CONSTANTINE, *Combinatorial Theory and Statistical Design*, John Wiley, New York, 1987.

[10] W. FULTON AND J. HARRIS, *Representation Theory*, Graduate Texts in Math., 129, Springer-Verlag, New York, 1991.

[11] F. HARARY, *Parallel concepts in graph theory*, Math. Comput. Modeling, 18 (1993), pp. 101–105.

[12] M. JACROUX, *On the construction of sets of integers with equal power sums*, J. Number Theory, 52 (1995), pp. 35–42.

SIAM J. DISCRETE MATH.
Vol. 18, No. 3, pp. 581–591

DETACHMENT OF VERTICES OF GRAPHS PRESERVING EDGE-CONNECTIVITY*

BALÁZS FLEINER†

Abstract. The detachment of vertex is the inverse operation of merging vertices $s_1, \dots, s_t$ into s. We speak about $\{d_1, \dots, d_t\}$-detachment if, for the detached graph G', the new degrees are specified as $d_{G'}(s_1) = d_1, \dots, d_{G'}(s_t) = d_t$. We call a detachment k-feasible if $d_{G'}(X) \geq k$ whenever X separates two vertices of $V(G) - s$. In our main theorem, we give a necessary and sufficient condition for the existence of a k-feasible $\{d_1, \dots, d_t\}$-detachment of vertex s. This theorem also holds for graphs containing 3-vertex hyperedges disjoint from s. From special cases of the theorem, we get a characterization of those graphs whose edge-connectivity can be augmented to k by adding γ edges and p 3-vertex hyperedges. We give a new proof for the theorem of Nash-Williams that characterizes the existence of a simultaneous detachment of the vertices of a given graph such that the resulting graph is k-edge-connected.

Key words. detachment, graph, edge-connectivity

AMS subject classification. 05C40

DOI. 10.1137/S0895480198341511

1. Introduction.

Throughout this paper, by "a graph," we mean an undirected, not necessarily simple graph (loops and multiple edges are allowed).

We denote the set of vertices and edges of hypergraph G by $V(G)$ and $E(G)$, respectively. The notation "$\subset$" means proper inclusion, and instead of $X \cup \{s\}$ or $X \setminus \{s\}$ we simply write $X + s$ or $X - s$. $\Gamma(s)$ denotes the set of neighbors of s, where each neighbor v is represented with multiplicity $d(s, v)$. Thus when we speak about some neighbors of s we do not necessarily mean that they are different. For $X \subset V(G)$, let $G[X]$ be the graph induced by X, and define $G - s := G[V(G) - s]$ ($s \in V$). $X \subseteq V(G)$ *covers* vertex v if $v \in X$ and *separates* vertices a and b if it covers exactly one of them. A *cut* of G is a proper subset X of V, and $d : 2^{V(G)} \to \mathbb{N}$ is the degree function, where $d(X)$ is the number of edges whose endvertices are separated by X:

$$\lambda(a, b) := \min\{d(X) \ : \ X \text{ separates } a \text{ and } b\},$$

$$\lambda(G) := \min\{d(X) \ : \ \emptyset \neq X \subset V(G)\}.$$

A cut X with $d(X) = \lambda(G)$ is a *mincut* of G.

Two cuts X and Y on the ground set V are *intersecting* if none of the sets $X - Y$, $Y - X$, and $X \cap Y$ are empty. If, in addition, $V - (X \cup Y) \neq \emptyset$, then they are called *crossing*.

DEFINITION 1.1. *Suppose that $d_1, \dots, d_t$ are positive integers, and that there is no loop incident to s and $d(s) = \sum_{i=1}^{t} d_i$. A $\{d_1, \dots, d_t\}$-detachment of $s \in V(G)$ is*

*Received by the editors July 6, 1998; accepted for publication (in revised form) December 17, 2003; published electronically February 25, 2005. This work was supported by Hungarian National Foundation for Scientific Research grant OTKA T029772.
http://www.siam.org/journals/sidma/18-3/34151.html

†Computer and Automation Institute, Hungarian Academy of Sciences, Lágymányosi u. 11., 1111 Budapest, Hungary. This research was done as part of the author's Ph.D. studies at Department of Operations Research, Eötvös Loránd University, Pázmány Péter sétány 1/C, 1117 Budapest, Hungary (fleinerb@cs.elte.hu).

the following operation: we delete vertex s and the edges incident to s, and connect new vertices $s_1, \dots, s_t$ to the neighbors of s by new edges such that, for the resulted graph G', $d_{G'}(s_i) = d_i$ and $d_{G'}(z) = d_G(z)$ if $z \in \Gamma(s)$. In other words, a detachment is the inverse operation of merging vertices s_i with degree d_i into vertex s.

Obviously, there is a natural one-to-one correspondence between the edges of G and of G'. We call $\{d_1, \dots, d_t\}$ the *degree specification* of the detachment. If there is a $k \in \mathbb{N}$ such that $\lambda_{G'}(x, y) \geq k$ for $x, y \in V(G) - s$, then the detachment is *k-feasible*. A graph H is the detachment of G if H can be obtained by detaching certain vertices of G, one after the other. If $g : V(G) \to \mathbb{N}$ is a function and we detach each vertex v into $g(v)$ vertices, then we obtain a *g-detachment* of G.

Lovász's edge-splitting theorem [3] asserts that a k-feasible $\{2, d(s) - 2\}$ detachment exists whenever $d(s)$ is even, $k \geq 2$, and g is k-edge-connected in $V - s$. Using this result, Frank [1] gave a short proof for the theorem of Watanabe and Nakamura that characterizes those graphs that may become k-edge-connected after adding a certain number of new edges.

If an arbitrary degree specification is imposed on s, then making a k-feasible detachment at s is not always possible. In section 3, our main theorem characterizes the existence of a k-feasible detachment for a given degree specification, generalizing the above-mentioned theorem of Lovász. It turns out that the general case can be reduced to the case where $\{d_1, \dots, d_{t-1}, d_t\} = \{3, \dots, 3, d(s) - 3(t-1)\}$.

In the proof of our main theorem, we have to introduce hyperedges. That is why our theorem is about 2-3-*graphs*, i.e., hypergraphs with edges of size two or three. It is straightforward to generalize the theoretic definitions of the graph above to 2-3-graphs. By definition, a 3-edge contributes to the degree of a cut X by 1 if two of its vertices are separated by X. In the definition of the detachment we demand that no 3-edge is incident to the vertex s.

In section 4, we apply our main theorem to generalize the theorem of Watanabe and Nakamura. We also give a new proof for the theorem of Nash-Williams [5] that characterizes those graphs for which the k-edge-connectivity property can be preserved by a g-detachment for a given g.

Before stating our main theorem, we summarize some properties of the degree function of 2-3-graphs.

2. The degree function. The degree function of a 2-3-graph is symmetric and submodular, i.e.,

$$d(X) = d(V(G) - X) \text{ and}$$

$$d(X) + d(Y) \geq d(X \cap Y) + d(X \cup Y) \quad (\text{for all } X, Y \subset V(G)).$$

There is also a useful inequality for three sets for the degree function of a 2-3-graph:

$$\begin{aligned} d(X) + d(Y) + d(Z) \geq d(X \cap Y \cap Z) &+ d(X - (Y \cup Z)) \\ &+ d(Y - (Z \cup X)) \\ &+ d(Z - (X \cup Y)) \\ &+ 2\overline{d}(X, Y, Z), \end{aligned} \tag{2.1}$$

where $\overline{d}(X, Y, Z)$ denotes the number of edges (of size two) between $X \cap Y \cap Z$ and $V - (X \cup Y \cup Z)$. The following inequality is for two sets:

$$d(X) + d(Y) \geq d(X - Y) + d(Y - X) + 2\overline{d}(X, Y) \quad (X, Y \subseteq V(G)). \tag{2.2}$$

Here, $\overline{d}(X,Y)$ stands for the number of 2-edges connecting $X \cap Y$ to $V - (X \cup Y)$. These well-known inequalities can be checked by enumerating the contribution of the different types of edges and 3-edges to the left and right sides of the inequality. The following lemma shows that 3-edges and vertices of degree 3 are interchangeable.

LEMMA 2.1. *Let s be a vertex of degree 3 of a 2-3-graph G and a, b, and c are the three neighbors of s. Let G' be the 2-3-graph obtained from G by deleting vertex s together with edges as, bs, cs and by adding the 3-edge abc. Then*

$$\lambda_G(x,y) = \lambda_{G'}(x,y)$$

for every $x, y \in G - s$.

Proof. It is enough to prove that for any $x-y$-cut C in G or in G', there is an $x-y$-cut C' such that the degree of C' is not more than the degree of C in the other 2-3-graph. By symmetry, we may assume that $|C \cap \{a,b,c\}| \leq 1$. It is easy to check that $C' := C - s$ suffices. □

3. The main theorem. The main result of this paper is the following.

THEOREM 3.1. *Given a 2-3-graph $G = (V,E)$ with a specified vertex $s \in V$ and a degree specification $\{d_1, \ldots, d_t\}$, $(d_i \geq 2,\ \sum_{i=1}^{t} d_i = d(s))$ for s. Assume $\lambda_G(x,y) \geq k \geq 2$ for every pair of vertices $x, y \in V - s$ and that there is no loop or 3-edge incident to s. Then there exists a k-feasible $\{d_1, \ldots, d_t\}$-detachment of s if and only if*

$$\lambda(G-s) \geq k - \sum_{i=1}^{t} \left\lfloor \frac{d_i}{2} \right\rfloor. \tag{3.1}$$

Proof. We prove the necessity first. Assume that G' is obtained by a k-feasible detachment from G and that the vertex s is split into vertices s_i $(1 \leq i \leq t)$ with $d(s_i) = d_i$. Take an arbitrary cut X of $G - s$ and define the set S by $S := \{s_i : d_{G'}(X, s_i) > \frac{d_i}{2}\}$. Now

$$k \leq d_{G'}(X \cup S) \leq d_{G-s}(X) + \sum_{i=1}^{t} \left\lfloor \frac{d_i}{2} \right\rfloor.$$

Hence $\lambda(G-s) \geq k - \sum_{i=1}^{t} \left\lfloor \frac{d_i}{2} \right\rfloor$.

We prove the sufficiency in case of certain degree specifications. Then we deduce the general theorem.

Case 1. $d(s) \geq 4$, and the degree specification is $\{2, d(s)-2\}$.

In this case, condition (3.1) is the consequence of the k-edge-connectivity assumption, hence it holds automatically. We apply the following theorem of Mader from [4].

THEOREM 3.2. *Given a graph $G = (V,E)$ and a specified vertex $s \in V$ with $d(s) \neq 3$, no cutting edge is incident to s. Then there exists a $\{2, d(s)-2\}$-detachment at s such that for the resulted graph G'*

$$\lambda_G(x,y) = \lambda_{G'}(x,y)$$

for every $x, y \in V - s$.

By Lemma 2.1, we can exchange each 3-edge $e = \alpha\beta\gamma$ into new edges $s_e\alpha$, $s_e\beta$, and $s_e\gamma$, where s_e is a new vertex. By Lemma 2.1, this does not change $\lambda(a,b)$

$(a, b \in V(G))$. Now take the $\{2, d(s)-2\}$-detachment provided by Theorem 3.2. Change the 3-stars back into 3-edges and we are done.

Let us point out that Case 1 contains Lovász's edge-splitting theorem in [3]. In the appendix, there is a self-contained proof for Case 1.

It is easy to check that we do not change the connectivity between vertices of $V - s$ if, for vertex s_1 of degree 2, we replace edges s_1u and s_1v by edge uv. Then we can apply Case 1 again on s_2. Iterating this justifies the following.

Case 2. $d(s) \geq 2q+2$, and the degree specification is $\{2, \ldots, 2, d(s)-2q\}$, where s is split into $q+1$ new vertices.

Case 3. $d(s) \geq 3p+2$, and the degree specification is $\{3, \ldots, 3, d(s)-3p\}$, where $p+1$ is the number of the new vertices.

We may assume that $d(s) \geq 6$ since $d(s) = 5$ is covered by Case 1. We use induction on p. If $p = 0$, then there is nothing to prove. Assume that for $(p_0 - 1)$ the theorem is proved. We verify it for p_0. For brevity, let $\Delta := \lfloor \frac{d(s)-p_0}{2} \rfloor$.

By the induction hypothesis, $\lambda(G - s) \geq k - \Delta$ holds. If there is a k-feasible $\{3, \ldots, 3, d(s) - 3p_0\}$-detachment, then we can carry out the detachment by making two successive detachments. The first one is a k-feasible $\{3, d(s)-3\}$-detachment of s. Next we apply the induction hypothesis for the resulting 2-3-graph and for the new vertex s_2 of degree $d(s) - 3$.

Let us change the vertex s_1 and the three edges incident to it to the 3-edge defined by $\Gamma(s_1)$. By Lemma 2.1, the k-edge-connectivity between the vertices of $G - s$ is maintained after this operation.

Now we make a k-feasible $\{3, \ldots, 3, d(s) - 3p_0\}$-detachment of s_2 in the resulting 2-3-graph G' into p_0 vertices. Changing back the previously introduced 3-edge, the obtained detachment is k-feasible by Lemma 2.1.

By the induction hypothesis, the latter k-feasible detachment exists if and only if

$$\lambda(G' - s_2) \geq k - \left\lfloor \frac{(d(s)-3) - (p_0 - 1)}{2} \right\rfloor = k - \Delta + 1. \tag{3.2}$$

Let us suppose that $\lambda(G - s) = k - \Delta$. Then the first detachment must satisfy two conditions:

- it is k-feasible and
- the addition of the 3-edge induced by the three neighbors of s_1 increases the edge-connectivity of $G - s$.

From now on, any three neighbors of s will be referred to as a *triad*. The *vertices of the triad* are the corresponding neighbors of s.

Consider the following family of inclusionwise minimal mincuts of $G - s$:

$$\mathcal{B} = \{B : B \subset V - s,\ d_{G-s}(B) = k - \Delta,\ \nexists A \subset B :\ d_{G-s}(A) = k - \Delta\}.$$

PROPOSITION 3.3. *The elements of $\mathcal{B}$ are disjoint and $|\mathcal{B}| = 2$ or $|\mathcal{B}| = 3$.*

Proof. The disjointness of the elements of $\mathcal{B}$ and $|\mathcal{B}| \geq 2$ follows from submodularity. From the disjointness we get $d(s) \geq \sum_{B_i \in \mathcal{B}} d(B_i, s) \geq |\mathcal{B}|\Delta$, and by

$$4\Delta \geq 4\left(\frac{d(s) - p_0}{2}\right) - 2 = d(s) + (d(s) - 2p_0 - 2) \geq d(s) + p_0 > d(s)$$

we conclude that $|\mathcal{B}| \leq 3$. □

Obviously, the $\{3, d(s)-3\}$-detachment satisfies (3.2) if and only if the 3-edge induced by the neighbors of the new vertex s_2 contributes to the degree of each mincut

X of $G-s$. In other words, this means that both X and $\overline{X}$ contain some neighbor of s_2. Since both a mincut X and its complement contain an element of $\mathcal{B}$, the important triads are those that have a vertex in every element of $\mathcal{B}$. We say that such triads are *transversal*.

If $\lambda(G-s) > k-\Delta$, then $|\mathcal{B}| = 0$, and thus there exists no transversal triad. For the sake of the unified approach, we choose two arbitrary disjoint sets B_1 and B_2 such that $d(B_i, s) \geq 1$ $(i = 1, 2)$ and $d(B_1 \cup B_2, s) \geq 2\Delta$, and define $\mathcal{B} := \{B_1, B_2\}$. This can always be done unless $|\Gamma(s)| = 1$. But then $\lambda(G-s) \geq k$ holds trivially, and any detachment with the given degree specification is k-feasible.

To finish the proof, we have to find a transversal triad that induces a k-feasible detachment. First, we study k-feasible $\{3, d(s)-3\}$-detachments.

LEMMA 3.4. *Suppose that the $\{3, d(s)-3\}$-detachment induced by triad T is not feasible. Then either*

- *at least two vertices of T are covered by a set $Y \subset V-s$ with $d_G(Y) = k$ or*
- *all three vertices of T are covered by a set $Y \subset V-s$ with $k+1 \leq d_G(Y) \leq k+2$.*

Proof. By infeasibility, there is a cut X of G' separating two vertices of $V-s$ with $d_{G'}(X) < k$. By taking the complement if necessary, we can assume that $s_1 \in X$. Then $s_2 \notin X$; otherwise $d_G(X - s_1 - s_2 + s) = d_{G'}(X) < k$ is a contradiction. If no vertex of T is in $X - s_1$, then it is easy to see that $d_{G'}(X) = d_{G'}(X - s_1) + 3$, so $k > d_{G'}(X) = d_{G'}(X - s_1) + 3 = d_G(X - s_1) + 3 \geq k + 3$, a contradiction. Similarly, if $|(X - s_1) \cap T| = 1$, then $k > d_{G'}(X) = d_{G'}(X - s_1) + 1 = d_G(X - s_1) + 1 \geq k + 1$.

If $X - s_1$ contains exactly two neighbors of s_1 in $X - s_1$, then, from $k > d_{G'}(X) = d_{G'}(X - s_1) - 1 = d_G(X - s_1) - 1 \geq k - 1$, we get $k = d_G(X - s_1)$.

The remaining case is that all the 3 neighbors are in $X - s_1$. It means that $k > d_{G'}(X) = d_{G'}(X - s_1) - 3 = d_G(X - s_1) - 3 \geq k - 3$, thus $k + 2 \geq d_G(X - s_1) \geq k$. ∎

Define

$$\mathcal{K} = \{X : \ \ X \subset V - s, \ \ d_G(X) = k\}.$$

We call a triad *legal* if no pair of its vertices are covered by a member of $\mathcal{K}$. If no k-feasible detachment exists, then it follows from Lemma 3.4 that each transversal legal triad is covered by a set L with $k + 2 \geq d_G(L) \geq k + 1$.

In what follows, we focus on transversal legal triads. Let $\mathcal{L}$ be a family of different sets on the ground-set $V - s$ such that, for every transversal legal triad T, there is a set X of $\mathcal{L}$ with $T \subseteq X$ and $k + 1 \leq d_G(X) \leq k + 2$. Choose $\mathcal{L}$ so that $|\mathcal{L}|$ is minimal.

Remark. At this point it is not obvious that a transversal legal triad exists but this will follow from the proof.

LEMMA 3.5. *If $L \in \mathcal{L}$ and $K \in \mathcal{K}$ and $d(L \cap K, s) \geq 1$, then $K \subset L$.*

Proof. Obviously, $L \not\subseteq K$ from the definition of $\mathcal{L}$. If $K \not\subset L$, then from inequality (2.2) we get

$$(k + 2) + k \geq d_G(L) + d_G(K) \geq d_G(L - K) + d_G(K - L) + 2d(K \cap L, s) \geq k + k + 2.$$

Thus $d_G(K) = d_G(L - K) = k$. This contradicts the legality of the transversal triad inside L because two of the vertices of this triad must be covered by K or by $L - K$. ∎

If $|\mathcal{B}| = 3$, then

$$d(s) \geq \sum_{B_i \in \mathcal{B}} d(B_i, s) \geq 3 \left\lfloor \frac{d(s) - p_0}{2} \right\rfloor \geq 3 \left(\frac{d(s) - p_0}{2} \right) - \frac{3}{2} \geq d(s) - \frac{1}{2},$$

and since $\lfloor \frac{d(s)-p_0}{2} \rfloor$ is integer, we get $d(B_i, s) = \Delta$. Therefore $d_G(B_i) = d_{G-s}(B_i) + d(B_i, s) = (k-\Delta)+\Delta = k$, and thus $B_i \in \mathcal{K}$ $(\forall B_i \in \mathcal{B})$. It follows from submodularity that the maximal elements of $\mathcal{K}$ are disjoint. Each of them contains at most Δ neighbors of s, therefore the elements of $\mathcal{B}$ are contained in different maximal elements of $\mathcal{K}$. Thus every transversal triad is legal. By Lemma 3.5, there is a set $X \in \mathcal{L}$ that contains all the elements of $\mathcal{B}$, i.e., $d(X, s) \geq \sum_{i=1}^{3} d(B_i, s) = 3\Delta > \Delta + 2$, a contradiction.

The remaining case is $\mathcal{B} = \{B_1, B_2\}$.

LEMMA 3.6. *If $X, Y \in \mathcal{L}$, then $d(X \cap Y, s) \leq 2$.*

Proof. From $d_G(X) \leq k + 2$ and $d_G(Y) \leq k + 2$, inequality (2.2) gives that $2 \geq \overline{d}(X, Y) \geq d(X \cap Y, s)$. (Here we used that $X \not\subset Y$ since $|\mathcal{L}|$ is minimal.) □

We shall construct legal transversal triads by choosing the corresponding vertices one by one. Let $a \in B_1$ be a neighbor of s. Since the maximal elements of $\mathcal{K}$ are disjoint and each of them contains at most Δ neighbors of s, there must exist a neighbor $b \in B_2$ of s outside the element of $\mathcal{K}$ that might cover vertex a. Two maximal elements of $\mathcal{K}$ can contain at most 2Δ neighbors of s. Hence there is a vertex c such that $\{a, b, c\}$ is a legal triad and is also transversal due to a and b.

If we cannot choose c from $B_1 \cup B_2$, then there exist two sets $a \in K_1 \in \mathcal{K}$ and $b \in K_2 \in \mathcal{K}$ such that $(B_1 \cup B_2) \cap \Gamma(s) \subseteq K_1 \cup K_2$. But then, set X of $\mathcal{L}$ covering legal transversal triad $\{a, b, c\}$ contains the sets K_1 and K_2 by Lemma 3.5. Thus there are too many edges from s to X because $d(X, s) \geq d(K_1 \cup K_2 \cup \{c\}, s) \geq d(B_1 \cup B_2 \cup \{c\}, s) \geq 2\Delta + 1 > \Delta + 2$, a contradiction.

By interchanging the notation, if necessary, we may assume that $a \in B_1$ and $b, c \in B_2$. Since $d(X, s) \leq \Delta + 2$ and $d(s) - (\Delta + 2) \geq 2$, there exist $d, e \in \Gamma(s)$ such that $d, e \notin X$.

We claim that $d \neq e$. Otherwise $d(d, s) \geq 2$. Moreover, $\{a, b, d\}$ and $\{a, c, d\}$ are legal transversal triads by the usual argument. Consider the sets of $\mathcal{L}$ that cover them and the set X. By Lemma 3.6, $\{a, b, d\}$ and $\{a, c, d\}$ have no common covering set. So we have two sets Y_1 and Y_2 with $d(Y_1 \cap Y_2, s) \geq d(a, s) + d(d, s) \geq 1 + 2 = 3$. This contradicts Lemma 3.6.

Let us choose the members X, Y, and Z of $\mathcal{L}$ that correspond to legal transversal triads $\{a, b, c\}$, $\{a, b, d\}$, and $\{a, b, e\}$, respectively.

(If these three sets are not different, say, if $Y = Z$, then instead of X, Y, and Z we choose members X, U, W of $\mathcal{L}$ that correspond to legal transversal triads $\{a, b, c\}$, $\{a, c, d\}$, and $\{a, c, e\}$, respectively. Obviously, $X \neq U$ and $X \neq W$. Moreover, $U \neq W$, since $U = W$ would contradict Lemma 3.6 by $d(U \cap Y, s) \geq 3$, since $\{a, d, e\} \subseteq U \cap Y$.)

It is clear from Lemma 3.6 that $c \in X - (Y \cup Z)$, $d \in Y - (Z \cup X)$, and $e \in Z - (X \cup Y)$. $d_G(X \cap Y \cap Z) \neq k$ since $\{a, b, c\}$ is legal.

From (2.1) we get

$$\begin{aligned}
&(k+2) + (k+2) + (k+2) \geq d_G(X) + d_G(Y) + d_G(Z) \\
&\geq d_G(\overbrace{X \cap Y \cap Z}^{a,b\in}) + d_G(\overbrace{X - (Y \cup Z)}^{c\in}) + d_G(\overbrace{Y - (Z \cup X)}^{d\in}) + d_G(\overbrace{Z - (X \cup Y)}^{e\in}) \\
&+ 2\overline{d}(X, Y, Z) \geq (k+1) + k + k + k + 4
\end{aligned}$$

since $a, b \in X \cap Y \cap Z$ and $s \in V - (X \cup Y \cup Z)$. From this it follows that $k \leq 1$.

The general case. The degree of specification is $\{d_1, \ldots, d_t\}$, and d_i is odd for $1 \leq i \leq p$ and d_i is even for $p + 1 \leq i \leq t$.

The condition is $\lambda(G-s) \geq k - \lfloor \frac{d(s)-p}{2} \rfloor$. By Case 3, there exists a k-feasible $\{3,\ldots,3,d(s)-3p\}$-detachment of s into $p+1$ vertices. Change the new vertices of degree 3 into 3-edges. Perform a $\{2,\ldots,2\}$-detachment of the new vertex of degree $d(s)-3p$ and change the 3-edges back to 3-stars. By this, we get a k-feasible $\{2,\ldots,2,3,\ldots,3\}$-detachment of the original 2-3-graph, where the number of new vertices of degree 3 is p. Merge at most one vertex of degree 3 with some others of degree 2 to get a vertex of degree d_1. By repeating this operation, we construct a k-feasible $\{d_1,\ldots,d_t\}$-detachment of the original 2-3-graph. □

One may ask for a necessary and sufficient condition for the existence of a detachment which preserves also the local edge-connectivities.

CONJECTURE 3.7. *Given a graph $G=(V,E)$ with a specified vertex $s \in V$ and a degree specification $\{d_1,\ldots,d_t\}$ $(d_i \geq 2,\ \sum_{i=1}^{t} d_i = d(s))$ for s. Assume there is no loop or cut-edge incident to s. Then there exists a $\{d_1,\ldots,d_t\}$-detachment of s such that $\lambda_G(x,y) = \lambda_{G'}(x,y)$ $(x,y \in V-s)$ if and only if*

$$\lambda_{G-s}(x,y) \geq \lambda_G(x,y) - \sum_i \left\lfloor \frac{d_i}{2} \right\rfloor \quad (x,y \in V-s).$$

Remark. Conjecture 3.7 is true. Motivated by this paper, a generalized form of this conjecture was proved by Jordán and Szigeti [2].

4. Applications. In this section, we apply Theorem 3.1 to deduce some well-known theorems whose standard proofs are based on Lovász's edge-splitting theorem [3]. The equivalent form of Lovász's theorem follows.

THEOREM 4.1. *$G=(V,E)$ is a given multigraph, $k \geq 2$, and $m: V \to \mathbb{N}$ is a function. There is a graph $H=(V,F)$ such that $d_H(v) = m(v)(\forall v \in V)$ and $G+H=(V,E \cup F)$ is k-edge-connected if and only if*

(i) *$m(V)$ is even,*

(ii) *$m(X) \geq k - d_G(X)$ for any proper subset X of V, where $m(X) := \sum_{x \in X} m(x)$.*

A generalization of Theorem 4.1 is the following.

THEOREM 4.2. *$G=(V,E)$ is 2-3-graph, $k \geq 2$, and $m: V \to \mathbb{N}$ is a function. There is a 2-3-graph $H=(V,F)$ such that $d_H(v) = m(v)(\forall v \in V)$, F contains exactly p 3-edges, and $G+H=(V,E \cup F)$ is k-edge-connected if and only if*

(i) *$3p \leq m(V)$,*

(ii) *$m(V)-3p$ is even,*

(iii) *$m(X) \geq k - d_G(X)$ for any nonempty proper subset of V,*

(iv) *$\lambda(G) \geq k - \lfloor \frac{m(V)-p}{2} \rfloor$.*

Proof. Conditions (i) and (ii) are necessary, because the total degree requirement of the 3-edges is not more than $m(V)$ and the requirement of the edges (of size 2) is even. (iii) is also needed since the edges of H increase the degree of every set to k. Condition (iv) is equivalent to the inequality $|F| \geq k - \lambda(G)$.

To prove the sufficiency we add to the 2-3-graph an extra vertex s $(s \notin V)$ and $m(v)$ new edges between s and v for every vertex $v \in V$. By (iii) and (iv), Theorem 3.1 can be applied; i.e., there is a k-feasible $\{2,\ldots,2,3,\ldots,3\}$-detachment of s such that the number of new vertices of degree 3 is exactly p. Now the neighbors of each s_i are the endvertices of an edge of H. □

Watanabe and Nakamura [6] gave a characterization of the graphs that can be made k-edge-connected by adding γ edges. We prove the extension of this result along the lines of Frank's proof in [1].

A family of sets $\{X_1,\ldots,X_r\}$ is a *subpartition* of V if $\emptyset \neq X_i \subset V$ $(1 \leq i \leq r)$ and $X_i \cap X_j = \emptyset$ $(i \neq j)$.

THEOREM 4.3. *The 2-3-graph $G = (V, E)$ can be made k-edge-connected by adding γ edges and p 3-edges if and only if*

(i) $\lambda(G) \geq k - (\gamma + p)$ *and*

(ii) $2\gamma + 3p \geq \sum_i (k - d(X_i))$ *holds for every subpartition $\{X_1, \ldots, X_r\}$ of V.*

Proof. If the required augmentation exists, then conditions (i) and (ii) follow from the facts that the addition of an edge can increase the edge-connectivity by at most one and an edge (of size 2) or a 3-edge can contribute to the degree of at most 2 or 3 disjoint sets having degree less than k.

Let $m : V \to \mathbb{N}$ be a function such that $m(V)$ is minimal and $k - d(X) \leq m(X)$ for every set $\emptyset \neq X \subset V$.

LEMMA 4.4. $m(V) \leq 2\gamma + 3p$.

Proof. We call a set $\emptyset \neq X \subset V$ *critical* if $k - d(X) = m(X)$. If $m(v) > 0$, then v is in a critical set. Let Y_v be the minimal critical set containing v. We claim that the maximal elements of $\{Y_v \,:\, v \in V\}$ are disjoint. Indirectly, let Y_v and Y_u be two maximal sets intersecting each other. Then

$$\begin{aligned} m(Y_u) + m(Y_v) &= k - d(Y_u) + k - d(Y_v) \leq k - d(Y_u - Y_v) + k - d(Y_v - Y_u) \\ &\leq m(Y_u - Y_v) + m(Y_v - Y_u) = m(Y_u) + m(Y_v) - 2m(Y_u \cap Y_v) \\ &\leq m(Y_u) + m(Y_v). \end{aligned}$$

The above inequalities are equalities, hence $m(Y_u \cap Y_v) = 0$. Further, the sets $Y_u - Y_v$ and $Y_v - Y_u$ are critical and cannot be minimal covering sets of vertex u and v, respectively.

Let $\{X_i\}$ be the subpartition of the maximal members of $\{Y_v \,:\, v \in V\}$. From condition (ii), it follows that

$$m(V) = \sum_i m(X_i) = \sum_i (k - d(X_i)) \leq 2\gamma + 3p.$$

If $m(V) < 2\gamma + 3p$, then increase $m(v)$ at some vertex v so that $m(V)$ is equal to $2\gamma + 3p$. Now $\gamma + p = \lfloor \frac{m(V) - p}{2} \rfloor$, thus the conditions of Theorem 4.2 are satisfied. □

The next theorem is valid for graphs that may contain loops. If there are loops at vertex s, then each loop increases the degree of s by 2. We also demand that after the detachment at s, any edge that comes from a loop must connect two new vertices or must remain in the loop.

THEOREM 4.5 (see Nash-Williams [5]). *Let $G = (V, E)$ be a multigraph, $|V| \geq 2$, $k \geq 2$, $g : V \to \mathbb{N}$ be a function, and $\zeta_v = \{\zeta_v^1, \ldots, \zeta_v^{g(v)}\}$ be a degree specification for each $v \in V$. Then there is a k-edge-connected g-detachment of G if and only if*

(i) *G is k-edge-connected,*

(ii) *$d(v) \geq k \cdot g(v)$,*

(iii) *if k is odd, then none of the following conditions are true:*

(a) *there is a cut-vertex s in G such that $d(s) = 2k$, $g(s) = 2$,*

(b) *$|V| = 2$, $d(v) = 2k$, and $g(v) = 2$ $(\forall v \in V)$, and there is no loop in G.*

Moreover, there is a k-edge-connected g-detachment of G with degree specification ζ if, in addition to the previous conditions, (ii)′ *is satisfied as follows:*

(ii)′ $\zeta_v^i \geq k$ $\quad (v \in V,\ 1 \leq i \leq g(v))$.

Proof. If there is such a detachment, then (i) and (ii)–(ii)′ must hold, since a detachment does not increase the edge-connectivity and every vertex has degree at least k in a k-edge-connected graph. By Theorem 3.1, there is no k-edge-connected

g-detachment if (iii)(a) is true. If (iii)(b) holds, then, by detaching only one vertex, (iii)(a) holds for the resulting graph. Thus the detachment cannot be completed.

In order to prove the sufficiency, we use induction on the number of vertices v for which $g(v) \geq 2$. We detach the vertices one by one. Our purpose is to detach only one vertex, maintaining conditions (i)–(iii).

Case 1. k is odd, and there is a vertex $s \in V$ such that there is no loop at s and $d(s) = 2k$, $g(s) = 2$.

We show a $\{k, k\}$-detachment of s into vertices s_1, s_2 such that the resulting graph satisfies conditions (i)–(iii).

Perform a k-feasible $\{k, k\}$-detachment of s that exists by Theorem 3.1. This implies that any cut of the resulting graph G' has degree at least k if it separates two vertices of $V(G') - s_1 - s_2$. The other cuts or their complements are the subsets of $\{s_1, s_2\}$, and since $d(s_1) = d(s_2) = k$, they also have degree at least k, and thus condition (i) is satisfied.

Assume that the detachment of s creates a cut-vertex s^* for which $d(s^*) = 2k$ and $g(s^*) = 2$. Since G' is k-edge-connected, $G' - s^*$ has exactly two components induced by subsets A and B of V. By (iii), s^* was not a cut-vertex in G, therefore s_1 and s_2 are in different components. We may assume that $s_1 \in A$ and $s_2 \in B$. The subgraphs $G'[A + s^*]$ and $G'[B + s^*]$ are k-edge-connected because all of their cuts that do contain s^* have the same degree as in G'. By (iii)(b), we may assume that $|A| \geq 2$ and s_1 has a neighbor $a \in A$. (If both A and B consist of only one vertex, then, by condition (iii), there must exist a loop, for example, at s_1, but this contradicts the condition of Case 1.) By k-edge-connectivity, there are k edge disjoint paths in $G'[A + s^*]$ connecting s_1 and a. Since a path contains 0 or 2 edges incident to s^*, and $d_{G'[A+s^*]}(s^*) = k$ and k is odd, there are at least $k - \left\lfloor \frac{k}{2} \right\rfloor \geq 2$ paths that are disjoint from s^*. We modify the detachment. Delete edge $e = s_1a$ and another one, $f = s_2b$, and add new edges $e^* = s_2a$ and $f^* = s_1b$ (see Figure 4.1). We claim that the constructed graph G^*

(a) satisfies the degree specification at s_1 and s_2,

(b) is k-edge-connected, and

(c) does not contain a cut-vertex z such that $d(z) = 2k$ and $g(z) = 2$.

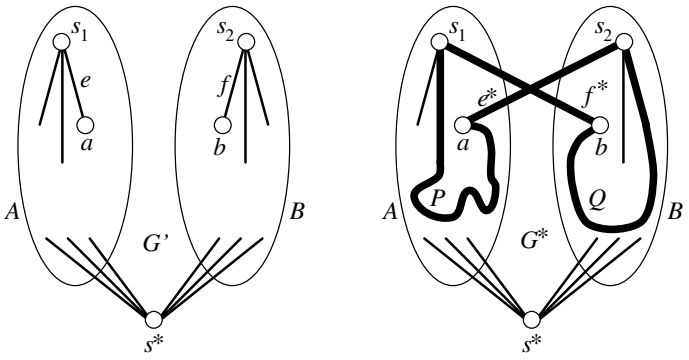


FIG. 4.1. *The graphs G' and G^*.*

Claim (a) follows directly from the construction.

We verify (b). By the above argument, there is a path P in $G^*[A]$ connecting s_1 and a. By the k-edge-connectivity of $G'[B + s^*]$, the subgraph $G'[B + s^*] - f = G^*[B + s^*]$ contains a path Q connecting s_2 and b that may contain vertex s^*.

Let X be an arbitrary cut of G^*. $A \cup B + s^* = V(G^*)$, thus X separates $A + s^*$ or $B + s^*$. Accordingly

$$\begin{aligned} d_{G^*}(X) &\geq d_{G^*[A+s^*]\cup(e^*+Q+f^*)}(X \cap V(G^*[A+s^*] \cup (e^*+Q+f^*))) \\ &\geq d_{G'[A+s^*]}(X \cap (A+s^*)) \geq k \end{aligned}$$

or the analogous inequality with B and P instead of A and Q holds, since $G'[A+s^*]$ and $G'[B+s^*]$ are k-edge-connected.

Suppose that there is a cut-vertex $z \in G^*$ such that $d_{G^*}(z) = 2k$ and $g(z) = 2$. The circle defined by the edges e^*, f^* and by the paths P and Q shows that s_1 and s_2 are in the same component of $G^* - z$. (Here we used the fact that $s^* \notin P$.) By merging s_1 and s_2 into one vertex, it turns out that z is a cut-vertex of G, and this is forbidden by (iii)(a).

Case 2. k is even, or no loopless vertex s exists with $d(s) = 2k$ and $g(s) = 2$.

Let z be a vertex with $g(z) \geq 2$. If no degree specification is imposed on z, then choose one that satisfies (ii)$'$.

Suppose that z has no loops. Since we are not discussing Case 1, $k - \sum_{i=1}^{g(z)} \lfloor \frac{\zeta_z^i}{2} \rfloor \leq 0$, and therefore there is a k-feasible $\{\zeta_z^1, \ldots, \zeta_z^{g(z)}\}$-detachment of z resulting in graph G' by Theorem 3.1. Then, for every cut X separating two vertices of $V(G') - z_1 - \cdots - z_{g(z)}$, inequality $d_{G'}(X) \geq k$ holds. The other cuts or their complements are the subsets of $\{z_1, \ldots, z_{g(z)}\}$, and thus, by (ii)$'$ and by the absence of loops, these cuts have degree at least k as well. Consequently G' satisfies condition (i). Conditions (ii) and (ii)$'$ hold trivially for the induced degree specification. Property (iii) is also valid since there is no vertex $s \in V(G)$ with $d(s) = 2k$ and $g(s) = 2$.

In the remaining cases we suppose that z has h loops.

If $2h \geq k$, then we subdivide each loop with a vertex and merge these vertices into new vertex q. The resulting graph G^q is k-edge-connected. (i) implies that $d(V - z, z) \geq k$. From this and from (ii)–(ii)$'$ and by a parity argument, we get that $k - \sum_{i=1}^{g(z)} \lfloor \frac{\zeta_z^i}{2} \rfloor \leq 0$. We make a k-feasible $\{\zeta_z^1, \ldots, \zeta_z^{g(z)}\}$-detachment of G^q at z, and after this, we perform a $\{2, \ldots, 2\}$-detachment at vertex q. This can be done because the original and the resulting graphs are clearly k-edge-connected, and we can apply Theorem 3.1 for them. Obviously, the final graph satisfies conditions (i)–(iii).

If $2h < k$, then delete the loops incident to z, and define new degree specification $\{\zeta_z^{*1}, \ldots, \zeta_z^{*g(z)}\}$ for the resulting graph G^*, with $\zeta_z^{*i} := \zeta_z^i - h$ for $i = 1, 2$ and $\zeta_z^{*i} := \zeta_z^i$ for $i > 2$. It is enough to construct a k-feasible detachment with respect to the new specification because connecting vertices z_1 and z_2 with h edges yields a k-feasible $\{\zeta_z^1, \ldots, \zeta_z^{g(z)}\}$-detachment at z.

If $g(z) \geq 4$, then $k - \sum_{i=1}^{g(z)} \lfloor \frac{\zeta_z^{*i}}{2} \rfloor \leq 0$ holds, and there is a k-feasible $\{\zeta_z^{*1}, \ldots, \zeta_z^{*g(z)}\}$-detachment of z in G^* by Theorem 3.1.

If $g(z) = 3$ and $k - \sum_{i=1}^{3} \lfloor \frac{\zeta_z^{*i}}{2} \rfloor \leq 0$, then we have a k-feasible $\{\zeta_z^{*1}, \zeta_z^{*2}, \zeta_z^{*3}\}$-detachment of G^* by Theorem 3.1. If $k - \sum_{i=1}^{3} \lfloor \frac{\zeta_z^{*i}}{2} \rfloor > 0$, then, from $d_{G^*}(V - z, z) \geq k$ and from (ii)–(ii)$'$, k is odd, $\zeta_z^{*1} + \zeta_z^{*2} = k + 1$, $\zeta_z^{*3} = k$, and $2h = k - 1$. Since ζ_z^{*3} is even, we can make a k-feasible $\{\zeta_z^{*1} + \zeta_z^{*2}, \zeta_z^{*3}\}$-detachment of G^* that reduces this case to $g(z) = 2$. If ζ_z^{*1} or ζ_z^{*2} is even, then Theorem 3.1 gives us the corresponding detachment. If both ζ_z^{*1} and ζ_z^{*2} are odd, then we add a loop to vertex z and divide the loop with vertex q. In the resulting graph, there is a k-feasible $\{2, \ldots, 2\}$-detachment of z by the successive use of Theorem 3.2. Merging some new vertices, we get a k-feasible $\{\zeta_z^{*1} + 1, \zeta_z^{*2} + 1\}$-detachment of z. Connecting z_1 and z_2 with the missing

$h-1$ edges, we obtain a k-feasible $\{\zeta_z^1, \zeta_z^2\}$-detachment of z. It follows from the constructions that the graphs obtained above satisfy conditions (i)–(iii). □

5. Appendix. We present a self-contained proof for Case 1 of Theorem 3.1.

Assume that every $\{2, d(s)-2\}$-detachment is infeasible. Hence, for every $x, y \in \Gamma(s)$, there is a set $x, y \in U \subset V - s$ with $d_G(U) \leq k+1$. For $x \in \Gamma(s)$, let $\mathcal{H}^x = \{U_1^x, \ldots, U_{l_x}^x\}$ be a family of sets with a minimum number of elements such that $\Gamma(s) \subseteq U_1^x \cup \cdots \cup U_{l_x}^x$, $x \in U_i^x$, and $d_G(U_i^x) \leq k+1$ $(1 \leq i \leq l_x)$.

Suppose that a is a neighbor of s with $l_a \geq 3$. Fix different sets X, Y, Z of $\mathcal{H}^a$. X cannot be omitted from $\mathcal{H}^a$, so there is a neighbor of s that is covered by X but not by Y or Z. Similarly we find that $Y - (Z \cup X) \neq \emptyset$ and $Z - (X \cup Y) \neq \emptyset$. We apply inequality (2.1) for 2-3-graph G:

$$\begin{aligned} d_G(X) + d_G(Y) + d_G(Z) \geq\ & d_G(X \cap Y \cap Z) + d_G(X - (Y \cup Z)) \\ & + d_G(Y - (Z \cup X)) + d_G(Z - (X \cup Y)) \\ & + 2\overline{d}(X, Y, Z). \end{aligned}$$

Each term on the left is at most $k+1$ and the first four terms on the right are at least k, and the last is at least 2 due to the edge sa. Hence, $k \leq 1$, a contradiction.

Hence, $l_x \leq 2$ for every $x \in \Gamma(s)$. Inequality

$$d_G(U_i^x) - 1 \leq k \leq d_G(U_i^x + s) = d_G(U_i^x) - 2d(U_i^x, s) + d(s)$$

implies that $d(U_i^x, s) \leq \lceil \frac{d(s)}{2} \rceil$. Thus, one U_i^x cannot contain all neighbors of s, that is, $l_x = 2$ for every $x \in \Gamma(s)$. Since $x \in U_1^x \cap U_2^x$, it follows that $d(s)$ is odd, $d(U_i^x, s) = \lceil \frac{d(s)}{2} \rceil$, $d_{G-s}(U_i^x) = k - \lfloor \frac{d(s)}{2} \rfloor$, $d_G(U_i^x) = k+1$, and $d(U_1^x \cap U_2^x, s) = 1$ $(\forall x \in \Gamma(s))$.

From these equalities and from inequality (2.2), we get that $d_G(U_1^x - U_2^x) = d_G(U_2^x - U_1^x) = k$. Consider an element X of $\mathcal{H}$ that covers a vertex $y \in \Gamma(s) \cup (U_1^x - U_2^x)$ and a vertex $z \in \Gamma(s) \cup (U_2^x - U_1^x)$. Since $d(X, s) = \lceil \frac{d(s)}{2} \rceil$, there exists a vertex $w \in \Gamma(s) \cap ((U_1^x - U_2^x) \cup (U_2^x - U_1^x) - X)$. We may assume that $w \in U_1^x - U_2^x$.

Apply inequality (2.2) to X (that has degree $k+1$) and to $U_1^x - U_2^x$ (that has degree k). Since edge sy connects $V - (X \cup (U_1^x - U_2^x))$ and $X \cap (U_1^x - U_2^x)$, we gain the contradiction $(k+1) + k \geq k + k + 2$.

Acknowledgment. I gratefully thank the anonymous referees for their helpful suggestions, comments, and observations, which enabled me to shorten some proofs and clarify the presentation.

REFERENCES

[1] A. Frank, *Augmenting graphs to meet edge-connectivity requirements*, SIAM J. Discrete Math., 5 (1992), pp. 22–53.
[2] T. Jordán and Z. Szigeti, *Detachments preserving local edge-connectivity of graphs*, SIAM J. Discrete Math., 17 (2003), pp. 72–87.
[3] L. Lovász, *Combinatorial Problems and Exercises*, North-Holland, Amsterdam, 1979, p. 52.
[4] W. Mader, *A reduction method for edge-connectivity in graphs*, Ann. Discrete Math., 3 (1978), pp. 145–164; 12 (1960), pp. 555–567.
[5] C. St. J. A. Nash-Williams, *Connected detachments of graphs and generalized Euler trails*, J. London Math. Soc. (2), 31 (1985), pp. 17–29.
[6] T. Watanabe and A. Nakamura, *Edge-connectivity augmentation problems*, J. Comput. System Sci., 35 (1987), pp. 96–144.

SIAM J. DISCRETE MATH.
Vol. 18, No. 3, pp. 592–607

AUTOMORPHISMS OF PROJECTIVE SPACES AND MIN-WISE INDEPENDENT SETS OF PERMUTATIONS*

MAXIM VSEMIRNOV†

Abstract. We study 4-restricted min-wise independent sets of permutations. In particular, we prove that the groups $\mathrm{PGL}(2,q)$ (for any prime power q) and $\mathrm{PGL}(3,q)$ (for any odd prime power q), permuting points of the projective line and the projective plane over the field $\mathbb{F}_q$, respectively, satisfy this property. We also show that (1) for any 4-restricted min-wise independent set $G \subseteq S_n$, its cardinality is at least $2n-2$, and (2) for any sufficiently large n, there exists a 4-restricted min-wise independent set $G \subseteq S_n$ of size $n^3 + o(n^3)$. The last two results improve previous bounds found by Itoh, Takei, and Tarui.

Key words. min-wise independent permutations, projective plane, projective linear group

AMS subject classifications. 20B25, 20B35, 05B25

DOI. 10.1137/S089548010241818X

1. Introduction.

Motivated by the problem of indexing Web documents, Broder et al. [3] introduced the notion of min-wise independent permutations. They served as a useful tool in detecting identical or almost identical documents. This notion also has some important theoretical applications, e.g., to derandomization [4, 16].

Let us recall the main definitions from [3]. Throughout this paper Ω denotes a finite set with some linear order $<$ on it and S_Ω denotes the symmetric group, i.e., the group of all permutations on Ω. For any finite set X, we denote its cardinality by $|X|$. If $|\Omega| = n$, then there is the natural isomorphism of ordered sets between Ω and $\{1, 2, \dots, n\}$ with the usual order on it. Sometimes we shall identify these sets and S_Ω will be identified with the group S_n of all permutations on $\{1, 2, \dots, n\}$.

DEFINITION 1.1. *A (multi)set G contained in S_Ω is called k-restricted min-wise independent if, for all $X \subseteq \Omega$ with $1 \le |X| \le k$ and for any $x \in X$, when a permutation π is chosen uniformly at random from G, we have*

$$\mathbf{Pr}_G(\min\{\pi(X)\} = \pi(x)) = \frac{1}{|X|}. \tag{1.1}$$

Here $\pi(X)$ denotes the set $\{\pi(y) : y \in X\}$.

As usual, the uniform distribution on a multiset means that the probability of choosing any element is proportional to its multiplicity. One may regard this as a very special case of nonuniform distributions. Sometimes it is useful to consider an arbitrary distribution on G.

DEFINITION 1.2. *We say that $G \subseteq S_\Omega$ is biased k-restricted min-wise independent if, for all $X \subseteq \Omega$ with $1 \le |X| \le k$ and for any $x \in X$, relation (1.1) holds when a permutation π is chosen from G at random with some biased distribution μ.*

Other variations of these notions, including approximate min-wise independence, are also possible. For details, see [3]. We just mention that approximate families are very useful in practical applications.

*Received by the editors November 20, 2002; accepted for publication (in revised form) May 9, 2004; published electronically February 25, 2005. This research was supported in part by grant 001 of the sixth competition of young scientists' research projects (Russian Academy of Sciences), by the Cariplo Foundation, and by INTAS (grant 2000-447).

http://www.siam.org/journals/sidma/18-3/41818.html

†Sidney Sussex College, Sidney Street, Cambridge, CB2 3HU, UK (m.vsemirnov@dpmms.cam.ac.uk, vsemir@pdmi.ras.ru).

One may regard min-wise independence as a weak analogue of k-wise independence. The latter notion has a lot of applications, e.g., in derandomization [13]. It turns out that in some cases min-wise independence may suffice [4, 16].

One of the main problems is to find exact and efficient constructions of min-wise independent sets of reasonable size, say, polynomial with respect to n. Broder et al. [3] proved (nonconstructively) the existence of only biased k-restricted min-wise independent sets of size $O(n^k)$. Later, in the case of the uniform distribution, an explicit construction of such sets of size $O(n^k e^{2k})$ was suggested by Itoh, Takei, and Tarui [10].

It is easy to show that S_Ω is $|\Omega|$-restricted min-wise independent and that the alternating group A_Ω is at least $(|\Omega|-2)$-restricted min-wise independent. Naturally, we can ask whether not only a subset but also a subgroup of S_Ω different from the whole symmetric group and its alternating subgroup can be k-restricted min-wise independent. Note that a similar question about nontrivial k-wise independent groups has, in general, a negative answer. More precisely, the property of k-wise independence is even stronger than k-transitivity. Recall that a group G acting on Ω is called k-transitive if, for any two k-tuples $x_1, \dots, x_k$ and $y_1, \dots, y_k$ such that $x_i \neq x_j$ and $y_i \neq y_j$ for $i \neq j$, there is some element $g \in G$ satisfying the condition $g(x_i) = y_i$, $i = 1, \dots, k$. Using the classification of finite simple groups, it is possible to show that, for $k \geq 6$, any finite k-transitive group must coincide either with a symmetric group or with an alternating group. Moreover, with the exception of the symmetric and alternating groups, the only finite 4- and 5-transitive groups are the Mathieu groups M_{11}, M_{12}, M_{23}, and M_{24}; see [5, Thm. 4.4] or [6, Chap. 7, sect. 3] and the references therein.

The problem of finding 2-restricted min-wise independent sets or groups of the smallest size is trivial. For instance, one can take only two permutations: the identical one and the permutation $(1, n)(2, n-1)\dots$, which reverses the order. An explicit example of 3-restricted min-wise independent groups appeared, e.g., in [16]. In this paper we present the first construction of "small" subgroups of S_Ω that are 4-restricted min-wise independent. However, even in the first nontrivial cases with $k = 3, 4$, the question whether these examples are optimal remains open.

For any prime power q, let $\mathbb{F}_q$ be the finite field with q elements and let $\mathbb{P}^d(\mathbb{F}_q)$ be the projective space of dimension d over $\mathbb{F}_q$. In our examples the set Ω will be the projective line $\mathbb{P}^1(\mathbb{F}_q)$ or the projective plane $\mathbb{P}^2(\mathbb{F}_q)$, and the groups will be the projective linear groups $\mathrm{PGL}(2, q)$ and $\mathrm{PGL}(3, q)$, respectively. Note that the natural action of $\mathrm{PGL}(2, q)$ on the projective line is 3-transitive and the action of $\mathrm{PGL}(3, q)$ on the projective plane is only 2-transitive. In section 4 we prove the following theorems.

THEOREM 1.3. *Let $\Omega = \mathbb{P}^1(\mathbb{F}_q)$ and $G = \mathrm{PGL}(2, q)$. Then G is 4-restricted min-wise independent with respect to any linear order on Ω.*

THEOREM 1.4. *Let $\Omega = \mathbb{P}^2(\mathbb{F}_q)$ and $G = \mathrm{PGL}(3, q)$, where q is odd. Then there exists a linear order on Ω such that G is 4-restricted min-wise independent with respect to this order.*

Remark 1. The required order will be described explicitly. Moreover, it can be effectively computed in time polynomial in n. This is important from the practical point of view, e.g., in applications to derandomization; see [16].

Several related open questions are stated at the end of section 4.

The action of $\mathrm{PGL}(2, q)$ on the projective line can be described by the transformations $z \mapsto (az + b)/(cz + d)$. Let us mention that the linear transformations $z \mapsto az + b \pmod p$ and their connections with min-wise independence were stud-

ied in [2]. Although, in general, the linear transformations are not exactly min-wise independent, they still can be used in practice; see [2, 3].

Note that $|\mathrm{PGL}(2,q)| = q(q^2-1)$ and $|\mathbb{P}^1(\mathbb{F}_q)| = q+1$. Thus, Theorem 1.3 provides examples of 4-restricted min-wise independent sets of the smallest size known today; compare with the result of Itoh, Takei, and Tarui [10] cited above. In Theorem 1.3 the size of the set Ω is $q+1$ for a prime power q. In section 5 we provide an explicit way of constructing "small" 4-restricted min-wise independent sets for almost all n.

THEOREM 1.5. *For any sufficiently large n, there exists a 4-restricted min-wise independent subset of S_n of cardinality $n^3 + o(n^3)$.*

Itoh, Takei, and Tarui [10] gave the first nontrivial lower bound for the size of a k-restricted min-wise independent set for $k \geq 3$. Namely, they proved that for such a set G of permutations on n symbols, $|G| \geq n-1$. In section 6 we slightly improve their result.

THEOREM 1.6. *Let $n \geq 4$ and $G \subseteq S_n$ be a biased 4-restricted min-wise independent set of permutations on $\{1,\dots,n\}$. Then $|G| \geq 2n-2$.*

Remark 2. One may expect better estimates for larger values of k. Exploring similar ideas, Norin [14] proved that the cardinality of any biased k-restricted min-wise independent set is at least $\binom{n-1}{\lfloor (k-1)/2 \rfloor}$. However, for $k=4$, Theorem 1.6 provides a better bound.[1]

2. Some auxiliary results. Here we present some results that are very useful in checking whether a given group of permutations is k-restricted min-wise independent. It should be noted that Lemma 2.2 also appeared in [7].

LEMMA 2.1. *Let Ω be a finite linearly ordered set. Given a subgroup G of S_Ω and a set $X \subseteq \Omega$, consider the set stabilizer*

$$G_{\{X\}} = \{\sigma \in G : \sigma(X) = X\}.$$

Let $\Gamma \subseteq X$ be an orbit of $G_{\{X\}}$. Then, with respect to the uniform distribution on G and the above order, the probabilities

$$\mathbf{Pr}_G(\min\{\pi(X)\} = \pi(x))$$

are one and the same for any $x \in \Gamma$.

Proof. Let us fix some linear order on Ω. Write $X = \{x_1,\dots,x_k\}$ for some k and fix $x \in \Gamma$. Let T be a left transversal for $G_{\{X\}}$ in G and

$$T(\Gamma) = \{\tau \in T : \min \tau(X) \in \tau(\Gamma)\}.$$

Take $\tau \in T$ and put $y_i = \tau(x_i)$, $i = 1,\dots,k$. Choose j such that y_j is the smallest among all y_i's with respect to the given order. Clearly, for $\sigma \in G_{\{X\}}$, we have that $\tau\sigma(x) = \min \tau\sigma(X)$ if and only if $\sigma(x) = x_j$. If $\tau \notin T(\Gamma)$, then $x_j \notin \Gamma$, and no such σ exists. For $\tau \in T(\Gamma)$, the number of required σ's is $|G_{\{X\}}|/|\Gamma|$ since by our

[1] During a revision of this paper, the author found that Itoh, Takei, and Tarui announced (Proceedings of the 35th ACM Symposium on Theory of Computing, 2003) further improvements of Norin's result for even values of k. For example, for $k=4$, the announced bound coincides with that given in Theorem 1.6.

assumptions $G_{\{X\}}$ is transitive on Γ. Therefore,

$$\begin{aligned}|\{\pi \in G : \min \pi(X) = \pi(x)\}| &= \sum_{\tau \in T} |\{\sigma \in G_{\{X\}} : \min \tau\sigma(X) = \tau\sigma(x)\}| \\ &= \sum_{\tau \in T(\Gamma)} |\{\sigma \in G_{\{X\}} : \min \tau\sigma(X) = \tau\sigma(x)\}| \\ &= \sum_{\tau \in T(\Gamma)} \frac{|G_{\{X\}}|}{|\Gamma|} = \frac{|T(\Gamma)| \cdot |G_{\{X\}}|}{|\Gamma|}. \end{aligned} \tag{2.1}$$

Clearly, the right-hand fraction does not depend on a particular choice of $x \in \Gamma$. □

LEMMA 2.2. *Under the hypothesis of Lemma* 2.1 *assume also that* $G_{\{X\}}$ *acts transitively on* X. *Then* (1.1) *holds for any* $x \in X$.

Proof. In this case, Γ coincides with X and $T(\Gamma)$ with T. Since $|T| = |G|/|G_{\{X\}}|$, the right-hand side of (2.1) becomes $|G|/|X|$. □

LEMMA 2.3. *Under the hypothesis of Lemma* 2.1 *assume that the set* X *splits into several* $G_{\{X\}}$*-orbits, say,* $\Gamma_1, \dots, \Gamma_s$, *where* $s > 1$. *If* (1.1) *holds for all* $x \in X \setminus \Gamma_1$, *then it also holds for any* $x \in \Gamma_1$.

Proof. Our assumption, together with (2.1), implies that

$$\frac{|T(\Gamma_i)| \cdot |G_{\{X\}}|}{|\Gamma_i|} = \frac{|G|}{|X|} \quad \text{for } i = 2, \dots, s. \tag{2.2}$$

On the other hand

$$|\Gamma_1| + |\Gamma_2| + \cdots + |\Gamma_s| = |X|$$

and

$$|T(\Gamma_1)| + |T(\Gamma_2)| + \cdots + |T(\Gamma_s)| = |T| = \frac{|G|}{|G_{\{X\}}|}.$$

It is easy to show that (2.2) is also true for $i = 1$. Together with (2.1), this proves the claim. □

3. The action of PGL$(d+1, q)$ on the points of $\mathbb{P}^d(\mathbb{F}_q)$. We start this section by recalling some facts about the group $G = \mathrm{PGL}(d+1, q)$ and its action on the points of $\mathbb{P}^d(\mathbb{F}_q)$. These facts can also be found in many standard textbooks, like [1, Chap. 3] and [11, Chap. 3]. Note that most of the statements below remain true for an arbitrary field instead of $\mathbb{F}_q$. In what follows we consider only configurations of *pairwise distinct* points. However, two different configurations considered simultaneously may have a nonempty intersection.

DEFINITION 3.1. *We say that* N *points* $A_1, \dots, A_N \in \mathbb{P}^d(\mathbb{F}_q)$ *are collinear if their projective hull* $\langle A_i : 1 \le i \le N \rangle$ *is a projective line.*

DEFINITION 3.2. *A set of points* $A_1, \dots, A_N \in \mathbb{P}^d(\mathbb{F}_q)$ *is called generic if, for any* $m \le \min\{N, d+1\}$ *and for all subsets* $S \subseteq \{1, \dots, N\}$ *with* $|S| = m$, *the (projective) dimension of the projective hull* $\langle A_i : i \in S \rangle$ *equals* $m - 1$.

In particular, any two distinct points in $\mathbb{P}^d(\mathbb{F}_q)$, as well as any three distinct points on the projective line, form a generic set. A triple on the projective plane is generic if and only if these points are not collinear. Four distinct points on the projective plane are generic if and only if any three of them are not collinear. There are two types of nongeneric quadruples on the projective plane:

- four collinear points;
- three points on one line with the fourth point not on this line.

Let us consider some (ordered) configuration of points in the projective space $\mathbb{P}^d(\mathbb{F}_q)$. To apply Lemma 2.2 we need a description of the orbit of this configuration under the action of $\mathrm{PGL}(d+1,q)$.

The following lemma gives the answer for all generic sets of cardinality $d+2$, where d is the dimension of the projective space. For the proof, see, e.g., [11, Chap. 3, sect. 8, Thm. 8].

LEMMA 3.3. *Let $A_1,\dots,A_{d+2}$ and $A'_1,\dots,A'_{d+2}$ be two generic sets of points in $\mathbb{P}^d(\mathbb{F})$. There exists a unique $\sigma\in\mathrm{PGL}(d+1,q)$ such that $\sigma(A_i)=A'_i$ for all $i=1,\dots,d+2$.*

Let X be a generic set of cardinality $d+2$. Since any permutation of X is again a generic set, it follows from Lemma 3.3 that the set stabilizer $\mathrm{PGL}(d+1,q)_{\{X\}}$ acts on X transitively. Together with Lemma 2.2 this implies the following corollary.

COROLLARY 3.4. *Let $G=\mathrm{PGL}(d+1,q)$ and $\Omega=\mathbb{P}^d(\mathbb{F}_q)$. If X is a generic set of cardinality $d+2$ and $x\in X$, then, for any linear order on Ω, relation* (1.1) *holds with respect to the uniform distribution on G.*

Now we describe the orbits of quadruples of collinear points. Let A, B, C, D be four distinct collinear points in $\mathbb{P}^d(\mathbb{F}_q)$. There are vectors $\mathbf{a},\mathbf{b}\in\mathbb{F}_q^{d+1}$ such that

$$\mathbf{a}\in A,\quad \mathbf{b}\in B,\quad \mathbf{a}+\mathbf{b}\in C.$$

Moreover, $\mathbf{a}$ and $\mathbf{b}$ are determined up to a common scalar multiple. In addition, there is a unique $s\in\mathbb{F}_q\setminus\{0,1\}$ such that

$$\mathbf{a}+s\mathbf{b}\in D.$$

It depends only on the quadruple, but not on the choice of $\mathbf{a}$ and $\mathbf{b}$ above. This s is called the *cross ratio* of the ordered quadruple A, B, C, D (e.g., see [1, Chap. 3, sect. 4] or [11, Chap. 3]). We denote it by $[A,B,C,D]$. The cross ratio remains invariant under the action of $\mathrm{PGL}(d+1,q)$.

LEMMA 3.5. *Let (A,B,C,D) and (A',B',C',D') be two quadruples of distinct collinear points on $\mathbb{P}^d(\mathbb{F}_q)$. Then $[A,B,C,D]=[A',B',C',D']$ if and only if there exists $\sigma\in\mathrm{PGL}(d+1,q)$ such that*

$$\sigma(A)=A',\quad \sigma(B)=B',\quad \sigma(C)=C',\quad \sigma(D)=D'.$$

For the proof see, e.g., [1, sect. III.4, Thm. 1b].

The cross ratio satisfies the following partial symmetry relations.

LEMMA 3.6. *If A, B, C, D are four distinct collinear points in $\mathbb{P}^d(\mathbb{F}_q)$, then*

$$[A,B,C,D]=[B,A,D,C]=[C,D,A,B]=[D,C,B,A]\,.$$

The reader can find the proof for this lemma in [1, Chap. 3, sect. 4] or in [11, Chap. 3].

Using Lemmas 3.5 and 3.6, we conclude that for any four distinct collinear points A, B, C, D, the ordered quadruples

$$(A,B,C,D)\,,\quad (B,A,D,C)\,,\quad (C,D,A,B)\,,\quad (D,C,B,A)$$

lie in one and the same orbit under the action of $G=\mathrm{PGL}(d+1,q)$. In particular, for $X=\{A,B,C,D\}$, the set stabilizer $G_{\{X\}}$ acts on X transitively. Combining with Lemma 2.2 we get the following corollary.

COROLLARY 3.7. *Let $G = \mathrm{PGL}(d+1, q)$ and $\Omega = \mathbb{P}^d(\mathbb{F}_q)$. If X is a set of four collinear points in Ω, then, with respect to the uniform distribution on G and any linear order on Ω, relation* (1.1) *holds for any $x \in X$.*

To complete this section, we describe the orbits of nongeneric noncollinear quadruples on the projective plane. This description is also known. However, it is rather difficult to find its proof in the literature. Thus, for the sake of completeness, we give a proof of the following lemma.

LEMMA 3.8. *Let (A, B, C, D) and (A', B', C', D') be two quadruples of distinct points on $\mathbb{P}^2(\mathbb{F}_q)$ such that*

- *A, B, C are collinear and D does not belong to their projective hull;*
- *A', B', C' are collinear and D' does not belong to their projective hull.*

Then there are exactly $q-1$ automorphisms $\sigma \in \mathrm{PGL}(3, q)$ such that

$$\sigma(A) = A', \quad \sigma(B) = B', \quad \sigma(C) = C', \quad \sigma(D) = D'.$$

Proof. Let us fix $\mathbf{a}$, $\mathbf{b}$, $\mathbf{d}$ and $\mathbf{a}'$, $\mathbf{b}'$, $\mathbf{d}'$ in $\mathbb{F}_q^3$ such that

$$\begin{array}{llll} \mathbf{a} \in A, & \mathbf{b} \in B, & \mathbf{a}+\mathbf{b} \in C, & \mathbf{d} \in D, \\ \mathbf{a}' \in A', & \mathbf{b}' \in B', & \mathbf{a}'+\mathbf{b}' \in C', & \mathbf{d}' \in D'. \end{array}$$

Note that $\mathbf{a}$, $\mathbf{b}$, $\mathbf{d}$ and $\mathbf{a}'$, $\mathbf{b}'$, $\mathbf{d}'$ are the bases of $\mathbb{F}_q^3$. Any required σ is induced by some linear automorphism $g \in \mathrm{GL}(3, q)$ such that

$$g(\mathbf{a}) = u\mathbf{a}', \quad g(\mathbf{b}) = v\mathbf{b}', \quad g(\mathbf{d}) = w\mathbf{d}', \quad g(\mathbf{a}+\mathbf{b}) = y(\mathbf{a}'+\mathbf{b}')$$

for some $u, v, w, y \in \mathbb{F}_q \setminus \{0\}$. In particular, these relations imply that $u = v = y$. Each choice of u and w uniquely determines g. Hence there are $(q-1)^2$ such g's. Since any projectivity $\sigma \in \mathrm{PGL}(3, q)$ is induced exactly by $q-1$ different linear automorphisms, there are $q-1$ required σ's. ∎

4. Proofs of Theorems 1.3 and 1.4. Now we are able to prove two main theorems of this paper.

Proof of Theorem 1.3. If $|X| = 4$, then (1.1) follows from Corollary 3.7. If $|X| = 3$, then (1.1) follows from Corollary 3.4. Since any two points A, B can be extended to a generic triple, Lemma 3.3 implies the existence of some $\sigma \in \mathrm{PGL}(2, q)$ that maps A to B and B to A. Therefore, by Lemma 2.2 we have (1.1) in the case $|X| = 2$. Finally, if $|X| = 1$, then (1.1) holds trivially. ∎

As we saw in Theorem 1.3, the group $\mathrm{PGL}(2, q)$ is 4-restricted min-wise independent for any order on the projective line. In general let the group $G = \mathrm{PGL}(d+1, q)$ act on the points of the space $\Omega = \mathbb{P}^d(\mathbb{F}_q)$ and let $n = |\mathbb{P}^d(\mathbb{F}_q)|$. Different linear orders on Ω give rise to different embeddings of $\mathrm{PGL}(d+1, q)$ into S_n. But from the algebraic point of view, these subgroups are similar since they are conjugate in S_n. However, as the following examples show, $\mathrm{PGL}(d+1, q)$ with $d > 1$ may be min-wise independent for one order and may not be for another.

Condition (1.1) can be restated in a purely combinatorial way. For instance, if $X = \{A, B, C, D\}$, where A, B, C are collinear, and $D \notin \langle A, B, C\rangle$ and $x = D$, then (1.1) becomes the following: exactly one-fourth of all configurations (A, B, C, D) described in Lemma 3.8 satisfies the additional property $D = \min\{A, B, C, D\}$.

Example 1. Let $q = 2$. Consider the following order on $\mathbb{P}^2(\mathbb{F}_2)$: $(1:0:0) < (1:1:0) < (1:0:1) < (1:1:1) < (0:1:0) < (0:1:1) < (0:0:1)$. There are $168 = 4 \cdot 42$ ways to choose an ordered quadruple (A, B, C, D) such that A, B,

C lie on one projective line and D does not belong to this line. By Lemma 3.8, any such configuration can be mapped (with one and the same probability) onto another. However, there are 48 configurations of the above type such that, with respect to this order, D is less than A, B, and C. Namely, they are

$$\begin{gathered}((0:0:1),(0:1:1),(0:1:0),(1:0:0)),\\ ((0:0:1),(0:1:1),(0:1:0),(1:1:0)),\\ ((0:0:1),(0:1:1),(0:1:0),(1:0:1)),\\ ((0:0:1),(0:1:1),(0:1:0),(1:1:1)),\\ ((0:0:1),(1:1:1),(1:1:0),(1:0:0)),\\ ((0:1:1),(1:0:1),(1:1:0),(1:0:0)),\\ ((0:1:0),(1:1:1),(1:0:1),(1:0:0)),\\ ((0:1:0),(1:1:1),(1:0:1),(1:1:0)),\end{gathered}$$

and all obtained from them by permuting A, B, C.

Similarly, if we revert the order, there are only 36 such configurations with $D = \min\{A, B, C, D\}$. Hence, for both these orders, $\mathrm{PGL}(3, 2)$ is not 4-restricted min-wise independent.

Example 2. Again take $q = 2$ and arrange the points of the projective plane as follows: $(1:0:0) < (1:1:0) < (1:0:1) < (0:0:1) < (0:1:0) < (0:1:1) < (1:1:1)$. A direct computation based on Lemmas 3.8 and 3.3 shows that, with respect to this order, the group $\mathrm{PGL}(3, 2)$ is 4-restricted min-wise independent.

Example 3. Let $q = 3$. Consider the following order on $\mathbb{P}^2(\mathbb{F}_3)$: $(1:0:0) < (1:0:1) < (1:0:2) < (1:1:0) < (1:1:1) < (1:1:2) < (1:2:0) < (1:2:1) < (1:2:2) < (0:1:0) < (0:1:1) < (0:1:2) < (0:0:1)$. There are $2808 = 4 \cdot 702$ ordered quadruples (A, B, C, D) such that A, B, C lie on one projective line while D does not belong to this line. However, for the given order, D is the smallest among A, B, C, D only in 648 cases. Similarly, for the reverse order, there are 756 such configurations with $D = \min\{A, B, C, D\}$. Hence, for both these orders, $\mathrm{PGL}(3, 3)$ is not 4-restricted min-wise independent.

Proof of Theorem 1.4. The case $|X| = 1$ is again trivial. If either $|X| = 2$ or $|X| = 3$ and the three points of X are not collinear, then X can be extended to a generic quadruple. It follows from Lemma 3.3 that any permutation of X is induced by some $\sigma \in \mathrm{PGL}(3, q)_{\{X\}}$. Hence $\mathrm{PGL}(3, q)_{\{X\}}$ acts transitively on X and (1.1) follows from Lemma 2.2. If $|X| = 3$ and the points of X are collinear, then we take a point D which does not belong to their projective hull. By Lemma 3.8 there is some $\sigma \in \mathrm{PGL}(3, q)$ that fixes D and acts on X as a given permutation. Thus, $\mathrm{PGL}(3, q)_{\{X\}}$ is again transitive on X and we can apply Lemma 2.2. If $|X| = 4$ and X is generic, then (1.1) follows from Corollary 3.4. If $|X| = 4$ and the four points are collinear, then we deduce (1.1) from Corollary 3.7. Moreover, in all the cases above, (1.1) holds for any order on Ω.

The only remaining case is when X consists of four noncollinear points, say, x_1, x_2, x_3, and x_4, such that the first three are collinear. It is the most difficult case, and the above examples show that not every order on Ω will suit us.

First, we define an order. There is a natural correspondence between linear orders on $\mathbb{P}^2(\mathbb{F}_q)$ and bijective maps from $\mathbb{P}^2(\mathbb{F}_q)$ onto $\{1, 2, \ldots, 1 + q + q^2\}$. It will be convenient to identify such an order and the corresponding map.

Let ω be any bijection from $\mathbb{F}_q$ onto $\{1, 2, \ldots, q\}$. Consider the following ordering

ψ of $\mathbb{P}^2(\mathbb{F}_q)$:

$$\begin{aligned}
\psi(1,a_1,a_2) &= \omega(a_2) + q(\omega(a_1)-1) \quad \text{if } \omega(a_1) \le \frac{q+1}{2},\\
\psi(0,0,1) &= \frac{q^2+q}{2} + 1,\\
\psi(1,a_1,a_2) &= \omega(a_2) + q(\omega(a_1)-1) + 1 \quad \text{if } \omega(a_1) \ge \frac{q+3}{2},\\
\psi(0,1,a) &= \omega(a) + q^2 + 1.
\end{aligned}$$

Now we prove that this order is suitable. There are two $G_{\{X\}}$-orbits in X, namely, $\{x_1,x_2,x_3\}$ and $\{x_4\}$. By Lemma 2.3 it is sufficient to check that (1.1) is true for $x = x_4$. Then it will be satisfied for $x = x_1$, x_2, or x_3 automatically. By Lemma 3.8 it is sufficient to count the number of ordered quadruples (A,B,C,D) such that A, B, C are collinear, but four points in common are not collinear, and D is less than A, B, and C with respect to the order defined above.

The projective plane $\mathbb{P}^2(\mathbb{F}_q)$ contains q^2+q+1 lines. We split them into three disjoint families as follows. For $u,v \in \mathbb{F}_q$, we define the lines

$$\begin{aligned}
L_1 &= \langle (0:1:0),(0:0:1)\rangle = \{(0:1:a) : a \in \mathbb{F}_q\} \cup \{(0:0:1)\},\\
L_2(u) &= \langle (1:u:0),(0:0:1)\rangle = \{(1:u:a) : a \in \mathbb{F}_q\} \cup \{(0:0:1)\},\\
L_3(u,v) &= \langle (1:0:u),(0:1:v)\rangle = \{(1:a:u+va) : a \in \mathbb{F}_q\} \cup \{(0:1:v)\}.
\end{aligned}$$

Each family will be considered separately. For any line L in $\mathbb{P}^2(\mathbb{F}_q)$, we define $N(L)$ as the number of ordered quadruples (A,B,C,D) such that $L = \langle A,B,C\rangle$, $D \in \mathbb{P}^2(\mathbb{F}_q) \setminus L$, and $\psi(D) = \min\{\psi(A),\psi(B),\psi(C),\psi(D)\}$. In addition, let $N_1 = N(L_1)$, $N_2 = \sum_u N(L_2(u))$, and $N_3 = \sum_{u,v} N(L_3(u,v))$.

Case 1. $\langle A,B,C\rangle = L_1$. There are $q(q-1)(q-2)$ ordered triples (A,B,C) with $(0,0,1) \notin \{A,B,C\}$. In this case one can choose D from q^2 points, namely, $D = (1:s:t)$, where $s,t \in \mathbb{F}_q$. On the other hand, there are $3q(q-1)$ triples (A,B,C) such that $(0:0:1) \in \{A,B,C\}$. In this case there are only $\frac{q(q+1)}{2}$ possibilities for D: $D = (1:s:t)$, where $\omega(s) \le \frac{q+1}{2}$ and $t \in \mathbb{F}_q$. Thus,

$$N_1 = \frac{q^2(q-1)(2q^2-q+3)}{2}. \tag{4.1}$$

Case 2. $\langle A,B,C\rangle = L_2(u)$. As above, there are $q(q-1)(q-2)$ triples (A,B,C) with $(0:0:1) \notin \{A,B,C\}$. In each case D can be chosen from $q\cdot(\omega(u)-1)$ points, namely, $D = (1:s:t)$, where $\omega(s) < \omega(u)$ and $t \in \mathbb{F}_q$. In addition, there are $3q(q-1)$ triples (A,B,C) such that $(0:0:1) \in \{A,B,C\}$. For these triples, D is one of $q\cdot\min(\omega(u)-1, \frac{q+1}{2})$ points $(1:s:t)$, where $\omega(s) \le \min(\omega(s)-1, \frac{q+1}{2})$ and $t \in \mathbb{F}_q$. Thus, $N(L_2(u)) = q^2(q-1)(q-2)(\omega(u)-1) + 3q^2(q-1)\cdot\min(\omega(u)-1, \frac{q+1}{2})$. Taking into account that $\omega(u)$ ranges over $1,\dots,q$ when u ranges over $\mathbb{F}_q$, we have

$$N_2 = \frac{q^2(q-1)^2(4q^2+q+9)}{8}. \tag{4.2}$$

Case 3. $\langle A,B,C\rangle = L_3(u,v)$. There are two kinds of triples $\{A,B,C\}$ here. The first type is

$$\{A,B,C\} = \{(1:a:u+va),(1:b:u+vb),(1:c:u+vc)\}$$

for some a, b, and $c \in \mathbb{F}_q$. Without loss of generality, we may assume that $\omega(a) < \omega(b) < \omega(c)$, so that for each set $\{a, b, c\}$ there are six different *ordered* triples (A, B, C). The second type is

$$\{A, B, C\} = \{(1 : a : u + va), (1 : b : u + vb), (0 : 1 : v)\}.$$

Again, we assume that $\omega(a) < \omega(b)$, so that there are six *ordered* triples (A, B, C). For both types of triples, there are the following possibilities for D:

- $D = (1 : s : t)$, where $\omega(s) < \omega(a)$ and $t \neq u + vs$ (the latter condition says that $D \notin \langle A, B, C\rangle$);
- $D = (1 : a : t)$, where $\omega(t) < \omega(u + va)$;
- $D = (0 : 0 : 1)$, provided $\omega(a) \geq (q + 3)/2$.

Therefore,

$$\begin{aligned}
N_3(u, v) = {} & 6(q-1)\sum_{a}\sum_{\substack{b\\ \omega(b)>\omega(a)}}\sum_{\substack{c\\ \omega(c)>\omega(b)}}(\omega(a)-1)\\
& +6\sum_{a}\sum_{\substack{b\\ \omega(b)>\omega(a)}}\sum_{\substack{c\\ \omega(c)>\omega(b)}}(\omega(u+va)-1))\\
& +6\sum_{\substack{a\\ \omega(a)\geq(q+3)/2}}\sum_{\substack{b\\ \omega(b)>\omega(a)}}\sum_{\substack{c\\ \omega(c)>\omega(b)}}1\\
& +6(q-1)\sum_{a}\sum_{\substack{b\\ \omega(b)>\omega(a)}}(\omega(a)-1)\\
& +6\sum_{a}\sum_{\substack{b\\ \omega(b)>\omega(a)}}(\omega(u+va)-1))\\
& +6\sum_{\substack{a\\ \omega(a)\geq(q+3)/2}}\sum_{\substack{b\\ \omega(b)>\omega(a)}}1\\
= {} & \frac{(q+1)(q-1)(2q^3-6q^2+5q-3)}{8}\\
& +3\sum_{a}(q-\omega(a))(q-\omega(a)+1)(\omega(u+va)-1).
\end{aligned}$$

To evaluate N_3, we sum $N_3(u, v)$ over all u and v. For any fixed v and a, if u ranges over $\mathbb{F}_q$, then $w = u + va$ also ranges over $\mathbb{F}_q$. Thus, we can change the order of summation:

$$\begin{aligned}
& 3\sum_{v}\sum_{u}\sum_{a}(q-\omega(a))(q-\omega(a)+1)(\omega(u+va)-1)\\
& \quad = 3\sum_{v}\sum_{a}\sum_{u}(q-\omega(a))(q-\omega(a)+1)(\omega(u+va)-1)\\
& \quad = 3\sum_{v}\sum_{a}\sum_{w}(q-\omega(a))(q-\omega(a)+1)(\omega(w)-1)\\
& \quad = \frac{(q+1)q^3(q-1)^2}{2}.
\end{aligned}$$

Therefore,

$$N_3 = \frac{(q+1)q^2(q-1)(2q^3-2q^2+q-3)}{8}. \tag{4.3}$$

Adding (4.1)–(4.3) together, we obtain that $N_1 + N_2 + N_3 = (q^7 + q^6 - q^4 - q^3)/4$, which is one-fourth of the total amount of noncollinear quadruples (A, B, C, D) such that A, B, and C are collinear. This completes the proof. □

We complete this section with some open questions. For reasons of simplicity, we proved Theorem 1.4 for odd prime powers q. Example 2 shows, however, that one may expect similar results for finite fields of characteristic 2.

Let $\le$ be a linear order on $\mathbb{P}^2(\mathbb{F}_q)$ and $\le^*$ be the reverse order, i.e., $A \le^* B$ if and only if $B \le A$. Consider ordered quadruples (A, B, C, D) such that A, B, C are collinear but D is not on the same line. One can check or deduce directly from Lemma 3.8 that the total number of such configurations is $q^3(q-1)(q+1)(q^2+q+1)$. Let $N_\le$ be the number of such quadruples with $D = \min(A, B, C, D)$. As we could see in Examples 1 and 3 above, $N_\le$ may differ from $\frac{q^3(q-1)(q+1)(q^2+q+1)}{4}$. However, in these examples,

$$N_\le + N_{\le^*} = \frac{q^3(q-1)(q+1)(q^2+q+1)}{2}. \tag{4.4}$$

This relation was also checked for some other orders on $\mathbb{P}^2(\mathbb{F}_2)$ and $\mathbb{P}^2(\mathbb{F}_3)$. It also holds for orders which occur in the proof of Theorem 1.4. This supports the following conjecture.

Conjecture 1. *For any linear order on $\mathbb{P}^2(\mathbb{F}_q)$, relation* (4.4) *holds.*

Finally, one can ask whether, for some linear order on $\Omega = \mathbb{P}^{k-2}(\mathbb{F}_q)$, where $k > 4$, the group $G = \mathrm{PGL}(k-1, q)$ is k-restricted min-wise independent. This question seems to be very difficult. In particular, one must consider each orbit under the action of G. Thus, we need some parameterization for nongeneric k-tuples, e.g., for k-tuples lying on one line as a special case. The action of $\mathrm{PGL}(2, q)$ on the set of k-tuples of points on the projective line $\mathbb{P}^1(\mathbb{F}_p)$ has attracted special interest. We mention a recent result [12, Thm. C].

5. 4-restricted min-wise independent sets for almost all n. Given a k-restricted min-wise independent set $F \subseteq S_m$ for some m, we can produce in a standard way min-wise independent subsets in S_n, where $n \le m$ (for instance, see [16]). Namely, consider the following "projection" from S_m onto S_n. Take any $\pi \in S_m$ and consider $\{\pi(1), \dots, \pi(n)\}$, which is a linearly ordered set. Thus, there is the (unique) map ϕ from this set to $\{1, \dots, n\}$, which preserves the order. Define $\pi' \in S_n$ by $\pi'(i) = \phi\pi(i)$, $i = 1, \dots, n$. Alternatively, π' can be defined as

$$\pi'(i) = |\{1, 2, \dots, \pi(i)\} \cap \{\pi(1), \pi(2), \dots, \pi(n)\}|.$$

Lemma 5.1. *Given a k-restricted min-wise independent set $F \subseteq S_m$, let F' be the image of F under the projection described above. Then the (multi)set F' is also k-restricted min-wise independent.*

Proof. For any $x, y \in \{1, \dots, n\}$ and any $\pi \in S_m$, we have $\pi'(x) < \pi'(y)$ if and only if $\pi(x) < \pi(y)$. Therefore, for any $X \subseteq \{1, 2, \dots, n\}$, $\min\{\pi'(X)\} = \pi'(x)$ if and only if $\min\{\pi(X)\} = \pi(x)$. Consequently,

$$\mathbf{Pr}_{F'}(\min\{\pi(X)\} = \pi(x)) = \mathbf{Pr}_F(\min\{\pi(X)\} = \pi(x)). \qquad \square$$

Unfortunately, different permutations may give one and the same projection. Thus, in general, F' will be a multiset only. If we want to find a min-wise independent *set* F', then we must take more care. In particular, the restriction of the above projection on F must be injective.

Example 4. For a prime p, consider an embedding of $\mathrm{PGL}(2,p)$ into S_{p+1} induced by the following order on the projective line:

$$\bar{1} < \bar{2} < \cdots < \bar{p}-\bar{1} < \bar{p} < \infty.$$

Let $\sigma_1 = \mathrm{id}$ and $\sigma_2 \in \mathrm{PGL}(2,p)$ be the transformation that maps z to $z+1$. As π_1 and $\pi_2 \in S_{p+1}$, we take permutations induced by σ_1 and σ_2, respectively, i.e., $\pi_1 = \mathrm{id}$ and $\pi_2 = (12\cdots p)$ is a cycle of length p. Then, for any $n \le p-1$, both π_1' and π_2' become the identity in S_n.

First, we describe briefly our construction of "small" 4-restricted min-wise independent sets for almost all n. Given an n, we find a prime $p > n$ and take $G \subseteq S_{p+1}$, which is an image of $\mathrm{PGL}(2,p)$ induced by the above order on the projective line (see Example 4). Then we explicitly construct $\pi \in S_{p+1}$ such that πG is still 4-restricted min-wise independent and the projection from πG into S_n is injective provided the difference $p-n$ is small enough (more precisely, $p-n = O(n^\theta)$ for some $\theta < 1$).

The following lemma shows that multiplication by a given permutation on the right preserves restricted min-wise independence.

LEMMA 5.2. *For any k-restricted min-wise independent set $G \subseteq S_\Omega$ and for any $\pi \in S_\Omega$, the set*

$$G\pi = \{\sigma \in S_n : \exists \tau \in G \text{ such that } \sigma = \tau\pi\}$$

is also k-restricted min-wise independent.

Proof. Let $X = \{x_1, \ldots, x_j\} \subseteq \Omega$, where $j \le k$, and let $x \in X$. Put $Y = \{\pi(x_1), \ldots, y_j = \pi(x_j)\}$ and $y = \pi(x) \in Y$. Since G is k-restricted min-wise independent, we have

$$\mathbf{Pr}_{G\pi}(\min\{\sigma(X)\} = \sigma(x)) = \mathbf{Pr}_G(\min\{\tau(Y)\} = \tau(y)) = \frac{1}{|Y|} = \frac{1}{|X|}. \qquad \square$$

In contrast with the previous lemma, multiplication by a given permutation on the left in general may not preserve min-wise independence. However, it still preserves it in the following special case.

LEMMA 5.3. *Let G be an image of $\mathrm{PGL}(2,q)$ embedded into S_{q+1} using the natural action of $\mathrm{PGL}(2,q)$ on $\mathbb{P}^1(\mathbb{F}_q)$. For any permutation $\pi \in S_{q+1}$, the set*

$$\pi G = \{\sigma \in S_n : \exists \tau \in G \text{ such that } \sigma = \pi\tau\}$$

is 4-restricted min-wise independent.

Proof. By Theorem 1.3, the corresponding embedding of $\mathrm{PGL}(2,q)$ into S_{q+1} is 4-restricted min-wise independent for any linear order on $\Omega = \mathbb{P}^1(\mathbb{F}_q)$. Changing the order on Ω leads to a conjugate embedding of $\mathrm{PGL}(2,q)$ into S_{q+1}. Therefore, $\pi G \pi^{-1}$ is also 4-restricted min-wise independent. Applying Lemma 5.2 to $\pi G \pi^{-1}$ and π we complete the proof. $\square$

Proof of Theorem 1.5. It is known that there exist $\theta < 1$ and $n_0(\theta)$ such that, for any integer $n > n_0(\theta)$, the interval $(n, n+n^\theta]$ contains at least one prime number. For example, one may take $\theta = 11/20 + \epsilon$ for any positive ϵ; see [8, 9]. For a historical

review of results of this type and current records, see also [15, Chap. 4]. Therefore, for any sufficiently large n, we can find a prime $p > n$ such that

$$n > 32d(\lfloor \log_2 d \rfloor + 1)^3, \quad \text{where } d = p + 1 - n. \tag{5.1}$$

We define G as the embedding of $\mathrm{PGL}(2, p)$ into S_{p+1}, induced by the same order on $\mathbb{P}^1(\mathbb{F}_p)$ as in Example 4. In particular, $|G| = p(p+1)(p-1) = n^3 + o(n^3)$. In what follows we identify projectivities with the corresponding permutations.

Now we describe a permutation π. Let p_i, $i = 1, 2, \ldots$, denote the ith prime number. Take $r = \lfloor \log_2 d \rfloor + 1$. Consider the following increasing sequence:

$$a_{4d(i-1)+j} = jp_i + 4d \sum_{u=1}^{i-1} p_u, \qquad i = 1, \ldots, r, \quad j = 1, \ldots, 4d.$$

In particular, if $i = 1$, then the last sum vanishes and we have

$$a_j = jp_1 = 2j \quad \text{for } j = 1, \ldots, 4d.$$

In other words, we arrange numbers in r groups, each group contains $4d$ elements, and within the ith group the difference between consecutive elements is p_i. Thus, our sequence begins with

$$2, 4, \ldots, 8d - 2, 8d, 8d + 3, \ldots, 20d - 3, 20d, 20d + 5, \ldots.$$

Note that $a_{4dr} = 4d(p_1 + \cdots + p_r) \le 4drp_r$. Using the rather rough bound $p_r \le 2r^2$, we conclude that elements of our sequence are bounded by $a = 8dr^3$. By (5.1), $p + 1 - a > a$. We define π as the following product of nonintersecting transpositions:

$$\pi = \prod_{s=1}^{4dr} (a_s, p + 2 - a_s).$$

Now take τ, $\sigma \in G$ and assume that $(\pi\tau)' = (\pi\sigma)'$. Our aim is to show that this assumption implies $\tau = \sigma$. Put

$$t_s = [(\pi\tau)']^{-1}(s) = [(\pi\sigma)']^{-1}(s), \qquad s = 1, 2 \ldots, n.$$

In particular, $\{t_1, t_2, \ldots, t_n\} = \{1, 2, \ldots, n\}$ and

$$(\pi\tau)'(t_1) < (\pi\tau)'(t_2) < \cdots < (\pi\tau)'(t_n),$$
$$(\pi\sigma)'(t_1) < (\pi\sigma)'(t_2) < \cdots < (\pi\sigma)'(t_n).$$

By the definition of the projection,

$$\pi\tau(t_1) < \pi\tau(t_2) < \cdots < \pi\tau(t_n),$$
$$\pi\sigma(t_1) < \pi\sigma(t_2) < \cdots < \pi\sigma(t_n).$$

Therefore,

$$a + 1 \le \pi\tau(t_{a+1}) < \pi\tau(t_{a+2}) < \cdots < \pi\tau(t_{n-a}) \le p + 1 - a,$$
$$a + 1 \le \pi\sigma(t_{a+1}) < \pi\sigma(t_{a+2}) < \cdots < \pi\sigma(t_{n-a}) \le p + 1 - a.$$

Since π acts trivially on $\{a + 1, a + 2, \ldots, p + 1 - a\}$, we conclude that

$$a + 1 \le \tau(t_{a+1}) < \tau(t_{a+2}) < \cdots < \tau(t_{n-a}) \le p + 1 - a, \tag{5.2}$$
$$a + 1 \le \sigma(t_{a+1}) < \sigma(t_{a+2}) < \cdots < \sigma(t_{n-a}) \le p + 1 - a. \tag{5.3}$$

Now we claim that there exists an s such that

(i) s, $s+1$, $s+2 \in \{a+1, a+2, \dots, n-a\}$;
(ii) $\tau(t_{s+2}) = \tau(t_{s+1}) + 1 = \tau(t_s) + 2$;
(iii) $\sigma(t_{s+2}) = \sigma(t_{s+1}) + 1 = \sigma(t_s) + 2$.

Let $s \in \{a+1, \dots, n-a-2\}$. If $\tau(t_{s+1}) > \tau(t_s) + 1$, then we have by (5.2) that $\tau(t_s)+1 = \tau(t)$ for some $t \notin \{t_{a+1}, t_{a+2}, \dots, t_{n-a}\}$. Note that $p+1-2a-(n-2a) = d$. Therefore, again by (5.2), the interval $[a+1, p+1-a]$ contains at most d integers not of the form $\tau(t_s)$, $s = a+1, a+2, \dots, n-a$. Hence, the inequality $\tau(t_{s+1}) > \tau(t_s)+1$ may hold for at most d values of s. Reasoning in the same way, we deduce from (5.2) or (5.3) that each of the inequalities $\tau(t_{s+2}) > \tau(t_{s+1}) + 1$, $\sigma(t_{s+1}) > \sigma(t_s) + 1$, and $\sigma(t_{s+2}) > \sigma(t_{s+1}) + 1$ holds for at most d values of s. Consequently, there are at most $4d$ values of s such that either (ii) or (iii) fails. On the other hand, by (5.1) and the choice of a, we have that

$$|\{a+1, a+2, \dots, n-a-2\}| = n - 2a - 2 > 16dr^3 - 2 > 4d.$$

Thus, a required s exists.

Put $u_1 = \tau(t_s)$, $u_2 = \sigma(t_s)$. Without loss of generality we may assume that $u_1 \geq u_2$; otherwise we change the role of τ and σ. Since any projectivity on a line is determined by images of any three points and $\tau\sigma^{-1}(u_2) = u_1$, $\tau\sigma^{-1}(u_2+1) = u_1+1$, and $\tau\sigma^{-1}(u_2+2) = u_1+2$, we have that $\tau\sigma^{-1}$ is induced by the shift $z \mapsto z+(u_1-u_2) \pmod p$. Hence $\tau(z) = \sigma(z) + (u_1 - u_2) \pmod p$.

On the other hand, $\tau(t_s) \leq p + 1 - n + s$ and $\sigma(t_s) \geq s$ by (5.2) and (5.3), respectively. Therefore, $0 \leq u_1 - u_2 = \tau(t_s) - \sigma(t_s) \leq p + 1 - n = d$.

Assume that $u_1 \neq u_2$. By the choice of r, we have $p_1 p_2 \cdots p_r \geq 2^r > d$. Hence, there exists at least one prime p_i which does not divide $u_1 - u_2$. Note that there are at most $d = p+1-n$ numbers h such that $\sigma^{-1}(h) \notin \{1, 2, \dots, n\}$. Consequently, among $d+2$ numbers $a_{4d(i-1)+j}$, $j = 1, 2, \dots, d+2$, there are at least two numbers, say, a_v and a_w, such that both $b_1 = \sigma^{-1}(a_v)$ and $b_2 = \sigma^{-1}(a_w)$ belong to $\{1, 2, \dots, n\}$. Since $\tau(b_i) = \sigma(b_i) + u_1 - u_2$, we have $a_{4d(i-1)+1} \leq \sigma(b_i) \leq \tau(b_i) \leq \sigma(b_i) + d \leq a_{4d(i-1)+d+2} + d \leq a_{4d(i-1)+4d}$. In particular, $\tau(b_1) = a_v + u_1 - u_2$ and $\tau(b_2) = a_w + u_1 - u_2$ do not coincide with any a_j, since the numbers $a_{4d(i-1)+1}, \dots, a_{4d(i-1)+4d}$ form an arithmetic progression with the difference equal to p_i. But in that case, $\pi\tau(b_1) < \pi\tau(b_2)$ if and only if $\pi\sigma(b_2) < \pi\sigma(b_1)$, a contradiction with the assumption $(\pi\tau)' = (\pi\sigma)'$. Hence $u_1 = u_2$ and $\tau = \sigma$. This completes the proof. □

6. Lower bounds. Consider a biased 4-restricted min-wise independent set $G \subseteq S_n$. For any distinct integers i, j, and l taken from $\{1, \dots, n\}$, we can evaluate the probability that $\pi(i) < \pi(j) < \pi(l)$. Namely,

$$\begin{aligned} \mathbf{Pr}_G(\pi(i) < \pi(j) < \pi(l)) &= \mathbf{Pr}_G(\pi(j) < \pi(l)) - \mathbf{Pr}_G(\min(\pi(i), \pi(j), \pi(l)) = \pi(j)) \\ &= \frac{1}{2} - \frac{1}{3} = \frac{1}{6}. \end{aligned} \tag{6.1}$$

Moreover, the same is true if G is only 3-restricted min-wise independent.

Let i, j, l, and m be four distinct integers taken from $\{1, \dots, n\}$ and

$$p_{ijlm} = \mathbf{Pr}_G(\pi(i) < \pi(j) < \pi(l) < \pi(m)).$$

In contrast with (6.1) we are not able to find p_{ijlm} exactly. However, there are some relations between them.

LEMMA 6.1. *If G is biased 4-restricted min-wise independent, then, for any distinct i, j, l, and m, we have $p_{ijlm} + p_{ijml} = \frac{1}{12}$ and $p_{ijlm} = p_{jiml}$.*

Proof. We start from the first relation:

$$\begin{aligned} p_{ijlm} + p_{ijml} &= \mathbf{Pr}_G(\min(\pi(j), \pi(l), \pi(m)) = \pi(j)) \\ &\quad - \mathbf{Pr}_G(\min(\pi(i), \pi(j), \pi(l), \pi(m)) = \pi(j)) \\ &= \frac{1}{3} - \frac{1}{4} = \frac{1}{12}. \end{aligned}$$

Now, applying the obtained relation to (i, l, j, m) and applying (6.1) to (i, m, l), we have

$$\begin{aligned} p_{ijlm} - p_{jiml} &= \mathbf{Pr}_G(\min\{\pi(i), \pi(j), \pi(l), \pi(m)\} = \pi(i)) \\ &\quad - \mathbf{Pr}_G(\pi(i) < \pi(m) < \pi(l)) - (p_{iljm} + p_{ilmj}) = \frac{1}{4} - \frac{1}{6} - \frac{1}{12} = 0, \end{aligned}$$

which completes the proof. □

Proof of Theorem 1.6. Let $\mathbf{a} = (a_1, a_2, a_3, a_4, a_5, a_6)$ be a complex vector whose coordinates will be specified later. For any $j = 3, \dots, n$, define random variables $V_j^{\mathbf{a}} : G \to \mathbb{C}$ as follows:

$$V_j^{\mathbf{a}}(\pi) = \begin{cases} a_1 \text{ if } \pi(1) < \pi(2) < \pi(j), \\ a_2 \text{ if } \pi(1) < \pi(j) < \pi(2), \\ a_3 \text{ if } \pi(2) < \pi(1) < \pi(j), \\ a_4 \text{ if } \pi(2) < \pi(j) < \pi(1), \\ a_5 \text{ if } \pi(j) < \pi(1) < \pi(2), \\ a_6 \text{ if } \pi(j) < \pi(2) < \pi(1). \end{cases}$$

By Lemma 6.1, we have, for any i, j $(2 < i \neq j \leq n)$,

$$\begin{array}{llll} p_{12ij} = \frac{1}{24} + \alpha_{ij}, & p_{12ji} = \frac{1}{24} - \alpha_{ij}, & p_{21ij} = \frac{1}{24} - \alpha_{ij}, & p_{21ji} = \frac{1}{24} + \alpha_{ij}, \\ p_{1i2j} = \frac{1}{24} + \beta_{ij}, & p_{1ij2} = \frac{1}{24} - \beta_{ij}, & p_{i12j} = \frac{1}{24} - \beta_{ij}, & p_{i1j2} = \frac{1}{24} + \beta_{ij}, \\ p_{1j2i} = \frac{1}{24} + \gamma_{ij}, & p_{1ji2} = \frac{1}{24} - \gamma_{ij}, & p_{j12i} = \frac{1}{24} - \gamma_{ij}, & p_{j1i2} = \frac{1}{24} + \gamma_{ij}, \\ p_{2i1j} = \frac{1}{24} + \delta_{ij}, & p_{2ij1} = \frac{1}{24} - \delta_{ij}, & p_{i21j} = \frac{1}{24} - \delta_{ij}, & p_{i2j1} = \frac{1}{24} + \delta_{ij}, \\ p_{2j1i} = \frac{1}{24} + \epsilon_{ij}, & p_{2ji1} = \frac{1}{24} - \epsilon_{ij}, & p_{j21i} = \frac{1}{24} - \epsilon_{ij}, & p_{j2i1} = \frac{1}{24} + \epsilon_{ij}, \\ p_{ij12} = \frac{1}{24} + \zeta_{ij}, & p_{ij21} = \frac{1}{24} - \zeta_{ij}, & p_{ji12} = \frac{1}{24} - \zeta_{ij}, & p_{ji21} = \frac{1}{24} + \zeta_{ij} \end{array}$$

for some parameters α_{ij}, β_{ij}, γ_{ij}, δ_{ij}, ϵ_{ij}, ζ_{ij}. A direct computation shows that

$$\mathbf{E}(V_j^{\mathbf{a}} V_j^{\mathbf{b}}) = \frac{a_1b_1 + a_2b_2 + a_3b_3 + a_4b_4 + a_5b_5 + a_6b_6}{6} \tag{6.2}$$

and, for $i \neq j$,

$$\begin{aligned} \mathbf{E}(V_i^{\mathbf{a}} V_j^{\mathbf{b}}) &= \frac{1}{24}\{(a_1 + a_2)(b_1 + b_2) + (a_1 + a_5)(b_1 + b_5) + (a_2 + a_5)(b_2 + b_5) \\ &\quad + (a_3 + a_4)(b_3 + b_4) + (a_3 + a_6)(b_3 + b_6) + (a_4 + a_6)(b_4 + b_6)\} \\ &\quad + \beta_{ij}(a_2 - a_5)(b_1 - b_2) + \gamma_{ij}(a_1 - a_2)(b_2 - b_5) \\ &\quad + \delta_{ij}(a_4 - a_6)(b_3 - b_4) + \epsilon_{ij}(a_3 - a_4)(b_4 - b_6). \end{aligned} \tag{6.3}$$

From now on, we restrict ourselves to vectors $\mathbf{a}$ and $\mathbf{b}$ that satisfy an additional requirement $a_2 = a_5$, $a_4 = a_6$, $b_2 = b_5$, and $b_4 = b_6$. In this case, extra terms in (6.3) depending on β_{ij}, γ_{ij}, δ_{ij}, and ϵ_{ij} vanish and $\mathbf{E}(V_i^{\mathbf{a}} V_j^{\mathbf{b}})$ becomes

$$\mathbf{E}(V_i^{\mathbf{a}} V_j^{\mathbf{b}}) = \frac{1}{12}\{(a_1 + a_2)(b_1 + b_2) + 2a_2b_2 + (a_3 + a_4)(b_3 + b_4) + 2a_4b_4\}. \tag{6.4}$$

The set $L(G)$ of all complex-valued random variables on G has the natural structure of a complex vector space. The dimension of this space is exactly $|G|$. Therefore, the cardinality of any linearly independent subset of $L(G)$ gives a lower bound for $|G|$. Thus, to prove the claim of the theorem it is sufficient to find $2n-2$ random variables $V_j^{\mathbf{a}}$ which are linearly independent over $\mathbb{C}$. We assert that the random variables $V_j^{(1)} = V_j^{(1,0,0,0,0,0)}$, $V_j^{(3)} = V_j^{(0,0,1,0,0,0)}$, $V^{(2)} = V_3^{(0,1,0,0,1,0)}$, $V^{(4)} = V_3^{(0,0,0,1,0,1)}$, where $j = 3, \dots, n$, are linearly independent. Let

$$S = \sum_{j=3}^{n} c_j^{(1)} V_j^{(1)} + c^{(2)} V^{(2)} + \sum_{j=3}^{n} c_j^{(3)} V_j^{(3)} + c^{(4)} V^{(4)}$$

and assume that

$$S = 0. \tag{6.5}$$

Let $Y = \sum_{i=3}^{n} V_i^{(n-3,-n+1,0,0,-n+1,0)}$. A direct computation based on (6.2), (6.4) shows that

$$\mathbf{E}(SY) = -c^{(2)} \frac{(n+1)(n-2)}{6}.$$

Since $n \geq 4$, assumption (6.5) implies $c^{(2)} = 0$. By a similar reasoning, $c^{(4)} = 0$ since, for $Z = \sum_{i=3}^{n} V_i^{(0,0,n-3,-n+1,0,-n+1)}$, we have

$$\mathbf{E}(SZ) = -c^{(4)} \frac{(n+1)(n-2)}{6}.$$

In addition,

$$\mathbf{E}(SV_j^{(1,-1,0,0,-1,0)}) = \begin{cases} \frac{1}{6} c_j^{(1)} - \frac{1}{6} c^{(2)} \text{ if } j \neq 3, \\ \frac{1}{6} c_j^{(1)} - \frac{1}{3} c^{(2)} \text{ if } j = 3, \end{cases}$$

$$\mathbf{E}(SV_j^{(0,0,1,-1,0,-1)}) = \begin{cases} \frac{1}{6} c_j^{(3)} - \frac{1}{6} c^{(4)} \text{ if } j \neq 3, \\ \frac{1}{6} c_j^{(3)} - \frac{1}{3} c^{(4)} \text{ if } j = 3. \end{cases}$$

Since $c^{(2)} = c^{(4)} = 0$, assumption (6.5) implies $c_j^{(1)} = c_j^{(3)} = 0$. Hence, the above family of random variables is linearly independent. This completes the proof. □

Remark 3. Let us note that

$$V_j^{(1,0,0,0,0,0)} + V_j^{(0,1,0,0,1,0)} = \begin{cases} 1 \text{ if } \pi(1) < \pi(2), \\ 0 \text{ if } \pi(2) < \pi(1), \end{cases}$$

$$V_j^{(0,0,1,0,0,0)} + V_j^{(0,0,0,1,0,1)} = \begin{cases} 0 \text{ if } \pi(1) < \pi(2), \\ 1 \text{ if } \pi(2) < \pi(1), \end{cases}$$

i.e., both of these sums do not depend on j. Therefore, for any vector $\mathbf{a}$ that satisfies additional requirements $a_2 = a_5$, $a_4 = a_6$, the random variable $V_j^{\mathbf{a}}$ can be expressed as linear combination of $V_j^{(1)}$, $V_3^{(1)}$, $V^{(2)}$, $V_j^{(3)}$, $V_3^{(3)}$, and $V^{(4)}$. In particular, the dimension of the space spanned by all such $V_j^{\mathbf{a}}$'s is exactly $2n-2$.

Acknowledgments. I am very grateful to C. Franci, E. A. Hirsch, L. Giuzzi, S. V. Ivanov, and M. C. Tamburini for many helpful discussions. My special thanks to the Catholic University of Brescia (Italy), where this paper was completed. I would also like to thank the anonymous referees for their comments and suggestions, which improved the quality of the presentation.

REFERENCES

[1] R. Baer, *Linear Algebra and Projective Geometry*, Academic Press, New York, 1952.
[2] T. Bohman, C. Cooper, and A. Frieze, *Min-wise independent linear permutations*, Electron. J. Combin., 7 (2000), research paper 26.
[3] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher, *Min-wise independent permutations*, J. Comput. System Sci., 60 (2000), pp. 630–659.
[4] A. Z. Broder, M. Charikar, and M. Mitzenmacher, *A derandomization using min-wise independent permutations*, in Randomization and Approximation Techniques in Computer Science, Barcelona, 1998, Lecture Notes in Comput. Sci. 1518, Springer-Verlag, Berlin, 1998, pp. 15–24.
[5] P. J. Cameron, *Permutation Groups*, London Math. Soc. Stud. Texts 45, Cambridge University Press, Cambridge, 1999.
[6] J. D. Dixon and B. Mortimer, *Permutation Groups*, Grad. Texts in Math. 163, Springer-Verlag, New York, 1996.
[7] C. Franchi and M. Vsemirnov, *Min-wise independent groups*, European J. Combin., 24 (2003), pp. 855–875.
[8] D. R. Heath-Brown and H. Iwaniec, *On the difference between consecutive primes*, Bull. Amer. Math. Soc. (N.S.), 1 (1979), pp. 758–760.
[9] D. R. Heath-Brown and H. Iwaniec, *On the difference between consecutive primes*, Invent. Math., 55 (1979), pp. 49–69.
[10] T. Itoh, Y. Takei, and J. Tarui, *On permutations with limited independence*, in Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, pp. 137–146.
[11] A. I. Kostrikin and Y. I. Manin, *Linear Algebra and Geometry*, Moscow State University Publishers, Moscow, 1980 (in Russian). English translation: Gordon and Breach, Amsterdam, 1997.
[12] A. López and E. Nart, *Classification of Goppa codes of genus zero*, J. Reine Angew. Math., 517 (1999), pp. 131–144.
[13] M. Luby and A. Wigderson, *Pairwise Independence and Derandomization*, Technical Report TR-95-035, International Computer Science Institute, 1995.
[14] S. Norin, *A polynomial lower bound for the size of k-min-wise independent set of permutations*, Zap. Nauchn. Sem. S.-Petersburg. Otdel. Mat. Inst. Steklov. (POMI), 277 (2001), pp. 104–116 (in Russian). English translation: J. Math. Sci. (N.Y.), 118 (2003), pp. 4994–5000.
[15] P. Ribenboim, *The New Book of Prime Number Records*, Springer-Verlag, New York, 1995.
[16] M. A. Vsemirnov, E. A. Hirsch, E. Y. Dantsin, and S. V. Ivanov, *Algorithms for SAT and upper bounds for their complexity*, Zap. Nauchn. Sem. S.-Petersburg. Otdel. Mat. Inst. Steklov. (POMI), 277 (2001), pp. 14–46 (in Russian). English translation: J. Math. Sci. (N.Y.), 118 (2003), pp. 4948–4962; also available as ECCC technical report via ftp://ftp.eccc.uni-trier.de/pub/eccc/reports/2001/TR01-012/index.html.

SIAM J. DISCRETE MATH.
Vol. 18, No. 3, pp. 608–625

A LINEAR PROGRAMMING FORMULATION AND APPROXIMATION ALGORITHMS FOR THE METRIC LABELING PROBLEM*

C. CHEKURI†, S. KHANNA‡, J. NAOR§, AND L. ZOSIN¶

Abstract. We consider approximation algorithms for the metric labeling problem. This problem was introduced in a paper by Kleinberg and Tardos [*J. ACM*, 49 (2002), pp. 616–630] and captures many classification problems that arise in computer vision and related fields. They gave an $O(\log k \log\log k)$ approximation for the general case, where k is the number of labels, and a 2-approximation for the uniform metric case. (In fact, the bound for general metrics can be improved to $O(\log k)$ by the work of Fakcheroenphol, Rao, and Talwar [*Proceedings of the* 35*th Annual ACM Symposium on Theory of Computing*, 2003, pp. 448–455].) Subsequently, Gupta and Tardos [*Proceedings of the* 32*nd Annual ACM Symposium on the Theory of Computing*, 2000, pp. 652–658] gave a 4-approximation for the truncated linear metric, a metric motivated by practical applications to image restoration and visual correspondence. In this paper we introduce an integer programming formulation and show that the integrality gap of its linear relaxation either matches or improves the ratios known for several cases of the metric labeling problem studied until now, providing a unified approach to solving them. In particular, we show that the integrality gap of our linear programming (LP) formulation is bounded by $O(\log k)$ for a general k-point metric and 2 for the uniform metric, thus matching the known ratios. We also develop an algorithm based on our LP formulation that achieves a ratio of $2+\sqrt{2} \simeq 3.414$ for the truncated linear metric improving the earlier known ratio of 4. Our algorithm uses the fact that the integrality gap of the LP formulation is 1 on a linear metric.

Key words. metric labeling, linear program, approximation algorithm, truncated linear metric

AMS subject classifications. 68Q25, 68W25, 90C59

DOI. 10.1137/S0895480101396937

1. Introduction.

Motivated by certain classification problems that arise in computer vision and related fields, Kleinberg and Tardos introduced the metric labeling problem [26]. In a typical classification problem, one wishes to assign labels to a set of objects so as to optimize some measure of the quality of the labeling. The metric labeling problem captures a broad range of classification problems where the quality of a labeling depends on the pairwise relations between the underlying set of objects. More precisely, the task is to classify a set V of n objects by assigning to each object a label from a set L of labels. The pairwise relationships between the objects are represented by a weighted graph $G = (V, E)$, where $w(u, v)$ represents the strength

*Received by the editors October 23, 2001; accepted for publication (in revised form) July 12, 2004; published electronically April 8, 2005. A preliminary version of this paper appeared in *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, Washington, DC, 2001, pp. 109–118.

http://www.siam.org/journals/sidma/18-3/39693.html

†Bell Labs, Lucent Technologies, 600 Mountain Ave., Murray Hill, NJ 07974 (chekuri@research.bell-labs.com).

‡Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104 (sanjeev@cis.upenn.edu). The research of this author was supported in part by an Alfred P. Sloan Research Fellowship.

§Computer Science Department, Technion, Haifa 32000, Israel (naor@cs.technion.ac.il). Most of this work was done while the author was at Bell Labs, Lucent Technologies, 600 Mountain Ave., Murray Hill, NJ 07974. The research of this author was supported in part by US-Israel BSF grant 2002276 and by EU contract IST-1999-14084 (APPOL II).

¶Bear Stearns & Company, One Federal St., 29th Floor, Boston, MA 02110 (LZosin@Bear.com). This work was done while the author was at NEC Research Institute, Princeton, NJ 08540.

of the relationship between u and v. The objective is to find a labeling, a function $f : V \to L$, that maps objects to labels, where the cost of f, denoted by $Q(f)$, has two components.

- For each $u \in V$, there is a nonnegative assignment cost $c(u, i)$ to label u with i. This cost reflects the relative likelihood of assigning labels to u.
- For each pair of objects u and v, the edge weight $w(u, v)$ measures the strength of their relationship. This models the assumption that strongly related objects should be assigned labels that are *close*. This is modeled in the objective function by the term $w(u, v) \cdot d(f(u), f(v))$, where $d(\cdot, \cdot)$ is a distance function on the labels L.

Thus

$$Q(f) = \sum_{u \in V} c(u, f(u)) + \sum_{(u,v) \in E} w(u, v) \cdot d(f(u), f(v)),$$

and the goal is to find a labeling of minimum cost. In the metric labeling problem, the distance function d is assumed to be a metric. We remark that if the distance function d is allowed to be arbitrary, the graph coloring problem can be reduced to the labeling problem.

A prototypical application of the metric labeling problem is the *image restoration* problem in computer vision [4, 5, 6]. In the image restoration problem, the goal is to take an image corrupted by noise and restore it to its "true" version. The image consists of pixels and these are the objects in the classification problem. Each pixel has an integer intensity value associated with it that is possibly corrupted and we would like to restore it to its true value. Thus the labels correspond to intensity values, and the goal is to assign a new intensity to each pixel. Further, neighboring pixels are assumed to be close to each other, since intensities typically change smoothly. Thus, neighboring pixels have an edge between them with a positive weight (assume a uniform value for concreteness). The original or observed intensities are assumed to be close to true values.[1] The assignment cost is some positive cost associated with changing the intensity from its original value to a new value; the larger the change, the larger the cost. To see how the cost function models the restoration, consider a pixel corrupted by noise as a result of which its observed intensity is very different from its neighboring pixels. By changing the corrupted pixel's intensity we incur a cost of relabeling, but that can be offset by the edge cost saved by being closer to its neighbors. The assignment costs weigh the labeling in favor of the original values since most of the intensities are likely to be correct (see also the footnote).

The metric labeling problem naturally arises in other applications in image processing and computer vision. Researchers in these fields have developed a variety of good heuristics that use classical combinatorial optimization techniques, such as network flow and local search [4, 21, 32, 17, 14, 12].

Kleinberg and Tardos [26] formalized the metric labeling problem and its connections to Markov random fields and other classification problems. (See [26] for a

[1] This assumption is based on the connection of the labeling problem to the theory of Markov random fields (MRFs) [27]. In this theory, the observed data or labeling of the objects is assumed to be obtained from a true labeling by adding independent random noise. The idea is to decide the most probable labeling given the observed data. An MRF can be defined by a graph on the objects with edges indicating dependencies between objects. Under the assumption that the probability distribution of an object's label depends only on its neighbors' labels, and if the MRF satisfies two standard assumptions of homogeneity and pairwise interactions, the labeling problem can be restated as the problem of finding a labeling f that maximizes the a posteriori probability $\mathbf{Pr}[f|f']$, where f' is the observed labeling. See [26] for more details on the connection of metric labeling to MRFs.

thorough description of the various connections.) Metric labeling also has rich connections to some well-known problems in combinatorial optimization. It is related to the quadratic assignment problem, an extensively studied problem in operations research. A special case of metric labeling is the 0-extension problem, studied by Karzanov [24, 25]. There are no assignment costs in this problem; however, the graph contains a set of terminals, $t_1, \ldots, t_k$, where the label of terminal t_i is fixed in advance to i and the nonterminals are free to be assigned to any of the labels. The 0-extension problem generalizes the well-studied multiway cut problem [13, 7, 23]. Karzanov [24, 25] showed that certain special cases (special metrics) of the 0-extension problem can be solved optimally in polynomial time. Another special case of the metric labeling problem is the *task assignment problem* in distributed computing [28]. Here, tasks of a modular program need to be assigned to the processors of a distributed system, while balancing between task execution cost and intertask communication cost.

Kleinberg and Tardos [26] obtained an $O(\log k \log\log k)$ approximation for the general metric labeling problem, where k denotes the number of labels in L, and a 2-approximation for the uniform metric. The approximation for general metrics improves to $O(\log k)$ by a recent result of Fakcheroenphol, Rao, and Talwar [16]. Metric labeling is Max-SNP hard; this follows from the Max-SNP hardness of the multiway cut problem which is a special case. Given the rich connections of this problem to other well-studied optimization problems and a variety of applications, a natural interesting question is to determine the approximability of the general problem as well as of the important special cases that arise in practice.

The truncated linear metric. Gupta and Tardos [18] considered the metric labeling problem for the truncated linear metric, a special case motivated by its direct applications to two problems in computer vision, namely, *image restoration* and *visual correspondence*. We briefly describe the application of this metric to the image restoration problem discussed earlier; see [18] for more details. Consider the case of gray scale images where the intensities are integer values. Our earlier description assumed that intensities of neighboring pixels should be similar since the image is typically smooth. This motivates a linear-like metric on the labels where the distance between two intensities i and j is $|i-j|$. However, at object boundaries (here we are referring to objects in the image) sharp changes in intensities happen. Thus, for the metric to be robust, neighboring pixels that are actually at object boundaries (and hence naturally differ in their intensities by large amounts) should not be penalized by arbitrarily large quantities. This motivated Gupta and Tardos to consider a truncated linear metric, where the distance between two intensities i and j is given by $d(i,j) = \min(M, |i-j|)$. Thus, the penalty is truncated at M; this is a natural (and nonuniform) metric for the problem at hand. A very similar reasoning applies to the visual correspondence problem, where the objective is to compare two images of the same scene for disparity. The labels here correspond to depth of the point in the image from the camera. For the truncated linear metric a 4-approximation algorithm was given in [18] using local search. The local moves in the algorithm make use of the flow network used in [4, 21], which gives an optimal solution to the linear metric case in polynomial time.

For the image restoration application, other distance functions have also been studied. In particular, the quadratic distance function $d(i,j) = |i-j|^2$ and its truncated version $d(i,j) = \min\{M, |i-j|^2\}$ have been considered (see [27] and [22]). Unfortunately, neither of these distance functions is a metric, and hence the algorithms for metric labeling problem cannot be used. However, as we discuss shortly,

we are able to provide nontrivial approximation ratios for them.

Results. In this paper we address the problem of obtaining improved approximation guarantees for the metric labeling problem. Kleinberg and Tardos [26] pointed out the difficulty of the general case as having to do with the absence of a "natural" integer programming (IP) formulation for the problem. They give an IP formulation for tree metrics and use Bartal's probabilistic tree approximations [2, 3] to reduce the problem with an arbitrary metric to that with a tree metric. In this work we give a natural IP formulation for the general problem. An advantage of our formulation is that it is applicable even to distance functions that are not metrics, for example, the quadratic distance function mentioned above. We substantiate the strength of this formulation by deriving both known results and new results using its linear relaxation. In particular, we show the following results on the integrality gap of our formulation:

- $O(\log k)$ for general metrics and a factor 2 for the uniform metric,
- 1 for the linear metric and distances on the line defined by convex functions (not necessarily metrics),
- $2 + \sqrt{2} \simeq 3.414$ for the truncated linear metric, and
- $O(\sqrt{M})$ for the truncated quadratic distance function.

The integrality gaps we show for our formulation either match or improve on the best previous approximation ratios for most of the cases known to us. (In addition to the above, we can also show that if G is a tree the integrality gap is 1.) Our formulation allows us to present these results in a unified fashion. In the process we improve the 4 approximation of Gupta and Tardos [18] for the important case of the truncated linear metric. We also show a reduction from the case with arbitrary assignment costs $c(u, i)$ to the case where $c(u, i) \in \{0, \infty\}$ for all u and i. The reduction preserves the graph G and the optimal solution but increases the size of the label space from k labels to nk labels. We also describe an alternative reduction of Chuzhoy [10] that preserves the label space but alters the graph to one with $n + nk$ vertices. We believe that our results and techniques are a positive step toward obtaining improved bounds for both the general metric labeling problem and interesting special cases.

Calinescu, Karloff, and Rabani [8] considered approximation algorithms for the 0-extension problem. They considered a linear programming (LP) formulation (which they called the *metric relaxation*) originally studied by Karzanov [24], where they associated a length function with every edge of the graph and required that (i) the distance between terminals t_i and t_j, for $1 \le i, j \le k$, is at least $d(i, j)$, and (ii) the length function is a semimetric. We note that their formulation does not apply to the general metric labeling problem. They obtained an $O(\log k)$-approximation algorithm for the 0-extension problem using this formulation and an $O(1)$-approximation for planar graphs. Our LP formulation, when specialized to the 0-extension problem, induces a feasible solution for the metric relaxation formulation, by defining the length of an edge to be its transshipment cost (see section 2). It is not hard to verify that this length function is a semimetric. Calinescu, Karloff, and Rabani [8] also showed a gap of $\Omega(\sqrt{\log k})$ on the integrality ratio of their formulation. Their lower bound proof does not seem to carry over in any straightforward way to our formulation. We note that the metric relaxation formulation optimizes over the set of all semimetrics, while our formulation optimizes only over a subset of the semimetrics. Whether our formulation is strictly stronger than the metric relaxation of [8] is an interesting open problem.

Outline. Section 2 describes our LP formulation for the general metric labeling problem. In section 3, we analyze our formulation for uniform and linear metrics.

Building on our rounding scheme for the linear metric, we design and analyze a rounding procedure for the truncated linear metric in section 4. In section 5, we study the general metric labeling problem and show that the integrality gap of our formulation is bounded by $O(\log k)$. We also describe here a transformation that essentially eliminates the role of the label cost assignment function.

2. The LP formulation. We present a new linear integer programming formulation of the metric labeling problem. Let $x(u,i)$ be a $\{0,1\}$-variable indicating that vertex u is labeled i. Let $x(u,i,v,j)$ be a $\{0,1\}$-variable indicating that for edge $(u,v) \in E$, vertex u is labeled i and vertex v is labeled j. See Figure 2.1 for the formulation.

Constraints (2.1) simply express that each vertex must receive some label. Constraints (2.2) force consistency in the edge variables: if $x(u,i) = 1$ and $x(v,j) = 1$, they force $x(u,i,v,j)$ to be 1. Constraints (2.3) express the fact that (u,i,v,j) and (v,j,u,i) refer to the same edge; the redundancy helps in notation. We obtain a linear relaxation of the above program by allowing the variables $x(u,i)$ and $x(u,i,v,j)$ to take any nonnegative value. We note that equality in (2.2) is important for the linear relaxation.

With each edge $(u,v) \in E$ we associate a complete bipartite graph $H(u,v)$. The vertices of $H(u,v)$ are $\{u_1,\ldots,u_k\}$ and $\{v_1,\ldots,v_k\}$, i.e., they represent all possible labelings of u and v. There is an edge (u_i,v_j) connecting the pair of vertices u_i and v_j, $1 \le i,j \le k$. In what follows, we refer to the edges of $H(u,v)$ as *links* to distinguish them from the edges of G. Suppose that the value of the variables $x(u,i)$ for all u and i has been determined. For an edge $(u,v) \in E$, we can interpret the variables $x(u,i,v,j)$ from a flow perspective. The contribution of edge $(u,v) \in E$ to the objective function of the linear program is the cost of the optimal transshipment of flow between $\{u_1,\ldots,u_k\}$ and $\{v_1,\ldots,v_k\}$, where (i) the supply of u_i is $x(u,i)$ and the demand of v_j is $x(v,j)$ for $0 \le i,j \le k$; (ii) the cost of shipping a unit flow from u_i to v_j is $d(i,j)$. (The choice of the supply side and the demand side is arbitrary.)

For the rest of the paper, the quantity $d_{LP}(u,v)$ refers to the LP distance between u and v and is defined to be the transshipment cost $\sum_{i,j} d(i,j) \cdot x(u,i,v,j)$. The LP

$$\text{(I)} \quad \min \sum_{u \in V} \sum_{i=1}^{k} c(u,i) \cdot x(u,i) + \sum_{(u,v) \in E} w(u,v) \sum_{i=1}^{k} \sum_{j=1}^{k} d(i,j) \cdot x(u,i,v,j)$$

subject to

$$\sum_{i=1}^{k} x(u,i) = 1 \qquad \forall\, v \in V \tag{2.1}$$

$$\sum_{j=1}^{k} x(u,i,v,j) - x(u,i) = 0 \qquad \forall\, u \in V,\ (u,v) \in E,\ i \in 1,\ldots,k, \tag{2.2}$$

$$x(u,i,v,j) - x(v,j,u,i) = 0 \qquad \forall\, u,v \in V,\ i,j \in 1,\ldots,k, \tag{2.3}$$

$$x(u,i) \in \{0,1\} \qquad \forall\, u \in V,\ i \in 1,\ldots,k, \tag{2.4}$$

$$x(u,i,v,j) \in \{0,1\} \qquad \forall\, (u,v) \in E,\ i,j \in 1,\ldots,k. \tag{2.5}$$

FIG. 2.1. *IP formulation.*

distance derived from an optimal (fractional) solution to (I) induces a metric on the graph, since for any $v_1, v_2, v_3 \in V$, the transshipment cost from v_1 to v_2 cannot be more than the sum of the transshipment costs from v_2 to v_3 and from v_3 to v_1. The transshipment problem between two distributions was introduced by Monge [29] and is also referred to as the Monge–Kantorovich mass transference problem and has several applications [31]. In the image processing literature [33, 30] this metric has also been referred to as the *earth mover's metric*.

A solution to the formulation has an interesting *geometric* interpretation. It defines an embedding of the graph into a k-dimensional simplex, where the distance between points in the simplex is defined by the earth mover's metric on the labels. Our formulation specializes to that of Kleinberg and Tardos [26] for the uniform metric case which in turn specializes to that of Călinescu, Karloff, and Rabani [7] for the multiway cut problem where the distance between points in the embedding is simply their ℓ_1 distance.

3. Uniform metrics and linear metrics. We now analyze the performance of our linear programming formulation on two natural special cases, namely, uniform metrics and linear metrics. Kleinberg and Tardos [26] showed a 2-approximation for the uniform metric case. Their approach is based on rounding the solution of a linear program formulated specifically for uniform metrics. We will show that our linear programming formulation dominates the one in [26] and thus also has an integrality gap of at most 2. For the case of linear metrics, Boykov, Veksler, and Zabih [4] and Ishikawa and Geiger [21] obtained exact algorithms by reducing the problem to a minimum $\{s, t\}$-cut computation. We show here that on linear metrics, our linear programming formulation gives an exact algorithm as well. Our analysis for the linear metric case plays an important role in the algorithm for the truncated linear metric case.

3.1. The uniform metric case. Kleinberg and Tardos [26] formulated a linear program, denoted (KT), for the uniform metric and gave the following iterative algorithm for rounding a solution to it. Initially, no vertex is labeled. Each iteration consists of the following steps: (i) choose a label uniformly at random from $1, \ldots, k$ (say, a label i); (ii) choose a real threshold θ uniformly at random from $[0, 1]$; (iii) for all unlabeled vertices $u \in V$, u is labeled i if $\theta \leq x(u, i)$. The algorithm terminates when all vertices are labeled. Kleinberg and Tardos [26] showed that the expected cost of a labeling obtained by this algorithm is at most twice the cost of the LP solution.

$$\text{(KT)} \quad \min \sum_{v \in V} \sum_{i=1}^{k} c(u, i) \cdot x(u, i) + \sum_{(u,v) \in E} w(u, v) \cdot \frac{1}{2} \sum_{i=1}^{k} |x(u, i) - x(v, i)|$$

subject to

$$\sum_{i=1}^{k} x(u, i) = 1 \quad \forall\, u \in V,$$

$$x(u, i) \geq 0 \quad \forall\, u \in V \text{ and } i \in 1, \ldots, k.$$

We show that applying the rounding algorithm to an optimal solution obtained from linear program (I) yields the same approximation factor. Let $\bar{x}$ be a solution to (I). Note that for both (I) and (KT) the variables $x(u, i)$ completely determine the cost. We will show that cost of (KT) on $\bar{x}$ is smaller than that of (I). Both linear

programs (I) and (KT) coincide regarding the labeling cost. Consider edge $(u, v) \in E$. We show that the contribution of (u, v) to the objective function of (I) is at least as large as the contribution to the objective function of (KT).

$$\begin{aligned}
\frac{1}{2} \cdot \sum_{i=1}^{k} |\bar{x}(u,i) - \bar{x}(v,i)| &= \frac{1}{2} \cdot \sum_{i=1}^{k} \left| \sum_{j=1}^{k} \bar{x}(u,i,v,j) - \sum_{j=1}^{k} \bar{x}(v,i,u,j) \right| \\
&\leq \frac{1}{2} \cdot \sum_{i=1}^{k} \left| \sum_{j=1,\, j\neq i}^{k} \bar{x}(u,i,v,j) + \sum_{j=1,\, j\neq i}^{k} \bar{x}(v,i,u,j) \right| \\
&= \frac{1}{2} \cdot \sum_{i=1}^{k} \sum_{j=1,\, j\neq i}^{k} 2 \cdot \bar{x}(u,i,v,j) \\
&= \sum_{i=1}^{k} \sum_{j=1}^{k} d(i,j) \cdot \bar{x}(u,i,v,j).
\end{aligned}$$

The penultimate equality in the above set of equations is true since $\bar{x}(u,i,v,j) = \bar{x}(v,j,u,i)$. The final equality follows from the fact that $d(i,i) = 0$ and $d(i,j) = 1$, $i \neq j$. The last term is the contribution of (u, v) to the objective function of (I). We note that the example used in [26] to show that the integrality gap of (KT) is at least $2 - 1/k$ can be used to show the same gap for (I) as well.

3.2. The line metric case. We now turn to the case of a linear metric and show that the value of an integral optimal solution is equal to the value of a fractional optimal solution of the LP (I). Without loss of generality, we can assume that the labels of the metric are integers $1, 2, \ldots, k$. If the label set contains nonconsecutive integers, we can add all the "missing" intermediate integers to the label set and set the cost of assigning them to every vertex to be infinite. In fact, the rounding can be generalized to the case where the labels are arbitrary points on the real line without any difficulty.

Rounding procedure. Let $\bar{x}$ be an optimal fractional solution to the linear program. We round the fractional solution as follows. Let θ be a real threshold chosen uniformly at random from $[0, 1]$. For all i, $1 \leq i \leq k$, let

$$\alpha(u,i) = \sum_{j=1}^{i} \bar{x}(u,j).$$

Each vertex $u \in V$ is labeled by the unique label i that satisfies $\alpha(u, i-1) < \theta \leq \alpha(u,i)$. Clearly all vertices are labeled since $\alpha(u,k) = 1$.

Analysis. We analyze the expected cost of the assignment produced by the rounding procedure. For each vertex $u \in V$, let $L(u)$ be a random variable whose value is the label assigned to u by the rounding procedure. It can be readily verified that the probability that $L(u) = i$ is equal to $\bar{x}(u,i)$. This means that the expected labeling cost of vertex v is equal to $\sum_{i=1}^{k} \bar{x}(u,i) \cdot c(u,i)$, which is precisely the assignment cost of u in the linear program (with respect to solution $\bar{x}$). We now fix our attention on the expected cost of the edges.

LEMMA 3.1. *Consider edge $(u, v) \in E$. Then,*

$$\mathbf{E}\left[d((L(u), L(v))\right] = \sum_{i=1}^{k} |\alpha(u,i) - \alpha(v,i)|.$$

Proof. For the sake of the analysis we define auxiliary binary random variable $Z_1, \ldots, Z_{k-1}$ as follows. The variable Z_i is 1 if $\min\{L(u), L(v)\} \leq i$ and $\max\{L(u), L(v)\} > i$, and is 0 otherwise. In other words Z_i is 1 if i is in the interval defined by $L(u)$ and $L(v)$. It is easy to see that

$$d(L(u), L(v)) = \sum_{i=1}^{k-1} Z_i.$$

Therefore $\mathbf{E}[d(L(u), L(v))] = \sum_{i=1}^{k-1} \mathbf{E}[Z_i]$. We claim that

$$\mathbf{E}[Z_i] = \mathbf{Pr}[Z_i = 1] = |\alpha(u, i) - \alpha(v, i)|.$$

The lemma easily follows from this claim. To prove the claim, assume without loss of generality (w.l.o.g.) that $\alpha(u, i) \geq \alpha(v, i)$. If $\theta < \alpha(v, i)$, it is clear that $L(u), L(v) \leq i$, and if $\theta > \alpha(u, i)$, then $L(u), L(v) > i$: in both cases $Z_i = 0$. If $\theta \in (\alpha(v, i), \alpha(u, i)]$, then $L(u) \leq i$ and $L(v) > i$, which implies that $Z_i = 1$. Thus $\mathbf{Pr}[Z_i = 1]$ is exactly $|\alpha(u, i) - \alpha(v, i)|$. ∎

We now estimate the contribution of an edge $(u, v) \in E$ to the objective function of the linear program. As indicated in section 2, the contribution is equal to the cost of the optimal transshipment cost of the flow in the complete bipartite graph $H(u, v)$ between $\{u_1, \ldots, u_k\}$ and $\{v_1, \ldots, v_k\}$, where the supply of u_i is $\bar{x}(u, i)$ and the demand of v_j is $\bar{x}(v, j)$ for $1 \leq i, j \leq k$. Recall that $d_{LP}(u, v) = \sum_{i,j} d(i, j) \cdot \bar{x}(u, i, v, j)$.

LEMMA 3.2. *For the line metric*

$$d_{LP}(u, v) \geq \sum_{i=1}^{k} |\alpha(u, i) - \alpha(v, i)|.$$

Proof. The crucial observation in the case of a linear metric is that flow can be *uncrossed*. Let $i \leq i'$ and $j \leq j'$. Suppose that ε amount of flow is sent from u_i to $v_{j'}$ and from $u_{i'}$ to v_j. Then, uncrossing the flow, i.e., sending ε amount of flow from u_i to v_j and from $u_{i'}$ to $v_{j'}$ will not increase the cost of the transshipment. This means that the amount of flow sent from $\{u_1, \ldots, u_i\}$ to $\{v_1, \ldots, v_i\}$ for all i, $1 \leq i \leq k$, is precisely $\min\{\alpha(u, i), \alpha(v, i)\}$. Therefore, $|\alpha(v, i) - \alpha(u, i)|$ amount of flow is sent "outside" the label set $1, 2, \ldots, i$ and can be charged one unit of cost (with respect to i). Applying this argument to all values of i, $1 \leq i \leq k$, we get that the cost of the *optimal* transshipment of flow is precisely $\sum_{i=1}^{k} |\alpha(u, i) - \alpha(v, i)|$. The lemma follows. ∎

Hence, together with Lemma 3.1, we get that the expected cost of edge $(u, v) \in E$ after the rounding is no more than its contribution to the objective function of the linear program.

The uncrossing of flow in the proof of Lemma 3.2 relies on the Monge property of the distances induced by points on a line. Hoffman [20], in his classical paper, pointed out that the Monge property can be exploited for transportation and related problems.

THEOREM 3.3. *The integrality gap of LP* (I) *for the line metric is* 1.

3.3. Distance functions on the line defined by convex functions. We now consider distance functions on the labels $1, \ldots, k$ on the integer line defined by strictly convex functions, that is, $d(i, j) = f(|i - j|)$, where f is convex and nondecreasing. Notice that such distance functions do not satisfy the metric property, since d is a

metric if and only if f is concave and increasing. Our motivation for studying these metrics comes from the quadratic function $d(i,j) = |i-j|^2$, which is of particular interest in the image restoration application [22, 27] described earlier. We note that for the special case where the assignment cost is also a convex function (of the label), efficient algorithms are given by [19]. We can show the following.

THEOREM 3.4. *For any distance function on the line defined by a convex function, the integrality gap of LP* (I) *is* 1.

We sketch the proof, since it is similar to the linear case. Consider a feasible solution $\bar{x}$ to the LP. For any edge (u,v), if f is convex, the optimal cost transshipment flow in $H(u,v)$ is *noncrossing*. Further, for the rounding that we described for the linear case, if the flow is noncrossing, $\mathbf{Pr}\left[L(u) = i \wedge L(v) = j\right] = \bar{x}(u,i,v,j)$. The theorem follows from this last fact trivially. Ishikawa and Geiger [22] showed that the flow graph construction for the line metric can be extended for convex functions to obtain an optimal solution. The advantage of our approach is that the solution is obtained from a general formulation. This allows us to extend the ideas to obtain the first nontrivial approximation for the truncated quadratic distance function.

4. Improved approximation for the truncated line metric. In this section we use LP (I) to give a $(2+\sqrt{2})$-approximation algorithm for the truncated line metric case. This improves the 4-approximation provided in [18]. We prefer to view the metric graph as a line with the truncation to M being implicit. This allows us to use, with appropriate modifications, ideas from section 3.2. We round the fractional solution $\bar{x}$ once again using only the values $\bar{x}(u,i)$. Let $M' \geq M$ be an integer parameter that we will fix later. We repeat the following iteration until all vertices are assigned a label:

- Pick an integer ℓ uniformly at random in $[-M'+2, k]$. Let I_ℓ be the interval $[\ell, \ell+M'-1]$.
- Pick a real threshold θ uniformly at random from $[0,1]$.
- Let u be an unassigned vertex. If there is a label $i \in I_\ell$ such that

$$\sum_{j=\ell}^{i-1} \bar{x}(u,j) < \theta \leq \sum_{j=\ell}^{i} \bar{x}(u,j),$$

we assign i to u. Otherwise u is unassigned in this iteration.

The above algorithm generalizes the rounding algorithms for the uniform and line metrics in a natural way. Once the index ℓ is picked, the rounding is similar to that of the line metric in the interval I_ℓ. The difference is that a vertex might not get a label in an iteration. If two vertices u and v get separated in an iteration, that is, only one of them gets assigned, our analysis will assume that their distance is M. This is similar to the analysis in [26] for the uniform metric case. Our improvement comes from a careful analysis of the algorithm that treats links differently, based on whether their distance is linear or truncated. The analysis guides the choice of M' to obtain the best guarantee.

Let $L(u)$ and $L(v)$ be random variables that indicate the labels that get assigned to u and v by the algorithm.

LEMMA 4.1. *In any given iteration, the probability of an unassigned vertex u getting a label i in that iteration is exactly $\bar{x}(u,i)\cdot M'/(k+M'-1)$. The probability of u getting assigned in the iteration is $M'/(k+M'-1)$. Therefore* $\mathbf{Pr}\left[L(u) = i\right] = \bar{x}(u,i)$.

Proof. If ℓ is picked in the first step of an iteration, the probability of assigning i to u is exactly $\bar{x}(u,i)$, if $i \in I_\ell$, and zero otherwise. The number of intervals that

contain i is M', and hence the probability of u getting i in an iteration is simply $\bar{x}(u,i) \cdot M'/(k+M'-1)$. □

It also follows from Lemma 4.1 that with high probability all vertices are assigned in $O(k \log n)$ iterations. The following lemma bounds the expected distance between $L(u)$ and $L(v)$ as a function of M'. Recall $d_{LP}(u,v) = \sum_{i,j} d(i,j) \cdot \bar{x}(u,i,v,j)$.

LEMMA 4.2. *The expected distance between $L(u)$ and $L(v)$ satisfies the following inequality:*

$$\mathbf{E}\left[d(L(u),L(v))\right] \leq \left(2 + \max\left\{\frac{2M}{M'}, \frac{M'}{M}\right\}\right) d_{LP}(u,v).$$

THEOREM 4.3. *There is a randomized $(2+\sqrt{2})$-approximation algorithm for the metric labeling problem when the metric is truncated linear.*

Proof. We note that the algorithm and the analysis easily generalizes to the case when M' is a real number. We choose $M' = \sqrt{2}M$ and the theorem follows from Lemmas 4.1 and 4.2. □

The algorithm that we described can be derandomized using the method of conditional probabilities. The proof uses standard ideas, and hence we omit it.

For the rest of the section, we restrict our attention to one particular edge $(u,v) \in E(G)$. We analyze the effect of the rounding on the expected distance with the goal of proving Lemma 4.2. To analyze the process, we consider an iteration before which neither u or v is assigned. If, in the current iteration, only one of u or v is assigned a label, that is, they are *separated*, we will pay a distance of M. If both of them are assigned, then we pay a certain distance based on their labels in the interval. The main quantity of interest is the expected distance between the labels of u and v in a single iteration conditioned under the event that both of them are assigned in this iteration. Recall that a link refers to the edges in the complete bipartite graph $H(u,v)$.

Given an interval $I_\ell = [\ell, \ell+M'-1]$, we partition the interesting links for I_ℓ into three categories, *internal*, *left crossing*, and *right crossing*. The internal links denoted by $\textsc{int}(I_\ell)$ are all the links (u,i,v,j) with $i,j \in I_\ell$; the left crossing links denoted by $\textsc{lcross}(I_\ell)$ are those with $\min\{i,j\} < \ell$ and $\max\{i,j\} \in I_\ell$; and the right crossing links denoted by $\textsc{rcross}(I_\ell)$ are those with $\min\{i,j\} \in I_\ell$ and $\max\{i,j\} > \ell + M' - 1$. It is clear that no link is both left and right crossing. Let $\textsc{cross}(I_\ell)$ denote $\textsc{lcross}(I_\ell) \cup \textsc{rcross}(I_\ell)$. See Figure 4.1 for an example. It is easy to see that e is not relevant to I_ℓ if $i,j \notin I_\ell$.

We set up some notation that we use for the rest of this section. For a link $e = (u,i,v,j)$ let $d_{\text{lin}}(e) = |i-j|$ and $d(e) = d(i,j) = \min(M, d_{\text{lin}}(e))$ be the linear distance and truncated linear distance, respectively, between i and j. The quantity $\bar{x}_e$ is compact notation for $\bar{x}(u,i,v,j)$. Consider a link e that crosses the interval I_ℓ and let i be the label of e that is internal to I_ℓ. We denote by $d_\ell(e)$ the quantity $(\ell + M' - 1 - i)$, the linear distance from i to the right end of the interval. The quantity $\bar{x}(u, I_\ell)$ refers to $\sum_{i \in I_\ell} \bar{x}(u,i)$, the flow of u assigned by the LP to labels in I_ℓ.

LEMMA 4.4. *The probability of u and v being separated given that ℓ was chosen in the first step of the iteration is at most $\sum_{e \in \textsc{cross}(I_\ell)} \bar{x}_e$.*

Proof. The probability of separation is exactly $|\bar{x}(u,I_\ell) - \bar{x}(v,I_\ell)|$, which can easily be seen to be upper bounded by $\sum_{e \in \textsc{cross}(I_\ell)} \bar{x}_e$. □

LEMMA 4.5. *For two vertices u and v, unlabeled before an iteration, let p_ℓ be the expected distance between them, conditioned on the event that ℓ was chosen in the*

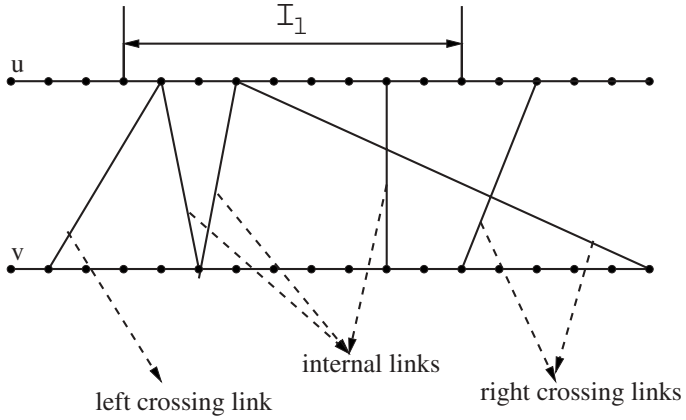


FIG. 4.1. *Illustration of interesting links for I_ℓ.*

first step of the iteration and both were assigned a label in I_ℓ. Then

$$p_\ell \leq \sum_{e \in \text{CROSS}(I_\ell)} d_\ell(e)\bar{x}_e + \sum_{e \in \text{INT}(I_\ell)} d_{lin}(e)\bar{x}_e.$$

We provide some intuition before giving the formal proof. Once ℓ is fixed, the rounding is exactly the same as that for the line metric when restricted to the interval I_ℓ. From Lemma 3.1 we know the exact expected cost of the rounding. However, we do not have an equivalent of Lemma 3.2 to bound the LP cost, because I_ℓ is only a subinterval of the full line and also because of the truncation. The main difficulty is with links that belong to CROSS(I_ℓ). By charging each of the crossing links e, an amount equal to $d_\ell(e)$ (instead of $d(e)$ that LP paid), we are able to pay for the expected cost of rounding. In other words, we charge the interesting links of I_ℓ to pay for the optimal linear metric transshipment flow induced by the fractional values $\bar{x}(u,i)$ and $\bar{x}(v,i)$, $i \in I_\ell$, when restricted to I_ℓ.

Proof. Fix an ℓ and w.l.o.g. assume that $\bar{x}(u, I_\ell) \geq \bar{x}(v, I_\ell)$. With probability $q = \bar{x}(v, I_\ell)$, both u and v get labels from I_ℓ. We analyze the expected distance conditioned on this event. Once the interval I_ℓ is fixed, the rounding is very similar to that of the linear metric case. For $0 \leq i < M'$ let $\alpha(u,i) = \sum_{j=\ell}^{j+i} \bar{x}(u,j)$. The quantity $\alpha(u,i)$ sums the amount of flow of u in the first $i+1$ labels of the interval I_ℓ. In the following analysis we assume that distances within I_ℓ are linear and ignore truncation. This can only hurt us. Following the reasoning in Lemma 3.1, the expected distance between u and v is equal to $\sum_{i=0}^{M'-1} |\min\{q, \alpha(u,i)\} - \alpha(v,i)|$, which we upper bound by $\sum_{i=0}^{M'-1} |\alpha(u,i) - \alpha(v,i)|$. We claim that

$$\sum_{i=0}^{M'-1} |\alpha(u,i) - \alpha(v,i)| \leq \sum_{e \in \text{CROSS}(I_\ell)} d_\ell(e)\bar{x}_e + \sum_{e \in \text{INT}(I_\ell)} d_{\text{lin}}(e)\bar{x}_e.$$

To prove this claim, we consider each link $e \in \text{CROSS}(I_\ell) \cup \text{INT}(I_\ell)$ and sum its contribution to the terms $q_i = |\alpha(u,i) - \alpha(v,i)|$, $0 \leq i < M'$. Let $e = (u,a,v,b) \in \text{INT}(I_\ell)$. It is clear that e contributes exactly $\bar{x}_e$ to q_i if $a \leq i$ and $b > i$ or if $a > i$ and $b \leq i$. Otherwise its contribution is 0. Therefore, the overall contribution of e to $\sum q_i$ is $\bar{x}_e|a-b| = \bar{x}_e d_{\text{lin}}(e)$.

Now suppose $e = (u, a, v, b) \in \text{LCROSS}(I_\ell)$. Assume w.l.o.g. that $a \geq \ell$ and $b < \ell$; the other case is similar. Link e will contribute $\bar{x}_e$ to $\alpha(u, i)$ for $a - \ell \leq i < M'$ and contributes 0 to $\alpha(v, i)$ for $0 \leq i < M'$ since b is outside the interval I_ℓ. Therefore, the contribution of e to q_i is $\bar{x}_e$ for $a - \ell \leq i < M'$ and 0 otherwise. The overall contribution of e to $\sum q_i$ is $\bar{x}_e|\ell + M' - 1 - a| = d_\ell(e)\bar{x}_e$. A similar argument holds for the case when $e \in \text{RCROSS}(I_\ell)$. □

Proof of Lemma 4.2. For a given iteration before which neither u nor v has a label, let $\mathbf{Pr}[u \wedge v]$, $\mathbf{Pr}[u \oplus v]$, and $\mathbf{Pr}[u \vee v]$ denote the probabilities that u and v are both assigned, exactly one of them is assigned, and at least one of them is assigned, respectively. We upper bound the quantity $\mathbf{E}[d(L(u), L(v))]$ as follows. If u and v are separated in some iteration, we upper bound their resulting distance by M. Using this simplification, $\mathbf{E}[d(L(u), L(v))]$ is bounded by the quantity below:

$$\frac{\mathbf{Pr}[u \oplus v] \cdot M + \mathbf{Pr}[u \wedge v] \cdot \mathbf{E}[d(L(u), L(v))|u \wedge v]}{\mathbf{Pr}[u \vee v]}.$$

We upper bound the above expression as follows:

- We lower bound $\mathbf{Pr}[u \vee v]$ by $\mathbf{Pr}[u]$, which by Lemma 4.1 is equal to $M'/(k + M' - 1)$.
- We upper bound $\mathbf{Pr}[u \oplus v]$ by $\frac{1}{k+M'-1} \sum_\ell \sum_{e \in \text{CROSS}(I_\ell)} \bar{x}_e$ using Lemma 4.4.
- By the definition of p_ℓ in Lemma 4.5 we get the following:

$$\mathbf{Pr}[u \wedge v]\, \mathbf{E}[d(L(u), L(v))|u \wedge v] = \frac{1}{k + M' - 1} \sum_\ell p_\ell.$$

Putting all these together and using Lemma 4.5 to bound p_ℓ,

$$\begin{aligned}\mathbf{E}[d(L(u), L(v))] &\leq \frac{1}{M'} \sum_\ell \left(\sum_{e \in \text{CROSS}(I_\ell)} (M + d_\ell(e))\bar{x}_e + \sum_{e \in \text{INT}(I_\ell)} d_{\text{lin}}(e)\bar{x}_e \right) \\ &\leq \frac{1}{M'} \sum_e \bar{x}_e \left(\sum_{\text{CROSS}(I_\ell) \ni e} (M + d_\ell(e)) + \sum_{\text{INT}(I_\ell) \ni e} d_{\text{lin}}(e) \right).\end{aligned}$$

Lemma 4.6 shows that

$$\sum_{\text{CROSS}(I_\ell) \ni e} (M + d_\ell(e)) + \sum_{\text{INT}(I_\ell) \ni e} d_{\text{lin}}(e) \leq \left(2M' + \max\{2M, (M')^2/M\}\right) d(e).$$

It follows that

$$\begin{aligned}\mathbf{E}[d(L(u), L(v))] &\leq \frac{1}{M'} \left(2M' + \max\{2M, (M')^2/M\}\right) \sum_e \bar{x}_e d(e) \\ &\leq (2 + \max\{2M/M', M'/M\}) \cdot d_{LP}(u, v).\end{aligned}$$

This finishes the proof. □

LEMMA 4.6. *Let $e = (u, i, v, j)$ be a link. Then*

$$\sum_{\text{CROSS}(I_\ell) \ni e} (M + d_\ell(e)) + \sum_{\text{INT}(I_\ell) \ni e} d_{lin}(e) \leq \left(2M' + \max\left\{2M, (M')^2/M\right\}\right) d(e).$$

Proof. Let $\bar{d}(e) = \sum_{\text{CROSS}(I_\ell) \ni e} (M + d_\ell(e)) + \sum_{\text{INT}(I_\ell) \ni e} d_{\text{lin}}(e)$. We evaluate $\bar{d}(e)$ separately for three different types of links based on their lengths. Recall that $M' \geq M$ and hence $d(e) \leq M \leq M'$ for all links e. Let e correspond to the link (u, i, v, j) in $H(u, v)$ and w.l.o.g. assume that $i \leq j$; the other case is similar. Also recall that $d_{\text{lin}}(e) = |i - j|$.

• $d_{\text{lin}}(e) \geq M'$. In this case it is clear that e is not an internal edge for any I_ℓ; hence $\sum_{\text{INT}(I_\ell)\ni e} d_{\text{lin}}(e) = 0$. Also $d(e) = M$. Therefore

$$\begin{aligned}
\bar{d}(e) &= \sum_{\text{CROSS}(I_\ell)\ni e} (M + d_\ell(e)) \\
&= \sum_{\text{LCROSS}(I_\ell)\ni e} (M + d_\ell(e)) + \sum_{\text{RCROSS}(I_\ell)\ni e} (M + d_\ell(e)) \\
&= \sum_{\ell=j-M'+1}^{j} (M + M' + \ell - 1 - i) + \sum_{\ell=i-M'+1}^{i} (M + M' + \ell - 1 - i) \\
&\leq M'(2M + M') \\
&= M'(2 + M'/M)M = M'(2 + M'/M)d(e).
\end{aligned}$$

• $d_{\text{lin}}(e) < M$. In this case $d(e) = d_{\text{lin}}(e)$.

$$\begin{aligned}
\bar{d}(e) &= \sum_{\text{CROSS}(I_\ell)\ni e} (M + d_\ell(e)) + \sum_{\text{INT}(I_\ell)\ni e} d_{\text{lin}}(e) \\
&= \sum_{\text{LCROSS}(I_\ell)\ni e} (M + d_\ell(e)) + \sum_{\text{RCROSS}(I_\ell)\ni e} (M + d_\ell(e)) \\
&\quad + \sum_{\text{INT}(I_\ell)\ni e} d_{\text{lin}}(e) \\
&= \sum_{\ell=i+1}^{j} (M + M' + \ell - 1 - i) + \sum_{\ell=i-M'+1}^{j-M'} (M + M' + \ell - 1 - i) \\
&\quad + \sum_{\ell=j-M'+1}^{i} d_{\text{lin}}(e) \\
&\leq (2M' + 2M - d_{\text{lin}}(e))d_{\text{lin}}(e) \\
&\leq (2M' + 2M)d_{\text{lin}}(e) \\
&= M'(2 + 2M/M')d(e).
\end{aligned}$$

• $M \leq d_{\text{lin}}(e) < M'$. In this case $d(e) = M$.

$$\begin{aligned}
\bar{d}(e) &= \sum_{\text{LCROSS}(I_\ell)\ni e} (M + d_\ell(e)) + \sum_{\text{RCROSS}(I_\ell)\ni e} (M + d_\ell(e)) + \sum_{\text{INT}(I_\ell)\ni e} d_{\text{lin}}(e) \\
&= \sum_{\ell=i+1}^{j} (M + M' + \ell - 1 - i) + \sum_{\ell=i-M'+1}^{j-M'} (M + M' + \ell - 1 - i) \\
&\quad + \sum_{\ell=j-M'+1}^{i} d_{\text{lin}}(e) \\
&= (2M' + 2M - d_{\text{lin}}(e))d_{\text{lin}}(e) \\
&\leq M'(2M + M') \\
&= M'(2 + M'/M)d(e). \qquad \square
\end{aligned}$$

4.1. The truncated quadratic distance on the line. Consider the label space $1, 2, \ldots, k$ on the line where the distance function $d(i,j) = \min\{M, |i-j|^2\}$. This is the truncated version of the quadratic distance. We note that the quadratic distance is not a metric. However, as mentioned earlier, this distance function arises in image processing applications. In subsection 3.3 we showed that our LP formulation gives an optimal solution for the quadratic distance on the line. For the truncated version of this distance we can use the algorithm from section 4. By choosing $M' = \sqrt{M}$ we can show the following theorem.

THEOREM 4.7. *The integrality gap of LP* (I) *for the truncated quadratic distance is* $O(\sqrt{M})$.

5. General metrics.

5.1. Integrality gap on general metrics. We now show that the integrality gap of the LP formulation is $O(\log k)$ on general metrics. This gives an alternative way to obtain the result of Kleinberg and Tardos [26]. The latter algorithm uses the approach of first approximating the given metric probabilistically by a hierarchically well-separated tree (HST) metric [2] and then using an LP formulation to solve the problem on tree metrics. The Kleinberg–Tardos LP formulation has only an $O(1)$ integrality gap on HST metrics. Since any arbitrary k-point metric can be probabilistically approximated by an HST metric with an $O(\log k)$ distortion [16], their result follows. In contrast, our approach is based on directly using our LP formulation on the given general metric. As a first step, we use the LP solution to identify a deterministic HST metric approximation of the given metric such that the cost of our fractional solution on this HST metric is at most $O(\log k)$ times the LP cost on the original metric. This first step is done by using the following proposition from [16]. A weaker version was shown earlier in [9, 3] with a bound of $O(\log k \log\log k)$.

PROPOSITION 5.1. *Let d be an arbitrary k-point metric and let α be a nonnegative function defined over all pairs of points in the metric. Then d can be deterministically approximated by an HST metric d_T such that*

$$\sum_{i,j} \alpha(i,j) \cdot d_T(i,j) \le O(\log k) \sum_{i,j} \alpha(i,j) \cdot d(i,j).$$

Given an optimal solution $\bar{x}$ to our LP formulation, we apply Proposition 5.1 with the weight function $\alpha(i,j) = \sum_{(u,v)\in E} w(u,v) \cdot \bar{x}(u,i,v,j)$ for $1 \le i,j \le k$. Thus, $\alpha(i,j)$ is the fractional weight of edges between i and j. Let d_T denote the resulting HST metric. Since we are changing only the metric on the labels and not the assignments provided by the LP, the fractional solution is a feasible solution for this new metric and has cost at most $O(\log k) \cdot C^*$, where C^* is the optimal LP cost for the original metric. Thus, if we can now round our fractional solution on d_T by introducing only a constant factor increase in the solution cost, we will obtain an $O(\log k)$-approximation algorithm. We prove this by showing that on any tree metric, our LP formulation is at least as strong as the Kleinberg–Tardos LP formulation (for tree metrics).

Given an edge weighted tree that defines the metric on the labels, we root the tree at some arbitrary vertex. Let T_a denote the subtree hanging off a vertex a in the rooted tree and let $\mathcal{T}$ denote the set of all such trees. For any tree $T \in \mathcal{T}$ we denote by $\ell(T)$ the length of the edge leaving the root of T to its parent. Let $x_T(u)$ be compact notation for $\sum_{i\in T} x(u,i)$, the fractional assignment of the LP to labels

in the subtree T. With this notation the LP formulation in [26] is as follows:

$$\text{(KT)} \quad \min \sum_{v \in V} \sum_{i=1}^{k} c(u,i) \cdot x(u,i) + \sum_{(u,v) \in E} w(u,v) \sum_{T \in \mathcal{T}} \ell(T) \cdot |x_T(u) - x_T(v)|$$

subject to

$$\sum_{i=1}^{k} x(u,i) = 1 \quad \forall\ u \in V,$$

$$x(u,i) \geq 0 \quad \forall\ u \in V \text{ and } i \in 1, \ldots, k.$$

Let $\bar{x}$ be a solution to our formulation (I). As we remarked in section 3.1, for both (I) and (KT) above, the values $\bar{x}(u,i)$ completely determine the cost. We will show that cost of (KT) on $\bar{x}$ is smaller than that of (I). Both linear programs (I) and (KT) coincide regarding the labeling cost. For each edge $(u,v) \in E$ we will show that LP distance for (u,v) is smaller in (I) than (KT). This is based on the following claim.

Claim 5.2. For any feasible solution $\bar{x}$ of (I),

$$\sum_{T \in \mathcal{T}} \ell(T) \cdot |\bar{x}_T(u) - \bar{x}_T(v)| \leq \sum_{i,j} d(i,j) \cdot \bar{x}(u,i,v,j).$$

The proof of the above claim is straightforward and simply relies on the fact that the distance between two vertices in a tree metric is defined by the unique path between them. We omit the details. We obtain the following theorem from the above discussion.

THEOREM 5.3. *The integrality gap of LP* (I) *on a k-point metric is $O(\log k)$.*

5.2. Reduction to zero-infinity assignment costs. We now describe a transformation for the general problem that essentially allows us to eliminate the label assignment cost function. This transformation reduces an instance with arbitrary label assignment cost function c to one where each label assignment cost is either 0 or ∞. We refer to an instance of this latter type as a *zero-infinity* instance. Our transformation exactly preserves the cost of any feasible solution but in the process increases the number of labels by a factor of n. This provides some evidence that the label cost assignment function does not play a strong role in determining the approximability threshold of the metric labeling problem. In particular, existence of a constant factor approximation algorithm for zero-infinity instances would imply a constant factor approximation algorithm for general instances as well.

From an instance $I = \langle c, d, w, L, G(V,E) \rangle$ of the general problem, we create an instance of the zero-infinity variant $I' = \langle c', d', w, L', G(V,E) \rangle$ as follows. We define a new label set $L' = \{i_u \mid i \in L \text{ and } u \in V\}$, i.e., we make a copy of L for each vertex in G. The new label cost assignment function is given by $c'(u, i_v) = 0$ if $v = u$ and ∞ otherwise. Thus, each vertex has its own copy of the original label set, and any finite cost solution to I' would assign each vertex a label from its own private copy.

Let $W_u = \sum_{(u,v) \in E} w(u,v)$ for any vertex $u \in V$. The new distance metric on L' is defined in terms of the original distance metric as well as the original label cost assignment function. For $i \neq j$ or $u \neq v$,

$$d'(i_u, j_v) = d(i,j) + \frac{c(u,i)}{W_u} + \frac{c(v,j)}{W_v},$$

and $d'(i_u, i_u) = 0$. It can be verified that d' is indeed a metric and that any solution to instance I can be mapped to a solution to instance I', and vice versa, in a cost-preserving manner. The proof of the following theorem follows in a straightforward manner from the above construction.

THEOREM 5.4. *If there exists a $f(n,k)$-approximation algorithm for zero-infinity instances of the metric labeling problem, then there exists a $f(n,nk)$-approximation algorithm for general instances.*

In fact, there is an even simpler reduction to zero-infinity instances, conveyed to us by Chuzhoy [10], which does not change the input metric but changes the graph in a very simple way. For each vertex v and label j such that $c(v,j) > 0$, add a new vertex z_{vj} to the graph for which

$$c(z_{vj}, \ell) = \begin{cases} \infty & \text{if } \ell = j, \\ 0 & \text{if } \ell \neq j. \end{cases}$$

A new edge (v, z_{vj}) is added to the graph. Let $\ell \neq j$ be the label that minimizes $d(j,\ell)$ such that $d(j,\ell) > 0$. The weight of the edge (v, z_{vj}) is set to

$$w(v, z_{vj}) = \frac{c(v,j)}{d(j,\ell)}.$$

Set $c(v,j) = 0$ for $1 \leq j \leq k$. We now obtain a new instance of the metric labeling problem by the above transformation. Note that the metric has not been altered. It is not hard to verify that this reduction preserves the value of an optimal solution and that an r-approximation to the new instance also yields an r-approximation to the original instance. Hence the following theorem is obtained.

THEOREM 5.5 (see Chuzhoy [10]). *If there is a $f(n,k)$-approximation algorithm for zero-infinity instances of metric labeling, then there is a $f(n+nk,k)$-approximation algorithm for general instances.*

6. Conclusions. As mentioned in section 1, our LP formulation has integrality gap 1 when G is a tree. We give a brief sketch of the idea. Consider an edge (u,v) in G where u is a leaf connected to v. If v is assigned a label i in some solution, then it is easy to see that an optimal assignment to u is to assign it a label j, where $c(u,j) + w(u,v)d(i,j) = \min_k(c(u,k) + w(u,v)d(i,k))$. Hence the assignment to u is completely fixed by the assignment to v. We can eliminate u from G and incorporate the cost of u in the assignment cost of v as follows: set $c'(v,i) = c(v,i) + \min_k(c(u,k) + w(u,v)d(i,k))$. This transformation can be repeated until there is only one node left and the optimal solution then is trivial. The same argument above can be used to show optimality of our LP formulation for trees. We leave the details to the reader.

Chuzhoy and Naor [11] recently obtained an $\Omega(\sqrt{\log k})$-factor hardness of approximation for the metric labeling problem. The 0-extension problem generalizes the multiway cut problem and is a special case of the metric labeling problem. As mentioned earlier, Calinescu, Karloff, and Rabani [8] established an $\Omega(\sqrt{\log k})$ lower bound on the integrality gap of the metric relaxation for the 0-extension problem. Fakcheroenphol et al. [15] improved the upper bound on the integrality gap of the metric relaxation to $O(\log k / \log\log k)$. It is worthwhile to study the integrality gap of our formulation for this restricted problem. See [1] for results in this direction.

The truncated quadratic distance function is of particular interest to applications in computer vision. Although this distance function does not form a metric, it is quite possible that a constant factor approximation is achievable. Here, too, our formulation might be of use in developing improved algorithms.

Acknowledgments. We thank Olga Veksler and Mihalis Yannakakis for useful discussions. We thank Julia Chuzhoy for allowing us to include her reduction in section 5.2.

REFERENCES

[1] A. Archer, J. Fakcharoenphol, C. Harrelson, R. Krauthgamer, K. Talwar, and É. Tardos, *Approximate classification via earthmover metrics*, in Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, 2004, pp. 1072–1080.

[2] Y. Bartal, *Probabilistic approximation of metric spaces and its algorithmic applications*, in Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science, 1996, pp. 184–193.

[3] Y. Bartal, *On approximating arbitrary metrics by tree metrics*, in Proceedings of the 30th Annual ACM Symposium on Theory of Computing, 1998, pp. 161–168.

[4] Y. Boykov, O. Veksler, and R. Zabih, *Markov random fields with efficient approximations*, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1998, pp. 648–655.

[5] Y. Boykov, O. Veksler, and R. Zabih, *Fast approximate energy minimization via graph cuts*, in Proceedings of the 7th IEEE International Conference on Computer Vision, 1999, pp. 377–384.

[6] Y. Boykov, O. Veksler, and R. Zabih, *A new algorithm for energy minimization with discontinuities*, in Proceedings of the International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, 1999.

[7] G. Călinescu, H. Karloff, and Y. Rabani, *An improved approximation algorithm for multiway cut*, J. Comput. System Sci., 60 (2000), pp. 564–574.

[8] G. Calinescu, H. Karloff, and Y. Rabani, *Approximation algorithms for the* 0*-extension problem*, SIAM J. Comput., 34 (2004), pp. 358–372.

[9] M. Charikar, C. Chekuri, A. Goel, and S. Guha, *Rounding via trees: Deterministic approximation algorithms for group steiner trees and k-median*, in Proceedings of the 30th ACM Symposium on Theory of Computing, 1998, pp. 114–123.

[10] J. Chuzhoy, *private communication*, 2001.

[11] J. Chuzhoy and J. Naor, *The hardness of metric labeling*, in Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, 2004, pp. 108–114.

[12] F. S. Cohen, *Modeling and applications of stochastic processes*, in The Markov Random Fields for Image Modeling and Analysis, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1986, pp. 243–272.

[13] E. Dalhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis, *The complexity of multiterminal cuts*, SIAM J. Comput., 23 (1994), pp. 864–894.

[14] R. Dubes and A. Jain, *Random field models in image analysis*, J. Appl. Stat., 16 (1989), pp. 131–164.

[15] J. Fakcharoenphol, C. Harrelson, S. Rao, and K. Talwar, *An improved approximation algorithm for the* 0*-extension problem*, in Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Baltimore, MD, 2003, pp. 257–265.

[16] J. Fakcharoenphol, S. Rao, and K. Talwar, *A tight bound on approximating arbitrary metrics by tree metrics*, in Proceedings of the 35th Annual ACM Symposium on Theory of Computing, 2003, pp. 448–455.

[17] D. Greig, B. Porteous, and A. Seheult, *Exact maximum a posteriori estimation for binary images*, J. Roy. Statist. Soc. Ser. B, 51 (1989), pp. 271–279.

[18] A. Gupta and E. Tardos, *Constant factor approximation algorithms for a class of classification problems*, in Proceedings of the 32nd Annual ACM Symposium on the Theory of Computing, 2000, pp. 652–658.

[19] D. Hochbaum, *An efficient algorithm for image segmentation, markov random fields and related problems*, J. ACM, 48 (2001), pp. 686–701.

[20] A. J. Hoffman, *On simple linear programming problems*, in Proceedings of the 7th Symposium in Pure Mathematics, Vol. 7, V. Klee, ed., AMS, Providence, RI, 1963, pp. 317–327.

[21] H. Ishikawa and D. Geiger, *Segmentation by grouping junctions*, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1998, pp. 125–131.

[22] H. Ishikawa and D. Geiger, *Mapping image restoration to a graph problem*, in Proceedings of the 1999 IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing, 1999.

[23] D. Karger, P. Klein, C. Stein, M. Thorup, and N. Young, *Rounding algorithms for a*

geometric embedding of multiway cut, in Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, 1999, pp. 668–678.

[24] A. Karzanov, *Minimum* 0*-extension of graph metrics*, European J. Combin., 19 (1998), pp. 71–101.

[25] A. Karzanov, *A combinatorial algorithm for the minimum* $(2, r)$*-metric problem and some generalizations*, Combinatorica, 18 (1999), pp. 549–569.

[26] J. Kleinberg and E. Tardos, *Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields*, J. ACM, 49 (2002), pp. 616–630.

[27] S. Z. Li, *Markov Random Field Modeling in Computer Vision*, 2nd ed., Springer-Verlag, New York, 2001.

[28] I. Milis, *Task assignment in distributed systems using network flow methods*, in Proceedings of the Conference on Combinatorics and Computer Science, Lecture Notes in Comput. Sci. 1120, Springer-Verlag, Berlin, 1996, pp. 396–405.

[29] G. Monge, *Déblai et Ramblai*, Mem. de l'Academie des Sciences, 1781.

[30] S. Peleg, M. Werman, and H. Rom, *A unified approach to the change of resolution: Space and gray-level*, IEEE Trans. Pattern Anal. Machine Intel., 11 (1989), pp. 739–742.

[31] S. Rachev, *The Monge-Kantorovich mass transference problem and its stochastic applications*, Theory Probab. Appl., 29 (1984), pp. 647–676.

[32] S. Roy and I. Cox., *A maximum-flow formulation of the N-camera stereo correspondence problem*, in Proceedings of the International Conference on Computer Vision, 1998, pp. 492–499.

[33] Y. Rubner, C. Tomasi, and L. Guibas, *The earth mover's distance as a metric for image retrieval*, Int. J. Comput. Vis., 40 (2000), pp. 99–121.

SIAM J. DISCRETE MATH.
Vol. 18, No. 3, pp. 626–637

A TIGHT LOWER BOUND FOR ONLINE MONOTONIC LIST LABELING*

PAUL F. DIETZ†, JOEL I. SEIFERAS‡, AND JU ZHANG§

Abstract. Maintaining a monotonic labeling of an ordered list during the insertion of n items requires $\Omega(n \log n)$ individual relabelings, in the worst case, if the number of usable labels is only polynomial in n. This follows from a lower bound for a new problem, *prefix bucketing*.

Key words. monotonic list labeling, order maintenance, density/congestion management and exploitation, load balancing, bucketing, online, lower bound, adversary argument

AMS subject classifications. 68R99, 68P05

DOI. 10.1137/S0895480100315808

1. Introduction. The *online list-labeling problem* can be viewed as one of linear density control. A sequence of n distinct items from some dense, linearly ordered set, such as the real numbers, is received one at a time, in no predictable order. Using "labels" from some discrete linearly ordered set of adequate but limited cardinality, the problem is to maintain an assignment of labels to the items received so far, so that the labels are ordered in the same way as the items they label. To make room for the next item received, it might be necessary to change the labels assigned to some of the items previously received. The *cost* is the total number of labelings and relabelings performed.

There are practical applications of online list labeling to the design of efficient data structures and algorithms. List labeling has been an especially fruitful approach to the *order maintenance problem* [Di82, Ts84, DS87, DZ90]. This problem involves the insertion and deletion of items into a linear list and response to online queries on the relative order of items currently in the list. A low-cost online list-labeling algorithm provides an efficient solution (or sometimes a component of an even more efficient solution) to the order maintenance problem, provided its computational overhead is also low. For further discussion of this and other specific applications, see the earlier papers by Dietz and his collaborators [Di82, DS87, DZ90]. In addition, it seems likely that our problem and related problems of dynamic density control will prove fundamental to the spatially structured maintenance in bounded media of changing data, such as text and pictures on a computer screen [Zh93].

When the total number of labels is large enough, it is possible to avoid completely any need to relabel. The algorithm is simply to assign to each arriving item a label

*Received by the editors June 7, 2000; accepted for publication (in revised form) July 12, 2004; published electronically April 8, 2005. A preliminary version of this paper appeared in *Proceedings of the 4th Scandinavian Workshop on Algorithm Theory* (*SWAT* '94), Lecture Notes in Comput. Sci. 824, Springer-Verlag, Berlin, 1994, pp. 131–142. It is based on a portion of the third author's doctoral dissertation [Zh93].

http://www.siam.org/journals/sidma/18-3/31580.html

†Software Technology Center, Motorola Land Mobile Products Sector, Schaumburg, IL 60196 (dietz@stc.corp.mot.com). The research of this author was supported in part by National Science Foundation grant CCR-8909667.

‡Computer Science Department, University of Rochester, Rochester, NY 14627-0226 (joel@cs.rochester.edu).

§Global Credit Risk Management, JPMorgan Chase, 245 Park Avenue, New York, NY 10167 (ju.zhang@jpmchase.com).

closest to midway through the range of unused labels that would work. In particular, if the total number of labels is at least $2^n - 1$, then there always will be at least one label in the needed range. (By induction on $i \leq n$, in fact, there will be at least $2^{n+1-i} - 1$ labels that would work for the ith item received.) But if the total number of labels is any less, then some relabeling will be required in the worst case (by a similar induction).

When the number of labels is merely at least $n^{1+\epsilon}$ for some $\epsilon > 0$, it is possible to limit the worst-case cost for online labeling of n items to $\mathcal{O}(n \log n)$ [Di82, Ts84, DS87]. Although the analyses are subtle, the best of the algorithms are both simple and fast, and hence practically useful. In this paper we show that the upper bound is tight and in fact that $\Omega(n \log n)$ relabelings are required even for an algorithm that is complicated and slow.

Our proof is a surprising adaptation of a lower-bound approach sketched by Dietz and Zhang [DZ90]. That approach seemed to be a dead end that addressed only strategies that satisfy the following "smoothness" property: The list items relabeled before each insertion form a contiguous neighborhood of the list position specified for the new item, and the new labels are as widely and equally spaced as possible. Although no good *non*smooth algorithms have been proposed or analyzed, it has seemed difficult to rule them out. (This is the usual sort of lower-bound predicament.) The key to our adaptation is to imagine appropriate dynamic "recalibrations" of the label space, in terms of which the arbitrary strategy does look fairly smooth.

To facilitate the elaboration and adaptation of the earlier argument, we formulate and separately attack variants of a previously unstated combinatorial "bucketing problem" that really lies at the heart of the argument. In the *unordered bucketing problem*, the challenge is to cheaply insert n items, one at a time, into k buckets. The cost of each insertion is the number of items (including the new one) in the bucket chosen for that insertion. The optimum total cost for the task as described so far is clearly $\Omega(n^2/k)$, but we allow an additional operation: Between insertions, we can redistribute the contents within any subset of the buckets, at a cost equal to the total number of items in those buckets. Now $\mathcal{O}(n \log n)$ is an upper bound on the required cost, by a well-known strategy,[1] provided k is $\Omega(\log n)$. On the other hand, we prove in section 3 that $\Omega(n \log n / \log k)$ is always a lower bound on the required cost, and we conjecture that $\Omega(n \log n)$ is a lower bound when k is $\mathcal{O}(\log n)$. We show in section 2 that either lower bound leads to a similar lower bound for the problem of primary interest, the n-item, polynomial-label labeling problem.

The *prefix bucketing problem* is like the unordered bucketing problem, except with the constraint that, in terms of some fixed linear order of the buckets, the subset for each redistribution must be a *prefix* of the bucket list. (The well-known strategy already cited[1] still works.) Under this constraint, each redistribution may as well move all items involved into the very last bucket of the chosen prefix. Section 2 shows that even a lower bound on this bucketing problem leads to a labeling lower bound. In section 4, we prove the needed lower bound on prefix bucketing.

[1] The strategy works even for the more restricted "prefix" version of the bucketing problem. Like the Hennie–Stearns simulation of multiple Turing machine tapes [HS66], it is modeled after binary counting: Just let the redistributions be forced by an exponentially growing sequence of bucket-capacity limits, such as the powers of 2. This particular choice would result in a sequence of involved-prefix lengths

$$1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 5, \ldots$$

that correspond to the propagation distances of binary "carries."

2. Relation to bucketing. This section is devoted to a proof of the following relation, which will yield labeling lower bounds when combined with the bucketing lower bounds of sections 3 and 4.

THEOREM 2.1. *When the number of labels is $\mathcal{O}(n^{1+\epsilon})$ for some $\epsilon > 0$, the worst-case cost for online labeling of n items is at least proportional to the cost for prefix bucketing of n items into $\mathcal{O}(\log n)$ buckets.*

A relabeling algorithm is *normalized* if the batch of items it (re)labels on each insertion, including the newly inserted item, forms a contiguous sublist of the list resulting from the insertion. Since *non*contiguous relabelings can safely be deferred until later, each labeling algorithm can be replaced at no additional cost by a normalized one.

To prove a lower bound for the labeling problem, we show that each normalized algorithm (and hence each algorithm of any kind) performs many relabelings when confronted with some bad-case sequence of insertion requests. That sequence, which will depend on the particular algorithm, will be determined by an "adversary" strategy that *interacts* with the algorithm—each next insertion will be into a gap that the adversary chooses based on the labeling decisions made by the algorithm in response to earlier insertion requests.

Intuitively, the most promising strategy for the adversary is to insert the next item into a gap between items in a part of the label space that is currently "relatively crowded." It seems difficult, however, to formulate an appropriate notion of crowdedness. Ordinary *density*, for instance, can vary depending on the size and choice of neighborhood. A more robust notion of a "dense point" is a gap all of whose neighborhoods (that include at least both the items delimiting the gap) are currently about as dense as the entire label space. The following general lemma and its corollaries show that such a dense point always does exist.

DENSE-POINT LEMMA. *Consider any nonnegative, integrable function f on the interval $[0,1]$. For each (nontrivial) subinterval I, define*

$$\rho(I) = \frac{1}{|I|} \int_I f(x)\,dx.$$

Then there is some point $x_0 \in [0,1]$ such that $\rho(I) \geq \frac{1}{2}\rho([0,1])$ holds whenever I includes x_0.

Proof. For the sake of argument, suppose not. Then, for each point x, select a spoiling interval that includes x and that is open in $[0,1]$. (An interval is "open in $[0,1]$" if it is the intersection of $[0,1]$ itself and an ordinary open interval of real numbers.) The selected open intervals cover the topologically compact set $[0,1]$, so some finite subfamily must do so. If any point lies in three or more intervals of the subfamily, then keep only the one that reaches farthest left and the one that reaches farthest right. This leaves a finite family $\mathcal{I}$ that covers each point in $[0,1]$ either once or twice, but each of whose members I satisfies $\rho(I) < \frac{1}{2}\rho([0,1])$. Therefore,

$$\begin{aligned}\int_{[0,1]} f(x)\,dx &\leq \sum_{I\in\mathcal{I}} \int_I f(x)\,dx = \sum_{I\in\mathcal{I}} \rho(I)|I| \\ &< \sum_{I\in\mathcal{I}} \frac{1}{2}\rho([0,1])|I| = \rho([0,1]) \sum_{I\in\mathcal{I}} \frac{1}{2}|I| \leq \rho([0,1]) = \int_{[0,1]} f(x)\,dx,\end{aligned}$$

a contradiction. ∎

In our application of the Dense-point Lemma, the intervals I are sets of consecutive labels, and the density $\rho(I)$ of such a set should be the fraction of those labels currently in use as labels of items. We speak of the labels in use as being "populated" by the items they label.

DENSE-POINT COROLLARY 1. *In each labeling, there is a label such that every label-space subinterval containing that label is at least half as dense as the entire label space. (The same applies to either of the two item-through-item gaps that include the distinguished label, since they themselves are qualifying subintervals.)*

Proof. If the total number of labels is m, then consider a function f that is constantly 1 or 0 on each subinterval $((i-1)/m, i/m)$, depending on whether the ith label is or is not in use, respectively. □

DENSE-POINT COROLLARY 2. *In each labeling, there is a populated label in the middle third of the populated labels (where rounding is in favor of that middle third) such that every label-space subinterval containing that label is at least one-sixth as dense as the entire label space.*

Proof. Remove the leftmost third and the rightmost third of the items (both rounded down), and apply Dense-point Corollary 1 to the resulting labeling. □

Although dense-point gaps always exist, it does not quite suffice for the adversary always to insert into just any such gap. For example, if the adversary selects gaps that satisfy the conclusion of Dense-point Corollary 1 in round-robin fashion for its insertions, then the algorithm that always assigns a label closest to midway through the requested gap will be able to maintain an essentially perfect spread without ever relabeling any items at all. The problem is that this adversary forfeits an opportunity to use its insertions to selectively increase congestion in a particular locality.

Here is a sketch of how our adversary will take advantage of its opportunity to create congestion: It will try to keep the relocation of its insertion point "commensurate with" the relabeling response by the algorithm. That is, unless it has forced the algorithm to move many items away from the insertion point, it will continue to insert into and add congestion to the same neighborhood. To this end, it will actually maintain an entire *nest* of $k = \mathcal{O}(\log n)$ distinct *intervals* of labels that converge down to the insertion point, in a way that guarantees that the population of the smallest enclosing one of the intervals will be proportional to the number of relabelings performed. This will "justify" relocation of the insertion point to any appropriate gap in that interval, since the relabeling could have reconcentrated the population arbitrarily within the interval.

In more detail, our adversary maintains a nest of label-space intervals $I_1 \supset I_2 \supset \cdots \supset I_k$ that satisfy the five conditions listed below. For each label interval I, we denote the number of currently assigned labels and the total number of labels by $\mathrm{pop}(I)$ (for "population") and $\mathrm{area}(I)$, respectively. If $I \supset I'$, then the difference $I - I'$ consists of (at most) a left interval and a right interval; we denote their respective populations by $\mathrm{leftpop}(I - I')$ and $\mathrm{rightpop}(I - I')$.

1. I_1 is the whole label space.
2. $\mathrm{pop}(I_k) = \mathcal{O}(1)$.
3. For every i, $\mathrm{area}(I_{i+1}) \le \mathrm{area}(I_i)/2$.
4. For every i, $\mathrm{pop}(I_{i+1}) = \Omega(\mathrm{pop}(I_i))$.
5. For every i, $\mathrm{leftpop}(I_i - I_{i+1}) = \Theta(\mathrm{rightpop}(I_i - I_{i+1}))$.

As promised, it follows that $k = \mathcal{O}(\log n)$ (by conditions 1–3) and that the population of the smallest one of the intervals that encloses an insertion's batch of relabelings is at most some constant times the number of relabelings in the batch (by conditions

4 and 5, since the algorithm is normalized). Therefore, if we consider the successive differences, $I_i - I_{i+1}$, and the innermost interval, I_k, to be the buckets, and if our adversary can avoid modifying intervals larger than the smallest one that encloses an insertion's batch of relabelings, then the algorithm solves the resulting (prefix) bucketing problem at a total cost that is at most some constant times the number of relabelings it performs. Since the former has to be $\Omega(n \log n / \log k) = \Omega(n \log n / \log \log n)$, for example (the lower bound in section 3), so does the latter.

Finally, the following lemma ensures that our adversary can appropriately restore the invariant conditions after each insertion's batch of relabelings by the algorithm, replacing only the intervals reached by the relabelings.

Restoration Lemma. *Each sufficiently long and populous interval I has a subinterval I' such that*

$\text{area}(I') \le \text{area}(I)/2$,
$\text{pop}(I') = \Omega(\text{pop}(I))$, *and*
$\text{leftpop}(I - I') = \Theta(\text{rightpop}(I - I'))$.

Proof. From a dense point in the middle population third of I (provided by Densepoint Corollary 2), move boundaries leftward and rightward through population at rates proportional to the total populations in those directions (which can differ by at most a factor of 2), until half the area is covered. (If this requires a fraction of an item in either direction, then just stop one label short of that item's label.) □

3. Lower bound for unordered bucketing. This section is devoted to the relatively easy proof of the following lower bound, which we conjecture can be tightened to $\Omega(n \log n)$ when k is $\mathcal{O}(\log n)$.

Theorem 3.1. *The cost for unordered bucketing of n items into k buckets is $\Omega(n \log n / \log k)$.*

The proof is based on the following measure of a configuration's complexity:

$$C = \sum_{i=1}^{k} n_i \log n_i,$$

where n_i is the number of items in bucket i. (Since $\lim_{x \downarrow 0} x \log x = 0$, it works well to define $0 \log 0$ to *be* 0.)

Complexity-range Lemma. *If $n_1 + \cdots + n_k = n$, where each n_i is nonnegative, then $\sum n_i \log n_i$ lies between $n \log n$ and $n \log n - n \log k$.*

Proof. It is easy to argue that the sum is maximized when some n_i equals n, and minimized when every n_i equals n/k. □

So C starts out at 0 and finally reaches a value no smaller than

$$F = n \log n - n \log k.$$

We show below, however, that no operation increases C by more than $\mathcal{O}(\log k)$ times the cost of the operation. Therefore, the total cost will have to be at least $F / \log k = \Omega(n \log n / \log k)$.[2]

The main operation to consider is the reorganization of $k' \le k$ buckets containing a total of $n' \le n$ items. By definition, the cost of the operation is n'. And, by the Complexity-range Lemma again, the increase in C is indeed at most

$$n' \log k' \le n' \log k = \mathcal{O}(n' \log k).$$

[2]For bucketing to be nontrivial, n and k have to be at least 2. In that case, $\log n$ and $\log k$ are safely positive if we use some logarithmic base strictly between 1 and 2.

The only other operation is insertion into an n'-item bucket. If $n' = 0$, then there is no change in C, so assume $n' \geq 1$. Then the cost is exactly $n' + 1$, and the increase in C is exactly

$$\begin{aligned}(n'+1)\log(n'+1) - n'\log n' \\ &= \log(n'+1) + n'(\log(n'+1) - \log n') \\ &= \log(n'+1) + n'\mathcal{O}(1/n'),\end{aligned}$$

which is certainly $\mathcal{O}((n'+1)\log k)$.

4. Tight lower bound for prefix bucketing. This section is devoted to a proof of the following tight lower bound.

THEOREM 4.1. *The cost for prefix bucketing of n items into $k = \mathcal{O}(\log n)$ buckets is $\Omega(n \log n)$.*

Recall that we need only consider algorithms each of whose redistributions moves all items involved into the very last bucket of the chosen prefix.

For our proof, it will be convenient to have terminology for the current configuration of (a prefix of) a bucket list and to generalize to allow fractional numbers of items in a bucket. If k is a positive integer and n is a positive real number, then an (n,k)-*configuration* is a list $L = (n_1, \ldots, n_k)$ of k nonnegative real numbers such that $\sum n_i = n$. We regard n_i as the (possibly fractional) number of items in bucket i.

For each (n,k)-configuration $L = (n_1, \ldots, n_k)$, we again define

$$C(L) = \sum_{i=1}^{k} n_i \log n_i,$$

as in the previous section's proof. The rest would be easy again if we could show now that no operation increases C by more than some constant times the cost of the operation. Unfortunately, there are counterexamples: The operation that transforms the configuration $(n/k, \ldots, n/k, n/k)$ to $(0, \ldots, 0, n)$, for example, has cost n but increases C by $n \log k$.

What we can show is weaker but still sufficient: Enough of the operations to account for most of the total increase in C do satisfy the desired condition. Intuitively, the sort of counterexample given above cannot be followed soon by similar gains, because the zeroed positions would have to grow back first. The idea for our proof, therefore, is to show that the troublesome operations use up some other sort of limited potential.

To this end, we define a second measure of complexity for each (n,k)-configuration $L = (n_1, \ldots, n_k)$:

$$M(L) = \sum_{i=1}^{k} i n_i.$$

When C makes its best progress (as in our counterexample above), M does even better (increasing by $\Omega(nk)$ in the example). But the potential for M to do better is limited by the fact that its total growth is not large compared to that of C. It follows that we can account for most of the increase in C by focusing on operations for which the increase in M is not very large compared to the increase in C. We will show that the very worst such cases involve very specific "before" configurations (and of course very specific "after" configurations, because the redistributions we consider move all

items to the last bucket involved), for which we can show by relatively straightforward calculation that the change in C is at most some constant times the number of items involved (i.e., the cost), as desired.

Now let us follow our plan more carefully. For n and k as in the theorem, let d be a constant so large that $k \le d \log n$. Assuming n is large enough so that $d \log n \le n^{1/2}$, the complexity C starts out at 0 and finally reaches a value

$$C_{\text{final}} \ge n \log n - n \log k \ge \frac{1}{2} n \log n.$$

The measure M starts out at 0, grows monotonically, and finally reaches a value

$$M_{\text{final}} \le kn \le dn \log n.$$

Over all, therefore, the increase in C is at least $\frac{1}{2d}$ times the increase in M. Consider the steps (both insertion steps and redistribution steps) on which we have

$$\Delta C < \frac{1}{4d} \Delta M,$$

where ΔC and ΔM are the respective associated increases in C and M on the step. Such steps can account for at most half the overall change in C. Therefore, we can restrict attention to the other steps, on each of which we must have

$$\Delta M \le 4d \Delta C.$$

We show that on each such step, regardless of its context, ΔC is at most some constant times the number of items involved, which is the bucketing cost, and hence that the total bucketing cost for such steps has to be $\Omega(n \log n)$. We saw at the end of section 3 that this fact holds for every insertion step, so we restrict further attention to the analysis of redistribution steps.

We directly analyze such a redistribution step only when the configuration L of the involved-bucket prefix is of a special form. (This is where fractional numbers of items are convenient.) The first sequence of lemmas below, culminating in Lemma 4.4, shows that it is no loss of generality to restrict attention to this form; and the final lemmas provide the needed estimates involving ΔC and ΔM when L is of this form. (Because it completely determines the redistribution (all items to the furthest involved bucket), L does determine both ΔC and ΔM, which we thus denote $(\Delta C)(L)$ and $(\Delta M)(L)$ in those final lemmas.)

Call an (n, k)-configuration $L = (n_1, \ldots, n_k)$ *nondecreasing* if $n_i \le n_{i+1}$ holds for every $i < k$, and call it *exponential, with ratio a and with k' initial 0's*, if $n_i = 0$ for every $i \le k'$ and $n_i = an_{i+1}$ for every $i \in \{k' + 1, \ldots, k - 1\}$.

LEMMA 4.2. *For each (n, k)-configuration L that is not nondecreasing, there is a nondecreasing (n, k)-configuration L' with $C(L') = C(L)$ and $M(L') > M(L)$.*

Proof. Reorder the configuration so that it is nondecreasing. □

LEMMA 4.3. *For each nondecreasing (n, k)-configuration L that is not exponential, there is a nondecreasing (n, k)-configuration L' with $C(L') < C(L)$ and $M(L') = M(L)$.*

Proof. First, note that we lose no generality if we assume $k = 3$. If $L = (n_1, \ldots, n_k)$ is nondecreasing but not exponential, then there has to be some $i \le k-2$ such that (n_i, n_{i+1}, n_{i+2}) is an $(n_i + n_{i+1} + n_{i+2}, 3)$-configuration with these same properties. It is clear from the definitions of C and M that the desired conclusion for (n_i, n_{i+1}, n_{i+2}) will yield the conclusion for L, too.

For $k = 3$, the idea is to take

$$L' = (n_1 - x, n_2 + 2x, n_3 - x)$$

for some nonzero x. Since L is not exponential and k is only 3, there can be no initial 0's. In the case that $n_1 > (n_2/n_3)n_2$, x must satisfy $0 < x < n_1$, and, in the case that $n_1 < (n_2/n_3)n_2$, it must satisfy $-n_2/2 < x < 0$. Whatever x is, we will have $M(L') = M(L)$. It remains only to show that some eligible x will yield $C(L') < C(L)$.

For each prospective x, let $C(x)$ denote the resulting value $C(L')$. It is enough to show that

$$\lim_{x \downarrow 0} C'(x) < 0 \qquad \text{if } n_1 > (n_2/n_3)n_2$$

and that

$$\lim_{x \uparrow 0} C'(x) > 0 \qquad \text{if } n_1 < (n_2/n_3)n_2.$$

Expressed more explicitly,

$$C(x) = f(n_1 - x) + f(n_2 + 2x) + f(n_3 - x),$$

where $f(x) = x \log x$. It is straightforward to check that the derivative $C'(x)$ does satisfy both requirements. □

LEMMA 4.4. *For each (n, k)-configuration L, there is a nondecreasing, exponential (n, k)-configuration L' with $C(L') \le C(L)$ and $M(L') \ge M(L)$.*

Proof. If the given configuration is not nondecreasing, then apply Lemma 4.2 one time. Then, calling the result L, consider the set $\mathcal{L}$ of nondecreasing (n, k)-configurations L' that satisfy $C(L') \le C(L)$ and $M(L') = M(L)$. Since C is continuous on the topologically compact set $\mathcal{L}$, there is some L' in $\mathcal{L}$ that minimizes C. By Lemma 4.3, that (n, k)-configuration must be exponential. □

Let $L_{n,k+k',a,k'}$ denote the nondecreasing, exponential $(n, k + k')$-configuration with ratio a and with k' initial 0's. Note that, for every k', $(\Delta C)(L_{n,k+k',a,k'})$ equals $(\Delta C)(L_{n,k,a,0})$, and $(\Delta M)(L_{n,k+k',a,k'})$ equals $\Delta M(L_{n,k,a,0})$. (The first equality is trivial, because the two "before" values of C are exactly the same and the two "after" values of C are exactly the same. In the case of M, however, both the "before" values and the "after" values do differ, but the differences are the same: nk' (k' for each full item).)

LEMMA 4.5. *For each n, k, and a,*

$$(\Delta C)(L_{n,k,a,0}) = \left(\log A - \frac{B}{A} \log a\right) n \quad \text{and}$$
$$(\Delta M)(L_{n,k,a,0}) = \left(\frac{B}{A} - 1\right) n,$$

where

$$A = \sum_{i=1}^{k} a^i \quad \text{and} \quad B = \sum_{i=1}^{k} i a^i.$$

Proof. The calculations are exact and easy, each change being to $(0, \ldots, 0, n)$ from $(a^k n/A, \ldots, a^2 n/A, an/A)$. The increase in M, for example, is

$$\begin{aligned} kn - \sum_{i=1}^{k} i a^{k+1-i} n/A &= kn - \sum_{i=1}^{k} (k+1-i) a^i n/A \\ &= kn - (k+1)An/A + Bn/A \\ &= \left(\frac{B}{A} - 1\right) n. \qquad \square \end{aligned}$$

COROLLARY 4.6. *If $a < 1$, then*

$$(\Delta C)(L_{n,k,a,0}) < \left(\log \frac{1}{1-a}\right) n.$$

Proof. Use the estimates

$$A < a/(1-a),\ B > a, \quad \text{and} \quad a \log a < 0. \qquad \square$$

COROLLARY 4.7. *For each fixed k,*

$$\begin{aligned} &\lim_{a \uparrow 1} (\Delta C)(L_{n,k,a,0})/n = (\Delta C)(L_{n,k,1,0})/n = \log k \quad \textit{and} \\ &\lim_{a \uparrow 1} \frac{(\Delta M)(L_{n,k,a,0})}{(\Delta C)(L_{n,k,a,0})} = \frac{(\Delta M)(L_{n,k,1,0})}{(\Delta C)(L_{n,k,1,0})} = \frac{k-1}{2 \log k}. \end{aligned}$$

LEMMA 4.8. *There exists a pair of thresholds, $a_0 < 1$ and k_0, such that, whenever $a_0 < a \le 1$ and $k > k_0$,*

$$\frac{(\Delta M)(L_{n,k,a,0})}{(\Delta C)(L_{n,k,a,0})} > 4d.$$

Proof. Choose k_0 large, and then choose $a_0 < 1$ large in terms of that k_0. The proof is by induction on $k \ge k_0$.

The base case, that

$$\frac{(\Delta M)(L_{n,k_0,a,0})}{(\Delta C)(L_{n,k_0,a,0})}$$

exceeds $4d$ whenever $a_0 < a \le 1$, follows from Corollary 4.7. For the induction step, it is enough to show that

$$\frac{(\Delta M)(L_{n,k+1,a,0}) - (\Delta M)(L_{n,k,a,0})}{(\Delta C)(L_{n,k+1,a,0}) - (\Delta C)(L_{n,k,a,0})}$$

exceeds $4d$ whenever $a_0 < a \le 1$ and $k \ge k_0$.

In terms of A and B, the goal is for the following to exceed $4d$:

$$\begin{aligned} E &= \frac{[(B + (k+1)a^{k+1}) - (A + a^{k+1})] - [B - A]}{[(A + a^{k+1}) \log(A + a^{k+1}) - (B + (k+1)a^{k+1}) \log a] - [A \log A - B \log a]} \\ &= \frac{k a^{k+1}}{A \log(1 + a^{k+1}/A) + a^{k+1} \log(A + a^{k+1}) - (k+1) a^{k+1} \log a}. \end{aligned}$$

Since $0 < a \le 1$ and k is large, $a^{k+1}/A \le 1/k$ is small enough that

$$\log_e(1 + a^{k+1}/A) < a^{k+1}/A \text{ or}$$
$$\log(1 + a^{k+1}/A) < (\log e)a^{k+1}/A,$$

where e is the base of the natural logarithms. Since $A \le k$ and $a^{k+1} \le 1$, we certainly have

$$\log(A + a^{k+1}) \le \log(k+1).$$

Substituting these estimates, and canceling a^{k+1}, we get

$$E > \frac{k}{\log e + \log(k+1) - (k+1)\log a}.$$

Since k and a are large, this estimate finally does clearly exceed $4d$. □

The following corollary is just what we need to complete the proof.

COROLLARY 4.9. *If (n,k)-configuration L satisfies*

$$\frac{(\Delta M)(L)}{(\Delta C)(L)} \le 4d,$$

then it also satisfies $(\Delta C)(L) = \mathcal{O}(n)$, where the implicit constant depends on neither n nor k.

Proof. By Lemma 4.4, since the conditions

$$C(L') \le C(L) \text{ and } M(L') \ge M(L)$$

respectively imply

$$(\Delta C)(L') \ge (\Delta C)(L) \text{ and } (\Delta M)(L') \le (\Delta M)(L),$$

it suffices to prove this when L is nondecreasing and exponential, say with ratio a, and with no initial 0's.

We deal with three separate cases: a "small" and k arbitrary, a "large" but k "small," and a and k both "large." First, however, we have to specify appropriate small-large thresholds. Recall the thresholds a_0 and k_0 from Lemma 4.8. For each $k \le k_0$ ($k \ge 2$), the first part of Corollary 4.7 lets us select a threshold $a_k < 1$ such that

$$(\Delta C)(L_{n,k,a,0})/n \le 1 + \log k$$

holds whenever $a_k < a \le 1$. Take $a_{\max} = \max_{0 \le k \le k_0} a_k$. We use $a_{\max}$ and k_0 as our small-large thresholds.

Whenever $a \le a_{\max}$, Corollary 4.6 yields

$$(\Delta C)(L) \le \left(\log \frac{1}{1-a}\right) n \le \left(\log \frac{1}{1-a_{\max}}\right) n = \mathcal{O}(n).$$

Whenever $k \le k_0$ and $a > a_{\max} \ge a_k$, we have made sure that

$$(\Delta C)(L) \le (1 + \log k)n \le (1 + \log k_0)n = \mathcal{O}(n).$$

And, whenever $k > k_0$ and $a > a_{\max} \ge a_0$, Lemma 4.8 ensures that the ratio $(\Delta M)(L)/(\Delta C)(L)$ exceeds $4d$. □

5. Further discussion. When the number of usable labels is not at least $n^{1+\epsilon}$ for some $\epsilon > 0$, the known upper bounds are not as low. With $\mathcal{O}(n)$ labels and exactly n labels, the respective bounds are $\mathcal{O}(n \log^2 n)$ [IKR81] and $\mathcal{O}(n \log^3 n)$ [AL90]. These bounds seem tight, and it can be shown that they are tight for smooth relabeling strategies [DZ90, Zh93, DSZ05]; however, we do not yet see how to extend these results to nonsmooth strategies, for which our $\Omega(n \log n)$ is still the only known lower bound. More generally, we would like a tight bound that is some nice *function*, say F, of the number n of usable labels, with $F(n) = \Theta(n \log^3 n)$, $F(cn) = \Theta(n \log^2 n)$ for each particular $c > 1$ and $F(n^{1+\epsilon}) = \Theta(n \log n)$ for each particular $\epsilon > 0$.

When the density of the labels in use grows large, the cost of further labeling becomes more closely related to an alternative natural cost measure: the number of labels spanned (rather than the number of items). Many of the same questions can be asked of this cost measure, and the answers and arguments might be independently interesting and enlightening. It turns out that the strongest version of the $\Omega(n \log^2 n)$ lower bound mentioned above (for smooth insertion of n items into a linearly bounded label space) is most natural in this setting, because then it turns out to hold regardless of the size of the label space; the lower bound on the standard cost follows as a corollary [DSZ05].

Even if it turns out that bucketing problems are not as closely related to online labeling for smaller numbers of labels, we would like to see tighter and more general analyses of their complexity as well. For $k = o(\log n)$ buckets, Jingzhong Zhang has proposed, via personal communication, a prefix-bucketing algorithm of cost $\mathcal{O}(n^{1+1/k}(k!)^{1/k})$. More careful analysis of his algorithm yields an expression that may be the exact optimum.

We also suspect that there are related continuous problems worthy of attention. Such problems, for example, might model the management of snow banks beside a path being plowed during an ongoing very large snow storm.

Acknowledgments. We thank Jun Tarui, Ioan Macarie, and two anonymous referees for their criticisms, corrections, and suggestions.

REFERENCES

[AL90] A. Andersson and T. W. Lai, *Fast updating of well-balanced trees*, in Proceedings of the 2nd Scandinavian Workshop on Algorithm Theory (SWAT '90), Lecture Notes in Comput. Sci. 447, Springer-Verlag, Berlin, 1990, pp. 111–121.

[Di82] P. F. Dietz, *Maintaining order in a linked list*, in Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing, ACM, New York, 1982, pp. 122–127.

[DS87] P. F. Dietz and D. D. Sleator, *Two algorithms for maintaining order in a list*, in Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing, ACM, New York, 1987, pp. 365–372.

[DSZ94] P. F. Dietz, J. I. Seiferas, and J. Zhang, *A tight lower bound for on-line monotonic list labeling*, in Proceedings of the 4th Scandinavian Workshop on Algorithm Theory (SWAT '94), Lecture Notes in Comput. Sci. 824, Springer-Verlag, Berlin, 1994, pp. 131–142.

[DSZ05] P. F. Dietz, J. I. Seiferas, and J. Zhang, *Lower Bounds for Smooth List Labeling*, manuscript.

[DZ90] P. F. Dietz and J. Zhang, *Lower bounds for monotonic list labeling*, in Proceedings of the 2nd Scandinavian Workshop on Algorithm Theory (SWAT '90), Lecture Notes in Comput. Sci. 447, Springer-Verlag, Berlin, 1990, pp. 173–180.

[HS66] F. C. Hennie and R. E. Stearns, *Two-tape simulation of multitape Turing machines*, J. Assoc. Comput. Mach., 13 (1966), pp. 533–546.

[IKR81] A. Itai, A. G. Konheim, and M. Rodeh, *A Sparse Table Implementation of Sorted Sets*, Research Report RC 9146, IBM Thomas J. Watson Research Center, Yorktown Heights, 1981.

[Ts84] A. K. Tsakalidis, *Maintaining order in a generalized linked list*, Acta Inform., 21 (1984), pp. 101–112.

[Zh93] J. Zhang, *Density Control and On-Line Labeling Problems*, Tech. Report 481 and Ph.D. thesis, University of Rochester, Rochester, NY, 1993.

SIAM J. DISCRETE MATH.
Vol. 18, No. 3, pp. 638–646

BRIDGING SEPARATIONS IN MATROIDS*

JIM GEELEN†, PETR HLINĚNÝ‡, AND GEOFF WHITTLE‡

Abstract. Let (X_1, X_2) be an exact k-separation of a matroid N. If M is a matroid that contains N as a minor and the k-separation (X_1, X_2) does not extend to a k-separation in M, then we say that M *bridges* the k-separation (X_1, X_2) in N. One would hope that a minor minimal bridge for (X_1, X_2) would not be much larger than N. Unfortunately there are instances in which one can construct arbitrarily large minor-minimal bridges. We restrict our attention to the class of matroids representable over a fixed finite field and show that here minor-minimal bridges are bounded in size.

Key words. matroids, connectivity, blocking sequences

AMS subject classification. 05B35

DOI. 10.1137/S089548010139638X

1. Introduction.

Seymour's Decomposition Theorem [4] states that any regular matroid can be obtained from graphic matroids, cographic matroids, and copies of R_{10} using 1-, 2-, and 3-sums. The main step in the proof of this remarkable theorem is to prove that

(1) *If M is a 3-connected regular matroid that is neither graphic nor cographic, then M contains a minor isomorphic to R_{10} or R_{12}.*

The matroids R_{10} and R_{12} are particular 3-connected regular matroids that are neither graphic nor cographic. It is easy to handle the regular matroids containing R_{10}.

(2) *If M is a 3-connected regular matroid that contains R_{10} as a minor, then $M = R_{10}$.*

Somewhat more complicated structures arise when considering R_{12}. Let N be a matroid with an exact k-separation (X_1, X_2), and let M be a matroid containing N as a minor. If there exists a k-separation (Y_1, Y_2) of M where $X_1 \subseteq Y_1$ and $X_2 \subseteq Y_2$, then we say that the k-separation (X_1, X_2) of N is *induced* in M. If (X_1, X_2) is not induced in M, then we say that M *bridges* the k-separation (X_1, X_2) in N.

(3) *R_{12} has a 3-separation (X_1, X_2) such that $|X_1|, |X_2| = 6$. Moreover, if M is a regular matroid that contains R_{12} as a minor, then the 3-separation (X_1, X_2) of R_{12} is induced in M.*

The proof of (2), and of results like (2), is reduced to an elementary finite case check by Seymour's Splitter Theorem [4]. However, there is no satisfactory analogue of Seymour's Splitter Theorem that can be applied to prove results like (3). We are interested in minor-minimal matroids that bridge the k-separation (X_1, X_2) in N. Unfortunately, in some cases such matroids are arbitrarily large. Nevertheless, Seymour [4] and Geelen, Gerards, and Kapoor [1] have shown that such matroids are highly structured (see Theorem 3.4). The main result of this paper is that when

*Received by the editors October 12, 2001; accepted for publication (in revised form) August 10, 2004; published electronically April 8, 2005.
http://www.siam.org/journals/sidma/18-3/39638.html

†Department of Combinatorics and Optimization, University of Waterloo, Waterloo N2L 3G1, ON, Canada (jfgeelen@math.uwaterloo.ca).

‡School of Mathematical and Computing Sciences, Victoria University, Wellington, New Zealand (hlineny@member.ams.org, whittle@mcs.vuw.ac.nz). Current address of Petr Hliněný: Department of Computer Science (FEI VŠB), Technical University Ostrava, 17. listopadu 15, 708 33 Ostrava, Czech Republic.

we restrict our attention to matroids over a fixed finite field, the situation improves significantly.

THEOREM 1.1. *For any finite field $\mathbb{F}$ and integer k there exists an integer n such that if (X_1, X_2) is an exact k-separation in an $\mathbb{F}$-representable matroid N and M is a minor-minimal $\mathbb{F}$-representable matroid that bridges the k-separation (X_1, X_2) in N, then $|E(M)| \leq |E(N)| + n$.*

We actually prove a stronger result, Theorem 7.1, that gives an explicit bound on n. This reduces the proof of (3) to a finite case check. Of course this is not practical as the number n is quite large. Nevertheless, we hope to use these results in subsequent papers to obtain excluded minor characterizations.

For 1- and 2-separations we obtain stronger results that are independent of representation. The first of these follows readily from results of Lemos and Oxley [3].

THEOREM 1.2. *If (X_1, X_2) is a separation in a matroid N and M is a minor-minimal matroid that bridges the separation (X_1, X_2) in N, then $|E(M)| \leq |E(N)| + 2$.*

THEOREM 1.3. *If (X_1, X_2) is an exact 2-separation in a matroid N and M is a minor-minimal matroid that bridges the 2-separation (X_1, X_2) in N, then $|E(M)| \leq |E(N)| + 5$.*

There is no analogue of Theorems 1.2 and 1.3 for 3-separations. Nevertheless, while there may be arbitrarily large minor-minimal bridging matroids, we can bound the branch-width of such matroids. (Branch-width is defined in section 8.)

THEOREM 1.4. *Let (X_1, X_2) be an exact k-separation in matroid N with branch-width n. If M is a minor-minimal matroid that bridges the k-separation (X_1, X_2) in N, then M has branch-width at most $n + k$.*

Let $(M_1, M_2, \ldots)$ be an infinite sequence of matroids each of which is representable over the same finite field and each with branch-width at most n. In [2] it is proved that there exists $i < j$ such that M_i is isomorphic to a minor of M_j. Combining this with Theorem 1.4, we can obtain a result similar to Theorem 1.1. However, there are two differences. First, in Theorem 1.1 we keep N as a minor while the other approach keeps a minor isomorphic to N. More importantly, we obtain an explicit upper-bound on the size of M, which cannot be done using the methods in [2].

2. Tutte's Linking Theorem. Let M be a matroid. For any subset A of $E(M)$ we let

$$\lambda_M(A) := r_M(A) + r_M(E(M) - A) - r_M(E(M));$$

λ_M is the *connectivity function* of M. For sets $A, B \subseteq E(M)$, we have

(i) $\lambda_M(A) = \lambda_M(E(M) - A)$,
(ii) $\lambda_M(A) \leq \lambda_M(A \cup \{e\}) + 1$ for each $e \in E(M)$, and
(iii) $\lambda_M(A) + \lambda_M(B) \geq \lambda_M(A \cup B) + \lambda_M(A \cap B)$.

If (X_1, X_2) is a partition of $E(M)$ such that $|X_1|, |X_2| \geq k$ and $\lambda_M(X_1) < k$, then we call (X_1, X_2) a *k-separation* of M. If, in addition, $\lambda_M(X_1) = k - 1$, then we call (X_1, X_2) an *exact k-separation* of M.

Let N be a minor of M and let (X_1, X_2) be an exact k-separation of N. We let $\kappa_M(X_1, X_2) = \min(\lambda_M(A) : X_1 \subseteq A \subseteq E(M) - X_2)$. Thus, M bridges (X_1, X_2) if and only if $\kappa_M(X_1, X_2) \geq k$. Note that if M' is a minor of M and $X_1, X_2 \subseteq E(M')$, then $\kappa_{M'}(X_1, X_2) \leq \kappa_M(X_1, X_2)$. The following theorem provides a good characterization for $\kappa(X_1, X_2)$; this theorem is in fact a generalization of Menger's theorem.

THEOREM 2.1 (Tutte's Linking Theorem [5]). *Let M be a matroid and let X_1, X_2 be disjoint subsets of $E(M)$. Then, there exists a minor M' of M such that $E(M') = X_1 \cup X_2$ and $\lambda_{M'}(X_1) = \kappa_M(X_1, X_2)$.*

We obtain the following easy corollary.

COROLLARY 2.2. *Let N be a matroid with an exact k-separation (X_1, X_2), and let M be a minor-minimal matroid that bridges the k-separation (X_1, X_2) in N. For any $e \in E(M) - E(N)$ either*

1. *$\kappa_{M\backslash e}(X_1, X_2) < k$ and N is not a minor of M/e, or*
2. *$\kappa_{M/e}(X_1, X_2) < k$ and N is not a minor of $M\backslash e$.*

By Corollary 2.2, there exists a unique partition (S, T) of $E(M) - E(N)$ such that $N = M\backslash S/T$. However, any minor can be obtained by contracting an independent set and deleting a coindependent set.

COROLLARY 2.3. *Let N be a matroid with an exact k-separation (X_1, X_2), and let M be a minor-minimal matroid that bridges the k-separation (X_1, X_2) in N. If $N = M\backslash S/T$, then S is coindependent and T is independent.*

We also require the following technical lemma.

LEMMA 2.4. *Let M be a matroid, let (Y_1, Y_2) be a partition of $E(M)$, and let $X_1 \subseteq Y_1$ and $X_2 \subseteq Y_2$. If $\kappa_M(X_1, Y_2) = \lambda_M(Y_2)$ and $\kappa_M(Y_1, X_2) = \lambda_M(Y_1)$, then $\kappa_M(X_1, X_2) = \lambda_M(Y_1)$.*

Proof. Let Y be a set such that $\lambda_M(Y) = \kappa_M(X_1, X_2)$ and $X_1 \subseteq Y \subseteq E(M) - X_2$. By submodularity, we have

$$\begin{aligned}
\kappa_M(X_1, X_2) &= \lambda_M(Y) \\
&\geq \lambda_M(Y \cap Y_1) + \lambda_M(Y \cup Y_1) - \lambda_M(Y_1) \\
&\geq \kappa_M(X_1, Y_2) + \kappa_M(Y_1, X_2) - \lambda_M(Y_1) \\
&= \lambda_M(Y_2) + \lambda_M(Y_1) - \lambda_M(Y_1) \\
&= \lambda_M(Y_2) \\
&\geq \kappa_M(X_1, X_2).
\end{aligned}$$

Thus, $\kappa_M(X_1, X_2) = \lambda_M(Y_1)$, as required. □

3. Blocking sequences. In this section we review results from [1], but we use slightly different notation; similar results are given in [4]. Let N be a minor of a matroid M, and let $X = E(N)$. Then there exists a coindependent set S and an independent set T such that $N = M\backslash S/T$. Therefore, there exists a basis B of M such that $T \subseteq B \subseteq E(M) - S$. For any subset Y of $E(M)$, we define

$$M[Y, B] := M\backslash(E(M) - (Y \cup B))/(B - Y).$$

Thus $M[Y, B]$ is the minor of M on ground set Y obtained by contracting only elements of B and deleting only elements of $E(M) - B$. In particular, $N = M[X, B]$.

Let (X_1, X_2) be an exact k-separation in N. A sequence $v_1, \ldots, v_p \in E(M)$ is a *blocking sequence* for the k-separation (X_1, X_2) of N, with respect to B, if

1.(a) $\lambda_{M[X \cup \{v_1\}, B]}(X_1) \geq k$,
(b) $\lambda_{M[X \cup \{v_p\}, B]}(X_1 \cup \{v_p\}) \geq k$,
(c) for all $i \in \{1, \ldots, p-1\}$, we have $\lambda_{M[X \cup \{v_i, v_{i+1}\}, B]}(X_1 \cup \{v_i\}) \geq k$, and
2. no proper subsequence of $v_1, \ldots, v_p$ satisfies 1.

If there is a blocking sequence for (X_1, X_2), then M clearly bridges (X_1, X_2). The converse is also true and is proved in [1, Theorem 4.14].

THEOREM 3.1. *Let B be a basis of the matroid M, let $N = M[X, B]$, and let (X_1, X_2) be an exact k-separation of N. Then, M bridges the k-separation (X_1, X_2) in N if and only if there exists a blocking sequence for (X_1, X_2) with respect to B.*

The following propositions give additional properties of blocking sequences; the first follows easily from the definitions, while the second is proved in [1, Proposition 4.15].

PROPOSITION 3.2. *Let B be a basis of the matroid M, let $N = M[X, B]$, and let $v_1, \ldots, v_p$ be a blocking sequence, with respect to B, for an exact k-separation (X_1, X_2) of N. Now, let $i, j \in \mathbb{Z}$, where $0 \leq i < j - 1 \leq p$; let $Y_1 \subseteq X_1 \cup \{v_1, \ldots, v_i\}$, where $X_1 \cup \{v_i\} \subseteq Y_1$; and let $Y_2 \subseteq X_2 \cup \{v_j, \ldots, v_p\}$, where $X_2 \cup \{v_j\} \subseteq Y_2$. Then, (Y_1, Y_2) is an exact k-separation in $M[Y_1 \cup Y_2, B]$, and $v_{i+1}, \ldots, v_{j-1}$ is a blocking sequence for this exact k-separation with respect to B.*

PROPOSITION 3.3. *Let B be a basis of the matroid M, let $N = M[X, B]$, and let $v_1, \ldots, v_p$ be a blocking sequence, with respect to B, for an exact k-separation (X_1, X_2) of N. Then, the sequence $v_1, \ldots, v_p$ alternates between elements of B and $E(M) - B$.*

In summary, we obtain the following theorem.

THEOREM 3.4. *Let N be a matroid with an exact k-separation (X_1, X_2), and let M be a minor-minimal matroid that bridges the k-separation (X_1, X_2) of N. Then there exists a unique partition (S, T) of $E(M) - E(N)$ such that $N = M \backslash S / T$. Moreover, there exists an ordering $v_1, \ldots, v_p$ of the elements in $E(M) - E(N)$ that alternates between elements of S and T such that, for each $i \in \{1, \ldots, p\}$,*

(i) *if $v_i \in S$, then $(X_1 \cup \{v_1, \ldots, v_{i-1}\}, X_2 \cup \{v_{i+1}, \ldots, v_p\})$ is a k-separation in $M \backslash v_i$, and*

(ii) *if $v_i \in T$, then $(X_1 \cup \{v_1, \ldots, v_{i-1}\}, X_2 \cup \{v_{i+1}, \ldots, v_p\})$ is a k-separation in M / v_i.*

4. Guts and coguts elements. We let $\mathrm{cl}_M(X)$ denote the closure of the set X in a matroid M. The *coclosure* of X, denoted $\mathrm{cl}^*_M(X)$, is the closure of X in M^*. If $e \notin X$, it is easy to show that $e \in \mathrm{cl}^*_M(X)$ if and only if $e \notin \mathrm{cl}_M(E(M) - (X \cup \{e\}))$. The following proposition is well known and straightforward.

PROPOSITION 4.1. *Let M be a matroid, let $X \subseteq E(M)$, and let $e \in E(M) - X$. Then*

(i) *$\lambda_{M/e}(X) < \lambda_M(X)$ if and only if $e \in \mathrm{cl}_M(X)$ and e is not a loop;*

(ii) *dually, $\lambda_{M \backslash e}(X) < \lambda_M(X)$ if and only if $e \in \mathrm{cl}^*_M(X)$ and e is not a coloop.*

Let (X_1, X_2) be an exact k-separation of M. An element e is in the *guts* of (X_1, X_2) if $e \in \mathrm{cl}_M(X_1 - \{e\})$ and $e \in \mathrm{cl}_M(X_2 - \{e\})$. Similarly, e is in the *coguts* of (X_1, X_2) if $e \in \mathrm{cl}^*_M(X_1 - \{e\})$ and $e \in \mathrm{cl}^*_M(X_2 - \{e\})$. Equivalently, e is in the coguts of (X_1, X_2) if $e \notin \mathrm{cl}_M(X_1 - \{e\})$ and $e \notin \mathrm{cl}_M(X_2 - \{e\})$.

The following proposition is also well known.

PROPOSITION 4.2. *Let M be a matroid, let (X_1, X_2) be a partition of $E(M)$, and let $e \in X_2$. Then*

(i) *$\lambda_M(X_1) < \lambda_M(X_1 \cup \{e\})$ if and only if $e \in \mathrm{cl}_M(X_2 - \{e\})$ and $e \notin \mathrm{cl}_M(X_1)$, and*

(ii) *$\lambda_M(X_1) = \lambda_M(X_1 \cup \{e\})$ if and only if e is either in the guts or in the coguts of (X_1, X_2).*

The following technical lemma is crucial.

LEMMA 4.3. *Let (X_1, X_2) be an exact k-separation of a matroid N, and let M be a minor-minimal matroid bridging the k-separation (X_1, X_2) of N. Moreover, let B be a basis of a matroid M such that $N = M[X_1 \cup X_2, B]$, let $v_1, \ldots, v_p$ be a blocking sequence for (X_1, X_2) with respect to B, and let $M' = M[X_1 \cup X_2 \cup \{v_2, \ldots, v_{p-1}\}, B]$.*

If $p \geq 2k+2$, then there exists $i \in \{2,3,\ldots,p-1\}$ such that $\kappa_{M'\backslash v_i}(X_1,X_2) = k-1$ and $\kappa_{M'/v_i}(X_1,X_2) = k-1$.

Proof. Given disjoint subsets A_1, A_2 of $E(M)$, we let $\sqcap_M(A_1,A_2) = r_M(A_1) + r_M(A_2) - r_M(A_1 \cup A_2)$. Thus, if (A_1,A_2) is a partition of $E(M)$, then $\sqcap_M(A_1,A_2) = \lambda_M(A_1)$. Moreover, it is straightforward to see that $\sqcap_M(A_1,A_2) \leq \kappa_M(A_1,A_2)$. We prove the stronger result that if $p \geq 2(k - \sqcap_M(X_1,X_2)) + 2$, then there exists $i \in \{2,\ldots,p-1\}$ such that $\kappa_{M'\backslash v_i}(X_1,X_2) = k-1$ and $\kappa_{M'/v_i}(X_1,X_2) = k-1$. By Proposition 3.2 and Lemma 2.4, we may assume that $p = 2(k - \sqcap_M(X_1,X_2)) + 2$. Note that $\sqcap_M(X_1,X_2) \leq k-1$, so $p \geq 4$.

By duality, we may assume that $v_1 \in B$; thus, by Proposition 3.3, $v_2 \notin B$ and $v_p \notin B$. Since N is a minor of $M'\backslash v_2$, we have $\kappa_{M'\backslash v_2}(X_1,X_2) = k-1$. Suppose that $\kappa_{M'/v_2}(X_1,X_2) < k-1$. Then there exists a $(k-1)$-separation (Y_1,Y_2) of M'/v_2 such that $X_1 \subseteq Y_1$ and $X_2 \subseteq Y_2$. Note that $\lambda_M(Y_1 \cup \{v_2\}) \leq \lambda_{M/v_2}(Y_1) + 1$ and $\kappa_M(X_1,X_2) = k$, so $(Y_1 \cup \{v_2\}, Y_2)$ is a k-separation of M'. Therefore, by the definition of a blocking sequence, $v_3 \in Y_1$. Similarly, we see that $v_4,\ldots,v_{p-1} \in Y_1$. Thus, $Y_1 = X_1 \cup \{v_3,\ldots,v_{p-1}\}$ and $Y_2 = X_2$.

By Proposition 4.2, $v_2 \in \mathrm{cl}_{M'}(X_2)$. Therefore, $v_2 \in \mathrm{cl}_M(X_2 \cup \{v_1\})$. Now, by Proposition 3.2, $(X_1 \cup \{v_1\}, X_2 \cup \{v_3,\ldots,v_p\})$ is a k-separation in $M\backslash v_2$. Thus, by Proposition 4.1, $v_2 \notin \mathrm{cl}_M(X_1 \cup \{v_1\})$. Similarly, $(X_1, X_2 \cup \{v_2,v_3,\ldots,v_p\})$ is a k-separation in M/v_1. Thus, by Proposition 4.1, $v_1 \in \mathrm{cl}_M(X_1)$. Let $X_1' = X_1 \cup \{v_2\}$, $X = X_1 \cup X_2$, and $X' = X \cup \{v_2\}$. Since $v_2 \notin \mathrm{cl}_M(X_1 \cup \{v_1\})$, we have $r_M(X_1') = r_M(X_1) + 1$ and, since $v_1 \in \mathrm{cl}_M(X_1)$ and $v_2 \in \mathrm{cl}_M(X_2 \cup \{v_1\})$, we have $r_M(X') = r_M(X)$. Hence, $\sqcap_M(X_1',X_2) > \sqcap_M(X_1,X_2)$. Moreover, by Proposition 3.2, $v_3,\ldots,v_p$ is a blocking sequence for the k-separation (X_1',X_2) in $M[X',B]$. Now let $M'' = M[X' \cup \{v_4,\ldots,v_{p-1}\}, B]$. Inductively, we find $i \in \{4,5,\ldots,p-1\}$ such that $\kappa_{M''\backslash v_i}(X_1',X_2) = k-1$ and $\kappa_{M''/v_i}(X_1',X_2) = k-1$. Now, the result follows by Lemma 2.4. ∎

5. Bridging 1- and 2-separations. In this section we prove Theorems 1.2 and 1.3.

Proof of Theorem 1.2. Let $X = E(N)$, let B be a basis of M such that $N = M[X,B]$, and let $v_1,\ldots,v_p$ be a blocking sequence for the separation (X_1,X_2) with respect to B. Suppose that $p \geq 3$, and let $M' = M[X \cup \{v_2\}]$. By the definition of a blocking sequence, $(X_1 \cup \{v_2\}, X_2)$ and $(X_1, X_2 \cup \{v_2\})$ are both separations of M'. Hence, N is a minor of both $M'\backslash v_2$ and M'/v_2. But then N is a minor of both $M\backslash v_2$ and M/v_2. So by Corollary 2.2, we obtain a contradiction. ∎

To prove Theorem 1.3 we require the following key lemma, whose proof we leave as an exercise.

LEMMA 5.1. *Let N be a minor of a matroid M, let (X_1,X_2) be an exact 2-separation of N, and suppose that $\lambda_M(X_1) = \lambda_M(X_2) = 1$. If N' is a minor of M such that $E(N') = X_1 \cup X_2$ and $\lambda_{N'}(X_1) = 1$, then $N' = N$.*

Proof of Theorem 1.3. Let $X = E(N)$, let B be a basis of M such that $N = M[X,B]$, and let $v_1,\ldots,v_p$ be a blocking sequence for the separation (X_1,X_2) with respect to B. Suppose that $p \geq 6$. By Proposition 3.2, we may assume that $p = 6$. Let $M' = M[X \cup \{v_2,v_3,v_4,v_5\}, B]$. By Lemma 4.3, there exists $i \in \{2,3,4,5\}$ such that $\kappa_{M'\backslash v_i}(X_1,X_2) = 1$ and $\kappa_{M'/v_i}(X_1,X_2) = 1$. Then, by Tutte's Linking Theorem and Lemma 5.1, N is a minor of both $M'\backslash v_i$ and M'/v_i. But then N is a minor of both $M\backslash v_i$ and M/v_i, contradicting Corollary 2.2. ∎

6. Bridging larger separations. In this section we give examples showing that there is no analogue of Theorems 1.2 and 1.3 for 3-separations. The same examples

also show that there is no analogue of Theorem 1.1 for infinite fields. In particular, we prove the following proposition.

PROPOSITION 6.1. *For any infinite field $\mathbb{F}$ and integer n, there exist $\mathbb{F}$-representable matroids N and M such that N has an exact 3-separation (X, Y), M is a minor-minimal matroid bridging this separation in N, and $|E(M)| \geq |E(N)| + n$.*

Let $p \geq n/2$ be an integer and let

$$A = \begin{array}{c} \\ x_1 \\ x_2 \\ y_3 \\ y_4 \\ u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_p \end{array} \begin{array}{c} \begin{array}{cccccccccc} y_1 & y_2 & x_3 & x_4 & v_1 & v_2 & \cdots & v_{p-1} & v_p \end{array} \\ \begin{pmatrix} 0 & 0 & -1 & 1 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 1 & 0 & \alpha_1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \alpha_2 & \alpha_3 & 0 & \ddots & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 & \alpha_{2p-2} & \alpha_{2p-1} \end{pmatrix} \end{array}.$$

Let $U = \{u_1, \ldots, u_p\}$ and $V = \{v_1, \ldots, v_p\}$, and let $D = A[U \cup \{y_3\}, V \cup \{x_3\}]$ (that is, D is the submatrix of A with rows indexed by $U \cup \{y_3\}$ and columns indexed by $V \cup \{x_3\}$). Now, choose $\alpha_1, \ldots, \alpha_{2p-1} \in \mathbb{F}$ so that each subdeterminant of D is nonzero unless it is identically zero as a polynomial in $\alpha_1, \ldots, \alpha_{2p-1}$.

Now, let M be the matroid represented over $\mathbb{F}$ by $[I|A]$, let $B = U \cup \{x_1, x_2, y_3, y_4\}$, let $X = \{x_1, x_2, x_3, x_4\}$, let $Y = \{y_1, y_2, y_3, y_4\}$, and let $N = M[B, X \cup Y]$. Note that $|E(M)| \geq |E(N)| + n$. Also, it is routine to check that (X, Y) is an exact 3-separation in N, and that $u_1, v_1, \ldots, u_p, v_p$ is a blocking sequence for (X, Y) with respect to B. Thus, M bridges the 3-separation (X, Y) in N; it remains to prove that M is minor-minimal with this property.

Claim. M is a minor-minimal matroid that bridges the 3-separation (X, Y) of N.

Proof. Suppose, for a contradiction, that there is a proper minor M' of M that bridges the 3-separation (X, Y) of N. Since M' is a minor of M, there exists a basis B' of M such that $M' = M[B', E(M')]$. Since N is a minor of M', we may assume without loss of generality that $E(N) \cap B' = \{x_1, x_2, y_3, y_4\}$. Since $u_1, v_1, \ldots, u_p, v_p$ is a blocking sequence with respect to B and M' is a *proper* minor of M, we see that $B \neq B'$. Now, since B' is a basis of M, $A[B - B', B' - B]$ is nonsingular. Note that $B - B' \subseteq U$ and $B' - B \subseteq V$. By our choice of $\alpha_1, \ldots, \alpha_{2p-1}$, we see that $A[(B - B') \cup \{y_3\}, (B' - B) \cup \{x_3\}]$ is nonsingular. Hence, $(B' - \{y_3\}) \cup \{x_3\}$ is a basis of M. So, $\{x_1, x_2, x_3, y_4\}$ is a basis of $M[B', E(N)] = N$. However, from A we can see that $\{x_1, x_2, x_3, y_4\}$ is not a basis of $M[B, E(N)] = N$. This contradiction completes the proof. □

7. Representation over finite fields. The difficulty when we go from 2-separations to 3-separations is that the analogue of Lemma 5.1 fails. Let N be a minor of a matroid M, let (X_1, X_2) be an exact 3-separation of N, and suppose that $\lambda_M(X_1) = \lambda_M(X_2) = 2$. If N' is a minor of M such that $E(N') = X_1 \cup X_2$ and $\lambda_{N'}(X_1) = 2$, then it need not be the case that $N' = N$. Lemma 5.1 essentially says that there is a unique way to compose two matroids at given points, but it is well known that there is no bound on the number of ways to compose two matroids along given lines. However, over a finite field, lines have bounded length, and hence there

is a bound on the number of ways to compose two representations along given lines. Similarly, there is also a bound on the number of ways to compose two representations on subspaces of any fixed dimension.

Throughout the remainder of this section we let $\mathbb{F}$ be a fixed finite field with q elements and we let $V(r,\mathbb{F})$ denote a vector space over $\mathbb{F}$ with rank r. Thus, the number of points in $V(r,\mathbb{F})$ is q^r. We let $n(r,q)$ denote the number of ordered bases of $V(r,\mathbb{F})$. It is straightforward to see that

$$n(r,q) = (q^r-1)(q^r-q)\cdots(q^r-q^{r-1}).$$

Let V_1 and V_2 be two rank-k subspaces of $V(r,\mathbb{F})$. Then the number of invertible linear transformations from V_1 to V_2 is $n(k,q)$. The following theorem is a strengthening of Theorem 1.1. (Sharper bounds can be obtained by using projective equivalence.)

THEOREM 7.1. *Let $\mathbb{F}$ be a finite field with q elements. If (X_1,X_2) is an exact k-separation in an $\mathbb{F}$-representable matroid N and M is a minor-minimal $\mathbb{F}$-representable matroid that bridges the k-separation (X_1,X_2) in N, then $|E(M)| \le |E(N)| + (2k+1)n(k-1,q)$.*

To make the proof of Theorem 7.1 rigorous, we need to be particular about the way we define representations. A *configuration over* $\mathbb{F}$ is a set of labelled elements in the vector space $V(r,\mathbb{F})$, for some integer r, where all labels are distinct, but a vector may receive more than one label. Two configurations are *isomorphic* if one can be obtained from the other by relabelling. Formally, a configuration is a pair $(E,\mathbb{V})$, where E is a finite set and $\mathbb{V} = V(r,\mathbb{F})$, with a function ψ from E to the set of vectors in $\mathbb{V}$. Let $\mathcal{L} : \mathbb{V} \to \mathbb{V}$ be a linear transformation. We let $\mathcal{L}(E,\mathbb{V})$ denote the configuration obtained by applying $\mathcal{L}$ to $\mathbb{V}$ and relabelling accordingly. That is, in $\mathcal{L}(E,\mathbb{V})$ an element $e \in E$ labels the vector $\mathcal{L}(\psi(e))$. If $\mathcal{L}$ is invertible, then we call $(E,\mathbb{V})$ and $\mathcal{L}(E,\mathbb{V})$ *equivalent*.

We associate a matroid M with a configuration $(E,\mathbb{V})$ in the natural way. That is, E is the ground set of M and for a set X of elements, $r_M(X)$ is the rank of X in $\mathbb{V}$. Thus, a matroid M is *representable* over a field $\mathbb{F}$ if it is induced by a configuration over $\mathbb{F}$ in this way. Note that equivalent configurations represent the same matroid.

The notion of minors extends naturally to configurations. Let $(E,\mathbb{V})$ be a configuration, let D and C be disjoint subsets of E, and let $\mathcal{L} : \mathbb{V} \to \mathbb{V}$ be a linear transformation whose kernel is the subspace spanned by C. We let $(E,\mathbb{V})\backslash D/C$ denote the configuration $\mathcal{L}(E-(D\cup C),\mathbb{V})$; we call any such configuration a *minor* of $(E,\mathbb{V})$. Obviously, if $(E,\mathbb{V})$ is a representation of M, then $(E,\mathbb{V})\backslash D/C$ is a representation of $M\backslash D/C$. Note that $\mathcal{L}$ is not uniquely defined, so $(E,\mathbb{V})\backslash D/C$ is not uniquely determined by D and C; but all such configurations are equivalent. When it is necessary to distinguish the particular linear transformation used, we say that $\mathcal{L}$ *projects* $(E,\mathbb{V})$ onto $(E,\mathbb{V})\backslash D/C$.

Proof of Theorem 7.1. Let $X = E(N)$, let $E = E(M)$, let B be a basis of M such that $N = M[X,B]$, and let $v_1,\dots,v_p$ be a blocking sequence for the separation (X_1,X_2) with respect to B. Let $n = n(k-1,q)$, and suppose that $p > n(2k+1)$. By Proposition 3.2, we may assume that $p = n(2k+1)+1$. For $i \in \{0,\dots,n\}$ let $u_i = v_{i(2k+1)+1}$ and, for $i \ne 0$, let $W_i = \{v_{(i-1)(2k+1)+2},\dots,v_{i(2k+1)}\}$. Thus $(\{u_0\},W_1,\{u_1\},\dots,W_n,\{u_n\})$ is a partition of $\{v_1,\dots,v_p\}$. Now let $E' = E - \{u_0,\dots,u_n\}$ and let $M' = M[E',B]$. For each $i \in \{0,\dots,n\}$, let $L_i = X_1 \cup (W_1 \cup \cdots \cup W_i)$ and $R_i = X_2 \cup (W_{i+1} \cup \cdots \cup W_n)$. Thus, (L_i,R_i) is a k-separation in M' for each $i \in \{0,\dots,n\}$. By Proposition 3.2 and Lemma 4.3, there exists $x_i \in W_i$ such that $\kappa_{M'\backslash x_i}(L_{i-1},R_i) = k-1$ and $\kappa_{M'/x_i}(L_{i-1},R_i) = k-1$ for each $i \in \{1,\dots,n\}$.

Consider a configuration $(E', \mathbb{V})$ representing M' over $\mathbb{F}$. For any $A \subseteq \mathbb{V}$ let $\langle A \rangle$ denote the span of A. Then, for $i \in \{0, \ldots, n\}$, let V_i denote $\langle L_i \rangle \cap \langle R_i \rangle$; thus V_i is a subspace of rank $k-1$. Now let $D_i = W_i - B$ and $C_i = W_i \cap B$. Note that N is a minor of $M' \backslash D_i / C_i$, so $\lambda_{M' \backslash D_i / C_i}(R_i) = k-1$. Note that, since $\lambda_{M'}(R_i) = k-1$, $\langle C_i \rangle$ intersects $\langle R_i \rangle$ trivially. Choose a linear transformation $\mathcal{L}_i$ such that the kernel of $\mathcal{L}_i$ is $\langle C_i \rangle$ and $\mathcal{L}_i$ acts as the identity on $\langle R_i \rangle$. Thus $\mathcal{L}_i$ projects $(E', \mathbb{V})$ onto $(E', \mathbb{V}) \backslash D_i / C_i$. Let π_i be the restriction of $\mathcal{L}_i$ to V_{i-1}. Note that π_i is an invertible linear transformation from V_{i-1} to V_i. Now let $\mathcal{L} = \mathcal{L}_n \mathcal{L}_{n-1} \cdots \mathcal{L}_0$. Thus, $\mathcal{L}(X, V)$ is a configuration representing N.

Recall that $\kappa_{M' \backslash x_i}(L_{i-1}, R_i) = k-1$ and $\kappa_{M'/x_i}(L_{i-1}, R_i) = k-1$. Therefore, there exists a partition (D_i', C_i') such that $\lambda_{M' \backslash D_i' / C_i'}(R_i) = k-1$ and x_i is in exactly one of D_i and D_i'. Choose a linear transformation $\mathcal{L}_i'$ such that the kernel of $\mathcal{L}_i'$ is $\langle C_i' \rangle$ and $\mathcal{L}_i'$ acts as the identity on R_i. Thus $\mathcal{L}_i'$ projects $(E', \mathbb{V})$ onto $(E', \mathbb{V}) \backslash D_i' / C_i'$. Let π_i' be the restriction of $\mathcal{L}_i'$ to V_{i-1}. Note that π_i' is an invertible linear transformation from V_{i-1} to V_i. Now, for $i \in \{0, \ldots, n\}$ we let $\sigma_i = (\pi_n' \cdots \pi_{i+1}')(\pi_i \cdots \pi_1)$. So σ_i is an invertible linear transformation from V_0 to V_{n-1}. The number of such distinct transformations is $n = n(k-1, q)$. Therefore, there exists $i > j$ such that $\sigma_i = \sigma_j$. Since each of these linear transformations is invertible, we see that $\pi_i' \pi_{i-1}' \cdots \pi_{j+1}' = \pi_i \pi_{i-1} \cdots \pi_{j+1}$; therefore

$$\pi_n \cdots \pi_{i+1} \pi_i' \cdots \pi_{j+1}' \pi_j \cdots \pi_1 = \pi_n \pi_{n-1} \cdots \pi_1.$$

Let

$$\mathcal{L}' = \mathcal{L}_n \cdots \mathcal{L}_{i+1} \mathcal{L}_i' \cdots \mathcal{L}_{j+1}' \mathcal{L}_j \cdots \mathcal{L}_1.$$

Now, $\mathcal{L}'(X, \mathbb{V})$ is equivalent to $\mathcal{L}(X, \mathbb{V})$, which is a representation of N. It follows that $M' \backslash x_i$ and M'/x_i both contain N as a minor. But then $M \backslash x_i$ and M/x_i both contain N as a minor, contradicting the minimality of M. □

8. Branch-width. A tree is *cubic* if all vertices have degree 1 or 3; the vertices with degree 1 are the *leaves*. A *branch-decomposition* of a matroid M on a finite ground set E is a cubic tree such that E labels a set of the leaves of T. (No leaf gets more than one label, but there may be unlabelled leaves.) The set *displayed* by a given subtree of T is the set of elements of E that label leaves of that subtree. A set of elements A of E is *displayed* by an edge e of T if it is displayed by one of the two components of $T \backslash e$; the *width* $\lambda(e)$ of the edge e of T is $\lambda_M(A) + 1$. The *width* of a branch-decomposition is the maximum of the widths of its edges, and the *branch-width* of M is the minimum among the widths of its branch-decompositions.

Let T be a branch-decomposition of M and let e be an edge of width k in T. There are two subsets A and B of E that are displayed by e. These two sets partition E, and, if $|A|, |B| \geq k$, then (A, B) is a k-separation of M; we say that such k-separations are *displayed* by T.

LEMMA 8.1. *Let (X_1, X_2) be an exact k-separation in a matroid N and let M be a minor-minimal matroid that bridges the k-separation (X_1, X_2) in N. If N has a branch-decomposition of width n that displays (X_1, X_2), then M has branch-width at most $n+1$.*

Proof. Let $X = E(N)$, let B be a basis of M such that $N = M[X, B]$, and let $v_1, \ldots, v_p$ be a blocking sequence for the separation (X_1, X_2) with respect to B. By duality we may assume that $v_1 \notin B$. Let T be a width-n branch-decomposition of N that displays (X_1, X_2) and let $e = ab$ be the edge of T that displays X_1 and

X_2. Now let T_a and T_b be the components of $T - e$ containing a and b, respectively. We construct a tree-decomposition T' of M as follows. Connect T_a to T_b with a path $P = (a, x_1, \ldots, x_p, b)$, and for each $i \in \{i, \ldots, p\}$ add a leaf, labelled v_i, adjacent to x_i. Note that e has width $k \leq n$ in T and, by Proposition 3.2, each edge of P has width $k+1$ in T'. The edges of T' incident with any of $v_1, \ldots, v_p$ all have width 2. Consider any other edge f of T'. By symmetry we may assume that f is an edge of T_a. Let A and B be the sets displayed by f in T', where $A \subseteq X_1$. Note that A is displayed by f in T, so $\lambda_N(A) \leq n$. By Proposition 3.2, $\lambda_N(X_1) = \lambda_{M\setminus v_1}(X_1)$. Therefore, since $A \subseteq X_1$, $\lambda_N(A) = \lambda_{M\setminus v_1}(A)$. Thus $\lambda_M(A) \leq \lambda_{M\setminus v_1}(A) + 1 = \lambda_N(A) + 1 \leq n + 1$. Therefore, T' has width at most $n + 1$, as required. □

Theorem 1.4 follows immediately from Lemma 8.1 and the next lemma.

LEMMA 8.2. *Let N be a matroid with branch-width n and let (X_1, X_2) be a k-separation of N. Then, there exists a branch-decomposition of N that displays (X_1, X_2) and that has width at most $n + k - 1$.*

Proof. Let T be a width-n branch-decomposition of N. We may assume that T has some unlabelled leaf r. Let s be the neighbor of r in T. Construct two copies T_1 and T_2 of $T - r$ such that for each vertex v of T the corresponding copies are labelled v_1 and v_2. Now construct a cubic tree T' by connecting T_1 and T_2 with the edge s_1s_2. We now make T' into a new branch-decomposition of N as follows. For each $i \in \{1, 2\}$ and $e \in X_i$, if e labels the leaf x in T, then we label the leaf x_i with e in T'. Therefore, X_1 and X_2 are displayed by s_1s_2 in T'.

Consider an edge f of $T - r$. Let A be the set that is displayed by the component of $T - f$ that does not contain s. Thus $A \cap X_1$ and $A \cap X_2$ are displayed by the copies of f in T'. Now, $\lambda_N(A \cap X_i) \leq \lambda_N(A) + \lambda_N(X_i) \leq n + (k - 1)$ for each $i \in \{1, 2\}$. Therefore, T' has width at most $n + k - 1$, as required. □

REFERENCES

[1] J. F. GEELEN, A. M. H. GERARDS, AND A. KAPOOR, *The excluded minors for GF(4)-representable matroids*, J. Combin. Theory Ser. B, 79 (2000), pp. 247–299.

[2] J. F. GEELEN, A. M. H. GERARDS, AND G. P. WHITTLE, *Branch-width and well-quasi-ordering in matroids and graphs*, J. Combin. Theory, Ser. B, 84 (2002), pp. 270–290.

[3] M. LEMOS AND J. OXLEY, *On packing minors into connected matroids*, Discrete Math., 189 (1998), pp. 283–289.

[4] P. D. SEYMOUR, *Decomposition of regular matroids*, J. Combin. Theory Ser. B, 28 (1980), pp. 305–359.

[5] W. T. TUTTE, *Menger's theorem for matroids*, J. Res. Nat. Bur. Standards Sect. B, 69 (1965), pp. 49–53.

SIAM J. DISCRETE MATH.
Vol. 18, No. 3, pp. 647–661

LOWER BOUNDS FOR h-VECTORS OF k-CM, INDEPENDENCE, AND BROKEN CIRCUIT COMPLEXES*

E. SWARTZ†

Abstract. We present a number of lower bounds for the h-vectors of k-Cohen–Macaulay (k-CM), broken circuit, and independence complexes. These lead to bounds on the coefficients of the characteristic and reliability polynomials of matroids. The main techniques are the use of series and parallel constructions on matroids and the short simplicial h-vector for pure complexes.

Key words. k-Cohen–Macaulay, matroid, h-vector, independence complex, broken circuit complex, short simplicial h-vector

AMS subject classifications. Primary, 05B35; Secondary, 52B05

DOI. 10.1137/S0895480102411358

1. Introduction.

Based on the ideas of Whitney [27] and Rota [21], the broken circuit complex of a graph was introduced by Wilf in "Which polynomials are chromatic?" [29]. Extended to matroids by Brylawski [9], its f-vector corresponds to the coefficients of the characteristic polynomial of the matroid. The h-vector encodes the same information in a different way. From the point of view of matroids, Wilf's original question becomes "What are the possible f-vectors, or, equivalently, h-vectors, of broken circuit complexes of matroids?"

Cohen–Macaulay complexes cover a wide variety of examples. In addition to the broken circuit and independence complexes of matroids covered here, Cohen–Macaulay complexes also include all triangulations of homology balls and spheres. In contrast to broken circuit complexes, the possible h-vectors (and hence f-vectors) of Cohen–Macaulay complexes have been completely characterized (see, for instance, [24, Theorem II.3.3, p. 59]). Introduced by Baclawski, doubly Cohen–Macualay complexes are Cohen–Macaulay complexes which lose neither a dimension nor the Cohen–Macaulay property when any vertex is removed. Spheres are doubly Cohen–Macaulay, but balls are not. More generally, a Cohen–Macaulay complex is k-Cohen–Macaulay (k-CM) if it retains its dimension and is still Cohen–Macaulay whenever $k-1$ or fewer vertices are removed. In addition to the independence complexes considered below, the order complex of a geometric lattice with the top and bottom points removed is k-CM if every line has at least k points [2].

The h-vectors of independence complexes of matroids are contained in the intersection of h-vectors of broken circuit complexes and k-CM complexes. Precisely, the cone on any independence complex is a broken circuit complex. In addition, if the smallest cocircuit of the matroid has cardinality k, then its independence complex is a k-CM complex. The close connection between h-vectors of independence complexes of matroids and reliability problems has been studied by a number of authors. See [13] for a recent survey.

Upper bounds on all of the above complexes have been studied. As they are all Cohen–Macaulay they share a common absolute upper bound of $h_i \leq \binom{n-r-1+i}{i}$,

*Received by the editors July 18, 2002; accepted for publication (in revised form) June 30, 2004; published electronically April 8, 2005. This work was partially supported by a VIGRE postdoc under NSF grant 9983660 to Cornell University.
http://www.siam.org/journals/sidma/18-3/41135.html
†Department of Mathematics, Cornell University, Ithaca, NY 14853 (ebs@math.cornell.edu).

where n is the number of vertices and $(r-1)$ is the dimension of the complex. In addition, they all satisfy the relative upper bound $h_{i+1} \leq h_i^{\langle i \rangle}$ (see section 4 for a definition of $h_i^{\langle i \rangle}$).

Our main purpose is to analyze absolute and relative lower bounds for the h-vectors of k-CM, broken circuit, and independence complexes. Section 2 contains the basic facts of the short-simplicial h-vector. The main tool for providing relative lower bounds is (2.6). The broken circuit and independence complexes of a matroid are described in section 3. Sections 4, 5, and 6 contain absolute and relative lower bounds for k-CM, broken circuit, and independence complexes, respectively.

Throughout the paper Δ is an $(r-1)$-dimensional simplicial complex with vertex set $V, |V| = n$. The link of a vertex $v \in V$ is $lk_\Delta v$, or just $lk\ v$ if no confusion is possible. We use $\Delta - v$ for the complex obtained by removing v and all of the faces which contain v from Δ. Similarly, if $A \subseteq V$, then $\Delta - A$ is the complex obtained by removing all of the vertices in A and any faces which contain one or more of those vertices.

2. Face enumeration. The combinatorics of a simplicial complex Δ can be encoded in several ways. The most direct is to let $f_i(\Delta)$ be the number of faces of cardinality i. For an $(r-1)$-dimensional complex the h-vector of Δ is the sequence $(h_0(\Delta), \ldots, h_r(\Delta))$, where

$$h_i(\Delta) = \sum_{j=0}^{i} (-1)^{i-j} \binom{r-j}{r-i} f_j(\Delta). \tag{2.1}$$

Equivalently,

$$f_j(\Delta) = \sum_{i=0}^{j} \binom{r-i}{r-j} h_i(\Delta). \tag{2.2}$$

By convention, $h_i(\Delta) = f_i(\Delta) = 0$ if $i < 0$ or $i > r$. The *short simplicial* h-vector was introduced in [16] as a simplicial analogue of the short cubical h-vector in [1]. It is the sum of the h-vectors of the links of the vertices. As far as we know, (2.5) was first stated in [17]. However, only a proof for shellable Δ was given there. So, we include a proof for arbitrary pure complexes for the sake of completeness.

DEFINITION 2.1. *Let Δ be a pure simplicial complex. Define*

$$\tilde{h}_i(\Delta) = \sum_{v \in V} h_i(lk\ v). \tag{2.3}$$

LEMMA 2.2 (see [16]). *Let Δ be a pure simplicial complex. For all i, $0 \leq i \leq r-1$,*

$$\tilde{h}_i(\Delta) = \sum_{j=0}^{i} (-1)^{i-j} (j+1) \binom{r-j-1}{r-i-1} f_{j+1}. \tag{2.4}$$

PROPOSITION 2.3. *Let Δ be a pure simplicial complex. Then,*

$$\tilde{h}_{i-1}(\Delta) = i\ h_i(\Delta) + (r-i+1) h_{i-1}(\Delta). \tag{2.5}$$

If $\dim(\Delta - v) = r-1$ for every vertex v, then

$$\sum_{v \in V} h_i(\Delta - v) = (n-i) h_i(\Delta) - (r-i+1) h_{i-1}(\Delta). \tag{2.6}$$

Proof. Combining (2.2) and (2.4),

$$\begin{aligned}\tilde{h}_{i-1}(\Delta) &= \sum_{j=0}^{i-1}(-1)^{i-j-1}(j+1)\binom{r-j-1}{r-i}\sum_{k=0}^{j+1}\binom{r-k}{r-j-1}h_k(\Delta)\\ &= \sum_{k=0}^{i}h_k(\Delta)\left\{\sum_{j=k-1}^{i-1}(-1)^{i-j-1}(j+1)\binom{r-j-1}{r-i}\binom{r-k}{r-j-1}\right\}\\ &= \sum_{k=0}^{i}h_k(\Delta)\left\{\sum_{j=k-1}^{i-1}(-1)^{i-j-1}(j+1)\binom{r-j-1}{i-j-1}\binom{r-k}{j+1-k}\right\}.\end{aligned}$$

Substituting $s = j-k+1$ and $t = i-j-1$,

$$\begin{aligned}\tilde{h}_{i-1}(\Delta) &= \sum_{k=0}^{i}h_k(\Delta)\left\{\sum_{s+t=i-k}(-1)^t(i-t)\binom{r+t-i}{t}\binom{r+s+t-i}{s}\right\}\\ &= \sum_{k=0}^{i}h_k(\Delta)\left\{\sum_{s+t=i-k}(-1)^t(i-t)\frac{A}{s!t!}\right\},\end{aligned}$$

where A is the falling factorial $(r-k)\cdot(r-k-1)\cdots(r-i+1)$.

For a fixed i, define c_k by

$$c_k = \sum_{s+t=i-k}(-1)^t(i-t)\frac{1}{s!t!}.$$

Equation (2.5) is equivalent to showing that $c_i = i$, $c_{i-1} = 1$, and $c_k = 0$ in all other cases. This can be seen by recognizing c_{i-k} as the kth term in the generating series for

$$(i+x)e^{-x}\cdot e^x = \left(\sum_{t=0}^{\infty}(-1)^t\frac{(i-t)}{t!}x^t\right)\left(\sum_{s=0}^{\infty}\frac{1}{s!}x^s\right).$$

In order to prove that (2.6) holds, we first notice that the hypothesis implies that $h_i(\Delta) = h_i(\Delta - v) + h_{i-1}(lk\ v)$ for every vertex v. Now sum this equation over all the vertices and apply (2.5). □

The above proposition makes precise the idea that, taken together, $h_{i-1}(\Delta)$ and $h_i(\Delta)$ measure the average contribution of $h_{i-1}(lk\ v)$ to $h_i(\Delta)$. Another consequence of (2.5) is that if the automorphism group of a pure $(r-1)$-dimensional complex Δ is transitive, or, more generally, if $h_{i-1}(lk\ v)$ is independent of v, then n divides $\{i\ h_i(\Delta) + (r-i+1)h_{i-1}(\Delta)\}$.

3. Broken circuit and independence complexes of matroids. We follow [19] for matroid terminology. Unless otherwise specified, M is always a rank r matroid with ground set E (or $E(M)$ if necessary) and $|E| = n$. There are many equivalent ways of defining matroids. The most convenient for us is the following.

A *matroid*, M, is a pair $(E, \mathcal{I}), E$ is a nonempty finite ground set, and $\mathcal{I}$ is a distinguished set of subsets of E. The members of $\mathcal{I}$ are called the *independent* subsets of M and are required to satisfy the following:

1. The empty set is in $\mathcal{I}$.
2. If B is an independent set and $A \subseteq B$, then A is an independent set.

3. If A and B are independent sets such that $|A| < |B|$, then there exists an element $x \in B - A$ such that $A \cup x$ is independent.

Matroid theory was introduced by Whitney [28]. The prototypical example of a matroid is a finite subset of a vector space with the canonical independent sets. Another example is the cycle matroid of a graph. Here the ground set is the edge set of the graph, and a collection of edges is independent if and only if it is acyclic.

An element e of a matroid is a *loop* if it is not contained in any independent set. The *circuits* of a matroid are its minimal dependent sets. Every loop of M is a circuit. A maximal independent set is called a *basis*, and any element which is contained in every basis is a *coloop* of the matroid. Every basis of M has the same cardinality. The *rank* of M, or $r(M)$, is that common cardinality. Similarly, the rank of a subset A of E is the cardinality of any maximal independent subset of A and is denoted $r(A)$. The *deletion* of M at e is denoted $M - e$. It is the matroid whose finite set is $E - e$ and whose independent sets are simply those members of $\mathcal{I}$ which do not contain e. The *contraction* of M at e is denoted M/e. It is a matroid whose ground set is also $E - e$. If e is a loop or a coloop of M, then $M/e = M - e$. Otherwise, a subset I of $E - e$ is independent in M/e if and only if $I \cup e$ is independent in M. Deletion and contraction for a subset A of E is defined by repeatedly deleting or contracting each element of A.

The *dual* of M is $M^\star$. It is the matroid whose ground set is the same as that of M and whose bases are the complements of the bases of M. For example, $U_{i,j}$ is the matroid defined by $E = \{1, 2, \ldots, j\}$ and $\mathcal{I} = \{A \subseteq E : |A| \leq i.\}$ So, $U^\star_{i,j} = U_{j-i,j}$.

Two nonloop elements $e, f \in E$ are *parallel* if they form a circuit. The relation "is parallel to" is an equivalence relation on E, and the corresponding equivalence classes are the parallel classes of M. If P is a parallel class of M, then for any $e \in P$ all of the members of $P - e$ are loops in M/e. A parallel class in $M^\star$ is a *series class* of M. If S is a series class of M, then for any $e \in S$, all of the members of $S - e$ are coloops in $M - e$.

Let $M = (E, \mathcal{I})$ and $M' = (E', \mathcal{I}')$ be two matroids with $E \cap E' = \emptyset$. Then $M \oplus M'$ is the direct sum of M and M'. It is the matroid whose ground set is $E \cup E'$ and whose independent sets are those subsets of the form $I \cup I', I \in \mathcal{I}, I' \in \mathcal{I}'$. A matroid is *connected* if it is not the direct sum of two smaller matroids. Every matroid can be written uniquely (up to order) as a direct sum $M = M_1 \oplus \cdots \oplus M_k$ of connected matroids. The *components* of M are the summands of this decomposition.

The *independence complex* of M is

$$\Delta(M) = \{A \subseteq E : A \text{ is independent}\}.$$

Evidently, $\Delta(M)$ is a pure $(r-1)$-dimensional complex, where r is the rank of M. In addition, $\Delta(M - e) = \Delta(M) - e$ and if e is not a loop of M, then $\Delta(M/e) = lk_{\Delta(M)}\, e$.

In order to define the broken circuit complex for M, we first choose a linear order ω on the elements of the matroid. Given such an order, a *broken circuit* is a circuit with its least element removed. The *broken circuit complex* is the simplicial complex whose simplices are the subsets of E which do not contain a broken circuit. We denote the broken circuit complex of M and ω by $\Delta^{BC}(M)$, or $\Delta^{BC}(M, \omega)$. Different orderings may lead to different complexes; see [3, Example 7.4.4]. However, $f_i(\Delta^{BC}(M, \omega))$ does not depend on ω (see Theorem 3.2 below). Conversely, distinct matroids can have the same broken circuit complex. For instance, let $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$, and let ω be the obvious order. Let M_1 be the matroid on E whose bases are all triples except $\{e_1, e_2, e_3\}$ and $\{e_4, e_5, e_6\}$ and let M_2 be the matroid on E whose bases are

all triples except $\{e_1, e_2, e_3\}$ and $\{e_1, e_5, e_6\}$. Then M_1 and M_2 are nonisomorphic matroids but their broken circuit complexes are identical.

In order to easily distinguish the h-vectors of $\Delta(M)$ and $\Delta^{BC}(M)$ we use the following notation.

DEFINITION 3.1. *Let M be a rank r matroid.*

- $h_i(M) = h_i(\Delta(M))$.
- $b_i(M) = h_{r-i}(\Delta^{BC}(M))$.
- $w_i(M) = f_{r-i}(\Delta^{BC}(M))$.
- $b_i^\star(M) = b_i(M^\star) = h_{n-r-i}(\Delta^{BC}(M^\star))$.

We will suppress the M when there is no danger of confusion. The invariants $h_i, b_i, w_i, b_i^\star$ are closely related to the Tutte polynomial of M. The *Tutte polynomial* is a two-variable polynomial invariant of M defined by

$$T(M; x, y) = \sum_{A \subseteq E} (x-1)^{r(M)-r(A)} (y-1)^{|A|-r(A)}.$$

THEOREM 3.2 (see [3]). *Suppose M has k components and j coloops. Then,*

(a) $T(M; x, 1) = h_0 x^r + h_1 x^{r-1} + \cdots + h_{r-j} x^j$;
(b) $T(M; x, 0) = b_r x^r + b_{r-1} x^{r-1} + \cdots + b_k x^k$;
(c) $T(M; 0, y) = b^\star_{n-r} y^{n-r} + \cdots + b^\star_k y^k$;
(d) $(-1)^r T(M; 1-x, 0) = w_0 x^r - w_1 x^{r-1} + \cdots + (-1)^r w_r$.

The w_i are the unsigned Whitney numbers of the first kind. The *characteristic polynomial* of M is $(-1)^r T(M; 1-x, 0)$. The characteristic polynomial of a matroid has a number of applications including graph coloring and flows, linear coding theory, and hyperplane arrangements. See [12] for a survey.

Properties (a)–(d) of b_i and h_i listed in Theorem 3.3 follow immediately from corresponding properties of the Tutte polynomial, which can be found in [11]. The parallel and series connection of two (pointed) matroids is described in [19, section 7.1].

THEOREM 3.3 (Tutte recursion).

(a) *If M has j coloops, then $h_i(M) = h_i(\tilde{M})$, where $\tilde{M}$ is M with the coloops deleted. In particular, $h_i(M) > 0$ if and only if $0 \le i \le r - j$.*
(b) *If M has k components and no loops, then $b_i > 0$ if and only if $k \le i \le r$.*
(c) *If e is neither a loop nor a coloop of M, then $h_i(M) = h_i(M - e) + h_{i-1}(M/e)$ and $b_i(M) = b_i(M - e) + b_i(M/e)$.*
(d) *If $M = M_1 \oplus M_2$, then $h_i(M) = \sum_{j+k=i} h_j(M_1) h_k(M_2)$ and $b_i(M) = \sum_{j+k=i} b_j(M_1) b_k(M_2)$.*
(e) *Suppose that P is a parallel class of M. Let $\tilde{M}$ be M with all but one element, say e, of P deleted. Then, $h_i(M) = h_i(\tilde{M}) + (|P| - 1) h_{i-1}(\tilde{M}/e)$.*
(f) *Let S be a series class of M. Let $\tilde{M}$ be M with all but one element, say e, of S contracted. Then $b_i(M) = b_i(\tilde{M}) + \sum_{j=1}^{|S|-1} b_{i-j}(\tilde{M} - e)$.*
(g) *Let M be a parallel connection of A and B, where the rank of A is $r(A)$ and the rank of B is $r(B)$. The rank of M is $r(A) + r(B) - 1$. In addition, $b_i(M) = \sum_{j+k=i+1} b_j(A) b_k(B)$. If A and B are connected, then M is also connected.*

Proof. Property (g) follows from the fact that if M is a parallel connection of A and B, then $T(M; x, 0) = T(A; x, 0) * T(B; x, 0)/x$ [11, pp. 179–182]. Both (e) and (f) are proved by deleting and contracting all the elements of the given parallel or series class except e. □

One of the consequences of (a) and (f) above is that if we increase the size of a series class of cardinality k in M by one, then $b_1, \ldots, b_k$ are unchanged, while b_i for $i > k$ may increase.

4. Cohen–Macaulay and k-CM complexes. There are several equivalent definitions of Cohen–Macaulay complexes. The following will suffice for our purposes.

DEFINITION 4.1. *A pure $(r-1)$-dimensional complex Δ is Cohen–Macaulay if for every face $F \in \Delta$ and $i < \dim(lk\ F)$, $\tilde{H}_i(lk\ F; \mathbb{Q}) = 0$.*

A numerical description of all possible h-vectors of Cohen–Macaulay complexes can be given using the following operator. Given any positive integers h and i, there is a unique way of writing

$$h = \binom{a_i}{i} + \binom{a_{i-1}}{i-1} + \cdots + \binom{a_j}{j}$$

so that $a_i > a_{i-1} > \cdots > a_j \geq j \geq 1$. Define

$$h^{\langle i \rangle} = \binom{a_i + 1}{i+1} + \binom{a_{i-1} + 1}{i} + \cdots + \binom{a_j + 1}{j+1}.$$

THEOREM 4.2 (see [24]). *A sequence of nonnegative integers $(h_0, \ldots, h_r)$ is the h-vector of some Cohen–Macaulay complex if and only if $h_0 = 1$ and $h_{i+1} \leq h_i^{\langle i \rangle}$ for all $1 \leq i \leq r-1$.*

The notion of k-CM complexes was introduced by Baclawski [2].

DEFINITION 4.3. *Let Δ be a pure $(r-1)$-dimensional simplicial complex with vertex set V and $k \geq 1$. We say that Δ is k-CM if for all $A \subseteq V$ with $|A| < k$, $\Delta - A$ is Cohen–Macaulay of dimension $(r-1)$.*

Examples of 2-CM complexes include order complexes of geometric lattices, finite buildings, and triangulations of spheres. Several examples and constructions involving k-CM complexes, especially for order complexes of posets, are contained in [2]. Since $lk_\Delta v - A = lk_{\Delta - A} v$, the link of any vertex of a k-CM complex is k-CM, and removing a vertex from a k-CM complex leaves a $(k-1)$-CM complex (as long as $k > 1$).

The independence and broken circuit complexes of a matroid are Cohen–Macaulay [23]. So, $\Delta(M)$ is k-CM if and only if every hyperplane of M has cardinality at most $n-k$. Equivalently, the smallest cocircuit of M has at least k elements. However, $\Delta^{BC}(M)$ is a cone on the least element; hence it is only 1-CM. If the cone point is removed, then the remaining complex is also Cohen–Macaulay but may still be only 1-CM. For example, let M be the cycle matroid of the theta-graph with three paths each of length 2. Direct computation shows that the h-vector of $\Delta^{BC}(M)$ is $(1, 2, 3, 1)$. Removing the cone point leaves a 2-dimensional complex with five points and the same h-vector. By Corollary 4.5 below, $(1, 2, 3, 1)$ is not the h-vector of any 2-dimensional 2-CM complex with five points.

Theorem 4.2 gives an upper bound for possible h-vectors of Cohen–Macaulay complexes. It also makes it clear that there are no lower bounds. For k-CM complexes we have the following absolute lower bound. Recall that $U_{r,n}$ is the rank r matroid with n elements such that every r-element subset is a basis.

PROPOSITION 4.4. *Let Δ be an $(r-1)$-dimensional k-CM complex. Then,*

$$h_i(\Delta) \geq h_i(U_{r,r+k-1}).$$

Proof. Induction on n and k. When $k = 1$, the theorem is simply the statement that $h_i(\Delta) \geq 0$ for $i \geq 1$, and $h_0(\Delta) \geq 1$. For fixed k, the definition of k-CM forces

$n \geq r + k - 1$. Suppose $n = r + k - 1$. Since the removal of any subset of vertices of cardinality $k - 1$ does not lower the dimension of Δ, every subset of vertices of cardinality r must be a face of Δ. So, $\Delta = \Delta(U_{r,r+k-1})$. For the induction step, let v be any vertex of Δ. Then

$$\begin{aligned} h_i(\Delta) &= h_i(\Delta - v) + h_{i-1}(lk_\Delta v) \\ &\geq h_i(U_{r,r+k-2}) + h_{i-1}(U_{r-1,r+k-2}) = h_i(U_{r,r+k-1}). \end{aligned}$$ □

Minimizing h-vectors is closely related to the problem of finding the least reliable graph. Let G be a connected graph with $r+1$ vertices and n edges. Thus $M(G)$, the cycle matroid of G, has rank r and cardinality n. Suppose that each edge of G has equiprobability p, $0 < p < 1$, of being deleted. Then the probability that G remains connected is $R_G(p) = (1-p)^r[h_0(M(G)^\star) + h_1(M(G)^\star)p + \cdots + h_{n-r}(M(G)^\star)p^{n-r}]$. Boesch, Satyanarayana, and Suffel [4] posed the problem of finding the minimum of $R_G(p)$ among all connected simple graphs with $r+1$ vertices and n edges. They also conjectured that a particular graph, which they called $L(r+1,n)$, would attain that lower bound. Brown, Colbourn, and Devitt [7] further conjectured that the h-vector of $L(r+1,n)$ would be an absolute lower bound for the h-vector of $M(G)^\star$ among all connected simple graphs with $r+1$ vertices and n edges. The original conjecture of Boesch, Satyanarayana, and Suffel [4] was confirmed for n greater than $\binom{r-1}{2}$ in [20]. The corresponding problem in the category of matroids is to find among all rank r cosimple matroids of cardinality n one which minimizes the h-vector. Since M is cosimple if and only if $\Delta(M)$ is 3-CM, the above proposition shows that $U_{0,n-r-2} \oplus U_{r,r+2}$ is the solution to this problem.

Combining the above proposition with (2.6) immediately gives a relative lower bound.

COROLLARY 4.5. *Let Δ be an $(r-1)$-dimensional k-CM complex with n vertices. Then,*

$$(n-i)h_i \geq (r-i+1)h_{i-1} + n\binom{i+k-3}{i}.$$

Proof. For every vertex v, $\Delta - v$ is $(k-1)$-CM. Now combine (2.6), Proposition 4.4, and the fact that $h_i(U_{r,r+k-2}) = \binom{i+k-3}{i}$. □

PROBLEM 4.6. *Given r, n, k, and i, what is the minimum of $h_i(\Delta)$ over all $(r-1)$-dimensional k-CM complexes with n vertices? Does there exist a Δ which attains these values?*

Conjecture II.6.2 in [24] would imply that for 2-CM complexes with n equal to $r+2$, $h_i(\Delta) \geq h_i(\Delta(U_{1,2} \oplus U_{r-1,r}))$. In section 6 we will give an answer to this problem for independence complexes of matroids when n is sufficiently large.

5. Broken circuit complexes. In this section we assume that M has no loops. An absolute upper bound for b_i when $1 \leq i \leq r$ is $\binom{n-i-1}{r-i}$, and this is achieved by $U_{n,r}$. Theorem 4.2 gives a relative upper bound of $b_{r-i} \leq b_{r-i+1}^{\langle i-1 \rangle}$. Absolute lower bounds for b_i were determined by Brylawski.

THEOREM 5.1 (see [10]). *If M is as above, then $b_i \geq n-r$ for all i, $2 \leq i \leq r-1$.*

In order to find relative lower bounds for b_1 we introduce the following definition.

DEFINITION 5.2. *Let S be a series class of a connected matroid M. Then S is a* regular series *class of M if $M - S$ is connected.*

PROPOSITION 5.3. *If M is connected and contains more than one series class, then M contains at least three regular series classes.*

Proof. The proof is by induction on m, the number of series classes in M. A matroid with exactly two series classes is not connected. If $m = 3$, then M is the cycle matroid of a theta-graph with exactly three paths. In this case all three of the series classes are regular.

For the induction step, let S be a series class which is not regular. Let $\tilde{M}$ be the matroid obtained by contracting all but one of the elements of S. Let e be the remaining element of S. Since $\tilde{M}$ is connected, but $\tilde{M} - e$ is not connected, $\tilde{M}$ is the series connection of two connected matroids A and B at e [19, Theorem 7.1.16]. Both A and B must contain more than one series class, otherwise they would be contained in S. Therefore, the induction hypothesis applies to A and B. Even if $\{e\}$ is contained in a regular series class in A and B, both A and B contain two other regular series classes. All four of these series classes are regular in M. □

THEOREM 5.4. *If M is connected and $1 \leq i \leq r$, then*

$$b_i \leq \binom{r-2}{i-1} b_1 + \binom{r-2}{i-2}. \tag{5.1}$$

Proof. The proof is by induction on n, the initial case being the three-point line. Let S be a series class of M. If S is the only series class of M, then M is a circuit and (5.1) holds. Otherwise, by the previous proposition, we may choose S to be a regular series class. In particular, $M - S$ is connected. We break the induction step into three cases.

1. $M - S$ and M/S are connected: Let $s = |S|$. If $s > i$, then $b_i(M) = b_i(\hat{M})$ and $b_1(M) = b_1(\hat{M})$, where $\hat{M}$ is M with S contracted down to a series class of cardinality i. So, we will assume that $s \leq i$. Let $\tilde{M}$ be M with S contracted down to a single element e. Since M is connected, e is neither a loop nor a coloop of M. Applying Tutte recursion to M and then again to $\tilde{M}$, we see that

$$b_i(M) = b_i(\tilde{M}/e) + \sum_{j=0}^{s-1} b_{i-j}(\tilde{M} - e).$$

Now, since $\tilde{M}/e = M/S$ is a rank $r-s$ connected matroid and $\tilde{M}-e = M-S$ is a rank $r - s + 1$ connected matroid, the induction hypothesis implies that the above expression is bounded above by

$$\binom{r-s-2}{i-1} b_1(\tilde{M}/e) + \binom{r-s-1}{i-2} + \sum_{j=0}^{s-1} \binom{r-s-1}{i-j-1} b_1(\tilde{M}-e) + \sum_{j=0}^{s-1} \binom{r-s-1}{i-j-2}$$

$$\leq \binom{r-2}{i-1} b_1(\tilde{M}/e) + \binom{r-2}{i-1} b_1(\tilde{M} - e) + \binom{r-2}{i-2}$$

$$+ \left\{ \binom{r-s-2}{i-1} - \binom{r-2}{i-1} \right\} b_1(\tilde{M}/e) + \binom{r-s-2}{i-2}.$$

Since $\tilde{M}/e$ is connected, $b_1(\tilde{M}/e) \geq 1$. Thus, the last row is nonpositive and (5.1) is satisfied. To see the last inequality, note that

$$\sum_{j=0}^{s-1} \binom{r-s-1}{i-j-1} \leq \sum_{j=0}^{s-1} \binom{r-s-1}{i-j-1} \binom{s-1}{j} = \binom{r-2}{i-1},$$

and similarly,

$$\sum_{j=0}^{s-1}\binom{r-s-1}{i-j-2} \le \sum_{j=0}^{s-1}\binom{r-s-1}{i-j-2}\binom{s-1}{j} = \binom{r-2}{i-2}.$$

2. $S=\{e\}$ and $M-e$ is connected, but M/e is not connected: Then, M is the parallel connection of two connected matroids A and B with $r(A)+r(B)-1=r$ [19, Theorem 7.1.16]. By Theorem 3.3 and the induction hypothesis,

$$\begin{aligned}
b_i(M) &= \sum_{j+k-1=i} b_j(A)b_k(B)\\
&\le \sum_{j+k-1=i}\left\{\binom{r(A)-2}{j-1}b_1(A)+\binom{r(A)-2}{j-2}\right\}\left\{\binom{r(B)-2}{k-1}b_1(B)\right.\\
&\qquad\left.+\binom{r(B)-2}{k-2}\right\}\\
&= \sum_{j+k-1=i}\left\{\binom{r(A)-2}{j-1}\binom{r(B)-2}{k-1}b_1(A)b_1(B)\right.\\
&\qquad\left.+\binom{r(A)-2}{j-1}\binom{r(B)-2}{k-2}b_1(A)\right\}\\
&\quad+\sum_{j+k-1=i}\left\{\binom{r(A)-2}{j-2}\binom{r(B)-2}{k-1}b_1(B)\right.\\
&\qquad\left.+\binom{r(A)-2}{j-2}\binom{r(B)-2}{k-2}\right\}\\
&= \binom{r-3}{i-1}b_1(A)b_1(B)+\binom{r-3}{i-2}b_1(A)+\binom{r-3}{i-2}b_1(B)+\binom{r-3}{i-3}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\binom{r-2}{i-1}b_1(M)+\binom{r-2}{i-2}-b_i(M)\\
&\quad\ge\left\{\binom{r-2}{i-1}-\binom{r-3}{i-1}\right\}b_1(A)b_1(B)+\left\{\binom{r-2}{i-2}-\binom{r-3}{i-3}\right\}\\
&\qquad-\binom{r-3}{i-2}\{b_1(A)+b_1(B)\}\\
&\quad=\binom{r-3}{i-2}(b_1(A)b_1(B)+1-b_1(A)-b_1(B))\ge 0.
\end{aligned}$$

3. Finally, suppose that S is a nontrivial series and that $M-S$ is connected, but M/S is not connected. Let $s, \tilde{M}$, and e be as above. Since $\tilde{M}/e$ is not

connected, $b_1(\tilde{M}) = b_1(\tilde{M} - e)$. Therefore,

$$\begin{aligned} b_i(M) &= b_i(\tilde{M}) + \sum_{j=1}^{s-1} b_{i-j}(\tilde{M} - e) \\ &\leq b_1(M) \left\{ \sum_{j=0}^{s-1} \binom{r-s-1}{i-j-1} \right\} + \sum_{j=0}^{s-1} \binom{r-s-1}{i-j-2} \\ &\leq \binom{r-2}{i-1} b_1(M) + \binom{r-2}{i-2}. \qquad \square \end{aligned}$$

COROLLARY 5.5. *Let M be a rank r matroid with k components, $r - k \geq 2$. Let $2 \leq i \leq r - k$. Then,*

$$b_{i+k-1}(M) \leq \binom{r-k-1}{i-1} b_k(M) + \binom{r-k-1}{i-2}. \tag{5.2}$$

Proof. Since $k = 1$ is the previous theorem, we assume that M is not connected. Let $M = M_1 \oplus \cdots \oplus M_k$ be a direct sum decomposition of M into connected matroids. Define $\tilde{M}_1 = M_1$. Given $\tilde{M}_i$, let $\tilde{M}_{i+1}$ be any parallel connection of $\tilde{M}_i$ and M_{i+1}. Then $\tilde{M}_k$ is a connected matroid of rank $r - k + 1$. Furthermore, by Theorem 3.3 $b_{i+k-1}(M) = b_i(\tilde{M}_k)$. Since (5.1) holds for the connected $\tilde{M}$, (5.2) holds for M. $\square$

When does equality occur in the above theorem? The proof shows that if equality occurs, then it must also occur in the minors of M used in the induction. Combining this with an induction argument shows that if $b_i(M) = \binom{r-2}{i-1} b_1(M) + \binom{r-2}{i-2}$, then $b_j(M) = \binom{r-2}{j-1} b_1(M) + \binom{r-2}{j-2}$ for all $1 \leq j \leq i$. Brylawski proved (5.1) for $i = r - 1$. He also showed that given b_1 and r, equality occurs if M is the parallel connection of a $(b_1 + 2)$-point line and $r - 1$ three-point lines. Hence, (5.1) is optimal, although a complete description of the matroids which satisfy equality in this corollary remains unknown [10].

The coefficient $b_1(M)$ is also known as $\beta(M)$, the *beta* invariant of M. Brylawski [8] identified matroids with beta invariant 1 as series-parallel matroids, while Oxley [18] classified matroids with $2 \leq \beta(M) \leq 4$.

THEOREM 5.6. *Assume $r \geq 2$ and let $\beta = b_1(M)$. Then, for all i, $0 \leq i \leq r$,*

$$w_i \leq \sum_{j=0}^{i} \binom{r-j}{r-i} \left\{ \binom{r-2}{r-i-1} \beta + \binom{r-2}{r-i-2} \right\}.$$

Proof. This follows immediately from (2.2) and Theorem 5.4. $\square$

It is also possible to estimate b_i in terms of $n - r$. For positive integers i and x define

$$\phi_i(x) = \binom{x-2}{i-1}\binom{x-1}{0} + \binom{x-2}{i-2}\binom{x}{1} + \cdots + \binom{x-2}{0}\binom{x+i-2}{i-1}.$$

THEOREM 5.7. *Suppose M is connected. Then,*

$$b_i(M) \leq \phi_i(n-r) b_1(M) + \phi_{i-1}(n-r). \tag{5.3}$$

Proof. We can assume that every series class of M has exactly i elements. Indeed, by (a) and (c) of Theorem 3.3, any series class with more than i elements can be

contracted down to cardinality i without changing either side of (5.3), while expanding any class with fewer than i elements may increase the left-hand side of (5.3) but will not alter the right-hand side. Let $\tilde{M}$ be the matroid obtained from M by contracting all of the series classes down to one element. The dual of the formula on the top of of [11, p. 185] is

$$T(M;x,0) = (x^{i-1}+\cdots+x+1)^{n-r}T\left(\tilde{M};x^i,\frac{x^{i-1}+\cdots+x}{x^{i-1}+\cdots+x+1}\right) \tag{5.4}$$

Using (5.4), we see that

$$b_i(M) = \sum_{j=1}^{i}\binom{n-r+i-j-1}{i-j}b_j^\star(\tilde{M}). \tag{5.5}$$

Since $b_1^\star(\tilde{M}) = b_1(M)$, (5.3) follows from (5.5) by applying (5.1) to $\tilde{M}^\star$. □

Inequality (5.3) is as optimal as can be expected in the sense that given $n-r$, i, and b_1, there are matroids which satisfy equality. Take any matroid which satisfies equality in (5.1) and expand every series class to cardinality i. Then, equality in (5.3) holds. Of course, since $b_r = 1$ and ϕ_i is increasing in i, no matroid can satisfy equality in (5.3) for all i.

6. Independence complexes. Suppose the smallest cocircuit of M has cardinality k. As pointed out in section 4, $\Delta(M)$ is a k-CM complex. So, we can apply those methods to $\Delta(M)$. In addition to the previously mentioned absolute upper bound $h_i(M) \le \binom{n-r+i-1}{i}$ and relative upper bound $h_{i+1} \le h_i^{\langle i\rangle}$, the h-vectors of independence complexes of matroids satisfy an analogue of the g-theorem for simplicial polytopes.

THEOREM 6.1 (see [25]). *Assume that M has no coloops. Let $g_i(M) = h_i(M) - h_{i-1}(M)$. Then for all i, $1 \le i \le (r+1)/2$,*

$$g_{i+1}(M) \le g_i^{\langle i\rangle}(M).$$

The above theorem was proved independently by Hausel and Sturmfels [15] for matroids representable over the rationals using toric hyperkähler varieties.

Relative lower bounds, also reminiscent of the g-theorem for simplicial polytopes, were originally established by Chari [14] using a PS-ear decomposition of $\Delta(M)$. See [14] for the definition of PS-ear decompositions and a proof of the following theorem.

THEOREM 6.2. *Suppose M has no coloops. Then for all i, $0 \le i \le r/2$,*

$$h_{i-1} \le h_i,$$

$$h_i \le h_{r-i}.$$

PROBLEM 6.3. *Do 2-CM complexes satisfy the inequalities in the previous two theorems?*

An affirmative answer to this question would, with the addition of the Dehn–Sommerville equations, give a complete description of all possible h-vectors of simplicial homology spheres [24, Conjecture II.6.2].

In [5] Brown and Colbourn conjectured that for cographic M, the complex zeros of $T(M;x,1)$ were contained in the closed unit disk. While this has since proved to be false [22], attempts to prove it led to a couple of relative lower bounds for h-vectors of independence complexes of any matroid.

THEOREM 6.4. *Suppose M has no coloops.*

1. *For all $i \leq r$ (see* [5]*),*

$$h_i \geq \sum_{j=1}^{i} (-1)^{j-1} h_{i-j}.$$

2. *Let I_j be the number of independent subsets of M of cardinality j (see* [26]*). Then for all $0 \leq k \leq r$,*

$$\sum_{j=k}^{r} \binom{j}{k} (-2)^{r-j} I_j \geq 0.$$

Stanley used the notion of a level ring to establish the relative lower bound $h_{j-i}(M) \leq h_i(M)h_j(M)$ whenever $0 \leq i, j \leq r$. In particular, setting $j = r$, we find that $h_{r-i}(M) \leq \binom{n-r+i-1}{i} h_r(M)$. By applying (2.6) we can obtain similar relative lower bounds for $h_{i-j}(M)$ in terms of $h_i(M)$ and we can also determine when equality occurs.

PROPOSITION 6.5. *Assume that M has no coloops. Then for all $i,\ 1 \leq j < i \leq r$,*

$$h_{i-j}(M) \leq \frac{\binom{n-i+j-1}{r-i+j}}{\binom{n-i-1}{r-i}} h_i(M). \tag{6.1}$$

Furthermore, equality occurs if and only if every series class of M has cardinality greater than $r - i + j$.

Proof. Since M has no coloops, $\Delta(M)$ is a 2-CM complex. Therefore, (2.6) implies $(r - i + 1)h_{i-1}(M) \leq (n - i)h_i(M)$. In order for equality to occur, $h_i(M - e)$ must be zero for every e in E. By Theorem 3.3(a), this is equivalent to every series class of M having cardinality greater than $r - i + 1$. The proposition follows by induction on j. □

In [6] Brown and Colbourn proved the relative lower bound $h_{r-1}(M) \leq rh_r(M)$, which involves only the rank of M. This can be improved using Theorem 5.4.

THEOREM 6.6. *Let M be a rank r matroid without coloops. Then,*

$$h_{r-i} \leq \binom{r-1}{i} h_r + \binom{r-1}{i-1}. \tag{6.2}$$

Proof. By [9], $h_i(M)$ equals b_{r-i+1} of the free coextension of M. Since the latter matroid has rank $r+1$ and is connected, (6.2) is an immediate consequence of Theorem 5.4. □

As in the case of Theorem 5.4, if $h_{r-i}(M) = \binom{r-1}{i} h_r(M) + \binom{r-1}{i-1}$, then $h_{r-j}(M) \leq \binom{r-1}{j} h_r(M) + \binom{r-1}{j-1}$ for all $0 \leq j \leq i$. A routine deletion-contraction induction shows that for a given r and h_r,

$$M = U_{1,h_r+1} \oplus \underbrace{U_{1,2} \oplus \cdots \oplus U_{1,2}}_{r-1}$$

satisfies equality in (6.2).

COROLLARY 6.7. *Let M be a rank r matroid without coloops. Let I_j be the number of independent subsets of M of cardinality j. Then,*

$$I_j \leq \sum_{i=0}^{j} \binom{r-i}{r-j} \left\{ \binom{r-1}{i} h_r + \binom{r-1}{i-1} \right\}.$$

Proof. Apply the above theorem to (2.1). □

In section 4 we posed the problem of finding absolute lower bounds for a k-CM complex given n and r. Here we examine this problem for independence complexes. Consider the special case of a rank two matroid M without loops. The simplification of M is isomorphic to $U_{2,m}$, where m is the number of parallel classes of M. Therefore, M is specified up to isomorphism by a partition $n = p_1 + \cdots + p_m$, where the p_i's are the sizes of the parallel classes of M. Since $h_0 = 1$ and $h_1 = n - r$, minimizing the h-vector of M is equivalent to minimizing the number of bases of M. As noted earlier, M is k-CM if and only if every hyperplane of M has cardinality at most $n-k$. Equivalently, each $p_i \leq n - k$. The number of bases of M is

$$\binom{n}{2} - \sum_{i=1}^{m} \binom{p_i}{2}.$$

This is minimized by setting $m = \lceil n/(n-k) \rceil, p_i = n - k$ for $i \leq m - 1$, and $p_m = n-(m-1)(n-k)$. Note that this implies that when $n \geq 2k$, $h_2(M)$ is bounded below by $h_2(U_{1,n-k} \oplus U_{1,k})$.

An independence complex is 2-CM if and only if it has no coloops. In [3] Björner showed that for any matroid without coloops $h_i \geq n - r$ for $0 < i < r$. While it is not specifically stated, the proof implies that $h_r \geq n - 2r + 1$. In general, given n and r there may be no single coloop-free matroid that achieves all of these bounds. For example, if $n = 8$ and $r = 4$, then the only matroid without coloops such that $h_4(M) = 1$ is $M = U_{1,2} \oplus U_{1,2} \oplus U_{1,2} \oplus U_{1,2}$. However, $h_2(M) = 6 > n-r$. If we restrict our attention to $i < r$, then $U_{1,n-r} \oplus U_{r-1,r}$ does satisfy $h_i = n - r$ for $0 < i < r$.

DEFINITION 6.8. $M(r,n,k) = U_{1,n-r-k+2} \oplus U_{r-1,r+k-2}$.

Direct computation shows that $h_i(M(r,n,k)) = \binom{k+i-2}{i} + (n-r-k+1)\binom{k+i-3}{i-1}$. In addition, $\Delta(M(r,n,k))$ is k-CM as long as $n \geq r + 2k - 2$.

THEOREM 6.9. *Fix $r \geq 2$ and $k \geq 3$. There exists $N(k,r)$ such that if M is a matroid without loops whose smallest cocircuit has cardinality at least k and $n \geq N(k,r)$, then for all i, $0 \leq i \leq r$,*

$$h_i(M) \geq h_i(M(r,n,k)). \tag{6.3}$$

Proof. First we show that if $n > k(r+1)$, then there exists $e \in M$ such that $\Delta(M - e)$ is still k-CM. Let $\mathbf{H}$ be the set of hyperplanes of M of cardinality $n - k$. If $\mathbf{H}$ is empty, then any e will do since no hyperplane of $M - e$ will have size greater than $n - k - 1$. Otherwise, let B be the intersection of all of the hyperplanes in $\mathbf{H}$. Since B is a flat of M there exists $H_1, \ldots, H_{r+1}$, not necessarily distinct, in $\mathbf{H}$ such that $H_1 \cap \cdots \cap H_{r+1} = B$. Therefore, $|B| \geq n - k(r+1)$, and B is not empty. But, for any $e \in B$, $\Delta(M - e)$ is k-CM.

As noted above, when $r = 2$, $N(2,k) = 2k$ works. So, assume that $r \geq 3$. Let M' be a contraction of M and let $n' = |E(M')|$. By Proposition 4.4, $h_i(M') \geq h_i(U_{r-1,r+k-2})$. In fact, if $n > r + k - 2$, then $h_i(M')$ is strictly greater than $h_i(U_{r-1,r+k-2})$ for $1 \leq i \leq r$. Indeed, this is proved by Tutte recursion as in Proposition 4.4. The base case compares the h-vectors of $U_{2,4}$ and any five-element rank two matroid whose smallest cocircuit has at least three elements. The h-vector of $U_{2,4}$ is $(1,2,3)$. From the discussion of rank two matroids, the h-vectors of the latter group of matroids are bounded below by $(1,3,4)$, the h-vector of the matroid whose simplification is $U_{2,3}$ and whose parallel classes have cardinality $2, 2$,

and 1. Note that this claim is not true when $k = 2$. In particular, $U_{1,2} \oplus U_{r-1,r}$ is a coloop-free matroid with $r + 2$ elements whose h_r is not strictly less than h_r of $U_{r,r+1}$.

To finish the proof, we find $N(r, k, i)$ such that the theorem holds for just h_i and then let $N(r, k)$ be the maximum of the all of the $N(r, k, i)$. Since $h_0(M) = 1$ and $h_1(M) = n - r$, $r + k - 1$ works for $N(r, k, 0)$ and $N(r, k, 1)$. So fix $i \geq 2$. Let N be the minimum of $h_i(\bar{M})$ for all loopless matroids $\bar{M}$ such that $|E(\bar{M})| = k(r + 1) + 1$, $r(\bar{M}) = r$, and the smallest cocircuit of $\bar{M}$ has at least k elements. Let $N(r, k, i) = k(r + 1) + 1 + h_i(M(r, k(r + 1) + 1, k)) - N$.

Claim. If $n \geq N(k, r, i)$, then $h_i(M) \geq h_i(M(r, n, k))$.

Proof of claim. Choose $e_1 \in M$ such that the smallest cocircuit of $M - e_1$ has cardinality greater than or equal to k. Given e_j, choose e_{j+1} so that the smallest cocircuit of $M - \{e_1, \ldots, e_j, e_{j+1}\}$ has size at least k. This can be done up to $j = n - k(r + 1) - 1$. Deleting and contracting on each deletion,

$$h_i(M) = h_i(\tilde{M}) + \sum_j h_{i-1}(M - \{e_1, \ldots, e_{j-1}\}/e_j),$$

where $\tilde{M}$ is $M - \{e_1, \ldots, e_{n-k(r+1)-1}\}$. By construction, $|E(\tilde{M})| = k(r+1)+1$, $r(\tilde{M}) = r$, and the smallest cocircuit of $\tilde{M}$ has at least k elements. In addition, the rank of each contraction is $r - 1$, and its independence complex is k-CM. There are two possibilities.

- Every contraction has more than $r + k - 2$ nonloop elements. In this case $h_i(M) \geq h_i(\tilde{M}) + (n - k(r + 1) - 1)[h_{i-1}(U_{r-1,r+k-2}) + 1]$. Compare this to computing $h_i(M(r, n, k))$ by deleting and contracting down to $U_{1,rk-k+3} \oplus U_{r-1,r+k-2}$. The definition of $N(r, k, i)$ ensures that $h_i(M)$ is bounded below by $h_i(M(r, n, k))$.
- At least one contraction, say $M - \{e_1, \ldots, e_{j-1}\}/e_j)$, has exactly $r + k - 2$ elements. Since this contraction is a rank $r - 1$ matroid whose smallest cocircuit has at least k elements, it must be equal to $U_{r-1,r+k-2}$. Therefore, $M - \{e_1, \ldots, e_{j-1}\}$ has one nontrivial parallel class which contains e_j, and the simplification of $M - \{e_1, \ldots, e_{j-1}\}$ is a one-element coextension of $U_{r-1,r+k-2}$. The one-element coextension of $U_{r-1,r+k-2}$ which minimizes $h_i(M - \{e_1, \ldots, e_{j-1}\})$ is the one obtained by adding a coloop to $U_{r-1,r+k-2}$. Hence, $h_i(M - \{e_1, \ldots, e_{j-1}\})$ is bounded below by $h_i(M(r, n - j, k))$. However, this implies that $h_i(M) \geq h_i(M(r, n - r, k) + j\ h_{i-1}(U_{r-1,r+k-2})) = h_i(M(r, n, k))$. □

Some lower bound on n is necessary in order for (6.3) to hold. For instance, let $M = U_{1,3} \oplus U_{1,3} \oplus U_{1,3}$. Then $r = 3, k = 3$, and $n = 9$. The h-vector of M is $(1, 6, 12, 8)$, while the h-vector of $M(3, 9, 3) = U_{1,5} \oplus U_{2,4}$ is $(1, 6, 11, 12)$. As usual, absolute lower bounds yield relative lower bounds via (2.6).

COROLLARY 6.10. *Fix $r \geq 2$ and $k \geq 3$. There exists $N(k, r)$ such that if M is a matroid without loops whose smallest cocircuit has cardinality k and $n \geq N(k, r)$, then for all i, $0 \leq i \leq r$,*

$$(r - i + 1)h_{i-1}(M) + n\ h_i(M(r, n - 1, k - 1)) \leq (n - i)h_i(M).$$

Acknowledgment. An anonymous referee's comments and suggestions dramatically improved the exposition in several places.

REFERENCES

[1] R. Adin, *A new cubical h-vector*, Discrete Math., 157 (1996), pp. 3–14.

[2] K. Baclawski, *Cohen-Macaulay connectivity and geometric lattices*, European J. Combin., 3 (1982), pp. 293–305.

[3] A. Björner, *The homology and shellability of matroids and geometric lattices*, in Matroid Applications, N. L. White, ed., Cambridge University Press, Cambridge, UK, 1992, pp. 226–283.

[4] F. Boesch, A. Satyanarayana, and C. Suffel, *Least reliable networks and the reliability domination*, IEEE Trans. Comm., 38 (1990), pp. 2004–2009.

[5] J. I. Brown and C. J. Colbourn, *Roots of the reliability polynomial*, SIAM J. Discrete Math., 5 (1992), pp. 571–585.

[6] J. Brown and C. Colbourn, *Non-Stanley bounds for network reliability*, J. Algebraic Combin., 5 (1996), pp. 13–36.

[7] J. Brown, C. Colbourn, and J. Devitt, *Network transformations and bounding network reliability*, Networks, 23 (1993), pp. 1–17.

[8] T. Brylawski, *A combinatorial model for series-parallel networks*, Trans. Amer. Math. Soc., 154 (1971), pp. 1–22.

[9] T. Brylawski, *The broken-circuit complex*, Trans. Amer. Math. Soc., 234 (1977), pp. 417–433.

[10] T. Brylawski, *Connected matroids with the smallest Whitney numbers*, Discrete Math., 18 (1977), pp. 243–252.

[11] T. Brylawski, *The Tutte polynomial.* I. *General theory*, in Matroid Theory and Its Applications (C.I.M.E., 1980), A. Barlotti, ed., Liguori, Naples, Italy, 1982, pp. 125–275.

[12] T. Brylawski and J. G. Oxley, *The Tutte polynomial and its applications*, in Matroid Applications, N. L. White, ed., Cambridge University Press, Cambridge, UK, 1992, pp. 123–225.

[13] M. Chari and C. Colbourn, *Reliability polynomials: A survey*, J. Combin. Inform. System Sci., 22 (1997), pp. 177–193.

[14] M. K. Chari, *Two decompositions in topological combinatorics with applications to matroid complexes*, Trans. Amer. Math. Soc., 349 (1977), pp. 3925–3943.

[15] T. Hausel and B. Sturmfels, *Toric hyperkähler varieties*, Doc. Math., 7 (2002), pp. 495–534.

[16] P. Hersh and I. Novik, *A short simplicial h-vector and the upper bound theorem*, Discrete Comput. Geom., 28 (2002), pp. 283–289.

[17] P. McMullen, *The maximum number of faces of a convex polytope*, Mathematika, 17 (1970), pp. 179–184.

[18] J. G. Oxley, *On Crapo's beta invariant for matroids*, Stud. Appl. Math., 66 (1982), pp. 267–277.

[19] J. G. Oxley, *Matroid Theory*, Oxford University Press, Oxford, UK, 1992.

[20] L. Petingi, J. Saccoman, and L. Schoppmann, *Uniformly least reliable graphs*, Networks, 27 (1996), pp. 125–131.

[21] G.-C. Rota, *On the foundations of combinatorial theory* I*: Theory of Möbius functions*, Z. Wahrscheinlichkeitstheorie und Verw. Gebiete, 2 (1964), pp. 340–368.

[22] G. Royle and D. Sokal, *The Brown-Colbourn conjecture on zeros of reliability polynomials is false*, J. Combin. Theory Ser. B, 91 (2004), pp. 345–360.

[23] R. P. Stanley, *Cohen–Macaulay complexes*, in Higher Combinatorics, NATO Adv. Study Inst. Ser., Ser. C: Math. and Phys. Sci. 31, M. Aigner, ed., Reidel, Dordrecht, The Netherlands, 1977, pp. 51–62.

[24] R. P. Stanley, *Combinatorics and commutative algebra*, Progr. Math. 41, Birkhäuser Boston, Boston, 1996.

[25] E. Swartz, *g-elements of matroid complexes*, J. Comb. Theory. Ser. B, 88 (2003), pp. 369–375.

[26] D. Wagner, *Zeros of reliability polynomials and f-vectors of matroids*, Combin. Probab. Comput., 9 (2000), pp. 167–190.

[27] H. Whitney, *A logical expansion in mathematics*, Bull. Amer. Math. Soc., 38 (1932), pp. 572–579.

[28] H. Whitney, *On the abstract properties of linear dependence*, Amer. J. Math., 57 (1935), pp. 509–533.

[29] H. Wilf, *Which polynomials are chromatic?*, in Colloquio Internazionale sulle Teorie Combinatorie (Rome, 1973), Atti dei Convegni Lincei 17, Accad. Naz. Lincei, Rome, 1976, pp. 247–256.

SIAM J. DISCRETE MATH.
Vol. 18, No. 4, pp. 663–678

COMBINATORIAL CONSTRUCTIONS FOR OPTIMAL SPLITTING AUTHENTICATION CODES*

GENNIAN GE†, YING MIAO‡, AND LIHUA WANG‡

Dedicated to Professor L. Zhu on the occasion of his 60th birthday

Abstract. The notion of a splitting authentication code is very important in the context of an authentication code with arbitration. Ogata et al. [*Discrete Math.*, 279 (2004), pp. 383–405] characterized an optimal splitting authentication code in terms of a splitting balanced incomplete block design (BIBD). A $(v, u \times c, 1)$-splitting BIBD is a pair $(\mathcal{V}, \mathcal{B})$, where $\mathcal{V}$ is a v-set of points and $\mathcal{B}$ is a collection of $u \times c$ arrays, called blocks, with entries from $\mathcal{V}$, such that any point of $\mathcal{V}$ can occur at most once in any block, and for any two distinct points x and y of $\mathcal{V}$, there is exactly one block of $\mathcal{B}$ in which x and y occur in different rows. In this paper, we describe various combinatorial constructions for splitting BIBDs (or, equivalently, optimal splitting authentication codes). We show that the necessary conditions for the existence of a $(v, u \times c, 1)$-splitting BIBD (or, equivalently, an optimal c-splitting authentication code with u source states and v messages) are also sufficient for

(1) $(u, c) = (2, 2t)$ for any positive integer t,
(2) $(u, c) = (2, 3)$ with a definite exception of $v = 10$,
(3) $(u, c) = (3, 2)$ with a definite exception of $v = 9$, and
(4) $(u, c) = (4, 2)$ with two possible exceptions of $v = 49, 385$.

Key words. graph design, optimal, splitting authentication code, splitting balanced incomplete block design, splitting group divisible design

AMS subject classifications. 94A62, 05B05, 05C70

DOI. 10.1137/S0895480103435469

1. Introduction.

Authentication codes were invented by Gilbert, MacWilliams, and Sloane [4] for protecting the integrity of information, which involve three active parties: a *transmitter* T, a *receiver* R, and an *opponent* O. The transmitter T transmits messages to the receiver R using a communication channel. The opponent O has access to this communication channel and can interfere with the contents of cryptograms transmitted via this channel. Two types of active attack from the opponent O, *impersonation* and *substitution*, are usually considered.

A game-theoretic model for authentication codes was developed by Simmons [10]. In this model, the transmitter T and the receiver R share a common *encoding rule* (or *key*) e. The key e is chosen from some *key space* E according to some specified probability distribution. Given a *source state* (or *plaintext*) s from some *source state space* S, the transmitter T computes a *message* $m = e(s) \in M$, where M is the *message space*, and sends $m \in M$ to the receiver R. The receiver R accepts or rejects the transmitted message $m \in M$ based on the key $e \in E$ which the receiver R shared with the transmitter T.

*Received by the editors October 1, 2003; accepted for publication (in revised form) August 10, 2004; published electronically April 22, 2005. This research was supported by National Natural Science Foundation of China 10471127, Zhejiang Provincial Natural Science Foundation of China, and Grant-in-Aid for Scientific Research (C) 14540100 of Japan.
http://www.siam.org/journals/sidma/18-4/43546.html

†Department of Mathematics, Zhejiang University, Hangzhou 310027, Zhejiang, People's Republic of China (gnge@zju.edu.cn).

‡Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba 305-8573, Ibaraki, Japan (miao@sk.tsukuba.ac.jp, wlh@cipher.risk.tsukuba.ac.jp).

We say that an authentication code has *perfect secrecy* if the opponent O has no information about the source state $s \in S$ given a message $m \in M$. In this paper, we will consider only authentication codes with perfect secrecy.

It is possible that more than one message can be used to communicate a particular source state $s \in S$; this is called *splitting*, a very important concept in the context of an authentication code with arbitration (see [11, 12, 6, 7]). In this case, a message $m \in M$ is computed as $m = e(s, r) \in M$, where r is some random number chosen from a specified finite set $\mathcal{R}$. If we define

$$e(s) = \{m \in M : m = e(s, r) \text{ for some } r \in \mathcal{R}\},$$

then splitting means that $|e(s)| > 1$ for some $e \in E$ and $s \in S$. Note also that for any $e \in E$, $e(s) \cap e(s') = \emptyset$ if $s \neq s'$, for otherwise decoding would be impossible. Let $\kappa(e) = \cup_{s \in S} e(s)$. We say that $e \in E$ *accepts* $m \in M$ if $m \in \kappa(e)$.

In an impersonation attack, the opponent O transmits a message $m \in M$ to the receiver R. The opponent O succeeds if $m \in \kappa(e)$. The *impersonation attack probability* P_I is defined as

$$P_I = \max_{m \in M} Pr(m \in \kappa(e)),$$

where the probability is computed over the key space E. In a substitution attack, the opponent O observes a message $m \in M$ transmitted by the transmitter T and then substitutes $m \in M$ with another message $m' \in M$. The opponent O succeeds if $m \in \kappa(e)$ and $m' \in e(s')$, where $s, s' \in S$, $s \neq s'$; in other words, the receiver R accepts $m' \in M$ as authentic and is misled to the false source state $s' \in S$. The *substitution attack probability* P_S is defined as

$$P_S = \sum_{m \in M} Pr(T \text{ sends } m) \max_{m' \in M} Pr(R \text{ accepts } m', s' \neq s, s, s' \in S \mid R \text{ accepts } m),$$

where the probability is computed over the key space E.

A splitting authentication code is called *c-splitting* if $|e(s)| = c$ for any $e \in E$ and any $s \in S$. In a c-splitting authentication code, for every $e \in E$, we know that $|\kappa(e)| = c|S|$. The following theorem describes the known bounds on attack probabilities and the number of keys in a c-splitting authentication code.

THEOREM 1.1 (see [9]). *For any c-splitting authentication code, the following two inequalities always hold:*

$$P_I \geq c|S|/|M|, \qquad P_S \geq c(|S| - 1)/(|M| - 1).$$

If in fact the above equalities are satisfied, then another inequality also holds:

$$|E| \geq |M|(|M| - 1)/(c^2|S|(|S| - 1)).$$

A c-splitting authentication code is said to be *optimal* if it satisfies all the equalities in Theorem 1.1.

In this paper, we will focus our attention on the combinatorial constructions and existence problems of optimal c-splitting authentication codes. An optimal c-splitting authentication code has been shown to be closely related to a combinatorial structure called splitting balanced incomplete block design (BIBD). As a consequence, to construct such an optimal c-splitting authentication code, we need only to construct its corresponding splitting balanced incomplete block design. Various recursive and

direct constructions are used to produce splitting BIBDs or, equivalently, optimal c-splitting authentication codes. Especially, we show that the necessary conditions for the existence of a $(v, u\times c, 1)$-splitting BIBD or, equivalently, an optimal c-splitting authentication code with u source states and v messages, are also sufficient for

(1) $(u, c) = (2, 2t)$ for any positive integer t,
(2) $(u, c) = (2, 3)$ with a definite exception of $v = 10$,
(3) $(u, c) = (3, 2)$ with a definite exception of $v = 9$, and
(4) $(u, c) = (4, 2)$ with two possible exceptions of $v = 49, 385$.

2. Splitting authentication codes, splitting BIBDs, and graph designs. Optimal c-splitting authentication codes are closely related to some combinatorial structures such as splitting BIBDs and balanced graph designs. Let v, u, c, λ be positive integers such that $v \geq uc$. A $(v, u \times c, \lambda)$-*splitting BIBD* is a pair $(\mathcal{V}, \mathcal{B})$, where

(1) $\mathcal{V}$ is a v-set of elements called *points*;
(2) $\mathcal{B}$ is a collection of $u \times c$ arrays, called *blocks*, with entries from $\mathcal{V}$, such that every point occurs at most once in each block;
(3) for every pair of distinct points $x, y \in \mathcal{V}$, there are exactly λ blocks in which x and y occur in different rows.

The following combinatorial properties of a splitting BIBD can be easily obtained.

LEMMA 2.1 (see [9]). *In a $(v, u \times c, \lambda)$-splitting BIBD $(\mathcal{V}, \mathcal{B})$, each point of $\mathcal{V}$ is contained in exactly*

$$r = \lambda(v - 1)/((u - 1)c)$$

blocks, and there are exactly

$$b = \lambda v(v - 1)/\big(u(u - 1)c^2\big)$$

blocks. Furthermore,

$$b \geq v/u.$$

Ogata et al. [9] showed the following relations between splitting authentication codes and splitting BIBDs. An *authentication matrix* of a c-splitting authentication code is a matrix with the rows indexed by the keys $e \in E$, the columns indexed by the source states $s \in S$, and entry (e, s) given by $e(s) \subseteq M$.

THEOREM 2.2 (see [9]). *If there exists an optimal c-splitting authentication code, then the rows of its authentication matrix form the blocks of an $(|M|, |S|\times c, 1)$-splitting BIBD, each source state in a row of the authentication matrix yielding a row in its corresponding block of the splitting BIBD.*

Conversely, starting from a $(v, u\times c, 1)$-splitting BIBD $(\mathcal{V}, \mathcal{B})$, we can put $M = \mathcal{V}$, $S = \{s_1, \ldots, s_u\}$, and for each block

$$\begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_u \end{pmatrix},$$

we define an encoding rule $e \in E$ such that $e(s_1) = B_1, e(s_2) = B_2, \ldots, e(s_u) = B_u$. Then we obtain the following result.

THEOREM 2.3 (see [9]). *If there exists a $(v, u \times c, 1)$-splitting BIBD, then there exists an optimal c-splitting authentication code such that*

(1) $|M| = v$, $|S| = u$;
(2) *each source state occurs with equal probability.*

As an immediate consequence of the above results [9], to construct optimal c-splitting authentication codes, we need only to construct their corresponding splitting balanced incomplete block designs.

We also noticed that splitting BIBDs are in fact a special kind of graph designs. Let $G = (V(G), E(G))$ be a graph, where $V(G)$ is the set of vertices of G and $E(G)$ is the set of edges of G. Let λK_n be the multiple complete graph with n vertices, that is, the graph with n vertices such that any two distinct vertices are incident with λ common edges, and let K_c^u be the complete u-partite graph with each part having c vertices, that is, the graph with the set of vertices being partitioned into u parts of size c each such that every vertex is adjacent to every vertex in a different part. A $(\lambda K_n, G)$-*graph design* is a partition of the edges of λK_n into subgraphs, called G-*blocks*, each of which is isomorphic to G. A $(\lambda K_n, G)$-graph design is *balanced* if each vertex of λK_n belongs to exactly the same number of G-blocks. Then, from the definitions, we can easily see that a $(v, u \times c, \lambda)$-splitting BIBD is equivalent to a $(\lambda K_v, K_c^u)$-balanced graph design.

Therefore, we can also investigate optimal c-splitting authentication codes from a graph theoretic point of view. However, in this paper, we will use only the combinatorial design theoretic approach.

3. Combinatorial constructions. Now we describe our combinatorial constructions for splitting BIBDs or, equivalently, optimal splitting authentication codes. These include recursive constructions, in which the new concept of a splitting group divisible design plays an important role, and direct constructions by difference method, some of which making use of Weil's theorem on character sums and Wilson's choice mapping.

3.1. Recursive constructions. Let K be a set of some positive integers. A *group divisible design*, denoted by K-GDD, is a triple $(\mathcal{V}, \mathcal{G}, \mathcal{B})$, where $\mathcal{V}$ is a set of elements called *points*, $\mathcal{G}$ is a partition of $\mathcal{V}$ into subsets called *groups*, and $\mathcal{B}$ is a collection of subsets of $\mathcal{V}$ called *blocks* such that

(1) $|B| \in K$ for any $B \in \mathcal{B}$;
(2) $|G \cap B| \leq 1$ for any $G \in \mathcal{G}$ and any $B \in \mathcal{B}$; and
(3) for any pair of points $\{x, y\}$, where x and y belong to distinct groups, there exists exactly one block of $\mathcal{B}$ in which x and y occur.

We define the *group type* (or *type*) of a K-GDD to be the multiset $(|G| : G \in \mathcal{G})$. The usual exponential notation will be used to describe types. Thus a GDD of type $t_1^{u_1} \dots t_n^{u_n}$ is one in which there are exactly u_i groups of size t_i, $1 \leq i \leq n$.

A $\{k\}$-GDD of type 1^v is commonly called a $(v, k, 1)$-BIBD.

Let $u \geq 2$ and $c \geq 2$ be integers. A *splitting GDD*, denoted by $u \times c$-splitting GDD, is a triple $(\mathcal{V}, \mathcal{G}, \mathcal{A})$ where $\mathcal{V}$ is a set of elements called *points*, $\mathcal{G}$ is a partition of $\mathcal{V}$ into subsets called *groups*, and $\mathcal{B}$ is a collection of $u \times c$ arrays with entries from $\mathcal{V}$, called *blocks*, such that

(1) any point of $\mathcal{V}$ can occur at most once in any block;
(2) for any pair of points $\{x, y\}$, where x and y belong to distinct groups, there exists exactly one block of $\mathcal{A}$ in which x and y occur in different rows.

We can define the *group type* (or *type*) of a $u \times c$-splitting GDD similarly to a K-GDD. Clearly, a $u \times c$-splitting GDD of type 1^v is equivalent to a $(v, u \times c, 1)$-splitting BIBD.

Splitting GDDs can be used to construct splitting BIBDs. In this section, we describe several recursive constructions for splitting GDDs and splitting BIBDs. They are analogues of the well-known recursive constructions for GDDs and BIBDs due to Wilson [14].

THEOREM 3.1 (see [14]). *Let $(\mathcal{V}, \mathcal{G}, \mathcal{B})$ be a GDD. Further let $w : \mathcal{V} \longrightarrow \mathcal{N} \cup \{0\}$ be a weight function, where $\mathcal{N}$ is the set of positive integers. For each $B \in \mathcal{B}$, suppose that there exists a K-GDD of type $(w(x) : x \in B)$, $(\cup_{x\in B} S(x), \{S(x) : x \in B\}, \mathcal{B}(B))$, where $S(x) = \{x_1, x_2, \ldots, x_{w(x)}\}$ for every $x \in \mathcal{V}$ and $\mathcal{B}(B)$ is the collection of blocks of the ingredient GDD. Then there exists a K-GDD of type $(\sum_{x\in G} w(x) : G \in \mathcal{G})$, $(\cup_{x\in\mathcal{V}} S(x), \{\cup_{x\in G} S(x) : G \in \mathcal{G}\}, \cup_{B\in\mathcal{B}} \mathcal{B}(B))$.*

THEOREM 3.2 (see [14]). *Let $(\mathcal{V}, \mathcal{G}, \mathcal{B})$ be a K-GDD. Further, let G_0 be a set of new points, that is, $G_0 \cap \mathcal{V} = \emptyset$, and suppose that for each group $G \in \mathcal{G}$, there exists a K-GDD $(G \cup G_0, \{G_0\} \cup \mathcal{H}_G, \mathcal{B}_G)$, where $\mathcal{H}_G$ is the set of groups except G_0 and $\mathcal{B}_G$ is the collection of blocks of the ingredient GDD. Then there exists a K-GDD $(\mathcal{V} \cup G_0, \{G_0\} \cup \{\mathcal{H}_G : G \in \mathcal{G}\}, \mathcal{B} \cup (\cup_{G\in\mathcal{G}} \mathcal{B}_G))$.*

A special case of Theorem 3.2 is the following, which is very useful in the construction of BIBDs.

COROLLARY 3.3 (see [14]). *Let $(\mathcal{V}, \mathcal{G}, \mathcal{B})$ be a $\{k\}$-GDD. Further, let ∞ be a new point, that is, $\{\infty\} \cap \mathcal{V} = \emptyset$, and suppose that for each group $G \in \mathcal{G}$, there exists a $(|G| + 1, k, 1)$-BIBD, $(G \cup \{\infty\}, \mathcal{B}_G)$, where $\mathcal{B}_G$ is the collection of blocks of the ingredient BIBD. Then there exists a $(|\mathcal{V}| + 1, k, 1)$-BIBD $(\mathcal{V} \cup \{\infty\}, \mathcal{B} \cup (\cup_{G\in\mathcal{G}} \mathcal{B}_G))$.*

Then we can state our recursive constructions for splitting GDDs and splitting BIBDs.

THEOREM 3.4. *Let $(\mathcal{V}, \mathcal{G}, \mathcal{B})$ be a GDD. Further, let $w : \mathcal{V} \longrightarrow \mathcal{N} \cup \{0\}$ be a weight function such that $w(x) = w(y)$ for any points $x, y \in \mathcal{V}$. For each block $B \in \mathcal{B}$, suppose that there exists a $u \times c$-splitting GDD of type $(w(x) : x \in B)$, $(\cup_{x\in B} S(x), \{S(x) : x \in B\}, \mathcal{A}(B))$, where $S(x) = \{(x, 1), (x, 2), \ldots, (x, w(x))\}$ for every point $x \in \mathcal{V}$. Then there exists a $u \times c$-splitting GDD of type $(\sum_{x\in G} w(x) : G \in \mathcal{G})$, $(\cup_{x\in\mathcal{V}} S(x), \{\cup_{x\in G} S(x) : G \in \mathcal{G}\}, \cup_{B\in\mathcal{B}} \mathcal{A}(B))$.*

A degenerate case of Theorem 3.4 is the following simple but very powerful construction.

COROLLARY 3.5. *Let $u \geq 2$ and $c \geq 2$ be two integers. Further, let $(\mathcal{V}, \mathcal{G}, \mathcal{B})$ be a $\{u\}$-GDD, and let $w : \mathcal{V} \longrightarrow \mathcal{N} \cup \{0\}$ be a weight function such that $w(x) = c$ for any point $x \in \mathcal{V}$. Then there exists a $u \times c$-splitting GDD of type $(\sum_{x\in G} w(x) : G \in \mathcal{G})$.*

Proof. For any block $B = \{b_1, b_2, \ldots, b_u\} \in \mathcal{B}$, there always exists a $u \times c$-splitting GDD of type c^u, $(\cup_{x\in B} S(x), \{S(x) : x \in B\}, \{B'\})$, where $S(x) = \{(x, 1), (x, 2), \ldots, (x, w(x))\}$ for every point $x \in \mathcal{V}$, and

$$B' = \begin{pmatrix} (b_1, 1) & \cdots & (b_1, c) \\ (b_2, 1) & \cdots & (b_2, c) \\ \cdots & \cdots & \cdots \\ (b_u, 1) & \cdots & (b_u, c) \end{pmatrix}.$$

Then apply Theorem 3.4. □

THEOREM 3.6. *Let $(\mathcal{V}, \mathcal{G}, \mathcal{A})$ be a $u \times c$-splitting GDD. Further, let G_0 be a set of new points, that is, $G_0 \cap \mathcal{V} = \emptyset$, and suppose that for each group $G \in \mathcal{G}$, there exists a $u \times c$-splitting GDD, $(G \cup G_0, \{G_0\} \cup \mathcal{H}_G, \mathcal{A}_G)$. Then there exists a $u \times c$-splitting GDD, $(\mathcal{V} \cup G_0, \{G_0\} \cup \{\mathcal{H}_G : G \in \mathcal{G}\}, \mathcal{A} \cup (\cup_{G\in\mathcal{G}} \mathcal{A}_G))$.*

Theorem 3.6 leads to the following corollary, which is very useful in the construction of splitting BIBDs.

COROLLARY 3.7. *Let $(\mathcal{V}, \mathcal{G}, \mathcal{B})$ be a $u \times c$-splitting GDD. If for each group $G \in \mathcal{G}$, there exists a $(|G|+1, u \times c, 1)$-splitting BIBD, then there exists a $(|\mathcal{V}|+1, u \times c, 1)$-splitting BIBD.*

3.2. Difference method direct constructions. To apply the recursive constructions described in subsection 3.1, we must first have some constructions to produce ingredient splitting BIBDs. The method of differences is the most widely used direct construction for many types of combinatorial designs. Splitting BIBDs can also be constructed by this method.

We first construct $(v, 2 \times c, 1)$-splitting BIBDs.

LEMMA 3.8. *There exists a $(2c^2t+1, 2 \times c, 1)$-splitting BIBD for any positive integers t and c.*

Proof. Let the set of points be Z_{2c^2t+1}. Then the blocks of the desired $(2c^2t+1, 2 \times c, 1)$-splitting BIBD can be obtained by developing the elements of Z_{2c^2t+1} in the following base blocks $+1$ modulo $2c^2t+1$, that is, for each base block in the following list, construct $2c^2t+1$ blocks by adding the elements $0, 1, 2, \ldots$ modulo $2c^2t+1$ to the given base block:

$$\begin{pmatrix} 1 & 2 & \cdots & c \\ 2c^2i-(2c^2-c)+1 & 2c^2i-(2c^2-c)+c+1 & \cdots & 2c^2i-(2c^2-c)+c(c-1)+1 \end{pmatrix},$$

where $i = 1, 2, \ldots, t$. □

LEMMA 3.9. *There exists a $(28, 2 \times 3, 1)$-splitting BIBD.*

Proof. Let the set of points be Z_{28}. The blocks of the desired $(28, 2 \times 3, 1)$-splitting BIBD can be obtained by developing the elements of Z_{28} in the following given base blocks $+4$ modulo 28, that is, for each base block in the following list, construct seven blocks by adding the elements $0, 4, 8, 12, 16, 20, 24$ modulo 28 to the given base block:

$$\begin{pmatrix} 1 & 17 & 20 \\ 2 & 3 & 25 \end{pmatrix}, \begin{pmatrix} 4 & 10 & 16 \\ 11 & 13 & 17 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 10 & 11 & 12 \end{pmatrix},$$
$$\begin{pmatrix} 4 & 20 & 22 \\ 16 & 18 & 19 \end{pmatrix}, \begin{pmatrix} 3 & 14 & 21 \\ 9 & 19 & 27 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 16 \\ 8 & 18 & 19 \end{pmatrix}.$$ □

Next we consider the construction of $(v, 3 \times 2, 1)$-splitting BIBDs. We list explicitly $(v, 3 \times 2, 1)$-splitting BIBDs for $v = 25, 33, 49, 57, 81$.

LEMMA 3.10. *There exists a $(v, 3 \times 2, 1)$-splitting BIBD for $v \in \{25, 33, 49, 57, 81\}$.*

Proof. For $v = 25$, let the set of points be Z_{25}. The blocks of the desired $(25, 3 \times 2, 1)$-splitting BIBD can be obtained by developing the elements of Z_{25} in the following given base block $+1$ modulo 25:

$$\begin{pmatrix} 0 & 1 \\ 2 & 4 \\ 12 & 20 \end{pmatrix}.$$

For $v = 33$, let the set of points be Z_{33}. The blocks of the desired $(33, 3 \times 2, 1)$-splitting BIBD can be obtained by developing the elements of Z_{33} in the following given base blocks $+3$ modulo 33:

$$\begin{pmatrix} 3 & 5 \\ 4 & 18 \\ 15 & 29 \end{pmatrix}, \begin{pmatrix} 1 & 18 \\ 10 & 20 \\ 14 & 31 \end{pmatrix}, \begin{pmatrix} 3 & 29 \\ 12 & 13 \\ 19 & 31 \end{pmatrix}, \begin{pmatrix} 1 & 11 \\ 3 & 29 \\ 8 & 30 \end{pmatrix}.$$

For $v = 49$, let the set of points be Z_{49}. The blocks of the desired $(49, 3 \times 2, 1)$-splitting BIBD can be obtained by developing the elements of Z_{49} in the following given base blocks $+1$ modulo 49:

$$\begin{pmatrix} 0 & 1 \\ 2 & 4 \\ 11 & 19 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 6 & 14 \\ 22 & 26 \end{pmatrix}.$$

For $v = 57$, let the set of points be Z_{57}. The blocks of the desired $(57, 3 \times 2, 1)$-splitting BIBD can be obtained by developing the elements of Z_{57} in the following given base blocks $+3$ modulo 57:

$$\begin{pmatrix} 0 & 49 \\ 4 & 50 \\ 44 & 51 \end{pmatrix}, \begin{pmatrix} 0 & 11 \\ 2 & 15 \\ 23 & 37 \end{pmatrix}, \begin{pmatrix} 0 & 28 \\ 5 & 38 \\ 34 & 49 \end{pmatrix}, \begin{pmatrix} 0 & 17 \\ 3 & 48 \\ 27 & 31 \end{pmatrix},$$
$$\begin{pmatrix} 0 & 55 \\ 1 & 25 \\ 16 & 17 \end{pmatrix}, \begin{pmatrix} 0 & 2 \\ 20 & 32 \\ 35 & 52 \end{pmatrix}, \begin{pmatrix} 0 & 27 \\ 11 & 46 \\ 13 & 39 \end{pmatrix}.$$

For $v = 81$, let the set of points be $Z_{27} \times \{0, 1, 2\}$. The blocks of the desired $(81, 3 \times 2, 1)$-splitting BIBD can be obtained by developing the first coordinates of elements of $Z_{27} \times \{0, 1, 2\}$ in the following given base blocks $+1$ modulo 27:

$$\begin{pmatrix} (0,0) & (26,0) \\ (0,1) & (2,1) \\ (4,1) & (7,1) \end{pmatrix}, \begin{pmatrix} (0,0) & (1,0) \\ (10,1) & (12,1) \\ (18,1) & (21,1) \end{pmatrix}, \begin{pmatrix} (0,0) & (1,0) \\ (14,1) & (16,1) \\ (0,2) & (4,2) \end{pmatrix}, \begin{pmatrix} (0,0) & (1,0) \\ (23,1) & (25,1) \\ (2,2) & (6,2) \end{pmatrix},$$
$$\begin{pmatrix} (0,0) & (13,1) \\ (7,0) & (15,2) \\ (26,1) & (14,2) \end{pmatrix}, \begin{pmatrix} (0,1) & (2,1) \\ (17,1) & (26,1) \\ (20,2) & (24,2) \end{pmatrix}, \begin{pmatrix} (0,1) & (13,1) \\ (0,2) & (12,2) \\ (5,2) & (9,2) \end{pmatrix}, \begin{pmatrix} (0,0) & (26,0) \\ (10,2) & (20,2) \\ (16,2) & (18,2) \end{pmatrix},$$
$$\begin{pmatrix} (0,0) & (26,0) \\ (8,2) & (12,2) \\ (22,2) & (24,2) \end{pmatrix}, \begin{pmatrix} (0,0) & (3,0) \\ (13,0) & (26,0) \\ (18,0) & (24,0) \end{pmatrix}. \quad \square$$

Now we consider the construction for $(v, 4 \times 2, 1)$-splitting BIBDs. Similarly, we list explicitly $(v, 4 \times 2, 1)$-splitting BIBDs for $v = 97, 145, 193, 241, 289, 337, 433$.

LEMMA 3.11. *There exists a* $(v, 4 \times 2, 1)$*-splitting BIBD for* $v \in \{97, 145, 193, 241, 289, 337, 433\}$.

Proof. We prove this lemma in a way similar to Lemma 3.10, but here we describe only the base blocks for each desired splitting BIBD.

For $v = 97$, let the set of points be Z_{97}. The base blocks of the desired $(97, 4 \times 2, 1)$-splitting BIBD are given below:

$$\begin{pmatrix} 1 & 2 \\ 5 & 29 \\ 37 & 48 \\ 22 & 0 \end{pmatrix}, \begin{pmatrix} 6 & 55 \\ 31 & 85 \\ 44 & 45 \\ 22 & 0 \end{pmatrix}.$$

For $v = 145$, let the set of points be Z_{145}. The base blocks of the desired $(145, 4 \times 2, 1)$-splitting BIBD are given below:

$$\begin{pmatrix} 3 & 88 \\ 37 & 76 \\ 96 & 98 \\ 119 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 68 \\ 6 & 92 \\ 36 & 106 \\ 133 & 0 \end{pmatrix}, \begin{pmatrix} 2 & 64 \\ 60 & 119 \\ 130 & 138 \\ 18 & 0 \end{pmatrix}.$$

For $v = 193$, let the set of points be Z_{193}. Multiplying the following two base blocks by $1, 81 \in Z_{193}$ yields the four base blocks of the desired $(193, 4 \times 2, 1)$-splitting BIBD:

$$\begin{pmatrix} 1 & 2 \\ 5 & 8 \\ 43 & 115 \\ 140 & 0 \end{pmatrix}, \begin{pmatrix} 11 & 126 \\ 39 & 160 \\ 140 & 147 \\ 56 & 0 \end{pmatrix}.$$

For $v = 241$, let the set of points be Z_{241}. Multiplying the following base block by $1, 87, 87^2, 87^3, 87^4 \in Z_{241}$ yields the five base blocks of the desired $(241, 4 \times 2, 1)$-splitting BIBD:

$$\begin{pmatrix} 1 & 2 \\ 6 & 85 \\ 50 & 227 \\ 116 & 0 \end{pmatrix}.$$

For $v = 289$, let the set of points be $GF(17^2)$. Choose one primitive polynomial $f(x) = x^2 + 11x + 6 \in GF(17)[x]$. Then multiplying the following two base blocks by $1, x^{96}, x^{192} \in GF(17^2)$ yields the desired six base blocks:

$$\begin{pmatrix} x & x^2 \\ x^3 & x^4 \\ x^5 & x^{127} \\ x^{136} & 0 \end{pmatrix}, \begin{pmatrix} x^{10} & x^{17} \\ x^{35} & x^{163} \\ x^{139} & x^{182} \\ x^{66} & 0 \end{pmatrix}.$$

For $v = 337$, let the set of points be Z_{337}. Multiplying the following base block by $1, 8, 8^2, 8^3, 8^4, 8^5, 8^6 \in Z_{337}$ yields the desired seven base blocks:

$$\begin{pmatrix} 1 & 2 \\ 5 & 94 \\ 74 & 205 \\ 108 & 0 \end{pmatrix}.$$

For $v = 433$, let the set of points be Z_{433}. Multiplying the following base block by $1, 27, 27^2, 27^3, 27^4, 27^5, 27^6, 27^7, 27^8 \in Z_{433}$ yields the desired nine base blocks:

$$\begin{pmatrix} 1 & 2 \\ 5 & 8 \\ 248 & 292 \\ 316 & 0 \end{pmatrix}. \qquad \square$$

We also describe a direct construction for $(v, 5 \times 2, 1)$-splitting BIBDs. By a computer search and by applying Weil's theorem on character sums, we can show the existence of $(p, 5 \times 2, 1)$-splitting BIBDs for some prime numbers $p \equiv 1 \pmod{80}$.

A *multiplicative character* of a finite field $GF(q)$ is a homomorphism from the multiplicative group of $GF(q)$ into the multiplicative group of complex numbers of absolute value 1. The following is the statement of Weil's theorem on multiplicative character sums cited from Theorem 5.41 in [8]. In the theorem it is understood that if χ is a multiplicative character of $GF(q)$, then $\chi(0) = 0$.

THEOREM 3.12. *Let χ be a multiplicative character of $GF(q)$ of order $m > 1$ and let f be a polynomial of $GF(q)[x]$ which is not of the form kg^m for some $k \in GF(q)$ and some $g \in GF(q)[x]$. Then we have*

$$\left| \sum_{x \in GF(q)} \chi(f(x)) \right| \leq (d-1)\sqrt{q},$$

where d is the number of distinct roots of $f(x)$ in its splitting field over $GF(q)$.

As an application of Theorem 3.12, Chang and Ji [1] obtained the following lemma, Lemma 3.13. To state their result and describe our own, we need to explain some definitions and notations in finite fields. Given a prime power $q \equiv 1 (\mathrm{mod}\, n)$ and a primitive element $\omega \in GF(q)$, C_0^n will denote the unique multiplicative subgroup $\{\omega^{in} : 0 \leq i < (q-1)/n\}$ of index n and order $(q-1)/n$, while C_j^n will denote the multiplicative coset of C_0^n represented by ω^j, i.e., $C_j^n = \omega^j \cdot C_0^n$. The multiplicative cosets $C_0^n, C_1^n, \ldots, C_{n-1}^n$ of C_0^n are the *cyclotomic classes* of index n in $GF(q)$. They evidently partition $GF(q) \setminus \{0\}$. The class of cosets $\{C_0^n, C_1^n, \ldots, C_{n-1}^n\}$ will be denoted by $\mathcal{C}^n$. Given a set of n distinct elements in $GF(q)$, if they belong to n distinct cyclotomic classes $C_0^n, C_1^n, \ldots, C_{n-1}^n$, then we say that this set of n elements forms a *system of distinct representatives of the cyclotomic classes* $C_0^n, C_1^n, \ldots, C_{n-1}^n$, and it is denoted by $\mathrm{SDRC}(\mathcal{C}^n)$.

LEMMA 3.13 (see [1]). *Let $p \equiv 1 (\mathrm{mod}\, q)$ be a prime number satisfying the inequality $p - [\sum_{0 \leq i \leq s-2} \binom{s}{i}(s-i-1)(q-1)^{s-i}]\sqrt{p} - sq^{s-1} > 0$. Then, for any given s-tuple $(j_1, j_2, \ldots, j_s) \in \{0, 1, \ldots, q-1\}^s$ and any given s-tuple $(c_1, c_2, \ldots, c_s)$ of pairwise distinct elements of $GF(p)$, there exists an element $x \in GF(p)$ such that $x + c_i \in C_{j_i}^q$ for each i, $i = 1, 2, \ldots, s$.*

Let $p = 80t + 1$ be a prime number, where t is a positive integer. Let w be a primitive element of $GF(p)$ and x be some element of $GF(p)$. We consider the list ΔA of the 80 differences from any two entries in different rows of the following 5×2 array:

$$A = \begin{pmatrix} 1 & x \\ w^{16t} & xw^{16t} \\ w^{32t} & xw^{32t} \\ w^{48t} & xw^{48t} \\ w^{64t} & xw^{64t} \end{pmatrix}.$$

It is straightforward to check that

$$\begin{aligned} \Delta A = \{&1, w^{8t}, w^{16t}, w^{24t}, w^{32t}, w^{40t}, w^{48t}, w^{56t}, w^{64t}, w^{72t}\} \\ &\times \{1 - w^{16t}, 1 - w^{32t}, x - w^{16t}, x - w^{32t}, x - w^{48t}, x - w^{64t}, \\ &\quad x(1 - w^{16t}), x(1 - w^{32t})\}. \end{aligned}$$

Now if we define $\mathcal{A} = \{A_i : 0 \leq i \leq t-1\}$, where

$$A_i = w^{8i} \cdot A = \begin{pmatrix} w^{8i} & xw^{8i} \\ w^{16t+8i} & xw^{16t+8i} \\ w^{32t+8i} & xw^{32t+8i} \\ w^{48t+8i} & xw^{48t+8i} \\ w^{64t+8i} & xw^{64t+8i} \end{pmatrix},$$

then

$$\begin{aligned}\Delta\mathcal{A} &= \cup_{0\le i\le t-1}\Delta A_i\\ &= \{1, w^8, \ldots, w^{8(t-1)}\}\times\{1, w^{8t}, w^{16t}, w^{24t}, w^{32t}, w^{40t}, w^{48t}, w^{56t}, w^{64t}, w^{72t}\}\\ &\quad\times\{1-w^{16t}, 1-w^{32t}, x-w^{16t}, x-w^{32t}, x-w^{48t}, x-w^{64t},\\ &\qquad x(1-w^{16t}), x(1-w^{32t})\}.\end{aligned}$$

Clearly, when $t \equiv 1, 3, 7, 9 \pmod{10}$, $C_0^8 = \{1, w^8, \ldots, w^{8(t-1)}\} \times \{1, w^{8t}, w^{16t}, w^{24t}, w^{32t}, w^{40t}, w^{48t}, w^{56t}, w^{64t}, w^{72t}\}$, and if $\{1-w^{16t}, 1-w^{32t}, x-w^{16t}, x-w^{32t}, x-w^{48t}, x-w^{64t}, x(1-w^{16t}), x(1-w^{32t})\}$ forms an SDRC($\mathcal{C}^8$), then $\Delta\mathcal{A} = GF(q)\setminus\{0\}$. The latter condition is equivalent to the following condition $(*)$:

$$\begin{cases} 1-w^{16t} \in C_a^8,\\ 1+w^{16t} \in C_b^8,\\ x-w^{16t} \in C_c^8,\\ x-w^{32t} \in C_d^8,\\ x-w^{48t} \in C_e^8,\\ x-w^{64t} \in C_f^8,\\ x \in C_g^8,\end{cases}$$

where $\{a, a+b, c, d, e, f, a+g, a+b+g\}$ should form a complete set of residues modulo 8.

Let $p = 80t+1$ be a prime number, where $t \equiv 1, 3, 7, 9 \pmod{10}$. We first investigate the existence of a primitive element w in $GF(p)$ such that $1+w^{16t} \notin C_0^8$.

LEMMA 3.14. *Let $p = 80t+1$ be a prime number, where t is a positive integer. Then in $GF(p)$, one and only one of the following two properties is satisfied:*

(1) $1+\theta^{16t} \in C_0^8$ *for any primitive element* $\theta \in GF(p)$.

(2) $1+\theta^{16t} \notin C_0^8$ *for any primitive element* $\theta \in GF(p)$.

Proof. Let $h = w^{16t}$, where w is a fixed primitive element of $GF(p)$. Then $h^5 = w^{80t} = 1$, which implies that 1, h, h^2, h^3, h^4 are exactly the five solutions to the equation $x^5 - 1 = 0$ in $GF(p)$. For any other primitive element $\theta \in GF(p)$, we also have $\theta^{16t} \neq 1$ and θ^{16t} satisfies the equation $x^5 - 1 = 0$. Thus, θ^{16t} must be one of the four elements h, h^2, h^3, h^4. Denote the set $\{h, h^2, h^3, h^4\}$ by H. Since $h = w^{16t}$, we can easily know that $H \subset C_0^8$. We prove that if $1+h \in C_0^8$, then $1+H \subset C_0^8$. In fact, since $h^5 = 1$, we know that $1+h+h^2+h^3+h^4 = 0$. Also, it is clear that $-1 = w^{40t} \in C_0^8$. Then $(1+h)(1+h^2) = 1+h+h^2+h^3 = -h^4 \in C_0^8$, which implies that $1+h^2 \in C_0^8$. Similarly, $(1+h)(1+h^3) = 1+h+h^3+h^4 = -h^2 \in C_0^8$, which implies that $1+h^3 \in C_0^8$. Finally, $1+h^4 = h^4(1+h) \in C_0^8$. Therefore, for a fixed primitive element $w \in GF(p)$, if $1+w^{16t} \in C_0^8$, then $1+\theta^{16t} \in C_0^8$ for any other primitive element $\theta \in GF(p)$, and if $1+w^{16t} \notin C_0^8$, then $1+\theta^{16t} \notin C_0^8$ for any other primitive element $\theta \in GF(p)$. ∎

As an immediate consequence, to show the existence of a primitive element $w \in GF(p)$ such that $1+w^{16t} \notin C_0^8$, we need only to show that the least primitive element w_0 of $GF(p)$ satisfies the above condition. The computational results show that in most cases with $p \le 10^7$, the least primitive element $w_0 \in GF(p)$ does satisfy the condition $1+w_0^{16t} \notin C_0^8$.

In fact, for a primitive element w of $GF(p)$, a proof similar to that of Lemma 3.14 shows that if $1+w^{16t} \in C_0^8$, then $1+y^{16t} \in C_0^8$ for any other element $y \notin C_0^5$, and if $1+w^{16t} \notin C_0^8$, then $1+y^{16t} \notin C_0^8$ for any other element $y \notin C_0^5$. Therefore, the problem of whether there is a primitive element w in $GF(p)$ such that $1+w^{16t} \notin C_0^8$ can be further reduced to an easier problem—whether there is an element $y \notin C_0^5$ such that $1+y^{16t} \notin C_0^8$.

Based on these observations, with the aid of a computer, we have determined one pair (w, x) satisfying the condition $(*)$ for any prime number $p \le 10^7$, $p = 80t + 1$, $t \equiv 1, 3, 7, 9 \pmod{10}$, as long as the least primitive element $w_0 \in GF(p)$ satisfies the condition $1 + w_0^{16t} \notin C_0^8$. To save the space, we list only a few examples with $p \le 5 \times 10^4$ in this paper.

LEMMA 3.15. *There exists a $(p, 5 \times 2, 1)$-splitting BIBD for each p listed in Appendix 1.*

Proof. For each p listed in Appendix 1, we have found a pair (w, x) which satisfies the condition $(*)$. Then $\mathcal{A} = \{A_i : 0 \le i \le t - 1\}$ forms the collection of base blocks of the desired $(p, 5 \times 2, 1)$-splitting BIBD. □

Now we consider large prime numbers $p = 80t + 1$ with $t \equiv 1, 3, 7, 9 \pmod{10}$. Under the assumption that in $GF(p)$ there is an element $y \notin C_0^5$ such that $1 + y^{16t} \notin C_0^8$, by applying Lemma 3.13 with $q = 8$ and $s = 5$, we know that for $p \ge 1.8 \times 10^{10}$, there always exists a pair (w, x), w a primitive element and x a nonzero element of $GF(p)$, which satisfies the condition $(*)$. This implies that there always exists a $(p, 5 \times 2, 1)$-splitting BIBD for each such p.

Summarizing the above, we obtain the following existence result for $(p, 5 \times 2, 1)$-splitting BIBDs.

THEOREM 3.16. *Let $p = 80t + 1$, $t \equiv 1, 3, 7, 9 \pmod{10}$, be a prime number such that in $GF(p)$ there is an element $y \notin C_0^5$ satisfying $1 + y^{16t} \notin C_0^8$. Then there exists a $(p, 5 \times 2, 1)$-splitting BIBD provided that $p \le 10^7$ or $p \ge 1.8 \times 10^{10}$.*

In the remainder of this subsection, we provide a construction for an infinite series of $(v, u \times c, 1)$-splitting BIBDs for any integers $u \ge 2$ and $c \ge 2$. Let q be a prime power, e a positive integer deviding $q - 1$, $r \ge 2$ an integer, and P_r the set of ordered pairs $\{(i, j) : 1 \le i < j \le r\}$. We define an *$r$-choice* (for e) to be any mapping $C : P_r \longrightarrow \mathcal{C}^e$, assigning to each pair $(i, j) \in P_r$ a cyclotomic class $C(i, j)$ of index e in $GF(q)$. An r-vector $(a_1, a_2, \ldots, a_r)$ of elements of $GF(q)$ is said to be *consistent* with the choice C if and only if $a_j - a_i \in C(i, j)$ for all $(i, j) \in P_r$.

LEMMA 3.17 (see [13]). *Let $q \equiv 1 \pmod{e}$ be a prime power and $q > e^{r(r-1)}$. Then for any r-choice $C : P_r \longrightarrow \mathcal{C}^e$, there exists an r-vector $(a_1, a_2, \ldots, a_r)$ of elements of $GF(q)$ consistent with C.*

THEOREM 3.18. *Let $u \ge 2$ and $c \ge 2$ be two integers. Then there exists a $(q, u \times c, 1)$-splitting BIBD for any prime power $q = 2u(u-1)c^2m + u(u-1)c^2 + 1$ satisfying $q > [u(u-1)c^2]^{uc(uc-1)}$, where m is some positive integer.*

Proof. Let $r = uc$ and $e = u(u-1)c^2$. We define a uc-choice $C_0 : P_{uc} \longrightarrow \mathcal{C}^{u(u-1)c^2}$ such that $\{C(sc + i, tc + j) : 0 \le s < t \le u - 1, 1 \le i, j \le c\} = \{C_0^{u(u-1)c^2}, C_1^{u(u-1)c^2}, \ldots, C_{\frac{1}{2}u(u-1)c^2-1}^{u(u-1)c^2}\}$. Then according to Lemma 3.17, we know that for any prime power $q = u(u-1)c^2n + 1$ satisfying $q > [u(u-1)c^2]^{uc(uc-1)}$, where n is some positive integer, there exists a uc-vector $(a_1, a_2, \ldots, a_{uc})$ of elements of $GF(q)$ consistent with the above defined uc-choice C_0. If we take $n = 2m+1$, then $-1 \in C_{\frac{1}{2}u(u-1)c^2}^{u(u-1)c^2}$. We then define a $u \times c$ array

$$B = \begin{pmatrix} a_1 & a_2 & \cdots & a_c \\ a_{c+1} & a_{c+2} & \cdots & a_{2c} \\ \cdots & \cdots & \cdots & \cdots \\ a_{(u-1)c+1} & a_{(u-1)c+2} & \cdots & a_{uc} \end{pmatrix}.$$

Clearly in this case, the differences $\Delta B = \mathrm{SDRC}(\mathcal{C}^{u(u-1)c^2})$. Therefore we know that $\{x \cdot B : x \in C_0^{u(u-1)c^2}\}$ forms the collection of base blocks of the desired $(q, u \times c, 1)$-splitting BIBD. □

Note that although the above proof is a constructive proof, which is usually more valuable than an existence proof, since we do not know explicitly the uc-vector $(a_1, a_2, \ldots, a_{uc})$, we expect strongly a more explicit direct construction for $(v, u\times c, 1)$-splitting BIBDs for any two integers $u \geq 2$ and $c \geq 2$.

4. Existence results. In this section, we will use the results obtained in the previous sections to establish the existence result of a $(v, u \times c, 1)$-splitting BIBD or, equivalently, an optimal c-splitting authentication code with u source states and v messages, for several small $u \geq 2$ and $c \geq 2$.

The following assertion can be easily proved by applying Lemma 2.1.

THEOREM 4.1. *The necessary conditions for the existence of a $(v, u\times c, 1)$-splitting BIBD or, equivalently, an optimal c-splitting authentication code with u source states and v messages, are that*

$$\begin{cases} v - 1 \equiv 0 \,(\mathrm{mod}\,(u-1)c), \\ v(v-1) \equiv 0 \,(\mathrm{mod}\, u(u-1)c^2). \end{cases}$$

Remembering that a $(v, u \times c, \lambda)$-splitting BIBD is in fact a $(\lambda K_v, K_c^u)$-balanced graph design, we can know, by a result due to Wilson [15] (see also [5]), that the above necessary conditions are also asymptotically sufficient for the existence of a $(v, u\times c, 1)$-splitting BIBD or, equivalently, an optimal c-splitting authentication code with u source states and v messages.

THEOREM 4.2 (see [15]). *For any given integers $u \geq 2$ and $c \geq 2$, there exists an integer $N(u, c)$ such that if $v > N(u, c)$, then the necessary conditions for the existence of a $(v, u \times c, 1)$-splitting BIBD are also sufficient.*

Unfortunately, the above is only an asymptotic existence result. We do not know the exact value of $N(u, c)$ for any given $u \geq 2$ and $c \geq 2$. In what follows, we determine the exact values of $N(u, c)$ for a few small $u \geq 2$ and $c \geq 2$. First we consider the case $u = 2$.

THEOREM 4.3. *Let $c \geq 2$ be any even integer. Then the necessary and sufficient condition for the existence of a $(v, 2\times c, 1)$-splitting BIBD is that $v-1 \equiv 0 \,(\mathrm{mod}\, 2c^2)$.*

Proof. According to Theorem 4.1, $v = cs + 1$ for some positive integer s. Then $cs(cs+1) \equiv 0 \,(\mathrm{mod}\, 2c^2)$, which implies $s(cs+1) \equiv 0 \,(\mathrm{mod}\, 2c)$, and $s \equiv 0 \,(\mathrm{mod}\, c)$. We may assume $s = ct$ for some positive integer t. Then $v = c^2t + 1$, and we know that $(c^2t+1)c^2t \equiv 0 \,(\mathrm{mod}\, 2c^2)$, which means $(c^2t+1)t \equiv 0 \,(\mathrm{mod}\, 2)$. Since c is even, t must be even, and therefore $v - 1 \equiv 0 \,(\mathrm{mod}\, 2c^2)$. This proves the necessity.

For the sufficiency, see Lemma 3.8. □

Complete existence results for odd c are not many. We have only completely solved the case $c = 3$.

THEOREM 4.4. *The necessary and sufficient condition for the existence of a $(v, 2 \times 3, 1)$-splitting BIBD is that $v \equiv 1 \,(\mathrm{mod}\, 9)$ except for $v = 10$.*

Proof. From Theorem 4.1, we can easily know that, for odd c, the necessary condition is $v - 1 \equiv 0 \,(\mathrm{mod}\, c^2)$. When $c = 3$, it becomes $v \equiv 1 \,(\mathrm{mod}\, 9)$.

We prove the sufficiency. The case $v \equiv 1 \,(\mathrm{mod}\, 18)$ can be solved by Lemma 3.8. Now we consider the case $v \equiv 10 \,(\mathrm{mod}\, 18)$. It is obvious that for any integers $t \geq 2$ and $c \geq 1$, there always exists a $\{2\}$-GDD of type $(2c)^{t-1}(3c)^1$, where any two points in different groups form a block. By applying Corollary 3.5, we obtain a $2\times c$-splitting GDD of type $(2c^2)^{t-1}(3c^2)^1$. Then applying Corollary 3.7 with a $(2c^2 + 1, 2 \times c, 1)$-splitting BIBD from Lemma 3.8, we know that there exists a $(2c^2t + c^2 + 1, 2 \times c, 1)$-splitting BIBD for any integer $t \geq 2$, provided that there exists a $(3c^2 + 1, 2 \times c, 1)$-

splitting BIBD. According to Lemma 3.9, there exists a $(28, 2 \times 3, 1)$-splitting BIBD, and therefore there exists an $(18t + 10, 2 \times 3, 1)$-splitting BIBD for any integer $t \geq 2$.

Finally, we prove the nonexistence of a $(10, 2 \times 3, 1)$-splitting BIBD. If such a splitting BIBD exists, then there would be in total five blocks, and every point would occur in exactly three blocks. Without loss of generality, we may assume that the set of points of this splitting BIBD is $\mathcal{V} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, and from the definition of a splitting BIBD, we may also assume that the three blocks containing the point $1 \in \mathcal{V}$ are

$$\begin{pmatrix} 1 & a & b \\ 2 & 3 & 4 \end{pmatrix}, \begin{pmatrix} 1 & c & d \\ 5 & 6 & 7 \end{pmatrix}, \begin{pmatrix} 1 & e & f \\ 8 & 9 & 10 \end{pmatrix}.$$

Note that the points in $\{2, 3, 4\}$, $\{5, 6, 7\}$, and $\{8, 9, 10\}$ can be permuted randomly within their set, respectively, up to isomorphism. So, without loss of generality, we need only to consider the cases $\{a, b\} = \{5, 6\}$ and $\{5, 8\}$, that is, $\{a, b\}$ in the same block and in different blocks.

(1) $\{a, b\} = \{5, 6\}$: From the definition of a splitting BIBD, it is easy to see that $\{c, d\}$ can be contained only in $\{8, 9, 10\}$ and $\{e, f\}$ can be contained only in $\{2, 3, 4\}$. Since the points in $\{8, 9, 10\}$ can be relabeled randomly up to isomorphism, we may assume $\{c, d\} = \{8, 9\}$. Similarly, we may assume $\{e, f\} = \{2, 3\}$. Now, consider the third block containing the point $2 \in \mathcal{V}$. This new block should be of the form below, up to isomorphism:

$$\begin{pmatrix} 2 & x & y \\ 3 & 4 & 7 \end{pmatrix}.$$

Counting the pairs containing the point $3 \in \mathcal{V}$, we know that x, y can be contained only in $\{2, 4, 7\}$. This is impossible since every point can occur at most once in any block. Therefore in this case, no $(10, 2 \times 3, 1)$-splitting BIBD can exist.

(2) $\{a, b\} = \{5, 8\}$: Similarly, we know that c, d can be contained only in $\{8, 9, 10\}$ and e, f can be contained only in $\{5, 6, 7\}$. Then it can be easily seen that at least one of the pairs of points from $\{5, 6, 7\} \times \{8, 9, 10\}$ occurs in

$$\begin{pmatrix} 1 & c & d \\ 5 & 6 & 7 \end{pmatrix}, \begin{pmatrix} 1 & e & f \\ 8 & 9 & 10 \end{pmatrix}$$

simultaneously. This leads to a contradiction to the definition of a splitting BIBD with $\lambda = 1$ that each pair of points should appear in different rows of exactly one block. Therefore, in this case, no $(10, 2 \times 3, 1)$-splitting BIBD can exist either.

This completes the proof. □

Next we investigate the existence of a $(v, 3 \times 2, 1)$-splitting BIBD.

LEMMA 4.5 (see [2]). *Let g, t, and u be nonnegative integers. Then there exists a $\{3\}$-GDD of type $g^t u^1$ if and only if the following conditions are satisfied:*

(1) *if $g > 0$, then $t \geq 3$, or $t = 2$ and $u = g$, or $t = 1$ and $u = 0$, or $t = 0$;*
(2) *$u \leq g(t - 1)$ or $gt = 0$;*
(3) *$g(t - 1) + u \equiv 0 \pmod 2$ or $gt = 0$;*
(4) *$gt \equiv 0 \pmod 2$ or $u = 0$;*
(5) *$\frac{1}{2}g^2t(t - 1) + gtu \equiv 0 \pmod 3$.*

THEOREM 4.6. *The necessary and sufficient condition for the existence of a $(v, 3 \times 2, 1)$-splitting BIBD is that $v \equiv 1, 9 \pmod{24}$ except for $v = 9$.*

Proof. The necessity is clear from Theorem 4.1. We are to prove the sufficiency.

From Lemma 4.5, there exists a $\{3\}$-GDD of type 12^t for any integer $t \geq 3$. Giving weight 2 to each point of this $\{3\}$-GDD and applying Corollary 3.5, we obtain a 3×2-splitting GDD of type 24^t for any integer $t \geq 3$. From Lemma 3.10, there exists a $(25, 3 \times 2, 1)$-splitting BIBD. Applying Corollary 3.7, we obtain a $(24t+1, 3 \times 2, 1)$-splitting BIBD for any integer $t \geq 3$. A $(49, 3 \times 2, 1)$-splitting BIBD also exists from Lemma 3.10. Therefore a $(24t + 1, 3 \times 2, 1)$-splitting BIBD exists for any positive integer t.

Similarly, from Lemma 4.5, there exists a $\{3\}$-GDD of type $12^t(2m)^1$ for any integers $t \geq 3$ and $0 \leq m \leq 6(t-1)$. Giving weight 2 to each point of this $\{3\}$-GDD and applying Corollary 3.5, we obtain a 3×2-splitting GDD of type $24^t(4m)^1$ for any integers $t \geq 3$ and $0 \leq m \leq 6(t-1)$. From Lemma 3.10, there exist a $(25, 3 \times 2, 1)$-splitting BIBD and a $(33, 3 \times 2, 1)$-splitting BIBD. Taking $m = 8$ and applying Corollary 3.7, we obtain a $(24t + 33, 3 \times 2, 1)$-splitting BIBD for any integer $t \geq 3$. Both a $(57, 3 \times 2, 1)$-splitting BIBD and an $(81, 3 \times 2, 1)$-splitting BIBD also exist from Lemma 3.10. Therefore a $(24t + 9, 3 \times 2, 1)$-splitting BIBD exists for any positive integer t.

Now we prove the nonexistence of a $(9, 3\times 2, 1)$-splitting BIBD. If such a splitting BIBD exists, then there would be in total three blocks, and every point would occur in exactly two blocks. Without loss of generality, we may assume that the set of points of this splitting BIBD is $\mathcal{V} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, and one of the three possible blocks of this splitting BIBD is

$$\begin{pmatrix} 1 & a \\ 2 & 3 \\ 4 & 5 \end{pmatrix},$$

where the point $a \in \mathcal{V}$ should be in the subset $\{6, 7, 8, 9\} \subset \mathcal{V}$ since any point cannot occur more than once in any block. We consider the other possible block containing the point $1 \in \mathcal{V}$. We may suppose, again without loss of generality, that this block is of the form

$$\begin{pmatrix} 1 & b \\ c & d \\ e & f \end{pmatrix},$$

where any point of $\{c, d, e, f\}$ must not be in $\{2, 3, 4, 5\}$, that is, the subset $\{c, d, e, f\}$ of points must be equal to $\{6, 7, 8, 9\}$. Now we focus our attention on the two points a and b. The point a should be in $\{6, 7, 8, 9\}$, and the point b should be in $\{2, 3, 4, 5\}$. As an immediate consequence, the pair $\{a, b\}$ of distinct points would appear in different rows of both the first and second possible blocks. This leads to a contradiction to the definition of a splitting BIBD with $\lambda = 1$ that each pair of points should appear in different rows of exactly one block. Therefore, there is no $(9, 3 \times 2, 1)$-splitting BIBD.

The proof is then completed. □

Finally, we consider the existence of a $(v, 4 \times 2, 1)$-splitting BIBD. We need the following result on $\{4\}$-GDDs due to Ge and Ling [3].

LEMMA 4.7 (see [3]). *There exists a $\{4\}$-GDD of type $12^t m^1$ for any integers $t \geq 4$ and $m \equiv 0 \pmod 3$ with $0 \leq m \leq 6(t-1)$.*

Then we can prove the following result.

THEOREM 4.8. *The necessary and sufficient condition for the existence of a $(v, 4 \times 2, 1)$-splitting BIBD is that $v \equiv 1 \pmod{48}$, with two possible exceptions $v = 49, 385$.*

Proof. The necessity is an easy corollary of Theorem 4.1. We need only to prove the sufficiency.

According to Lemma 4.7, there exists a $\{4\}$-GDD of type $12^t m^1$ for any integers $t \geq 4$ and $m \equiv 0 \pmod 3$ with $0 \leq m \leq 6(t-1)$. Giving weight 4 to each point of this $\{4\}$-GDD and applying Theorem 3.1 with the well-known ingredient $\{4\}$-GDD of type 4^4, we obtain a $\{4\}$-GDD of type $48^t(4m)^1$ for any integers $t \geq 4$ and $m \equiv 0 \pmod 3$ with $0 \leq m \leq 6(t-1)$. Again giving weight 2 to each point of the resultant $\{4\}$-GDD, and applying Corollary 3.5, we obtain a 4×2-splitting GDD of type $96^t(8m)^1$ for any integers $t \geq 4$ and $m \equiv 0 \pmod 3$ with $0 \leq m \leq 6(t-1)$. From Lemma 3.11, there exists a $(97, 4 \times 2, 1)$-splitting BIBD. If there exists an $(8m+1, 4 \times 2, 1)$-splitting BIBD, where $m \equiv 0 \pmod 3$ with $0 \leq m \leq 6(t-1)$, then by applying Corollary 3.7, we obtain a $(96t + 8m + 1, 4 \times 2, 1)$-splitting BIBD for any integer $t \geq 4$. A $(145, 4 \times 2, 1)$-splitting BIBD also exists by Lemma 3.11. Taking $m = 12, 18$, we obtain a $(v, 4 \times 2, 1)$-splitting BIBD for any positive integer $v \equiv 1 \pmod{48}$ except possibly for $v = 49, 193, 241, 289, 337, 385, 433$. All of these remaining splitting BIBDs except for $v = 49, 385$ have been constructed in Lemma 3.11. Therefore a $(v, 4 \times 2, 1)$-splitting BIBD exists for any positive integer $v \equiv 1 \pmod{48}$ except possibly for $v = 49, 385$. □

5. Concluding remarks. In this paper, we studied optimal c-splitting authentication codes from a combinatorial viewpoint. We described various combinatorial constructions for optimal c-splitting authentication codes. We showed that the necessary conditions for the existence of an optimal c-splitting authentication code with u source states and v messages are also sufficient for $(u, c) = (2, 2t)$ for any positive integer t, $(u, c) = (2, 3)$ with a definite exception of $v = 10$, $(u, c) = (3, 2)$ with a definite exception of $v = 9$, and $(u, c) = (4, 2)$ with two possible exceptions of $v = 49, 385$. However, many problems are left open. For example, what is the explicit expression of $N(u, c)$ as a function of u and c in Theorem 4.2? Another challenging problem, which might be more important from a cryptographic point of view, is how to directly and explicitly construct a $(u(u-1)c^2t + 1, u \times c, 1)$-splitting BIBD or, equivalently, an optimal c-splitting authentication code with u source states and $c^2u(u-1)t + 1$ messages, for any given $t \geq 1$ and $c \geq 2$, whenever $c^2u(u-1)t + 1$ is a prime power.

Appendix 1.

(p, w, x)	(p, w, x)	(p, w, x)	(p, w, x)
(241,7,159)	(881,3,231)	(1361,3,651)	(2161,23,1090)
(3121,7,2156)	(3761,3,1396)	(4241,3,3024)	(4561,11,3205)
(4721,6,3614)	(5521,11,574)	(6481,7,6313)	(6961,13,2063)
(7121,3,718)	(9041,3,349)	(9521,3,729)	(10321,7,5522)
(12241,7,6768)	(12721,13,10138)	(13681,22,7398)	(13841,6,10800)
(14321,3,3392)	(15121,11,10850)	(15761,3,12740)	(16561,7,4126)
(17041,7,13872)	(17681,3,15559)	(18481,13,7135)	(19121,6,729)
(19441,13,8488)	(21521,3,8696)	(21841,11,8995)	(22481,3,11916)
(22961,6,8634)	(23761,7,3574)	(26161,13,21228)	(26321,3,19049)
(27281,6,6490)	(28081,19,501)	(30161,3,27)	(31121,3,18682)
(32561,6,30665)	(34961,3,21425)	(36721,37,18918)	(37361,3,11291)
(38321,3,33069)	(39761,3,16151)	(40241,3,12654)	(41521,22,8753)
(42641,3,9053)	(42961,11,9487)	(43441,11,12459)	(45361,11,3117)
(45841,7,18310)	(47441,3,11350)	(49681,17,36263)	

Acknowledgments. The authors express their heartfelt gratitude to Professor L. Zhu for his many constructive discussions and suggestions. Without his unselfish help, this paper could not be in the present form. The authors also thank the two anonymous referees and Professor Kevin T. Phelps, the editor, for their helpful comments.

REFERENCES

[1] Y. Chang and L. Ji, *Optimal* $(4up, 5, 1)$ *optical orthogonal codes*, J. Combin. Des., 12 (2004), pp. 346–361.

[2] C. J. Colbourn, D. G. Hoffman, and R. Rees, *A new class of group divisible designs with block size three*, J. Combin. Theory Ser. A, 59 (1992), pp. 73–89.

[3] G. Ge and A. C. H. Ling, *Group divisible designs with block size four and group type* $g^u m^1$ *for small* g, Discrete Math., 285 (2004), pp. 97–120.

[4] E. N. Gilbert, F. J. MacWilliams, and N. J. A. Sloane, *Codes which detect deception*, Bell System Tech. J., 53 (1974), pp. 405–424.

[5] K. Heinrich, *Graph decompositions and designs*, in The CRC Handbook of Combinatorial Designs, CRC Press, Boca Raton, FL, 1996, pp. 361–366.

[6] T. Johansson, *Lower bounds on the probability of deception in authentication with arbitration*, IEEE Trans. Inform. Theory, 40 (1994), pp. 1573–1585.

[7] K. Kurosawa, *New bound on authentication code with arbitration*, in Advances in Cryptology—CRYPTO'94, Lecture Notes in Comput. Sci. 839, Springer, Berlin, 1994, pp. 140–149.

[8] R. Lidl and H. Niederreiter, *Finite Fields*, Cambridge University Press, Cambridge, UK, 1997.

[9] W. Ogata, K. Kurosawa, D. R. Stinson, and H. Saido, *New combinatorial designs and their applications to authentication codes and secret sharing schemes*, Discrete Math., 279 (2004), pp. 383–405.

[10] G. J. Simmons, *Authentication theory/coding theory*, in Advances in Cryptology—CRYPTO'84, Lecture Notes in Comput. Sci. 196, Springer, Berlin, 1985, pp. 411–431.

[11] G. J. Simmons, *Message authentication with arbitration of transmitter/receiver disputes*, in Advances in Cryptology—EUROCRYPT'87, Lecture Notes in Comput. Sci. 304, Springer, Berlin, 1988, pp. 150–165.

[12] G. J. Simmons, *A Cartesian product construction for unconditionally secure authentication codes that permit arbitration*, J. Cryptology, 2 (1990), pp. 77–104.

[13] R. M. Wilson, *Cyclotomy and difference families in elementary abelian groups*, J. Number Theory, 4 (1972), pp. 17–47.

[14] R. M. Wilson, *Constructions and uses of pairwise balanced designs*, in Combinatorics Part I: Theory of Designs, Finite Geometry and Coding Theory, Math. Centre Tracts 55, Math. Centrum, Amsterdam, 1974, pp. 18–41.

[15] R. M. Wilson, *Decompositions of complete graphs into subgraphs isomorphic to a given graph*, Congr. Numer., 15 (1976), pp. 647–659.

SIAM J. DISCRETE MATH.
Vol. 18, No. 4, pp. 679–696

LOWER BOUNDS FROM TILE COVERS FOR THE CHANNEL ASSIGNMENT PROBLEM*

J. C. M. JANSSEN†, T. E. WENTZELL†, AND S. L. FITZPATRICK‡

Abstract. A method to generate lower bounds for the channel assignment problem is given. The method is based on the reduction of the channel assignment problem to a problem of covering the demand in a cellular network by preassigned blocks of cells called tiles. This tile cover approach is applied to networks with a cosite constraint and two different constraints between cells. A complete family of lower bounds is obtained, which include a number of new bounds that improve or include almost all known clique bounds. When applied to an example from the literature, the new bounds give better results.

Key words. channel assignment, graph labeling, polyhedral lower bounds

AMS subject classifications. 05C78, 05C90, 90C57

DOI. 10.1137/S0895480101384402

1. Introduction.

Finding an optimal assignment of communication channels in a cellular network is a difficult combinatorial optimization problem which has received considerable attention over the last decade. This is due to the explosive growth of wireless communications and the scarcity of the radio spectrum. The channel assignment problem (CAP) is NP-complete even in a drastically simplified form, and, consequently, most efforts have gone toward the development of good heuristics. (Recently, integer programming techniques which can lead to exact solutions have been used. See, for example, [12].) Lower bounds play an important role in the evaluation of any heuristic or approximation algorithm. Moreover, lower bounds can help to identify the structures that form the bottleneck for a particular instance, and this information can, in turn, be used to find better assignments.

A basic model for a cellular network describes it in terms of the *demand* for channels in each cell and a set of *reuse constraints* which prescribe minimal separations that must exist between channels assigned to certain cells in order to avoid interference. The goal of the CAP is to assign channels (represented by integers) to the cells such that each cell receives as many channels as its demand requires while respecting the reuse constraints. Here, the objective is to minimize the *span* of the assignment, which is the difference between the highest and the lowest channel assigned. (An alternative objective, when a limited span is given, can be to minimize the number of violated interference constraints.)

Cellullar networks can be modeled as graphs where the nodes of the graph represent the cells, and two nodes are adjacent precisely when there exists a (nonzero) reuse constraint between them. The demands are given by a *weight vector* indexed by the nodes, and the reuse constraints are given by a vector indexed by the nodes and

*Received by the editors February 5, 2001; accepted for publication (in revised form) August 10, 2004; published electronically April 22, 2005. This research was supported by the Natural Science and Engineering Research Council of Canada.
http://www.siam.org/journals/sidma/18-4/38440.html
†Department of Mathematics and Statistics, Dalhousie University, Halifax B3H 3J5, NS, Canada (janssen@mathstat.dal.ca, tania@mathstat.dal.ca).
‡Department of Mathematics and Computer Science, University of Prince Edward Island, Charlottetown C1A 4P3, PE, Canada (slfitzpatric@upei.ca).

edges. When all reuse constraints are 1, the CAP reduces to the problem of finding a coloring of a weighted graph.

The minimal span needed for any assignment will generally be determined by the cells with highest demand. It is reasonable to assume that these cells will often be geographically close, corresponding, for example, to a business district or a city center. Since interference also tends to be highest between cells that are close, these cells will often form a clique in the underlying graph.

Most lower bounds for the CAP are therefore based on cliques. The simplest clique bound, mentioned in [6] but generally considered folklore, is found by assuming that all edge constraints and cosite constraints are equal to the lowest constraint in the clique. A first refinement was obtained in [6] by considering two different constraints. A second refinement, similar to the situation studied here, was considered in [18]. In all of these cases, bounds were obtained using ad hoc methods.

In this paper, we study networks where the reuse constraint between different cells can take only three values, one of which is reserved for the *cosite constraint*. The cosite constraint is the reuse constraint between channels assigned to the same cell, or node. Naturally, any bounds obtained from this approach can also be used in networks with more general constraints by reducing the constraints in any particular set of edges to the lowest constraint in that set.

We describe how lower bounds can be generated from an approach based on reducing the CAP to a covering problem. The crucial step is to show that any channel assignment can be broken down into small blocks called *tiles*. A *tile cover* is a collection of tiles such that the number of tiles covering a node equals the number of channels assigned to that node. The conversion of the CAP to a tile cover problem brings the advantage that tile covers can be easily analyzed using linear programming (LP) duality and polyhedral methods. A similar tile cover method, applied to the simpler case of cliques with one cosite constraint and one edge constraint, can be found in [10]. This particular result is used in our paper as the base case for the induction which forms the proof of our main theorem. In [13], heuristic channel assignment methods using preassigned "tiles" of assigned channels are applied successfully to a number of CAP instances.

We apply the tile cover approach to configurations which we call *nested cliques*. These are cliques consisting of an *inner* clique and an *outer* clique where all edge constraints involving an inner clique node take the larger constraint value, while all edge constraints containing only nodes from the outer clique take the smaller value (see section 2 for a more precise definition). Nested cliques arise naturally from the geographical layout of cellular networks and from the fact that interference levels are generally lower between transmitters that are at greater distance from each other. Hence, it will be common to find a cluster of cells with high interference constraints between them surrounded by an outer shell of cells at greater distance and thus with weaker interference constraints. Such a situation will form a nested clique in the interference graph.

Using the tile cover approach on nested cliques, we derive a comprehensive family of general "second generation" clique bounds. This family includes all bounds from [6] and improves the bound obtained in [18]. We also show, using an example, how the approach can be used directly to obtain specific lower bounds for any specific set of parameters.

There are two types of clique bounds that cannot be derived directly from our approach. In [15], [8], and [16], it was shown how the traveling salesman problem

and its linear program relaxation can be used to derive lower bounds for cliques. This approach is most effective when the cosite constraint is relatively low. In [2], an integer programming approach for obtaining upper and lower bounds is given, which is based on *d-walks*, i.e., walks that cover each node more than once. This method is somewhat related to the tile cover method, since paths between successive visits of a node in the walk can be seen as tiles.

In [19] a lower bounding method is described which is based on network flows. We will show that our tile cover bounds give an improvement of 13% when applied to the example given in this paper.

Since it is NP-hard to find a maximum weight clique in a graph, it will also be hard to find the nested clique that gives the best bound. However, clique enumeration procedures such as the *Carraghan–Pardalos* algorithm (see [5]) give good performance in practice. The reduction of the CAP to a tile cover problem leads to an easy way of computing the lower bound for any particular clique by way of a linear program. Alternatively, any particular network can be analyzed in advance using our method, and a complete family of easily computable lower bounds can be obtained. Therefore, we expect the computation of the best tile cover clique bound to be feasible and realistic.

The layout of the paper is at follows. After introducing some formal definitions related to channel assignment in section 2, we introduce and define the concepts involved in the tile cover method in section 3. At the end of this section we also state our main result, namely, that each channel assignment can be reduced to a tile cover, such that the cost of the cover is no larger than the span of the assignment. In section 4, we develop lower bounds for tile covers using an LP formulation and we show how they translate into bounds for the CAP. In section 5, the proof of the main theorem is given.

2. Preliminaries. For the basic definitions of graph theory we refer to [4]. A (simple) graph G is a pair (V, E) of a node set V and an edge set E, where each edge $e \in E$ is an unordered pair of nodes. A *clique* in a graph is a set of nodes of which every pair is adjacent.

In this paper, we will use the following notation for integer vectors: if $y \in \mathbb{Z}^V$ for some set V, then $y(v)$ is the coordinate of y indexed by v. Sets will often be represented by their characteristic vectors. Given a set V and $A \subseteq V$, the characteristic vector $\chi^A \in \mathbb{Z}_+^V$ is defined as follows:

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

Conversely, given a vector $y \in \mathbb{Z}_+^V$, the *support* of y, denoted by $V(y)$, is the set of all nodes in V indexing nonzero coordinates of y, so

$$V(y) = \{v \in V : y(v) > 0\}.$$

A *constrained graph* $G = (V, E, s, e)$ is a graph with node set V, edge set E, and positive integer constraint vectors $s \in \mathbb{Z}_+^V$, $e \in \mathbb{Z}_+^E$. Vectors s and e represent the channel reuse constraints: vector s represents the *cosite constraints*, the required separation between channels assigned to the same node, and e represents the *edge constraints*, the required separation between channels assigned to the two endpoints of an edge.

A *constrained, weighted graph* is a pair (G, w) where G is a constrained graph and w is a positive integral weight vector indexed by the nodes of G. The coordinate of

w corresponding to node u is denoted by $w(u)$ and called the *weight* of node u. The weight of node u represents the number of channels needed at node u.

A *channel assignment* for a constrained, weighted graph (G, w) where $G = (V, E, s, e)$ is an assignment f of sets of nonnegative integers (which will represent the channels) to the nodes of G that satisfies the conditions

$$\begin{array}{ll} |f(u)| = w(u) & (u \in V), \\ i \in f(u) \text{ and } j \in f(v) \Rightarrow |i - j| \geq e(uv) & (uv \in E,\ u \neq v), \\ i, j \in f(u) \text{ and } i \neq j \Rightarrow |i - j| \geq s(u) & (u \in V). \end{array}$$

For reasons of brevity, throughout this paper we will use the notation $f(V)$ to denote $f(V) = \bigcup_{u \in V} f(u)$, in deviation from the standard definition of $f(V) = \{f(u) \,|\, u \in V\}$.

The *span* $S(f)$ of a channel assignment f of a constrained weighted graph is the difference between the lowest and the highest channel assigned by f, in other words, $S(f) = \max_{v \in V} f(v) - \min_{v \in V} f(v)$. The span $S(G, w)$ of a constrained, weighted graph G and a positive integer vector w indexed by the nodes of G is the minimum span of any channel assignment for (G, w).

We will consider complete graphs with constraints that have a special, nested structure. A constrained graph $G = (V, E, s, e)$ is a *nested clique* with parameters (k, u, a), where $k \geq u > a$ if $s(v) \geq k$ for all $v \in V$, and V can be partitioned into two sets Q and R such that $e(vw) \geq a$ if $v, w \in R$, and $e(vw) \geq u$ otherwise. The parameters k, u, and a are always assumed to be positive integers.

3. Tile covers. In this paper, we reduce the channel assignment problem for nested cliques to a tile covering problem. The tiles that may be used for a tile cover are defined in this section. We can think of these tiles as partial assignments, or "building blocks," from which any possible assignment can be constructed.

We assume that a particular nested clique G with node partition (Q, R) and parameters (k, u, a) is given. We define the set $\mathcal{T}$ of all possible tiles that may be used in a tile cover of G. All tiles are defined as vectors indexed by the nodes of G. For reasons of brevity we will sometimes identify a tile with its support and thus think of tiles as node sets. It is this representation that allows mention of "the nodes in tile t."

In order to facilitate the definition and the proof of Theorem 5.1, we distinguish various categories of tiles. So

$$\mathcal{T} = \mathcal{T}_Q \cup \mathcal{T}_R \cup \mathcal{T}_{QR} \cup \mathcal{T}_{QR}^{big}.$$

The tiles in each category are defined as

$$\begin{array}{lcl} \mathcal{T}_Q & = & \{\chi^A : A \subseteq Q\}, \\ \mathcal{T}_R & = & \{\chi^B : B \subseteq R\}, \\ \mathcal{T}_{QR} & = & \{\chi^A + \chi^B : A \subseteq Q,\ B \subseteq R, \text{ where } A \neq \emptyset,\ B \neq \emptyset\}, \\ \mathcal{T}_{QR}^{big} & = & \{\chi^{A \cup B} + \chi^{A_2 \cup B_2} : A_2 \subseteq A \subseteq Q,\ B_2 \subseteq B \subseteq R,\ A_2 \neq \emptyset,\ B_2 \neq \emptyset\}. \end{array}$$

The tiles in $\mathcal{T}_{QR}^{big}$ will be called *big* tiles. Note that all coefficients of tiles in $\mathcal{T}_Q$, $\mathcal{T}_R$, and $\mathcal{T}_{QR}$ have value either zero or 1, while for tiles in $\mathcal{T}_{QR}^{big}$, the coefficients indexed by nodes in A_2 and B_2 have value 2.

A *tiling* is a collection of tiles from $\mathcal{T}$ (multiplicities are allowed). We represent a tiling by a nonnegative integer vector $y \in \mathbb{Z}_+^{\mathcal{T}}$, where $y(t)$ represents the number of copies of tile t present in the tiling. A *tile cover* of a weighted nested clique (G, w) is a tiling y such that $\sum_{t\in\mathcal{T}} y(t)t(v) \geq w(v)$ for each node v of G.

With each tile $t \in \mathcal{T}$ we associate a cost $c(t)$. The costs of the tiles in each category are given in Table 1. The cost of each tile t is derived from the span of a channel assignment for (G, t) plus a "link-up" cost of connecting the assignment to a following tile. This link-up cost is calculated using the assumption that the same assignment will be repeated. For example, $t = \chi^A$, where $A = \{v_0, \ldots, v_{j-1}, v_j\}$, is a tile of $j + 1$ distinct vertices in Q. Then the minimum span of (G, t) is u, and an assignment of minimum span would be $f(v_i) = iu$ for all i. However, if this assignment is repeated, the next channel that can be assigned will be $(j + 1)u$, which is u more than the highest channel in the assignment. Hence the link-up cost of this assignment equals u.

It will follow from Theorem 5.1 that our choice of the costs is justified.

TABLE 1
Costs of tiles.

Category	Number of nodes in Q	Number of nodes in R	Cost
$\mathcal{T}_Q$	n	0	$\max\{k, nu\}$
$\mathcal{T}_R$	0	m	$\max\{k, ma\}$
$\mathcal{T}_{QR}$	n	m	$\max\{k, nu + ma + u - a\}$
$\mathcal{T}_{QR}^{big}$	n, of which n_2 have value 2	m, of which m_2 have value 2	$\max\{k, nu\} + \max\{k, ma\}$ $+ n_2u + m_2a + u - a$

Formally, the cost of a tile t is such that for any constant α the minimum span of $(G, \alpha t)$ equals $\alpha c(t)$ minus a small constant, or

$$\frac{S(G, \alpha t)}{\alpha} \to c(t) \text{ as } \alpha \to \infty.$$

The *cost* of a tiling y, denoted by $c(y)$, is the sum of the cost of the tiles in the tiling. So $c(y) = \sum_{t\in\mathcal{T}} y(t)c(t)$. The minimum cost of a tile cover of a weighted nested clique (G, w) will be denoted by $\tau(G, w)$.

4. Polyhedral bounds from tile covers. In section 5 we will prove the following theorem.

THEOREM 5.1. *Let G be a nested clique with node partition (Q, R) and parameters (k, u, a). Then for any weight vector w for G,*

$$S(G, w) \geq \tau(G, w) - k.$$

In this section, we will demonstrate how this theorem, combined with polyhedral methods, leads to new lower bounds for $S(G, w)$.

The problem of finding a minimum cost tile cover of (G, w) can be formulated as an integer program (IP):

$$\begin{aligned}
\text{Minimize} \quad & \sum_{t\in\mathcal{T}} c(t)y(t) \\
\text{subject to:} \quad & \sum_{t\in\mathcal{T}} t(v)y(t) \geq w(v) \qquad (v \in V), \\
& y(t) \geq 0 \qquad\qquad\qquad (t \in \mathcal{T}), \\
& y \text{ integer.}
\end{aligned}$$

We obtain the LP relaxation of this IP by removing the requirement that y must be integral. Any feasible solution to the resulting linear program is called a *fractional* tile cover. The minimum cost of a fractional tile cover gives a lower bound on the minimum cost of a tile cover. The dual of this LP is formulated as follows:

$$\begin{aligned} \text{Maximize} \quad & \sum_{v \in V} w(v)x(v) \\ \text{subject to:} \quad & \sum_{v \in V} t(v)x(v) \leq c(t) \qquad (t \in \mathcal{T}), \\ & x(v) \geq 0 \qquad\qquad\qquad (v \in V). \end{aligned}$$

By LP duality, the maximum of the dual is equal to the minimum cost of a fractional tile cover. Thus, any vector that satisfies the inequalities of the dual program gives a lower bound on the cost of a minimum fractional tile cover, and therefore also on the span of the corresponding complete constrained, weighted graph. The maximum is achieved by one of the vertices of the polytope $TC(G)$ defined as follows:

$$TC(G) = \left\{ x \in \mathbb{Q}_+^V : \sum_{v \in V} t(v)x(v) \leq c(t) \text{ for all } t \in \mathcal{T} \right\}.$$

A classification of the vertices of this polytope will therefore lead to a comprehensive set of lower bounds that can be obtained from fractional tile covers. The next theorem demonstrates the strength of the tile cover approach, by giving a family of bounds for nested cliques with parameters $(k, u, 1)$.

THEOREM 4.1. *Let G be a nested clique with node partition (Q, R) and parameters $(k, u, 1)$. Let $w \in \mathbb{Z}_+^V$ be a weight vector for G, and let w_{Qmax} be the maximum weight of any node in Q, and w_{Rmax} the maximum weight of any node in R. Then*

$$\tau(G, w) \geq (\lambda_1 - \lambda_2)w_{Qmax} + \lambda_2 \sum_{v \in Q} w(v) + (\lambda_3 - \lambda_4)w_{Rmax} + \lambda_4 \sum_{v \in R} w(v)$$

for each 4-tuple $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$, where λ_1, λ_2, λ_3, and λ_4 can take the following values:

λ_1	λ_2	λ_3	λ_4	Case
k	0	0	0	(1)
0	0	k	0	(2)
$k-(\mu-1)\delta$	δ	δ	0	(3)
δ	δ	$k-(\mu-1)\delta$	0	(4)
$k-(\mu-1)\delta$	δ	ϵ	ϵ	(5)
u	u	1	1	(6)
u	u	u	$\frac{k-u}{k-1}$	(7)
$2u-1$	ν	1	1	(8)

where

$$\begin{aligned} &\mu = \lfloor \tfrac{k}{u} \rfloor, \\ &\delta = (\mu+1)u - k, \\ &\epsilon = \begin{cases} 1 & \text{if } \mu = 1, \\ \min\left\{ \frac{\delta}{k-2u+1}, \frac{2u+\mu\delta-\delta}{k+1}, 1 \right\} & \text{otherwise,} \end{cases} \\ &\nu = \begin{cases} 1 & \text{if } \mu = 1, \\ u - \max\left\{ \frac{u-1}{\mu}, \frac{\delta-1}{\mu-1} \right\} & \text{otherwise.} \end{cases} \end{aligned}$$

Proof. For the proof we consider feasible points in $TC(G)$ that are of the form

$$\lambda_1\chi^{\{q\}} + \lambda_2\chi^{Q-\{q\}} + \lambda_3\chi^{\{r\}} + \lambda_4\chi^{R-\{r\}}, \quad \text{where } q \in Q,\ r \in R,\ \lambda_1 \geq \lambda_2,\ \lambda_3 \geq \lambda_4.$$

For such points, the inequality system that defines $TC(G)$ reduces to the following form:

$$\lambda_1 + (\mu-1)\lambda_2 \leq k, \tag{1}$$

$$\lambda_1 + \mu\lambda_2 \leq (\mu+1)u, \tag{2}$$

$$\lambda_3 + (k-1)\lambda_4 \leq k, \tag{3}$$

$$\lambda_1 + (\mu-2)\lambda_2 + \lambda_3 + (k-\mu u)\lambda_4 \leq k, \tag{4}$$

$$\lambda_1 + (\mu-1)\lambda_2 + \lambda_3 \leq (\mu+1)u, \tag{5}$$

$$2\lambda_1 + (\mu-1)\lambda_2 + 2\lambda_3 + (k-1)\lambda_4 \leq 2k + 2u, \tag{6}$$

$$2\lambda_1 + \mu\lambda_2 + 2\lambda_3 + (k-1)\lambda_4 \leq k + (\mu+3)u, \tag{7}$$

$$\lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0. \tag{8}$$

Inequalities (1) and (2) are obtained by choosing tiles of size μ and $\mu+1$, respectively, from $\mathcal{T}_Q$. Inequality (3) is derived from a tile of size k from $\mathcal{T}_R$.

Inequalities (4) and (5) are derived from tiles in $\mathcal{T}_{QR}$. Inequality (4) is derived from a tile with $\mu-1$ nodes in Q and $k-\mu u-u+1$ nodes in R, and inequality (5) from a tile with μ nodes in Q and one node in R.

Inequalities (6) and (7) are obtained by choosing tiles from $\mathcal{T}_{QR}^{big}$, where nodes q and r have weight 2, all other nodes have weight 1, $m = k$, and $n = \mu$ or $n = \mu+1$, respectively.

Note that inequalities (2) and (3) imply that $\lambda_2 \leq u$ and $\lambda_4 \leq 1$. Using this fact, it is easy to see that all inequalities that correspond to tiles other than those mentioned are implied by inequalities (1)–(7).

It can be verified that each of the points provided in the statement of the theorem provides a feasible solution to the system. Note that each of the feasible solutions satisfies at least one inequality with equality. So, for each vector $x = \lambda_1\chi^{\{q\}} + \lambda_2\chi^{Q-\{q\}} + \lambda_3\chi^{\{r\}} + \lambda_4\chi^{R-\{r\}}$ with $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ as given, and q and r any nodes in Q and R, respectively, it holds that $x \in TC(G)$. Therefore, $\tau(G, w) \geq \sum_{v\in V} w(v)x(v)$. Since $\lambda_1 \geq \lambda_2$ and $\lambda_3 \geq \lambda_4$, $\sum_{v\in V} w(v)x(v)$ is maximized when we choose q and r to be the nodes of maximum weight in Q and R, respectively. With this choice of q and r, $\sum_{v\in V} w(v)x(v) = (\lambda_1-\lambda_2)w_{Qmax} + \lambda_2 \sum_{v\in Q} w(v) + (\lambda_3-\lambda_4)w_{Rmax} + \lambda_4 \sum_{v\in R} w(v)$, and the result follows. □

Theorem 4.1 leads to a family of bounds, since each case of values for the parameters $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ as given in the table leads to a different bound. Some of these bounds are new, while others have been obtained before by conventional methods.

The bounds derived from cases (5), (7), and (8) are new. From case (7), where $(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (u, u, u, \frac{k-u}{k-1})$, we obtain the bound

$$S(G, w) \geq u\left(\sum_{v\in Q} w(v) + w_{R\max}\right) + \frac{k-u}{k-1} \sum_{v\in R, v\neq v_{Rmax}} w(v) - k.$$

This bound strengthens the bound $S(G, w) \geq u\sum_{v\in C} w(v) - u$ (first mentioned in [6]), which holds for any clique C, where all edge constraints have value at least u.

From case (8), which uses the point $(2u-1, \nu, 1, 1)$, we obtain the new bound

$$S(G,w) \geq (2u-1)w_{Qmax} + \nu \sum_{v \in Q, v \neq v_{Qmax}} w(v) + \sum_{v \in R} w(v) - k.$$

In [17] a bound of $(2u-1)w_{Qmax} + \sum_{v \in R} w(v) - \kappa$ (where κ is a small constant) is given for nested cliques with the special property that $|Q| = 1$. The bound resulting from case (8) can be seen as a generalization of this bound for nested cliques where Q contains more than one node.

Case (5) uses the point $(k - (\mu - 1)\delta, \delta, \epsilon, \epsilon)$ and leads to the bound

$$S(G,w) \geq (k - \mu\delta)w_{Qmax} + \delta \sum_{v \in Q} w(v) + \epsilon \sum_{v \in R} w(v) - k.$$

The new bound from case (5) can be seen as an extension of the bound $S(G,w) \geq (k - \mu\delta)w_{max} + \delta \sum_{v \in C} w(v) - \kappa$ (κ is a small constant) that was given for cliques with cosite constraint k and uniform edge constraint u in [6].

Using the clique $Q \cup \{v_{Rmax}\}$ (with edge constraint at least u), our method also gives the bound

$$S(G,w) \geq (k - \mu\delta)w_{max} + \delta \left(\sum_{v \in Q} w(v) + w_{Rmax} \right) - k.$$

We simply use case (3) or (4), depending on whether $w_{max} = w_{Qmax}$ or $w_{max} = w_{Rmax}$, respectively.

The bound from case (6), namely,

$$S(G,w) \geq u \sum_{v \in Q} w(v) + \sum_{v \in R} w(v) - k,$$

was the first bound treating nested cliques specifically. It was derived in [6] using ad hoc methods.

The bound derived from cases (1) and (2) is the well-known bound

$$S(G,w) \geq k w_{max} - k.$$

In all these results, we have used the general rule, stated in Theorem 5.1, that $S(G,w) \geq \tau(G,w) - k$. A careful reading of the proof of Theorem 5.1 will show that in most cases the extra term k is too pessimistic. In principle, it is possible to find a more precise additive term by a more precise, and hence more complicated, analysis. Since our main interest here lies in showing a *method* by which lower bounds can be derived rather than in finding the best possible lower bounds, we content ourselves with the additive factor of k. However, this may cause our bounds to differ slightly from the older bounds.

The preceding theorems show how new lower bounds can be generated for any particular choice of parameters. In practice, it will often be useful to apply the tile cover method directly to the exact parameters of the particular network. For any specific nested clique, a classification of all extreme points of $TC(G)$ can be obtained by using vertex enumeration software, for example, the package `lrs`, developed by Avis [3]. In general, we can use the dual program to obtain families of vertices, and hence bounds, for certain choices of parameters.

This approach is demonstrated in the following example. The example is taken from [19], where it was used to demonstrate a lower bound derived from network flows. We will see that our tile cover approach gives a significant improvement.

Example 4.1. Consider the cellular network layout as shown Figure 1. The circled number in each cell represents the label of the cell; the node associated with the cell with label i is called v_i. The larger number in each cell gives the demand in the cell, i.e., the weight of the associated node. The particular hexagonal cell layout of this example is that of the "Philadelphia problem" [1], which has frequently been used as a benchmark for algorithms and lower bounds for the channel assignment problem (see, for example, [6, 7, 9, 13, 14, 20, 2]).

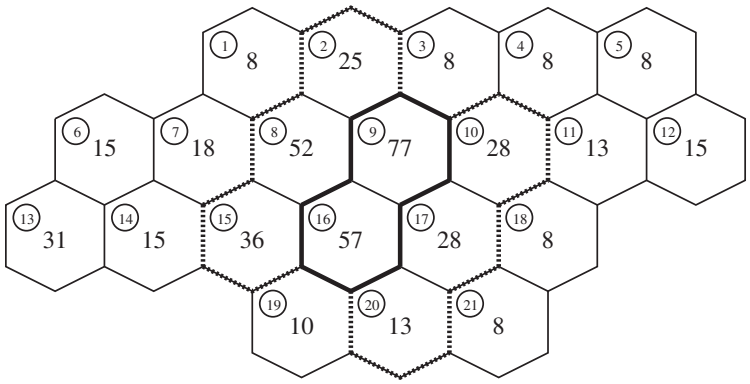


FIG. 1. *The layout of the example.*

The cosite constraint $s(v_i) = 5$ for each node v_i. The edge constraints are described in terms of the distance d_{ij} between the centers of cells v_i and v_j, where the unit is the distance between the centers of adjacent cells:

$$e(v_i, v_j) = \begin{cases} 0 & \text{if } d_{ij} > 3, \\ 1 & \text{if } \sqrt{3} < d_{ij} \leq 3, \\ 2 & \text{if } 0 < d_{ij} \leq \sqrt{3}. \end{cases}$$

This layout contains nested cliques of size 8, with 2 nodes in Q and 6 nodes in R, and nested cliques of size 7, with 1 node in Q and 6 nodes in R. The nested cliques have parameters $(5, 2, 1)$.

For a nested clique with bipartition (Q, R), where $|Q| = 2$ and $|R| = 6$, we derived a set of lower bounds using the software `lrs`. We looked for points of the form $(x_1, x_2, y_1, y_2, y_3, y_4, y_5, y_6)$, where x_1 and x_2 correspond to nodes of Q and $x_1 \geq x_2$, and $y_1, \ldots, y_6$ correspond to the nodes of R and $y_1 \geq y_2 \geq \cdots \geq y_6$. The inequality system that defines $TC(G)$ reduces to the following:

$$\begin{aligned}
&x_1 + x_2 \leq 5, \\
&y_1 + y_2 + y_3 + y_4 + y_5 \leq 5, \\
&x_1 + y_1 + y_2 \leq 5, \\
&x_1 + y_1 + y_2 + y_3 \leq 6, \\
&x_1 + y_1 + y_2 + y_3 + y_4 \leq 7, \\
&x_1 + x_2 + y_1 \leq 6, \\
&x_1 + x_2 + y_1 + y_2 \leq 7, \\
&x_1 + x_2 + y_1 + y_2 + y_3 \leq 8, \\
&x_1 + x_2 + y_1 + y_2 + y_3 + y_4 \leq 9, \\
&x_1 \geq x_2 \geq 0,\ y_1 \geq y_2 \geq \cdots \geq y_6 \geq 0.
\end{aligned}$$

Given this system, `lrs` returned a set of vertices, 14 of which could be used to generate lower bounds (the other vertices could be obtained from those 14 by dropping some coordinates to zero).

We applied these bounds to the nested clique formed by the cells as indicated in Figure 1. Here $Q = \{v_9, v_{16}\}$, and $R = \{v_2, v_8, v_{10}, v_{15}, v_{17}, v_{20}\}$. To obtain the best possible results, the nodes of larger weight in Q and R were matched with larger coordinates x_i or y_i, respectively. The best result was obtained by the point $(3, 2, 1, 1, 1, 1, 1, 1)$. The corresponding lower bound is

$$\begin{aligned} S(G, w) &\geq 3w(v_9) + 2w(v_{16}) + \sum_{v \in R} w(v) - 5 \\ &= 3 \cdot 77 + 2 \cdot 57 + (52 + 36 + 28 + 28 + 25 + 13) - 5 \\ &= 522. \end{aligned}$$

This improves by 13% the lower bound of 460 obtained in [19].

Example 4.2. Our second example also involves a variation of the Philadelphia problem. It should be noted that this example incorporates many properties of real-life problems: a regular planar layout of the base stations, derived from the ideal packing formed by the hexagonal grid, as well as edge constraints that diminish as the distance between base stations increases. This example again uses the layout of Figure 1. The cosite constraint for this example is 7, while the edge constraints are as follows:

$$e(v_i, v_j) = \begin{cases} 0 & \text{if } d_{ij} > 3, \\ 1 & \text{if } \sqrt{7} d_{ij} \leq 3, \\ 2 & \text{if } \sqrt{3} < d_{ij} \leq 2, \\ 3 & \text{if } 0 < d_{ij} \leq \sqrt{3}. \end{cases}$$

In this case, the network contains a nested clique (Q, R), where $Q = \{v_8.v_9, v_{16}\}$ and $R = \{v_2, v_{15}, v_{17}\}$, with parameters $(7, 3, 2)$ (other nested cliques exist in similar configurations). Assume that the demand in this nested clique is as follows:

Node	v_2	v_8	v_9	v_{15}	v_{16}	v_{17}
Demand	10	15	30	30	15	10

Consider the following dual solution to the tile cover problem: $x = (2, 2, 4, 3, 2, 3)$ (the ordering of the components in the vector refers to the order of the nodes as given in the table above), and a tile cover consisting of 10 copies of the tile $(1, 1, 2, 2, 1, 1) \in \mathcal{T}_{QR}^{big}$, and 5 copies each of tiles $(0, 1, 1, 1, 0, 0)$ and $(0, 0, 1, 1, 1, 0)$, both in $\mathcal{T}_Q$. It can be easily checked that these primal/dual solutions have the same value, namely, 310. This leads to a lower bound for the span of 303. This lower bound can be refined to 307 if the tile cover method is extended to include patches, as explained later in this paper. Moreover, the optimal tile cover can be converted into a matching channel assignment (in the last line, i takes values from 0 to 9):

v_2	v_8	v_9	v_{15}	v_{16}	v_{17}
	6,15,...,42	0,9,18,...,81	3,12,21,...,84	51,60,...,87	
$104 + 22i$	$93 + 22i$	$90, 99 + 22i$	$102, 109 + 22i$	$96 + 22i$	$106 + 22i$

Example 4.3. In [2], a small assignment problem of only 7 nodes is presented (instances M1 and M2). The problem was formed to test the limits of the method

proposed in the paper. Indeed, for this instance there is a gap of 3 between upper and lower bounds found by the method of Avenali, Mannino, and Sassano [2]. The cosite constraint is 5 for each node, while the edge constraints are as given in the following table.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	5	2	3	4	1	0	0
v_2		5	1	4	1	4	2
v_3			5	1	1	2	0
v_4				5	0	1	0
v_5					5	1	1
v_6						5	2
v_7							5

This example is highly irregular, but it does contain some small nested cliques. For example, there is a nested clique (Q, R) with parameters $(5, 4, 2)$, where $Q = \{v_6\}$ and $R = \{v_1, v_2\}$. One of the tiles for this nested clique is the tile from $\mathcal{T}_{QR}$ consisting of all vertices of (Q, R), with cost $2 \cdot 4 + 2 = 10$. In the examples of [2], the demand of all nodes is equal. Combining 10 such tiles gives a tile cover of cost 100, which leads to a lower bound of 95 for the case where the demand on all nodes equals 10. Following the more precise method outlined later in this paper, we can replace one of the tiles by a patch with cost 6, which gives a lower bound of 96 when the demand on all nodes equals 10, and 106 when the demand on all nodes equals 11. This reduces the gap between upper and lower bounds to 1. (Note that this particular case, where $|Q| = 1$, can also be solved with the bound from [17].)

5. From channel assignments to tile covers. In this section we give the proof of the following theorem.

THEOREM 5.1. *Let G be a nested clique with node partition (Q, R) and parameters (k, u, a). Then for any weight vector w for G,*

$$S(G, w) \geq \tau(G, w) - k.$$

This theorem will follow as a corollary from a more technical lemma. The lemma reduces any channel assignment to a tiling that uses only tiles from $\mathcal{T}$, except for one extra tile called a *patch*. (In subsequent proofs, we will specify a specific tile to act as the patch of any given tiling.) A patch is added to take care of the highest channels assigned, for which there is no link-up cost. Patches are defined as follows.

Given a nested clique G with node bipartition (Q, R) and constraints (k, u, a), the patch set $\mathcal{P}$ is defined as

$$\mathcal{P} = \mathcal{P}_Q \cup \mathcal{P}_R \cup \mathcal{P}_{QR} \cup \mathcal{P}_{QR}^{big}.$$

The patches in each category are defined below:

$$\begin{aligned}
\mathcal{P}_Q &= \{\chi^A : A \subseteq Q\}, \\
\mathcal{P}_R &= \{\chi^B : B \subseteq R\}, \\
\mathcal{P}_{QR} &= \{\chi^A + \chi^B : A \subseteq Q,\ B \subseteq R,\ A \neq \emptyset,\ B \neq \emptyset\}, \\
\mathcal{P}_{QR}^{big} &= \{\chi^{A \cup B} + \chi^{A_2 \cup B_2} : A_2 \subseteq A \subseteq Q,\ B_2 \subseteq B \subseteq R,\ A_2 \neq \emptyset,\ B_2 \neq \emptyset\}.
\end{aligned}$$

The cost of a patch p is denoted by $c'(p)$. The definition of the cost of a tile cover $y \in \mathbb{Z}^{\mathcal{T}\cup\mathcal{P}}$ is adjusted to account for patch cost:

$$c(y) = \sum_{t\in\mathcal{T}} c(t)y(t) + \sum_{p\in\mathcal{P}} c'(p)y(p).$$

Patch costs for each category are given in Table 2.

TABLE 2
Costs of patches.

Category	Number of nodes in Q	Number of nodes in R	Cost
$\mathcal{P}_Q$	n	0	$(n-1)u$
$\mathcal{P}_R$	0	m	$(m-1)a$
$\mathcal{P}_{QR}$	n	m	$nu + (m-1)a$
$\mathcal{P}_{QR}^{big}$	n, of which n_2 have weight 2	m, of which m_2 have weight 2	$(n+n_2)u + (m_2-1)a +$ $\max\{k, ma\}$

When we reduce a channel assignment to a tiling, a patch from $\mathcal{P}_R$ will be used only when the first channel is assigned to a node in R, and a patch from either $\mathcal{P}_Q$ or $\mathcal{P}_{QR}^{big}$ will be used only if the first channel is assigned in Q.

For the rest of this section we will adopt the following terminology. Suppose f is a channel assignment for a constrained graph G with node set V, where $f(V) = \{h_0, h_1, \ldots, h_f\}$, with $h_0 \le h_1 \le \cdots \le h_f$. We say that a tiling y of G *covers* channels h_i to h_j (where $j \ge i$) if y is a tile cover of the subgraph induced by the nodes of G that were assigned channels between h_i and h_j. More precisely, y covers channels $\{h_i, \ldots, h_j\}$ if for each node $v \in V$,

$$\sum_{t\in\mathcal{T}} y(t)t(v) \ge |f(v) \cap \{h_i, \ldots, h_j\}|.$$

Also, when y is a tiling and t is a patch or tile, we use $y + \{t\}$ to mean the tiling where one more copy of t is added, i.e., strictly speaking, the tiling $y + \chi^{\{t\}}$.

We start by stating a lemma that proves that any channel assignment can be reduced to a tile cover for the cliques where there is only one edge constraint and a cosite constraint.

LEMMA 5.2 (see [10]). *Let G be a clique with cosite constraint k and edge constraint u. Let Q be the node set of G, and let the tile set $\mathcal{T}_Q$ and patch set $\mathcal{P}_Q$ be as defined above. Let f be a channel assignment for G, where $f(V) = \{h_0, h_1, \ldots, h_f\}$, $h_0 < h_1 < \cdots < h_f$. Then there exists a tile cover $y \in \mathbb{Z}_+^{\mathcal{T}_Q\cup\mathcal{P}_Q}$ of (G, w) which contains exactly one patch p, covers all channels $\{h_0, \ldots, h_f\}$, and has cost at most $h_f - h_0$. Moreover, the support of p consists of the nodes that are assigned channels $h_{f-n}, \ldots, h_f$, where $n = |V(p)|$ and*

$$c(y - \{p\}) \le h_{f-n} - h_0.$$

The proof of Lemma 5.2 provides the following method of constructing the tile cover y, with patch p. Begin by finding a tile containing the set of nodes that are assigned channels in the range $[h_0, h_0 + k)$. Let t_0 denote that tile. For $j \ge 1$, we recursively find a tile t_j containing the nodes assigned channels in the range $[h_{e_j}, h_{e_j} + k)$, where h_{e_j} is the first channel not covered by the tiling $y_{j-1} = \chi^{\{t_0, t_1, \ldots, t_{j-1}\}}$.

Tile t_j is chosen so that the cost of $y_j = y_{j-1} + \{t_j\}$ is at most $h_{e_{j+1}} - h_0$, where $h_{e_{j+1}}$ is the first channel not covered by y_j. This process continues until the only channels not covered by the current tiling y_ℓ form a patch p. The cost of this patch is $c'(p) = h_f - h_{e_{\ell+1}}$, where $h_{\ell+1}$ is the first channel not covered by y_ℓ. The required tile cover y is formed by adding p to y_ℓ.

We are now ready to state and prove the technical lemma from which Theorem 5.1 will follow. The proof of this lemma uses a straightforward induction on the number of times the channel assignment "crosses over" from Q to R or vice versa. By invocation of Lemma 5.2, tilings are obtained for the channel assignment up to the first crossover and between the first and second crossovers, respectively. Then induction is used to obtain a tiling of the channel assignment that includes all channels after the second crossover. These tilings are then combined to obtain one new tiling which satisfies the induction hypothesis. The difficulties arise mainly from the fact that three different patches must be combined. As a result, there are a number of cases to be considered. Once the appropriate combinations of tiles and patches are described, verifying the cost of the tiling merely involves finding the appropriate substitutions. This, together with the fact that numerous cases are analogous, compels us to omit the details of the proof in many cases. For a complete treatment of the proof, we refer the reader to [11].

LEMMA 5.3. *Let G be a nested clique with node partition (Q, R) and integer constraints (k, u, a), and let $\mathcal{T}$ and $\mathcal{P}$ be the tile and patch set for G. Let f be a channel assignment for G, where $f(V) = \{h_0, h_1, \ldots, h_f\}$, $h_0 < h_1 < \cdots < h_f$. Then there exists a tile cover $y \in \mathbb{Z}_+^{\mathcal{T}\cup\mathcal{P}}$ of (G, w) which contains one patch p, covers all channels $\{h_0, \ldots, h_f\}$, and has cost at most $h_f - h_0$. Furthermore, if h_0 is assigned to a node in Q, then $p \notin \mathcal{P}_R$, and if h_0 is assigned to a node in R, then $p \notin \mathcal{P}_Q \cup \mathcal{P}_{QR}^{big}$.*

Proof. Let G be a nested clique and f be a channel assignment, as defined in the statement of the lemma. A crossover is defined to be a pair of channels (h_i, h_{i+1}), where the nodes that receive channels h_i and h_{i+1} are in different parts of the bipartition (Q, R). We now proceed with induction on the number of crossovers.

If f has no crossovers, then the statement follows directly from Lemma 5.2.

Suppose f has exactly one crossover and h_0 is assigned to a node in Q. Let h_ℓ be the first channel in R greater than h_0. By Lemma 5.2, we can cover the channels in $\{h_0, \ldots, h_{\ell-1}\}$ with a tiling y_Q, containing one patch $p_Q \in \mathcal{P}_Q$, with cost at most $h_{\ell-1} - h_0$. Likewise, the channels in $\{h_\ell, \ldots, h_f\}$ can be covered with a tiling y_R of cost at most $h_f - h_\ell$, containing one patch $p_R \in \mathcal{P}_R$. Combining the two patches into one, we form a new patch $p = p_Q + p_R \in \mathcal{P}_{QR}$ with cost $nu + (m-1)a$, where $n = |V(p_Q)|$ and $m = |V(p_R)|$. So $c'(p) = c'(p_Q) + c'(p_R) + u$. Moreover, $h_\ell - h_{\ell-1} \geq u$ since $h_{\ell-1}$ is assigned to a node in Q, and h_ℓ to a node in R.

Our final tiling is $y = y_Q - \{p_Q\} + y_R - \{p_R\} + \{p\}$ with cost

$$\begin{aligned} c(y) &= c(y_Q) - c'(p_Q) + c(y_R) - c'(p_Q) + c'(p) \\ &\leq (h_{\ell-1} - h_0) + (h_f - h_\ell) + u \\ &\leq h_f - h_0. \end{aligned}$$

When h_0 is assigned to a node in R, the proof is analogous.

For the induction step, assume that f is a channel assignment with g crossovers, where $g \geq 2$, and assume that the lemma holds for any channel assignment with less than g crossovers.

Case 1. Channel h_0 is assigned to a node in Q.

Let h_ℓ be the first channel assigned to a node in R, and let h_j be the first channel greater than h_ℓ assigned to a node in Q. So $(h_{\ell-1}, h_\ell)$ and (h_{j-1}, h_j) are the first two crossovers of f. Note that $h_\ell \geq h_{\ell-1} + u$ and $h_j \geq h_{j-1} + u$.

By Lemma 5.2, we can find a tiling y_Q (with one patch, $p_Q \in \mathcal{P}_Q$) which covers channels $\{h_0, \ldots, h_{\ell-1}\}$ in Q and has cost at most $h_{\ell-1} - h_0$, and a tiling y_R (with one patch, $p_R \in \mathcal{P}_R$) which covers channels $\{h_\ell, \ldots h_{j-1}\}$ and has cost at most $h_{j-1} - h_\ell$.

Define n and m to be the number of nodes in $V(p_Q)$ and $V(p_R)$, respectively. By Lemma 5.2, $V(p_Q)$ consists of the nodes that receive channels $\{h_{\ell-n}, \ldots, h_{\ell-1}\}$, and $V(p_R)$ consists of the nodes that receive channels $\{h_{j-m}, \ldots, h_{j-1}\}$. Note that $c'(p_Q) = (n-1)u$. $c'(p_R) = (m-1)a$.

Case 1A. Tiling y_R contains only the patch p_R (no other tiles).

In this case, patch p_R covers all channels from h_ℓ to h_{j-1}.

(i) Suppose $h_j - h_{\ell-n} \geq k$. Let $y = y_Q - \{p_Q\} + \{t\} + y_{end}$, where $t = p_Q + p_R$, and y_{end} is a tiling covering channels $\{h_j, \ldots, h_f\}$ with cost at most $h_f - h_j$ and a patch that is not in $\mathcal{P}_R$. (By induction, such a tiling y_{end} exists.) The new tiling y has the patch of y_{end} as its patch. It is clear that y covers all channels from h_0 to h_f and has a patch of the required type. It now remains to be proved that $c(y) \leq h_f - h_0$.

Since the channels from $h_{\ell-n}$ to h_j cover $n+1$ nodes in Q and m nodes in R, with two crossovers, we have $h_j \geq h_{\ell-n} + (n-1)u + (m-1)a + 2u$. Also, by assumption, $h_j - h_{\ell-n} \geq k$. Therefore, $h_j - h_{\ell-n} \geq \max\{nu + ma + u - a, k\} = c(t)$, and

$$\begin{aligned} c(y) &= c(y_Q - \{p_Q\}) + c(t) + c(y_{end}) \\ &\leq (h_{\ell-n} - h_0) + (h_j - h_{\ell-n}) + (h_f - h_j) \\ &= h_f - h_0. \end{aligned}$$

(ii) Suppose $h_j - h_{\ell-n} < k$. If there is a channel in the range $[h_{\ell-n}+k, h_{\ell-n}+k+u)$ which has been assigned to a node in Q, let h_i denote that channel. (The choice of h_i is unique, since the given range has length less than u.) Otherwise, let h_i be the first channel greater than or equal to $h_{\ell-n} + k + u$. If no such h_i can be chosen, then let $i = f + 1$, so $h_{i-1} = h_f$ and h_i is undefined. Note that it is always the case that $h_{i-1} < h_{l-n} + k + u$.

We form the final tile cover y as follows:

Step 1. Let A be the set of all nodes that receive channels from $\{h_{\ell-n}, \ldots, h_{i-1}\}$. Also, let $n_1 = |Q \cap A|$, and let $m_1 = |R \cap A|$.

Step 2. Find a tiling y_{end} which covers channels $\{h_i, \ldots, h_f\}$ and has cost at most $h_f - h_i$. Let p_{end} be the patch of y_{end}. (In the case that $h_{i-1} = h_f$, both y_{end} and p are empty.)

Step 3. If $p_{end} \in \mathcal{P}_Q \cup \mathcal{P}_{QR} \cup \mathcal{P}_{QR}^{big}$, form tile $t = \chi^A \in \mathcal{T}_{QR}$ and let $y = y_Q - \{p_Q\} + \{t\} + y_{end}$.

Step 4. If $p_{end} \in \mathcal{P}_R$, then

4(a) pick a node $v \in A \cap Q$,

4(b) form patch $p = p_{end} + \chi^{\{v\}} \in \mathcal{P}_{QR}$,

4(c) form tile $t = \chi^A - \chi^{\{v\}}$,

4(d) form tile cover $y = y_Q - \{p_Q\} + \{t\} + y_{end} - \{p_{end}\} + \{p\}$.

Step 5. If p_{end} is empty, then

5(a) form patch $p = \chi^A \in \mathcal{P}_{QR}$,

5(b) form tile cover $y = y_Q - \{p_Q\} + \{p\}$.

As before, it is easy to see that y covers all channels from h_0 to h_f. Steps 3, 4, and 5 guarantee that the patch of y is not in $\mathcal{P}_R$, as required. We prove that in all cases, $c(y) \leq h_f - h_0$.

CLAIM 5.4. *No two channels in* $S = \{h_{l-n}, \ldots, h_{i-1}\}$ *are assigned to the same node.*

Proof of claim. Assume two channels h_α and h_β, $l - n \le \alpha < \beta \le i - 1$, are assigned to the same node. Now, by combining our cosite constraint with a previous remark, we have $k \le h_\beta - h_\alpha \le h_{i-1} - h_{l-n} < k + u$.

Since h_{i-1} is in the interval $[h_{l-n} + k, h_{l-n} + k + u)$, it follows from the choice of h_i that no channel in $[h_{l-n} + k, h_{l-n} + k + u)$ is assigned to a node in Q. Hence, all channels from S assigned to Q are in the interval $[h_{l-n}, h_{l-n} + k)$. Since the range of this interval is less than k, it cannot be the case that h_α and h_β are assigned to a node in Q.

Suppose h_α and h_β are both assigned to nodes in R. Since h_{l-n} is assigned to a node in Q, $h_{l-n} + u \le h_\alpha$ due to our adjacency constraints. Since $h_\beta < h_{l-n} + k + u$, $h_\beta - h_\alpha < k$, which is a contradiction. Hence, no node receives two channels from C. □

CLAIM 5.5. *When a channel* h_i *can be chosen,* $h_i - h_{\ell-n} \ge \max\{n_1 u + m_1 a + u - a, k\}$.

Proof of claim. Suppose h_i is assigned to a node in Q. Since $\{h_{\ell-n}, \ldots, h_i\}$ is covered by $n_1 + 1$ nodes in Q and m_1 nodes in R and contains at least two crossovers, we have $h_i - h_{\ell-n} \ge (n_1 - 1)u + (m_1 - 1)a + 2u = n_1 u + m_1 a + u - a$.

If h_i is assigned to a node in R, then $\{h_{\ell-n}, \ldots, h_i\}$ covers n_1 nodes of Q and $m_1 + 1$ nodes in R and contains at least three crossovers. Hence, $h_i - h_{\ell-n} \ge (n_1 - 2)u + (m_1 - 1)a + 3u = n_1 u + m_1 a + u - a$.

Whenever h_i is chosen, it is done in such a way that $h_i \ge h_{l-n} + k$. Hence, $h_i - h_{\ell-n} \ge \max\{k + u, n_1 u + m_1 a + u - a\}$. □

In Step 3, we have $t = \chi^A \in \mathcal{T}_{QR}$ and $c(t) = \max\{k, n_1 u + m_1 a + u - a\} \le h_i - h_{\ell-n}$. Therefore,

$$c(y) = c(y_Q - \{p_Q\}) + c(t) + c(y_{end}) \le h_f - h_0.$$

In Step 4, a new patch $p = p_{end} + \chi^{\{v\}} \in \mathcal{P}_{QR}$ is formed, since p_{end} is not of the required type. The cost of this new patch is $c'(p) = c'(p_{end}) + u$. In finding the cost of t there are two possibilities to consider. If $n_1 > 1$, then $t = \chi^A - \chi^{\{v\}} \in \mathcal{T}_{QR}$ and $c(t) = \max\{k, (n_1 - 1)u + m_1 a + u - a\}$. If $n_1 = 1$, then $t \in \mathcal{T}_R$ and $c(t) = \max\{k, m_1 a\} \le \max\{k, (n_1 - 1)u + m_1 a + u - a\}$. Now, since $h_i - h_{\ell-n} \ge \max\{k + u, n_1 u + m_1 a + u - a\}$, it follows that $c(t) \le h_i - h_{\ell-n} - u$. Since $c'(p) - c'(p_{end}) \le u$, it follows that

$$c(y) = c(y_Q - \{p_Q\}) + c(y_{end}) + (c'(p) - c'(p_{end})) + c(t) \le h_f - h_0.$$

In Step 5, we have $h_{i-1} = h_f$. Since $p \in \mathcal{P}_{QR}$, we have $c'(p) = n_1 u + (m_1 - 1)a$. Furthermore, since $\{h_{\ell-n}, \ldots, h_f\}$ contains n_1 nodes from Q, m_1 nodes from R, and at least two crossovers, $h_f - h_{\ell-n} \ge n_1 u + (m_1 - 1)a = c'(p)$. Therefore,

$$c(y) = c(y_Q - \{p_Q\}) + c'(p) \le h_f - h_0.$$

Case 1B. y_R contains a tile other than p_R.

By Lemma 5.2, patch p_R covers channels $\{h_{j-m}, \ldots, h_{j-1}\}$, and these channels are all assigned to nodes in R, so $j - m \ge \ell$. Since $(h_{\ell-1}, h_\ell)$ is a crossover, the assignment of channels $\{h_{j-m}, \ldots, h_f\}$ has $g - 1$ crossovers. Then, by induction, there exists a tiling y_{end} that covers all channels in $\{h_{j-m}, \ldots, h_f\}$, contains a patch $p_{end} \in \mathcal{P}_{QR} \cup \mathcal{P}_R$, and has cost at most $h_f - h_{j-m}$.

Let $V_Q = V(p_{end}) \cap Q$, $V_R = V(p_{end}) \cap R$. Also let $n^p = |V_Q|$ and $m^p = |V_R|$. Note that $c'(p_{end}) = n^p u + (m^p - 1)a$ if $p_{end} \in \mathcal{P}_{QR}$, and $c'(p_{end}) = (m^p - 1)a$ if $p_{end} \in \mathcal{P}_R$.

Choose t_R to be any tile from y_R other than p_R. Let $V_t = V(t_R)$ and $m^t = |V_t|$. Note that $t_R \in \mathcal{T}_R$ and $c(t_R) = \max\{k, m^t a\}$. Let $V_{p_Q} = V(p_Q)$. Recall that $|V_{p_Q}| = n$ and $c'(p_Q) = (n-1)u$. In Table 3, we show how to combine p_Q, p_{end}, and t_R into a new tile t and a new patch p.

TABLE 3
Combining patches.

Case	Condition	Tile t	Patch p
(1)	$p_{end} \in \mathcal{P}_{QR}$		
(1.1)	(1) and $V_Q \cap V_{p_Q} = \emptyset$	t_R	$p_{end} + p_Q$
(1.2)	(1) and $V_Q \cap V_{p_Q} \neq \emptyset$		
(1.2.1)	(1.2) and $V_R \cap V_t = \emptyset$	$p_{end} + t_R$	p_Q
(1.2.2)	(1.2) and $V_R \cap V_t \neq \emptyset$	there is no t	$t_R + p_{end} + p_Q$
(2)	$p_{end} \in \mathcal{P}_R$	t_R	$p_{end} + p_Q$

Case	Cost $c(t)$	$t \in$	Cost $c'(p)$	$p \in$
(1.1)	$c(t_R)$	$\mathcal{T}_R$	$(n+n^p)u + (m^p-1)a$	$\mathcal{P}_{QR}$
(1.2.1)	$\max\{k, n^p u + m^p a + m^t a + u - a\}$	$\mathcal{T}_{QR}$	$c'(p_Q)$	$\mathcal{P}_Q$
(1.2.2)	–	–	$(n+n^p)u + \lvert V_R \cap V_t\rvert a - a + \max\{k, \lvert V_R \cup V_t\rvert a\}$	$\mathcal{P}_{QR}^{big}$
(2)	$c(t_R)$	$\mathcal{T}_R$	$nu + (m^p-1)a$	$\mathcal{P}_{QR}$

In Cases (1.1), (1.2.1), and (2), we form the new tiling

$$y = y_Q - \{p_Q\} + y_R - \{p_R\} - \{t_R\} + y_{end} - \{p_{end}\} + \{t\} + \{p\}.$$

In Case (1.2.2), there is no t, so we take the tiling

$$y = y_Q - \{p_Q\} + y_R - \{p_R\} - \{t_R\} + y_{end} - \{p_{end}\} + \{p\}.$$

In all cases, it is straightforward to verify that y covers all channels and has a patch of the required type, and that $c(y) \leq h_f - h_0$.

Case 2. Channel h_0 is assigned to a node in R.

Since this case is very similar to Case 1, we omit the details of the proof. And, unless otherwise stated, the same terminology will apply.

Let h_ℓ be the first channel assigned to a node in Q, and h_j the first channel greater than h_ℓ assigned to a node in R. As in Case 1, by Lemma 5.2, we can find tilings y_R and y_Q of the required cost, which together cover all channels in $\{h_0, \ldots, h_{j-1}\}$. Furthermore, by induction, we can find the appropriate tiling y_{end} (with patch p_{end}) to cover the remaining channels.

We now provide the method for finding a new tiling y that covers all channels in the assignments and has cost at most $h_f - h_0$.

Case 2A. Tiling y_Q contains only the patch p_Q.

(i) If $h_j - h_{l-m} \geq k$, then let $y = y_R - \{p_R\} + \{t\} + y_{end}$, where $t = p_Q + p_R$.

(ii) Suppose $h_j - h_{\ell-m} < k$. Channel h_i is chosen in a manner similar to that in Case 1A. Simply replace "Q" and "n" with "R" and "m," respectively, in the description. Similarly, A denotes the set of nodes receiving channels from $\{h_{\ell-m}, \ldots, h_{i-1}\}$.

If $p_{end} \in \mathcal{P}_R \cup \mathcal{P}_{QR}$, then let $y = y_R - \{p_R\} + \{t\} + y_{end}$, where $t = \chi^A \in \mathcal{T}_{QR}$.

TABLE 4
Combining patches.

Case	Condition	Tile t	Patch p
(1)	$p_{end} \in \mathcal{P}_{QR}^{big}$		
(1.1)	(1) and $V_t \cap V_Q = \emptyset$	$p_{end} + t_Q$	p_R
(1.2)	(1) and $V_t \cap V_Q \neq \emptyset$	$p_{end} + \chi^{V_t - V_Q}$	$\chi^{V_t \cap V_Q} + p_R$
(2)	$p_{end} \in \mathcal{P}_{QR}$	$t_Q + \chi^{V_R}$	$\chi^{V_Q} + p_R$
(3)	$p_{end} \in \mathcal{P}_Q$	t_Q	$p_{end} + p_R$

Case	Cost $c(t)$	$t \in$	Cost $c'(p)$	$p \in$
(1.1)	$\max\{k, (n^p + n^t)u\} + \max\{k, m^p a\}$ $+n_2^p u + m_2^p a + u - a$	$\mathcal{T}_{QR}^{big}$	$c'(p_R)$	$\mathcal{P}_R$
(1.2)	$\max\{k, \lvert V_t \cup V_Q \rvert u\} + \max\{k, m^p a\}$ $+n_2^p u + m_2^p a + u - a$	$\mathcal{T}_{QR}^{big}$	$\lvert V_t \cap V_Q \rvert u + c'(p_R)$	$\mathcal{P}_{QR}$
(2)	$\max\{k, n^t u + m^p a + u - a\}$	$\mathcal{T}_{QR}$	$n^p u + c'(p_R)$	$\mathcal{P}_{QR}$
(3)	$c(t_Q)$	$\mathcal{T}_Q$	$n^p u + c'(p_R)$	$\mathcal{P}_{QR}$

If $p_{end} \in \mathcal{P}_Q$, $y = y_R - \{p_R\} + \{p\} + y_{end} - \{p_{end}\} + \{t\}$, where $p = p_{end} + \chi^{\{v\}}$, $v \in A \cap R$, and $t = \chi^{A - \{v\}}$.

If $p_{end} \in \mathcal{P}_{QR}^{big}$, then let $t = \chi^{V_Q} + \chi^{V_R}$, $t' = \chi^A$, and $p = \chi^{B_Q} + \chi^{B_R}$. In this case, let $y = y_R - \{p_R\} + y_{end} - \{p_{end}\} + \{t\} + \{t'\} + \{p\}$.

For Cases 2A(i) and 2A(ii), it is straightforward to show that $c(y) \leq h_f - h_0$.

Case 2B. y_Q contains tiles other than p_Q.

Choose t_Q to be any tile from $y_Q - \{p_Q\}$. In Table 4, we show how we will combine p_R, p_{end}, and t_Q into a new tile t and a new patch p. Note that $n^t = |V(t_Q)|$ in this case. In all instances, we form the new tiling

$$y = y_R - \{p_R\} + y_Q - \{p_Q\} - \{t_Q\} + y_{end} - \{p_{end}\} + \{t\} + \{p\}.$$

It is straightforward to verify that y covers all channels and has cost at most $h_f - h_0$. This completes the proof. □

6. Conclusions. We have given a new general method of obtaining lower bounds for the channel assignment problems. When applied to the specific example of nested cliques, this leads to a complete family of lower bounds. This family includes almost all known clique-bounds. The bounds are easy to compute, and give improved results when applied to an example from the literature.

Nested cliques occur naturally in many CAPs. Radio signals decay with distance, so edge constraints between transmitters that are close together are usually stricter than constraints between transmitters that are farther apart. For the same reason, cosite constraints are usually the most restrictive. In this situation, a tight cluster of transmitters in a central area such as a city center, surrounded by a wider ring of more sparsely placed transmitters, will typically form a nested clique. The examples in this paper illustrate this situation.

Further work should address the computational issues related to lower bounds. A computational study comparing the performance of tile cover bounds to the lower bounds from previous work discussed in the introduction on a number of realistic CAP instances would be a valuable addition to this theoretical analysis.

Another interesting question is whether the tile cover approach can be used to obtain good channel assignments. Knowledge about which lower bound is most restrictive for any particular instance could be used to determine which tiles were most suited to build the best assignment.

Acknowledgment. The authors wish to thank the anonymous referees for their comments, which greatly improved the presentation of the paper.

REFERENCES

[1] L. Anderson, *A simulation study of some dynamic channel assignment algorithms in a high capacity mobile telecommunications system*, IEEE Trans. Commun., 21 (1973), pp. 1294–1301.

[2] A. Avenali, C. Mannino, and A. Sassano, *Minimizing the span of d-walks to compute optimum frequency assignments*, Math. Program., 91 (2002), pp. 357–374.

[3] D. Avis, *lrs: A revised implementation of the reverse search vertex enumeration algorithm*, in Polytopes—Combinatorics and Computation, Birkhäuser-Verlag, Basel, Switzerland, 2000, pp. 177–198; also available from http://cgm.cs.mcgill.ca/~avis/C/lrs.html.

[4] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, North–Holland, New York, 1976.

[5] R. Carraghan and P. Pardalos, *An exact algorithm for the maximum clique problem*, Oper. Res. Lett., 9 (1990), pp. 375–382.

[6] A. Gamst, *Some lower bounds for a class of frequency assignment problems*, IEEE Trans. Veh. Technol., 35 (1986), pp. 8–14.

[7] S. Hurley, S. Thiel, and D. Smith, *A comparison of local search algorithms for radio link frequency*, in Proceedings of the ACM Symposium on Applied Computing, Philadelphia, 1996, pp. 251–257.

[8] J. Janssen and K. Kilakos, *Polyhedral Analysis of Channel Assignment Problems:* (I) *Tours*, Tech. report CDAM-96-17, London School of Economics, LSE, London, 1996.

[9] J. Janssen and K. Kilakos, *An optimal solution to the "Philadelphia" channel assignment problem*, IEEE Trans. Veh. Technol., 48 (1999), pp. 1012–1014.

[10] J. Janssen and K. Kilakos, *Tile covers, closed tours and the radio spectrum*, in Telecommunications Network Planning, B. Sansó and P. Soriano, eds., Kluwer Academic Publishers, Boston, Dordrecht, London, 1999, pp. 257–270.

[11] J. Janssen, T. Wentzell, and S. Fitzpatrick, *Lower Bounds from Tile Covers for the Channel Assignment Problem*, Tech. report CS-2003-05, Computer Science, Dalhousie University, Halifax, NS, Canada, 2003; also available online from http://www.cs.dal.ca/research/techreports/2003/CS-2003-05.html.

[12] B. Jaumard, O. Marcotte, and C. Meyer, *Mathematical models and exact methods for channel assignment in cellular networks*, in Telecommunications Network Planning, B. Sansó and P. Soriano, eds., Kluwer Academic Publishers, Boston, 1999, pp. 239–255.

[13] R. Leese, *Tiling methods for channel assignment in radio communication networks*, Z. Angew. Math. Mech., 76 (1996), pp. 303–306.

[14] K. Sivarajan, R. McEliece, and J. Ketchum, *Channel assignment in cellular radio*, in Proceedings of the 39th IEEE Conference on Vehicular Technology, 1989, pp. 846–850.

[15] D. Smith and S. Hurley, *Bounds for the frequency assignment problem*, Discrete Math., 167/168 (1997), pp. 571–582.

[16] D. Smith, S. Hurley, and S. Allen, *A new lower bound for the channel assignment problem*, IEEE Trans. Veh. Technol., 49 (2000), pp. 1265–1272.

[17] C. Sung and W. Wong, *Sequential packing algorithm for channel assignment under cochannel and adjacent channel interference constraint*, IEEE Trans. Veh. Technol., 46 (1997), pp. 676–686.

[18] C.-W. Sung and W.-S. Wong, *A graph-theoretical approach to the channel assignment problem*, in Proceedings of the 45th IEEE Conference on Vehicular Technology, vol. 2, 1995, pp. 604–608.

[19] D.-W. Tcha, Y.-J. Chung, and T.-J. Choi, *A new lower bound for the frequency assignment problem*, IEEE ACM Trans. Networking, 5 (1997), pp. 34–39.

[20] W. Wang and C. K. Rushford, *An adaptive local-search algorithm for the channel-assignment problem (CAP)*, IEEE Trans. Veh. Technol., 45 (1996), pp. 459–466.

SIAM J. DISCRETE MATH.
Vol. 18, No. 4, pp. 697–712

LEARNING A HIDDEN SUBGRAPH*

NOGA ALON† AND VERA ASODI‡

Abstract. We consider the problem of learning a labeled graph from a given family of graphs on n vertices in a model where the only allowed operation is to query whether a set of vertices induces an edge. Questions of this type are motivated by problems in molecular biology. In the deterministic nonadaptive setting, we prove nearly matching upper and lower bounds for the minimum possible number of queries required when the family is the family of all stars of a given size or all cliques of a given size. We further describe some bounds that apply to general graphs.

Key words. hidden subgraphs, stars, complete graphs

AMS subject classifications. Primary, 68W99; Secondary, 05C90, 05C35

DOI. 10.1137/S0895480103431071

1. Introduction.

Let $\mathcal{H}$ be a family of labeled graphs on the set $V = \{1, 2, \ldots, n\}$, and suppose $\mathcal{H}$ is closed under isomorphism. Given a hidden copy of some $H \in \mathcal{H}$, we have to identify it by asking queries of the following form. For $F \subseteq V$, the query Q_F is, Does F contain at least one edge of H? Our objective is to identify H by asking as few queries as possible. We say that a family $\mathcal{F}$ solves the $\mathcal{H}$-problem if, for any two distinct members H_1 and H_2 of $\mathcal{H}$, there is at least one $F \in \mathcal{F}$ that contains an edge of one of the graphs H_i and does not contain any edge of the other. Obviously, any such family enables us to learn an unknown member of $\mathcal{H}$ deterministically and nonadaptively, by asking the questions Q_F for each $F \in \mathcal{F}$. Note that for any family $\mathcal{H}$, the set of all pairs of vertices solves the $\mathcal{H}$-problem. Note also that the information theoretic lower bound implies that we need at least $\log |\mathcal{H}|$ queries, where here and throughout the paper, all logarithms are in base 2, unless otherwise specified, and we omit all floor and ceiling signs when these are not crucial.

There are some families of graphs for which the above problem has been studied, motivated by applications in molecular biology. These include matchings [1] and Hamiltonian cycles [5, 6]. The biological problem is to find, given a set of molecules, pairs that react with each other. Here the vertices correspond to the molecules, the edges correspond to the reactions, and the queries correspond to experiments of putting a set of molecules together in a test tube and determining whether a reaction occurs. The problem of finding a hidden matching is the one encountered by molecular biologists when they apply multiplex PCR in order to close the gaps left in a DNA strand after shotgun sequencing. See [1] and its references for more details.

The previous works in this field study the minimum number of queries needed to identify a hidden graph, from various families of graphs. Some of these works consider

*Received by the editors July 10, 2003; accepted for publication (in revised form) July 13, 2004; published electronically April 22, 2005. A preliminary version of this paper appeared in *Proceedings of the 31st International Colloquium on Automata, Languages and Programming*, Turku, Finland, 2004, pp. 110–121.

http://www.siam.org/journals/sidma/18-4/43107.html

†Department of Mathematics, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel (nogaa@post.tau.ac.il). This author's research was supported in part by a USA–Israeli BSF grant, by the Israel Science Foundation, and by the Hermann Minkowski Minerva Center for Geometry at Tel Aviv University.

‡Department of Computer Science, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel (veraa@post.tau.ac.il).

query models other than the one described above. The authors of [1] study the hidden subgraph problem for the family of matchings. In that paper it is shown that under the deterministic and nonadaptive model, the minimum number of queries that one has to ask in order to identify a hidden matching is $\Theta(n^2)$; that is, one can do better than the trivial algorithm of asking all pairs only by a constant factor. It is also proved that $\Omega(n^2)$ queries are needed in order to find a hidden copy of any bounded-degree graph with a linear size matching. The authors further present randomized nonadaptive algorithms that use $\Theta(n \log n)$ random queries, and deterministic k-round algorithms that ask $O(n^{1+1/(2(k-1))}\text{polylog}\, n)$ queries. Grebinski and Kucherov [5, 6] study the family of Hamiltonian cycles. A few query models are discussed in those papers. Besides the model presented above, they consider the additive model, in which the answer to a query is not just "yes" or "no" but the number of edges in the subset. Both models are considered also when the size of the queries is bounded. They present matching lower and upper bounds under each of these models, where some of the upper bounds are achieved by 2-round algorithms, and the other algorithms are fully adaptive. In [7], Grebinski and Kucherov study the problem for low degree graphs and prove matching lower and upper bounds under the additive nonadaptive model.

In the present paper we consider only the deterministic nonadaptive model, where the answers are only "yes" or "no." The main families considered are families of stars and families of cliques. We study both families of stars or cliques of a given size and the families of all cliques or all stars. It is shown that the trivial upper bound of $\binom{n}{2}$ is tight up to a $1 + o(1)$-multiplicative term for the families of stars of k edges, for all $n^{\frac{2}{3}} \log^{\frac{2}{3}} n \ll k \leq n - 2$. For smaller stars, we show that fewer queries suffice and we give upper and lower bounds on the minimum number of queries needed. These bounds are tight up to some $\text{polylog}\, n$ factor for all sizes of stars, and they are of order k^3, up to the $\text{polylog}\, n$ factors. We show that the problem is easier when the hidden subgraph is a clique. In fact, even for the family of all cliques, the problem can be solved using $O(n \log^2 n)$ queries. We study, as in the case of stars, the problem of a hidden clique of size k, for all values of k. In all cases, we prove upper and lower bounds that are tight up to some $\text{polylog}\, n$ factor and are of order k^2, up to the $\text{polylog}\, n$ factors. We also consider the case where the family of graphs consists of all the graphs isomorphic to a given general graph G and give a lower bound that depends on the maximum size of an independent set in G. From this general bound, we obtain a lower bound of $\Omega(\frac{n^2}{\log^2 n})$ for the random graph $G(n, \frac{1}{2})$.

In section 2 we study the hidden subgraph problem where the hidden graph is a star, in section 3 we consider the case where the hidden graph is a clique, and in section 4 we prove a result for general graphs. Section 5 contains some concluding remarks and open problems.

2. Stars. In this section we consider the case where the graphs in $\mathcal{H}$ are stars. Denote by $\mathcal{S}_k$ the family of all graphs on $V = \{1, 2, \ldots, n\}$ that consist of a copy of $K_{1,k}$ and $n - k - 1$ isolated vertices. Let $\mathcal{S} = \cup_{k=1}^{n-1} \mathcal{S}_k$. We begin with the following simple claim.

PROPOSITION 2.1. *The minimum size of a family $\mathcal{F}$ that solves the $\mathcal{S}$-problem is exactly $\binom{n}{2}$.*

Proof. We show that any family $\mathcal{F}$ that solves the $\mathcal{S}$-problem must contain all pairs of vertices. Let u and v be two distinct vertices in V. Let S_1 be the star whose center is u and whose leaves are all other vertices of V, and let S_2 be the star whose

TABLE 1
Bounds on the size of a family that solves the $\mathcal{S}_k$-problem.

k	Lower bound	Upper bound
$k \le \sqrt{n}$	$\Omega(\frac{k^3 \log n}{\log k})$	$O(k^3 \log n)$
$\sqrt{n} < k < \frac{n^{2/3}}{(2\log n)^{1/3}}$	$\Omega(\frac{k^3}{\log^2 n})$	$O(k^3 \log n)$
$\frac{n^{2/3}}{(2\log n)^{1/3}} \le k \le O(n^{2/3}\log^{2/3} n)$	$\Omega(\frac{k^3}{\log^2 n})$	$\lceil \frac{n(n-2)}{2} \rceil$
$k = \omega(n^{2/3}\log^{2/3} n)$, $k \le n-3$	$(1-o(1))\binom{n}{2}$	$\lceil \frac{n(n-2)}{2} \rceil$
$k = n-2$	$\lceil \frac{n(n-2)}{2} \rceil$	$\lceil \frac{n(n-2)}{2} \rceil$
$k = n-1$	$\lceil \log n \rceil$	$\lceil \log n \rceil$

center is u and whose leaves are all other vertices of V except for v. Clearly, the answer to a query Q_F where F does not contain u is "no" for both S_1 and S_2, and the answer to a query Q_F with F containing u and another vertex of both stars is "yes" in both cases. Therefore $\mathcal{F}$ must contain the query $\{u, v\}$ or otherwise it cannot distinguish between S_1 and S_2. □

Note that the proof actually shows that even the solution of the $\mathcal{S}_{n-2} \cup \mathcal{S}_{n-1}$-problem requires $\binom{n}{2}$ queries. We now consider the case where the size of the star is known and prove the following theorem, which gives lower and upper bounds on the minimum size of a family that solves the $\mathcal{S}_k$-problem. These bounds are tight in all cases up to some polylog n factor.

THEOREM 2.2. *For all $k \le n-2$ and $n > 2$, there exists a family of size $\min(\lceil \frac{n(n-2)}{2} \rceil, O(k^3 \log n))$ that solves the $\mathcal{S}_k$-problem, and every family that solves the $\mathcal{S}_k$-problem either contains $(1-o(1))\binom{n}{2}$ pairs or is of size at least $\Omega(\frac{k^3}{\log^2 n})$. Moreover, if $k \le \sqrt{n}$, then the size of any family that solves the $\mathcal{S}_k$-problem is at least $\Omega(\frac{k^3 \log n}{\log k})$. For $k = n-1$, the minimum size of such a family is exactly $\lceil \log n \rceil$.*

The best bounds we get, for various values of k, are summarized in Table 1. In the rest of this section we prove these results.

PROPOSITION 2.3. *For all $n > 2$, the minimum size of a family $\mathcal{F}$ that solves the $\mathcal{S}_{n-1}$-problem is exactly $\lceil \log n \rceil$.*

Proof. The family $\mathcal{S}_{n-1}$ is of size n, so we clearly need at least $\lceil \log n \rceil$ queries. To prove that $\lceil \log n \rceil$ queries suffice, we construct the following family that solves the $\mathcal{S}_{n-1}$-problem. Note that we need only identify the center of the star. Assign distinct vectors of length $\lceil \log n \rceil$ over $\{0, 1\}$ to the vertices in V. For all $1 \le i \le \lceil \log n \rceil$, let F_i be the set of all vertices whose ith bit is 1, and let $\mathcal{F} = \{F_i \mid 1 \le i \le \lceil \log n \rceil\}$. The answer to a query Q_{F_i} is "yes" if and only if F_i contains the center. Thus, for all i, we can obtain the ith bit of the center from the answer to Q_{F_i}. □

PROPOSITION 2.4. *The minimum size of a family $\mathcal{F}$ that solves the $\mathcal{S}_{n-2}$-problem is 2 for $n = 3$, 5 for $n = 4$, and $\lceil \frac{n(n-2)}{2} \rceil$ for all $n \ge 5$*

Proof. Let $\mathcal{F}$ be a family that solves the $\mathcal{S}_{n-2}$-problem. Let u, v, and w be three distinct vertices in V. Let S_1 be the star whose center is u and in which the isolated vertex is v, and let S_2 be the star whose center is u and the isolated vertex is w. The only sets that distinguish between S_1 and S_2 are $\{u, v\}$ and $\{u, w\}$; hence $\mathcal{F}$ must contain at least one of them. Thus $\mathcal{F}$ must contain all pairs of vertices but a matching, and hence $|\mathcal{F}| \ge \lceil \frac{n(n-2)}{2} \rceil$. Moreover, it is easy to check directly that for $n = 3$, 2 pairs are necessary and suffice, and so are 5 pairs for $n = 4$.

On the other hand, assume $n \geq 5$, take a maximum matching M on V, and let $\mathcal{F}$ be the family of all pairs of vertices besides those in M. Since $|M| = \lfloor \frac{n}{2} \rfloor$, $|\mathcal{F}| = \lceil \frac{n(n-2)}{2} \rceil$. All the edges but those of M are obtained directly from the queries. Since $n \geq 5$, the center of the star can be identified by these edges, and then the only edge that may be in the graph and was not asked is the edge in M incident with the center, if there is such an edge in M. We can now decide whether this edge exists or not by the size of the star that was found. □

Note that the above upper bound holds for all $\mathcal{S}_k$, where $3 \leq k \leq n-2$.

We now give some general upper and lower bounds on the minimum size of a family that solves the $\mathcal{S}_k$-problem. These bounds are tight up to some $\operatorname{polylog} n$ factor. From now on we assume, throughout the section, that n is large.

PROPOSITION 2.5. *For every k, there exists a family $\mathcal{F}$ of size $O(k^3 \log n)$ that solves the $\mathcal{S}_k$-problem.*

Proof. Let $m = ck^3 \log n$ for some absolute constant c, and let $F_1, F_2, \ldots, F_m$ be m random subsets of V, chosen independently as follows. For every F_i, every $v \in V$ is chosen to be in F_i independently with probability $\frac{1}{k}$. Let S_1 and S_2 be two stars of size k, such that $|E(S_1) \setminus E(S_2)| = |E(S_2) \setminus E(S_1)| = 1$. Let u be the center of S_1 and S_2, let $u_1, \ldots, u_{k-1}$ be the other common vertices, and let v be the additional vertex of S_1, and w the additional vertex of S_2. F_i distinguishes between S_1 and S_2 if and only if $u \in F_i$, $u_j \notin F_i$ for all $1 \leq j \leq k-1$, and exactly one vertex among v and w is in F_i. Thus the probability that F_i distinguishes between S_1 and S_2 is

$$\frac{2}{k^2}\left(1 - \frac{1}{k}\right)^k = \Omega\left(\frac{1}{k^2}\right).$$

Therefore, the probability that no F_i distinguishes between S_1 and S_2 is

$$\left[1 - \Omega\left(\frac{1}{k^2}\right)\right]^m < n^{-(2k+2)},$$

provided c is sufficiently large. For two stars that differ in more edges, this probability is smaller. The number of pairs of stars is smaller than n^{2k+2}, and hence, there is a family $\mathcal{F} = \{F_1, F_2, \ldots, F_m\}$ that solves the $\mathcal{S}_k$-problem. □

We show that the upper bound given in Proposition 2.5 is tight up to a factor of $\operatorname{polylog} n$. More precisely, we show that for every $k \leq n-2$, a family $\mathcal{F}$ that solves the $\mathcal{S}_k$-problem either contains $(1 - o(1))\binom{n}{2}$ pairs or is of cardinality $\Omega(\frac{k^3}{\operatorname{polylog} n})$.

PROPOSITION 2.6. *For every $k \leq n-2$, if $\mathcal{F}$ is a family that solves the $\mathcal{S}_k$-problem, then $\mathcal{F}$ either contains $(1 - o(1))\binom{n}{2}$ pairs or is of cardinality at least $\Omega(\frac{k^3}{\log^2 n})$.*

Proof. Let $\mathcal{F}$ be a family that solves the $\mathcal{S}_k$-problem. Then, for every $u \in V$ and $A, B \subseteq V \setminus \{u\}$ such that $|A| = 2$, $|B| = k-1$, and $A \cap B = \emptyset$, there exists a set $F \in \mathcal{F}$ such that $u \in F$, $|F \cap A| = 1$, and $F \cap B = \emptyset$. Indeed, otherwise $\mathcal{F}$ would not distinguish between the two stars whose center is u, which share the vertices of B, and where the additional vertex of one star is one vertex of A, and the additional vertex of the other one is the other vertex of A. Denote by $\mathcal{F}_0$ the family of all sets $F \in \mathcal{F}$ of size 2. Let $m = c \cdot \frac{n \log n}{k}$, and define $\mathcal{F}_1 = \{F \in \mathcal{F} \mid 2 < |F| \leq m\}$ and $\mathcal{F}_2 = \mathcal{F} \setminus (\mathcal{F}_0 \cup \mathcal{F}_1)$. We show that for any constant $\epsilon > 0$, if $|\mathcal{F}_0| \leq (1-\epsilon)\binom{n}{2}$, then $|\mathcal{F}_1 \cup \mathcal{F}_2| > c_1 \epsilon^3 \cdot \frac{k^3}{\log^2 n}$ for some constant c_1 that depends only on c. Suppose $|\mathcal{F}_0| \leq (1-\epsilon)\binom{n}{2}$ and $|\mathcal{F}_1 \cup \mathcal{F}_2| \leq c_1 \epsilon^3 \cdot \frac{k^3}{\log^2 n}$. For every $u \in V$, denote by V_u the set of vertices $v \in V \setminus \{u\}$ such that $\{u, v\} \notin \mathcal{F}_0$. Let $V' = \{u \in V \mid |V_u| \geq \frac{\epsilon}{2}(n-1)\}$.

Since $|\mathcal{F}_0| \leq (1-\epsilon)\binom{n}{2}$, $|V'| \geq \frac{\epsilon}{2}n$. Otherwise, since the pairs of vertices that are not in $\mathcal{F}_0$ are pairs $\{u,v\}$ such that $v \in V_u$, and since $v \in V_u$ if and only if $u \in V_v$, we have

$$\begin{aligned}
|\mathcal{F}_0| &= \binom{n}{2} - \frac{1}{2}\sum_{u\in V}|V_u| \\
&= \binom{n}{2} - \frac{1}{2}\left(\sum_{u\in V'}|V_u| + \sum_{u\in V\setminus V'}|V_u|\right) \\
&> \binom{n}{2} - \frac{1}{2}\left[|V'|(n-1) + |V\setminus V'|\frac{\epsilon}{2}(n-1)\right] \\
&> \binom{n}{2} - \frac{1}{2}\left[\frac{\epsilon}{2}n(n-1) + n\frac{\epsilon}{2}(n-1)\right] \\
&= (1-\epsilon)\binom{n}{2}.
\end{aligned}$$

Choose uniformly a vertex $u \in V'$ and then choose uniformly a subset $A = \{v,w\} \subseteq V_u$. Define $\mathcal{F}_1' = \{F \in \mathcal{F}_1 \mid u \in F, |F \cap A| = 1\}$. For each $F \in \mathcal{F}_1$

$$Pr(F \in \mathcal{F}_1') \leq \frac{|F|(|F|-1)(n-|F|)}{\frac{\epsilon}{2}n\binom{\frac{\epsilon}{2}(n-1)}{2}} \leq \frac{16}{\epsilon^3}\cdot\frac{|F|^2}{n^2}.$$

Therefore,

$$E[|\mathcal{F}_1'|] \leq \frac{16}{\epsilon^3}\sum_{F\in\mathcal{F}_1}\frac{|F|^2}{n^2} \leq \frac{16}{\epsilon^3}|\mathcal{F}_1|\frac{m^2}{n^2},$$

and hence, there is a choice of u and A such that

$$\begin{aligned}
|\mathcal{F}_1'| &\leq \frac{16}{\epsilon^3}|\mathcal{F}_1|\frac{m^2}{n^2} \\
&\leq 16c_1c^2\cdot\frac{k^3}{\log^2 n}\cdot\frac{n^2\log^2 n}{k^2n^2} \\
&\leq \frac{k}{2} - 1,
\end{aligned}$$

provided c_1c^2 is sufficiently small. Thus, there exists a subset $B_1 \subseteq V\setminus(\{u\}\cup A)$ of size $\frac{k}{2}-1$ that intersects every $F \in \mathcal{F}_1'$. Choose a random subset $B_2 \subseteq V$ of size $\frac{k}{2}$. For every $F \in \mathcal{F}_2$

$$\begin{aligned}
Pr(F\cap B_2 = \emptyset) &= \frac{\binom{n-|F|}{\frac{k}{2}}}{\binom{n}{\frac{k}{2}}} \\
&\leq \left(1-\frac{|F|}{n}\right)^{\frac{k}{2}} \\
&\leq e^{-\frac{km}{2n}} \\
&= n^{-c_2}
\end{aligned}$$

for some constant $c_2 = \Theta(c)$. Therefore, if c is sufficiently large, with high probability $u \notin B_2$, $A \cap B_2 = \emptyset$, and for all $F \in \mathcal{F}_2$, $F \cap B_2 \neq \emptyset$. Denote $B' = B_1 \cup B_2$. $B' \subseteq V \setminus (\{u\} \cup A)$, and $|B'| \leq k-1$. Let B be an arbitrary extension of B' to a subset of $V \setminus (\{u\} \cup A)$ of size $k-1$. Consider the following two stars S_1 and S_2; u is the center of S_1 and S_2, they share the vertices of B, the additional vertex of S_1 is v, and the additional vertex of S_2 is w. Since A was chosen from V_u, the pairs $\{u, v\}$ and $\{u, w\}$ are not in $\mathcal{F}_0$, and thus no set in $\mathcal{F}_0$ can distinguish between S_1 and S_2. Neither can the sets in $\mathcal{F}_1$ that do not contain u, nor those whose intersection with A is not of size 1. All other sets in $\mathcal{F}_1$ (i.e., sets $F \in \mathcal{F}_1$ such that $u \in F$ and $|F \cap A| = 1$) and all the sets in $\mathcal{F}_2$ contain a vertex of B, so they cannot distinguish between these two stars either. Thus $\mathcal{F}$ cannot distinguish between S_1 and S_2, contradicting the assumption that it solves the $\mathcal{S}_k$-problem. □

We now prove a better lower bound for $k \leq \sqrt{n}$. This bound is tight up to a factor of $\log k$. For the proof of this bound, we need a variant of a lemma proved in [8, 4].

DEFINITION 2.7. *Let $\mathcal{A}$ be a family of subsets of a set S. We say that $\mathcal{A}$ is* r-cover-free *if no set in $\mathcal{A}$ is contained in the union of any r other sets in $\mathcal{A}$.*

LEMMA 2.8. *Let S be a set of size m, and let $\mathcal{A}$ be a family of n subsets of S. Suppose $\mathcal{A}$ is r-cover-free, where $r \leq 2\sqrt{n}$. Then,*

$$m > \frac{r^2 \log(n - \frac{r}{2})}{10 \log r}.$$

In [8], it is proved that for fixed r and large $n \geq n(r)$, $\frac{\log n}{m} \leq 8 \cdot \frac{\log r}{r^2}$. A simple modification of that proof described below shows that the lemma as stated above holds for every $r \leq 2\sqrt{n}$.

Proof. Let S and $\mathcal{A}$ be as defined in the lemma, and suppose that $m \leq \frac{r^2 \log(n - \frac{r}{2})}{10 \log r}$. As long as $\mathcal{A}$ contains a set A of size greater than $\frac{2m}{r}$, we remove A from $\mathcal{A}$, and its members from S and from all other sets in $\mathcal{A}$. Since $|S| = m$, this process stops after at most $\frac{r}{2}$ steps. Thus, we now have a subset S' of S and a family $\mathcal{A}'$ of subsets of S', where each subset is of size at most $\frac{2m}{r}$. Denote by m' the size of S', and by n' the size of $\mathcal{A}'$. Clearly, $n' \geq n - \frac{r}{2}$. No set $A \in \mathcal{A}'$ is contained in the union of $\frac{r}{2}$ others, or otherwise, the original set from which A was obtained would be contained in the union of these $\frac{r}{2}$ sets and the sets that were removed. Thus, every set in $\mathcal{A}'$ has a subset of size at most $\lceil \frac{4m}{r^2} \rceil$ that is not contained in any other set in $\mathcal{A}'$. Otherwise, if there were a set $A \in \mathcal{A}'$ for which the above did not hold, then, since $|A| \leq \frac{2m}{r}$, the set A would have been covered by $\frac{r}{2}$ other sets in $\mathcal{A}'$, which is impossible. Therefore, there are n' distinct sets of size at most $\lceil \frac{4m}{r^2} \rceil$. Thus,

$$n' \leq \binom{m'}{\lceil \frac{4m}{r^2} \rceil}.$$

If $\frac{4m}{r^2} < 1$, then the right-hand side of this inequality is m', and thus we have $n' \leq m'$ and hence $n \leq m$, contradicting the assumption that

$$m \leq \frac{r^2 \log(n - \frac{r}{2})}{10 \log r} \leq \frac{4n}{5}(1 + o(1)).$$

Thus, $\frac{4m}{r^2} \geq 1$, and we have

$$n - \frac{r}{2} \leq n' \leq \binom{m'}{\lceil \frac{4m}{r^2} \rceil} \leq \binom{m}{\lceil \frac{4m}{r^2} \rceil} < 2^{\frac{10m \log r}{r^2}}$$

and hence

$$m > \frac{r^2 \log(n - \frac{r}{2})}{10 \log r},$$

contradicting the assumption. □

We use the above lemma to improve the lower bound for $k \leq \sqrt{n}$.

PROPOSITION 2.9. *For every $k \leq \sqrt{n}$, if $\mathcal{F}$ is a family that solves the $\mathcal{S}_k$-problem, then $|\mathcal{F}| = \Omega(\frac{k^3 \log n}{\log k})$.*

Proof. Let $\mathcal{F}$ be a family that solves the $\mathcal{S}_k$-problem. Choose, randomly, $A, B \subseteq V$, such that $|A| = 2$, $|B| = \frac{k}{2} - 1$, and $A \cap B = \emptyset$. Define $\mathcal{G} = \{F \in \mathcal{F} \mid |F \cap A| = 1, F \cap B = \emptyset\}$. Clearly,

$$\begin{aligned} Pr(F \in \mathcal{G}) &= \frac{|F|(n - |F|)}{\binom{n}{2}} \frac{\binom{n-|F|-1}{\frac{k}{2}-1}}{\binom{n-2}{\frac{k}{2}-1}} \\ &= \frac{2|F|}{n} \frac{\binom{n-|F|}{\frac{k}{2}}}{\binom{n-1}{\frac{k}{2}}} \\ &\leq \frac{2|F|}{n} \left(1 - \frac{|F| - 1}{n - 1}\right)^{\frac{k}{2}} \\ &\leq \frac{2|F|}{n} e^{-\frac{k|F|}{4n}}. \end{aligned}$$

If $|F| \leq \frac{4n}{k}$, then

$$Pr(F \in \mathcal{G}) \leq \frac{8}{k}.$$

If $|F| > \frac{4n}{k}$, denote $x = \frac{k|F|}{4n}$. Since $x > 1$ we have

$$Pr(F \in \mathcal{G}) \leq \frac{8}{k} x e^{-x} < \frac{8}{ek}.$$

Hence, for all F,

$$Pr(F \in \mathcal{G}) \leq \frac{c}{k}$$

for some constant c, and thus the expected size of $\mathcal{G}$ is $c \cdot \frac{|\mathcal{F}|}{k}$. Therefore, there exists a choice of A and B for which $|\mathcal{G}| \leq c \cdot \frac{|\mathcal{F}|}{k}$. Denote $V' = V \setminus (A \cup B)$ and consider the family $\mathcal{G}' = \{F \cap V' \mid F \in \mathcal{G}\}$. Since $\mathcal{F}$ solves the $\mathcal{S}_k$-problem, for all $u \in V'$, and

every $C \subseteq V' \setminus \{u\}$ of size $\frac{k}{2}$, there is a set $F \in \mathcal{G}'$ such that $u \in F$ and $F \cap C = \emptyset$. Otherwise, $\mathcal{F}$ would not distinguish between the two stars whose center is u, that share the $k-1$ vertices of $B \cup C$, and for which the additional vertex of one of them is one element of A and the additional vertex of the other one is the other element of A. Let $m = |\mathcal{G}'|$, let $n' = |V'| = n - \frac{k}{2} - 1$, and let M be the m by n' matrix whose rows are the incidence vectors of the sets in $\mathcal{G}'$. Now let us look at the columns of M as the incidence vectors of subsets of another set, of size m. For every column i and every set J of $\frac{k}{2}$ columns such that $i \notin J$, there exists a row in which the ith coordinate is 1, and for all $j \in J$, the jth coordinate is 0. Thus, no subset corresponding to a column is contained in the union of $\frac{k}{2}$ subsets corresponding to any other $\frac{k}{2}$ columns, and by Lemma 2.8,

$$|\mathcal{G}'| = m > \frac{(\frac{k}{2})^2 \log(n' - \frac{k}{4})}{10 \log \frac{k}{2}} = \Omega\left(\frac{k^2 \log n}{\log k}\right),$$

and hence,

$$|\mathcal{F}| \geq \Omega(k|\mathcal{G}|) \geq \Omega(k|\mathcal{G}'|) \geq \Omega\left(\frac{k^3 \log n}{\log k}\right). \qquad \square$$

3. Complete graphs. In this section we consider the case where the hidden graphs are complete graphs. Denote by $\mathcal{C}_k$ the family of all graphs on $V = \{1, 2, \ldots, n\}$ that consist of a copy of K_k and $n - k$ isolated vertices. Let $\mathcal{C} = \cup_{k=2}^{n} \mathcal{C}_k$.

In the following theorem, we prove lower and upper bounds on the minimum size of a family that solves the $\mathcal{C}$-problem.

THEOREM 3.1. *Any family that solves the $\mathcal{C}$-problem is of size at least $\Omega(n \log n)$, and there exists a family of size $O(n \log^2 n)$ that solves the $\mathcal{C}$-problem.*

PROPOSITION 3.2. *The minimum size of a family $\mathcal{F}$ that solves the $\mathcal{C}$-problem is at least $\Omega(n \log n)$.*

Proof. Let $\mathcal{F}$ be a family that solves the $\mathcal{C}$-problem. Let $u \in V$, and let $V_1 = V \setminus \{u\}$. In order to distinguish between the complete graph on V_1 and the complete graph on $V_1 \cup \{u\}$, $\mathcal{F}$ must contain a query $F_1(u) = \{u, v_1\}$ for some $v_1 \in V_1$. Now let $V_2 = V_1 \setminus \{v_1\}$. In order to distinguish between the complete graph on V_2 and the complete graph on $V_2 \cup \{u\}$, $\mathcal{F}$ must contain a query $F_2(u)$ such that $u \in F_2(u)$ and $|F_2(u) \cap V_2| = 1$. Denote by v_2 the vertex in $F_2(u) \cap V_2$. We can continue in this way and define for all $1 \leq i \leq n-2$ a set $V_i = V_{i-1} \setminus \{v_{i-1}\}$ and find a set $F_i(u) \in \mathcal{F}$ that distinguishes between the complete graph on V_i and the complete graph on $V_i \cup \{u\}$. Then $u \in F_i(u)$, and $|F_i(u) \cap V_i| = 1$. Denote by v_i the vertex in $F_i(u) \cap V_i$. For all $1 \leq i \leq n-2$, $|V_i| = n - i$, and since $|F_i(u) \cap V_i| = 1$, $|F_i(u)| \leq i + 1$. Furthermore, all the sets $F_i(u)$ for $1 \leq i \leq n-2$ are distinct, since the vertices v_i are distinct, and $v_i \in F_i(u)$, but for all $j < i$, $v_i \notin F_j(u)$. $\mathcal{F}$ contains these sets $F_i(u)$ for all $u \in V$. For every vertex $u \in V$ and all $1 \leq i \leq n-2$, assign a weight to the pair (u, i), defined by $w(u, i) = \frac{1}{|F_i(u)|}$. For a set $F \in \mathcal{F}$, there are at most $|F|$ vertices u (the vertices in F) such that $F = F_i(u)$ for some i. Thus the total weight corresponding to a set $F \in \mathcal{F}$ is at most 1; that is,

$$\sum_{(u,i):F_i(u)=F} w(u, i) \leq |F| \cdot \frac{1}{|F|} = 1.$$

Therefore,

$$\begin{aligned}|\mathcal{F}| &\geq \sum_{F\in\mathcal{F}} \sum_{(u,i):F_i(u)=F} w(u,i)\\ &= \sum_{u\in V}\sum_{i=1}^{n-2} w(u,i)\\ &= \sum_{u\in V}\sum_{i=1}^{n-2} \frac{1}{|F_i(u)|}\\ &\geq \sum_{u\in V}\sum_{i=1}^{n-2} \frac{1}{i+1}\\ &= \Omega(n\log n). \qquad \square\end{aligned}$$

PROPOSITION 3.3. *There exists a family $\mathcal{F}$ of cardinality $O(n\log^2 n)$ that solves the $\mathcal{C}$-problem.*

Proof. We construct the family $\mathcal{F}$ recursively as follows. First, the set V is in $\mathcal{F}$. Now partition V into two halves V_1 and V_2 and find the part of the clique in each half. The clique is the union of the cliques found in V_1 and V_2. This works as long as the part of the clique in each V_i is of size 0 or of size at least 2. But if the part of the clique in V_i is of size 1, then the answer to Q_{V_i} is "no," and we need some additional queries to find this vertex. Suppose that the clique has one vertex in V_1. We show that we can find this vertex by the following queries. Assign distinct vectors of length $\lceil\log|V_1|\rceil$ over $\{0,1\}$ to the vertices in V_1. For all $1\leq i\leq\lceil\log|V_1|\rceil$, $j\in\{0,1\}$ and $u\in V_2$, we have the following set $F(i,j,u)=\{v\in V_1 \mid \text{the } i\text{th bit of } v \text{ is } j\}\cup\{u\}$ in $\mathcal{F}$. If the answer to Q_V is "yes" and the answer to Q_{V_1} is "no," then there is at least one vertex u of the clique in V_2. If there are no vertices of the clique in V_1, then the answers to all $Q_{F(i,j,u)}$ are "no." Otherwise, there is precisely one vertex v of the clique in V_1. The answer to $Q_{F(i,j,u)}$ is "yes" if and only if u is in the clique, and the ith bit of v is j. Since there is at least one vertex of the clique in V_2, we can obtain v from these queries. We should have similar queries for the case that V_2 contains one vertex of the clique. Denote by $f(n)$ the number of queries needed for n vertices. Then, by the above discussion,

$$f(n)\leq 4\cdot\frac{n}{2}\cdot\log\frac{n}{2}+2f\left(\frac{n}{2}\right)+1=O(n\log^2 n). \qquad \square$$

We now give upper and lower bounds for cliques of a given size. These results are tight up to a factor of polylog n for all admissible sizes.

THEOREM 3.4. *For every k, there exists a family $\mathcal{F}$ of size $O(k^2\log n)$ that solves the $\mathcal{C}_k$-problem, and every family that solves the $\mathcal{C}_k$-problem either contains $\Omega(n)$ pairs or is of size at least $\Omega(\frac{k^2}{\log n})$. Moreover, for all $k\leq n^{\frac{1}{3}}$, the size of any family that solves the $\mathcal{C}_k$-problem is at least $\Omega(\frac{k^2\log n}{\log k})$, and for all $k\leq\sqrt{n}$ it is at least $\Omega(k^2)$. In addition, for all s, there exists a family of size $(s+1)\lceil\frac{n}{2}\rceil$ that solves the $\mathcal{C}_{n-s}$-problem.*

The best bounds we have, for various values of k, are summarized in Table 2. In the rest of this section we prove these results.

PROPOSITION 3.5. *For every k, there exists a family $\mathcal{F}$ of size $O(k^2\log n)$ that solves the $\mathcal{C}_k$-problem.*

Proof. Let $m=ck^2\log n$ for some absolute constant c, and let $F_1,F_2,\ldots,F_m$ be m random subsets of V, chosen independently as follows. For every F_i, every $v\in V$ is

TABLE 2
Bounds on the size of a family that solves the $\mathcal{C}_k$-problem.

k	Lower bound	Upper bound
$k \le n^{\frac{1}{3}}$	$\Omega(\frac{k^2 \log n}{\log k})$	$O(k^2 \log n)$
$n^{\frac{1}{3}} < k \le \sqrt{n}$	$\Omega(k^2)$	$O(k^2 \log n)$
$\sqrt{n} < k < \sqrt{n \log n}$	$\Omega(\frac{k^2}{\log n})$	$O(k^2 \log n)$
$\sqrt{n \log n} \le k \le n - \log^2 n$	$\Omega(n)$	$O(n \log^2 n)$
$k = n - s,\ s < \log^2 n$	$\Omega(n)$	$(s+1)\lceil \frac{n}{2} \rceil$

chosen to be in F_i independently with probability $\frac{1}{k}$. Let C_1 and C_2 be two complete graphs of size k such that $|V(C_1) \setminus V(C_2)| = |V(C_2) \setminus V(C_1)| = 1$. Let $v_1, \ldots, v_{k-1}$ be the common vertices of C_1 and C_2, and let u_i be the additional vertex of C_i for $i = 1, 2$. F_i distinguishes between C_1 and C_2 if and only if exactly one vertex among u_1 and u_2 and exactly one vertex among $v_1, \ldots, v_{k-1}$ are in F_i. Thus the probability that F_i distinguishes between C_1 and C_2 is

$$\frac{2}{k} \cdot \frac{k-1}{k} \left(1 - \frac{1}{k}\right)^{k-1} = \Omega\left(\frac{1}{k}\right).$$

Therefore, the probability that no F_i distinguishes between C_1 and C_2 is

$$\left[1 - \Omega\left(\frac{1}{k}\right)\right]^m \le n^{-2k}$$

for an appropriate value of c. For two cliques that differ in more vertices, this probability is smaller. The number of pairs of cliques is smaller than n^{2k}, and hence, there is a family $\mathcal{F} = \{F_1, F_2, \ldots, F_m\}$ that solves the $\mathcal{C}_k$-problem. □

PROPOSITION 3.6. *For every k, if $\mathcal{F}$ is a family the solves the $\mathcal{S}_k$-problem, then $\mathcal{F}$ either contains $\Omega(n)$ pairs or is of cardinality at least $\Omega(\frac{k^2}{\log n})$.*

Proof. Clearly, we may assume that $k^2 > \log n$, since otherwise there is nothing to prove. Let $\mathcal{F}$ be a family that solves the $\mathcal{C}_k$-problem. Then, for all $A, B \subseteq V$ such that $|A| = 2$, $|B| = k - 1$, and $A \cap B = \emptyset$, there exists a set $F \in \mathcal{F}$ such that $|F \cap A| = 1$ and $|F \cap B| = 1$. Indeed, otherwise $\mathcal{F}$ would not distinguish between the complete graph on B and one vertex of A, and the complete graph on B and the other vertex of A. Denote by $\mathcal{F}_0$ the family of all sets $F \in \mathcal{F}$ of size 2. Let $m = c \cdot \frac{n \log n}{k}$, and define $\mathcal{F}_1 = \{F \in \mathcal{F} \mid 2 < |F| \le m\}$ and $\mathcal{F}_2 = \mathcal{F} \setminus (\mathcal{F}_0 \cup \mathcal{F}_1)$. We show that if, say, $|\mathcal{F}_0| \le \frac{1}{10}n$, then $|\mathcal{F}_1 \cup \mathcal{F}_2| > c_1 \cdot \frac{k^2}{\log n}$ for some constant c_1 that depends only on c. Suppose $|\mathcal{F}_0| \le \frac{1}{10}n$ and $|\mathcal{F}_1 \cup \mathcal{F}_2| \le c_1 \cdot \frac{k^2}{\log n}$. Choose uniformly a subset $A = \{u, v\} \subseteq V$, and define $\mathcal{F}_1' = \{F \in \mathcal{F}_1 \mid |F \cap A| = 1\}$. For each $F \in \mathcal{F}_1$

$$Pr(F \in \mathcal{F}_1') = 2 \cdot \frac{|F|}{n} \cdot \frac{n - |F|}{n-1} \le 2 \cdot \frac{|F|}{n}.$$

Therefore,

$$E[|\mathcal{F}_1'|] \le 2 \sum_{F \in \mathcal{F}_1} \frac{|F|}{n} \le 2|\mathcal{F}_1|\frac{m}{n}.$$

By Markov's inequality, the probability that $|\mathcal{F}_1'| > 4|\mathcal{F}_1|\frac{m}{n}$ is at most $\frac{1}{2}$. Since $|\mathcal{F}_0| \leq \frac{1}{10}n$, the probability that there is a set $F \in \mathcal{F}_0$ such that $F \cap A \neq \emptyset$ is less than $\frac{2}{5}$. Thus, there exists a choice of A such that for all $F \in \mathcal{F}_0$, $F \cap A = \emptyset$ and $|\mathcal{F}_1'| \leq 4|\mathcal{F}_1|\frac{m}{n}$. For such a choice of A and appropriate values of c and c_1,

$$\begin{aligned}|\mathcal{F}_1'| &\leq 4|\mathcal{F}_1| \cdot \frac{m}{n} \\ &\leq 4c_1 c \cdot \frac{k^2}{\log n} \cdot \frac{n \log n}{kn} \\ &\leq \frac{k-1}{4}.\end{aligned}$$

Thus, there exists a subset $B_1 \subseteq V \setminus A$ of size $\frac{k-1}{2}$ such that for every $F \in \mathcal{F}_1'$, $|F \cap B_1| \geq 2$. Choose a random subset $B_2 \in V$ of size $\frac{k-1}{2}$. For all $F \in \mathcal{F}_2$

$$\begin{aligned}Pr(|F \cap B_2| \leq 1) &= \frac{\binom{n-|F|}{\frac{k-1}{2}}}{\binom{n}{\frac{k-1}{2}}} + \frac{k-1}{2} \cdot \frac{|F|}{n} \cdot \frac{\binom{n-|F|}{\frac{k-1}{2}-1}}{\binom{n-1}{\frac{k-1}{2}-1}} \\ &\leq \left(1 - \frac{|F|}{n}\right)^{\frac{k-1}{2}} + \frac{k-1}{2} \cdot \frac{|F|}{n} \cdot \left(1 - \frac{|F|-1}{n-1}\right)^{\frac{k-1}{2}-1} \\ &\leq e^{-\frac{k-1}{2} \cdot \frac{m}{n}} + \frac{k-1}{2} \cdot \frac{n}{n} \cdot e^{-\left(\frac{k-1}{2}-1\right)\frac{m-1}{n-1}} \\ &\leq n^{-c_2}\end{aligned}$$

for some constant $c_2 = \Theta(c)$. Therefore, if c is sufficiently large, then, with high probability, $A \cap B_2 = \emptyset$ and for every $F \in \mathcal{F}_2$, $|F \cap B_2| \geq 2$. Denote $B' = B_1 \cup B_2$. $B' \subseteq V \setminus A$, and $|B'| \leq k-1$. Let B be an arbitrary extension of B' to a subset of $V \setminus A$ of size $k-1$. Let C_1 be the complete graph on $B \cup \{u\}$, and let C_2 be the complete graph on $B \cup \{v\}$. Since for every $F \in \mathcal{F}_0$, $u, v \notin F$, no set in $\mathcal{F}_0$ can distinguish between C_1 and C_2. Neither can sets in $\mathcal{F}_1$ that contain both u and v, or that contain neither of them. All other sets in $\mathcal{F}_1$, i.e., sets that contain exactly one vertex among u and v, and all the sets in $\mathcal{F}_2$ contain at least two vertices of B, so they cannot distinguish between these two cliques either. Thus $\mathcal{F}$ cannot distinguish between C_1 and C_2, contradicting the assumption that it solves the $\mathcal{C}_k$-problem. ∎

We now prove a better lower bound for $k \leq n^{\frac{1}{3}}$. This bound is tight up to a factor of $\log k$.

LEMMA 3.7. *Let S be a set of size m, and let $\mathcal{A}$ be a family of n subsets of S. Suppose that there are no distinct $A, B_1, \ldots, B_r, C_1, \ldots, C_r \in \mathcal{A}$ for which*

$$A \subseteq \bigcup_{i=1}^{r} B_i$$

and

$$A \subseteq \bigcup_{i=1}^{r} C_i,$$

where $r \leq n^{\frac{1}{3}}$. Then $m = \Omega(\frac{r^2 \log n}{\log r})$.

Proof. Let $\mathcal{B} = \emptyset$. As long as there exist $A, B_1, \ldots, B_r \in \mathcal{A}$ such that

$$A \subseteq \bigcup_{i=1}^{r} B_i,$$

remove $A, B_1, \ldots, B_r$ from $\mathcal{A}$, and add A to $\mathcal{B}$. Let $\mathcal{A}'$ be the family obtained from $\mathcal{A}$ at the end of this process, and denote the size of $\mathcal{B}$ by l. Then, $|\mathcal{A}'| = n - l(r+1)$, and both $\mathcal{A}'$ and $\mathcal{B}$ are r-cover-free. $\mathcal{A}'$ is clearly r-cover-free, or otherwise the above process would not stop. $\mathcal{B}$ is also r-cover-free, because if there were $A, C_1, \ldots, C_r \in \mathcal{B}$ such that

$$A \subseteq \bigcup_{i=1}^{r} C_i,$$

then there would also be $B_1, \ldots, B_r \in \mathcal{A}$ that were removed from $\mathcal{A}$ together with A, such that

$$A \subseteq \bigcup_{i=1}^{r} B_i,$$

contradicting the assumption. If $l \geq \frac{n^{\frac{2}{3}}}{4}$, then, since $r \leq n^{\frac{1}{3}}$, we have, by Lemma 2.8,

$$m > \frac{r^2 \log(l - \frac{r}{2})}{10 \log r} = \Omega\left(\frac{r^2 \log n}{\log r}\right).$$

Otherwise $l < \frac{n^{\frac{2}{3}}}{4}$, and thus, since $r < n^{\frac{1}{3}}$, $|\mathcal{A}'| = n - l(r+1) > \frac{n}{2}$. Hence, by Lemma 2.8,

$$m > \frac{r^2 \log(\frac{n}{2} - \frac{r}{2})}{10 \log r} = \Omega\left(\frac{r^2 \log n}{\log r}\right). \qquad \square$$

PROPOSITION 3.8. *For every $k \leq n^{\frac{1}{3}}$, if $\mathcal{F}$ is a family that solves the $\mathcal{C}_k$-problem, then $|\mathcal{F}| = \Omega(\frac{k^2 \log n}{\log k})$.*

Proof. Let $\mathcal{F}$ be a family that solves the $\mathcal{C}_k$-problem. Define $m = |\mathcal{F}|$, and let M be the m by n matrix whose rows are the incidence vectors of the sets in $\mathcal{F}$. Consider the columns of M as the incidence vectors of subsets of another set, of size m. For $1 \leq i \leq n$, let G_i be the subset corresponding to the ith column of M. Define the family $\mathcal{G}$ as $\mathcal{G} = \{G_{2i-1} \cup G_{2i} \mid 1 \leq i \leq \frac{n}{2}\}$. We claim that there are no distinct sets $A, B_1, \ldots, B_{\frac{k-1}{4}}, C_1, \ldots, C_{\frac{k-1}{4}} \in \mathcal{G}$ such that

$$A \subseteq \bigcup_{i=1}^{\frac{k-1}{4}} B_i \tag{3.1}$$

and

$$A \subseteq \bigcup_{i=1}^{\frac{k-1}{4}} C_i. \tag{3.2}$$

Suppose there were such sets. A is the union of two subsets corresponding to two distinct columns of M. Let u and v be the vertices corresponding to these columns. Similarly, let $w_1, \ldots, w_{k-1}$ be the vertices corresponding to $B_1, \ldots, B_{\frac{k-1}{4}}, C_1, \ldots, C_{\frac{k-1}{4}}$.

The members of A are the queries that contain u or v. Since (3.1) and (3.2) hold, each such query contains at least two vertices from $w_1, \ldots, w_{k-1}$. Thus, no query distinguishes between the complete graph on $u, w_1, \ldots, w_{k-1}$ and the complete graph on $v, w_1, \ldots, w_{k-1}$. Hence, there are no such sets in $\mathcal{G}$, and therefore, by Lemma 3.7, with $r = \frac{k-1}{4}$ and $\mathcal{A} = \mathcal{G}$,

$$|\mathcal{F}| = m = \Omega\left(\frac{k^2 \log n}{\log k}\right). \qquad \square$$

We now prove that for all $n^{\Omega(1)} \leq k \leq \sqrt{n}$, any family that solves the $\mathcal{C}_k$-problem is of size at least $\Omega(k^2)$.

DEFINITION 3.9. *Let A be a subset of a set S, and let $\mathcal{A}$ be a family of subsets of S. We say that A is* covered twice *by $\mathcal{A}$ if for all $a \in A$, there are at least two sets in $\mathcal{A}$ that contain a.*

LEMMA 3.10. *Let S be a set of size m, and let $\mathcal{A}$ be a family of n subsets of S. Suppose that no set in $\mathcal{A}$ is covered twice by any r other sets in $\mathcal{A}$, where $n^{\Omega(1)} \leq r \leq \sqrt{n}$. Then $m = \Omega(r^2)$.*

Proof. Suppose $m \leq \epsilon r^2$, for some small constant $\epsilon > 0$. We show that if ϵ is sufficiently small, then there is a set $A \in \mathcal{A}$ that is covered twice by some r other sets in $\mathcal{A}$. As long as there exists $a \in S$ that belongs to one or two sets in $\mathcal{A}$, remove these sets from $\mathcal{A}$. After removing these sets, a belongs to no set in $\mathcal{A}$. Therefore, this process stops after at most m steps, and then every $a \in S$ belongs to zero or at least three sets. Let $\mathcal{A}'$ be the family of the remaining sets, and denote its size by n'. Thus $n' \geq n - 2m \geq n - 2\epsilon r^2 \geq (1 - 2\epsilon)n$. If there exists a set $A \in \mathcal{A}'$ such that $|A| \leq \frac{r}{2}$, then it is covered twice by a family of at most r sets in $\mathcal{A}' \setminus \{A\}$, consisting of two arbitrarily chosen sets that contain each member of A. Suppose now that every set $A \in \mathcal{A}'$ is of size greater than $\frac{r}{2}$. Choose randomly $\frac{r}{2}$ sets $B_1, \ldots, B_{\frac{r}{2}} \in \mathcal{A}'$. Let C be the set of all $a \in S$ that belong to at most one set from $B_1, \ldots, B_{\frac{r}{2}}$. Now choose randomly another set $A \in \mathcal{A}'$. If $|A \cap C| \leq \frac{r}{4}$, then for all $a \in A \cap C$, choose two sets in $\mathcal{A}' \setminus \{A\}$ that contain a. These sets, together with $B_1, \ldots, B_{\frac{r}{2}}$, form a family of at most r sets that cover A twice. We now show that $E[|A \cap C|] \leq \frac{r}{5}$, and hence there exists a choice of $B_1, \ldots, B_{\frac{r}{2}}$ and $A \neq B_1, \ldots, B_{\frac{r}{2}}$ for which $|A \cap C| \leq \frac{r}{4}$. Therefore A is covered twice by r other sets, contradicting the assumption. Let $a \in S$, and let k be the number of sets in $\mathcal{A}'$ that contain a. The probability that $a \in A \cap C$ is at most

$$\begin{aligned}
\frac{k}{n'}\left[\frac{\binom{n'-k}{\frac{r}{2}}}{\binom{n'}{\frac{r}{2}}} + \frac{k\binom{n'-k}{\frac{r}{2}-1}}{\binom{n'}{\frac{r}{2}}}\right] &= \frac{k}{n'}\left[\frac{\binom{n'-k}{\frac{r}{2}}}{\binom{n'}{\frac{r}{2}}} + \frac{kr}{2n'}\frac{\binom{n'-k}{\frac{r}{2}-1}}{\binom{n'-1}{\frac{r}{2}-1}}\right] \\
&\leq \frac{k}{n'}\left(1 - \frac{k}{n'}\right)^{\frac{r}{2}} + \frac{k^2 r}{2n'^2}\left(1 - \frac{k-1}{n'-1}\right)^{\frac{r}{2}-1} \\
&\leq \frac{k}{n'}e^{-\frac{kr}{2n'}} + \frac{k^2 r}{2n'^2}e^{-\frac{kr}{4n'}}.
\end{aligned}$$

We now show that this probability is at most $\frac{c}{r}$ for some constant c. Let us first consider the term $\frac{k}{n'}e^{-\frac{kr}{2n'}}$. If $k \leq \frac{2n'}{r}$, then this term is at most $\frac{2}{r}$. If $k > \frac{2n'}{r}$, denote $x = \frac{kr}{2n'}$. Since $x > 1$ we have $\frac{k}{n'}e^{-\frac{kr}{2n'}} = \frac{2}{r}xe^{-x} < \frac{2}{er}$. Consider now the term $\frac{k^2 r}{2n'^2}e^{-\frac{kr}{4n'}}$. If $k \leq \frac{8n'}{r}$, then this term is at most $\frac{32}{r}$. If $k > \frac{8n'}{r}$, then denote $x = \frac{kr}{4n'}$. Then $x > 2$, and $\frac{k^2 r}{2n'^2}e^{-\frac{kr}{4n'}} = \frac{8}{r}x^2e^{-x}$. It is easy to check that x^2e^{-x} is decreasing for all $x > 2$, and hence $\frac{8}{r}x^2e^{-x} < \frac{32}{e^2 r}$.

Thus the probability that $a \in A \cap C$ is at most $\frac{c}{r}$ for some constant c. Therefore, we have

$$E[|A \cap C|] \leq \frac{cm}{r} \leq \frac{c\epsilon r^2}{r} \leq \frac{r}{5},$$

provided ϵ is sufficiently small, completing the proof of the lemma. ∎

PROPOSITION 3.11. *For every $n^{\Omega(1)} \leq k \leq \sqrt{n}$, if $\mathcal{F}$ is a family that solves the $\mathcal{C}_k$-problem, then $|\mathcal{F}| = \Omega(k^2)$.*

Proof. For $n^{\Omega(1)} \leq k \leq n^{1/3}$ the result follows from Proposition 3.8. We thus assume that $k > n^{1/3}$. Let $\mathcal{F}$ be a family that solves the $\mathcal{C}_k$-problem. Define $m = |\mathcal{F}|$, and let M be the m by n matrix whose rows are the incidence vectors of the sets in $\mathcal{F}$. Consider the columns of M as the incidence vectors of subsets of another set, of size m. For $1 \leq i \leq n$, let G_i be the subset corresponding to the ith column of M. Define $\mathcal{G} = \{G_{2i-1} \cup G_{2i} \mid 1 \leq i \leq \frac{n}{2}\}$. We claim that there are no distinct sets $A, B_1, \ldots, B_{\frac{k-1}{2}} \in \mathcal{G}$ such that A is covered twice by $B_1, \ldots, B_{\frac{k-1}{2}}$. Suppose there were such sets. A is the union of two subsets corresponding to two distinct columns of M. Let u and v be the corresponding vertices. Similarly, let $w_1, \ldots, w_{k-1}$ be the vertices corresponding to $B_1, \ldots, B_{\frac{k-1}{2}}$. The members of A are the queries that contain u or v. Since A is covered twice by $B_1, \ldots, B_{\frac{k-1}{2}}$, each such query contains at least two vertices from $w_1, \ldots, w_{k-1}$. Thus, no query distinguishes between the complete graph on $u, w_1, \ldots, w_{k-1}$ and the complete graph on $v, w_1, \ldots, w_{k-1}$. Hence, there are no such sets in $\mathcal{G}$, and therefore, by Lemma 3.10,

$$|\mathcal{F}| = m = \Omega(k^2). \qquad \blacksquare$$

We conclude the section with a simple upper bound, which improves our estimate for cliques that contain almost all the vertices.

PROPOSITION 3.12. *For every s, there exists a family of size at most*

$$(s+1)\left\lceil \frac{n}{2} \right\rceil$$

that solves the $\mathcal{C}_{n-s}$-problem.

Proof. For each $u \in V$, ask $s+1$ pairs that contain u. u is in the clique if and only if the answer to at least one of these queries is "yes." ∎

4. General graphs. In this section we consider families that contain all the graphs on V isomorphic to a graph G. Denote by $\mathcal{H}_G$ the family of all graphs isomorphic to G.

THEOREM 4.1. *Let $G = (V, E)$ be a graph on n vertices, and suppose that there are three vertices $u, v, w \in V$ such that for every two of them, the sets of their neighbors except these vertices themselves are distinct, i.e., $N(u) \setminus \{v\} \neq N(v) \setminus \{u\}$, $N(u) \setminus \{w\} \neq N(w) \setminus \{u\}$, and $N(v) \setminus \{w\} \neq N(w) \setminus \{v\}$. Then, the size of any family that solves the $\mathcal{H}_G$-problem is at least $\Omega(\frac{n^2}{\alpha^2(G)})$, where $\alpha(G)$ is the maximum size of an independent set in G.*

Proof. For any two vertices $x, y \in V$, denote by $A(x, y)$ the set of vertices $z \in V \setminus \{x, y\}$ such that z is a neighbor of both x and y, or of neither of them. We show that there are two vertices among u, v, and w, for which the size of this set is at least $\frac{1}{3}n - 1$. Suppose that $A(u, v) < \frac{1}{3}n - 1$. Then, $V \setminus (A(u, v) \cup \{u, v, w\}) > \frac{2}{3}n - 2$, and each one of these vertices is a neighbor of exactly one vertex among u and v. Thus, each one of these vertices is in $A(u, w)$ or in $A(v, w)$, and hence at least one of these sets is of size at least $\frac{1}{3}n - 1$. Assume, without loss of generality, that $|A(u, v)| \geq \frac{1}{3}n - 1$.

Let $\mathcal{F}$ be a family that solves the $\mathcal{H}_G$-problem, and let $\alpha = \alpha(G)$. Assume that $|\mathcal{F}| < \frac{n^2}{12\alpha^2}$. Every set $F \in \mathcal{F}$ is of size at most α, or otherwise the answer to Q_F is "yes" (and is known in advance). For every $x \in V$, denote by $f(x)$ the number of sets $F \in \mathcal{F}$ such that $x \in F$. Note that

$$\sum_{x \in V} f(x) = \sum_{F \in \mathcal{F}} |F| \leq \alpha |\mathcal{F}| < \frac{n^2}{12\alpha}. \tag{4.1}$$

Let $V' = \{x \in V \mid f(x) < \frac{n}{6\alpha}\}$. Then $|V'| \geq \frac{n}{2}$, since otherwise

$$\sum_{x \in V} f(x) \geq \sum_{x \in (V \setminus V')} f(x) \geq \frac{n}{2} \cdot \frac{n}{6\alpha} = \frac{n^2}{12\alpha},$$

contradicting (4.1). For $x \in V'$, the number of vertices $z \in V$ such that there exists a set $F \in \mathcal{F}$ that contains both x and z is at most

$$\sum_{F : x \in F} |F| \leq f(x)\alpha < \frac{n}{6}.$$

Let $x, y \in V'$, and let A be the set of all vertices $z \in V$ such that there exists a set $F \in \mathcal{F}$ that contains x or y, and z. Then

$$|A| \leq \sum_{F : x \in F} |F| + \sum_{F : y \in F} |F| < \frac{n}{3}.$$

Let G_1 be a graph isomorphic to G, where u is mapped to x, v is mapped to y, and only vertices from $A(u, v)$ are mapped into A. Let G_2 be the graph in which u is mapped to y, v is mapped to x, and the rest of it is identical to G_1. The only queries that could distinguish between G_1 and G_2 are queries Q_F where F contains x or y, but then all the other vertices in F are in $A(u, v)$ and thus, the answer to Q_F is the same for G_1 and G_2. Therefore, $\mathcal{F}$ cannot distinguish between G_1 and G_2, contradicting the assumption that it solves the $\mathcal{H}_G$-problem. □

COROLLARY 4.2. *Let $G = G(n, \frac{1}{2})$ be the random graph on n vertices. Then, almost surely, any family that solves the $\mathcal{H}_G$-problem is of size at least $\Omega(\frac{n^2}{\log^2 n})$.*

Proof. The corollary follows from Theorem 4.1 since almost surely $\alpha(G) = O(\log n)$ (see, for example, [3] or [2]), and since there are almost surely three vertices u, v, and w with distinct sets of neighbors, as defined in the theorem. □

5. Concluding remarks.

- It will be interesting to close the polylogarithmic gaps between the upper and the lower bounds proved in this paper.
- Another intriguing challenge is to obtain a general way to estimate, for every graph G, the number of queries needed to identify a hidden graph isomorphic to G. In particular, the problem of characterizing all graphs for which the trivial upper bound of $O(n^2)$ is best possible seems interesting. Our results enable us to prove an $\Omega(n^2)$ lower bound for the number of queries required to identify a hidden copy of any graph with at least one isolated vertex, containing a vertex of degree 1 which is adjacent to a vertex of high degree. We omit the details.
- The problems considered here can be investigated when more than one round is allowed and in the case when the algorithms are fully adaptive.

REFERENCES

[1] N. Alon, R. Beigel, S. Kasif, S. Rudich, and B. Sudakov, *Learning a hidden matching*, SIAM J. Comput., 33 (2004), pp. 487–501.
[2] N. Alon and J. H. Spencer, *The Probabilistic Method*, 2nd ed., Wiley, New York, 2000.
[3] B. Bollobás, *Random Graphs*, Academic Press, London, 1985.
[4] A. G. Dyachkov and V. V. Rykov, *Bounds on the length of disjunctive codes*, Problemy Peredachi Informatsii, 18 (1982), pp. 158–166.
[5] V. Grebinski and G. Kucherov, *Optimal query bounds for reconstructing a Hamiltonian cycle in complete graphs*, in Proceedings of the 5th Israeli Symposium on Theory of Computing and Systems, IEEE Computer Society, 1997, pp. 166–173.
[6] V. Grebinski and G. Kucherov, *Reconstructing a Hamiltonian cycle by querying the graph: Application to DNA physical mapping*, Discrete Appl. Math., 88 (1998), pp. 147–165.
[7] V. Grebinski and G. Kucherov, *Optimal reconstruction of graphs under the additive model*, Algorithmica, 28 (2000), pp. 104–124.
[8] M. Ruszinkó, *On the upper bound of the size of the r-cover-free families*, J. Combin. Theory Ser. A, 66 (1994), pp. 302–310.

SIAM J. DISCRETE MATH.
Vol. 18, No. 4, pp. 713–727

SET SYSTEMS WITH RESTRICTED CROSS-INTERSECTIONS AND THE MINIMUM RANK OF INCLUSION MATRICES*

PETER KEEVASH† AND BENNY SUDAKOV‡

Abstract. A set system is L-intersecting if any pairwise intersection size lies in L, where L is some set of s nonnegative integers. The celebrated Frankl–Ray-Chaudhuri–Wilson theorems give tight bounds on the size of an L-intersecting set system on a ground set of size n. Such a system contains at most $\binom{n}{s}$ sets if it is uniform and at most $\sum_{i=0}^{s}\binom{n}{i}$ sets if it is nonuniform. They also prove modular versions of these results.

We consider the following extension of these problems. Call the set systems $\mathcal{A}_1,\dots,\mathcal{A}_k$ L-*cross-intersecting* if for every pair of *distinct* sets A, B with $A \in \mathcal{A}_i$ and $B \in \mathcal{A}_j$ for some $i \neq j$ the intersection size $|A \cap B|$ lies in L. For any k and for $n > n_0(s)$ we give tight bounds on the maximum of $\sum_{i=1}^{k}|\mathcal{A}_i|$. It is at most $\max\{k\binom{n}{s}, \binom{n}{\lfloor n/2\rfloor}\}$ if the systems are uniform and at most $\max\{k\sum_{i=0}^{s}\binom{n}{i}, (k-1)\sum_{i=0}^{s-1}\binom{n}{i}+2^n\}$ if they are nonuniform. We also obtain modular versions of these results.

Our proofs use tools from linear algebra together with some combinatorial ideas. A key ingredient is a tight lower bound for the rank of the inclusion matrix of a set system. The s^*-*inclusion matrix* of a set system $\mathcal{A}$ on $[n]$ is a matrix M with rows indexed by $\mathcal{A}$ and columns by the subsets of $[n]$ of size at most s, where if $A \in \mathcal{A}$ and $B \subset [n]$ with $|B| \leq s$, we define M_{AB} to be 1 if $B \subset A$ and 0 otherwise. Our bound generalizes the well-known result that if $|\mathcal{A}| < 2^{s+1}$, then M has full rank $|\mathcal{A}|$. In a combinatorial setting this fact was proved by Frankl and Pach in the study of null t-designs; it can also be viewed as determining the minimum distance of the Reed–Muller codes.

Key words. set systems, restricted intersections, inclusion matrices

AMS subject classification. 05D05

DOI. 10.1137/S0895480103434634

1. Introduction.

Extremal problems on set systems with restricted intersections have been an important part of combinatorics in the last half-century. One of the first such results was obtained by Majumdar [11] and rediscovered by Isbell [8]. Extending earlier results of Fischer, they showed that a set system on $[n] = \{1,\dots,n\}$ in which the intersection of any pair of sets has the same cardinality t can have at most $n+1$ sets, and if $t \neq 0$ it can have at most n sets. This is commonly known as the nonuniform Fischer inequality. (A set system is *uniform* if all of its sets have the same size.)

Throughout this paper L will denote a set of s nonnegative integers. We say that a set system $\mathcal{A}$ is L-*intersecting* if for every $A, B \in \mathcal{A}$ we have $|A \cap B| \in L$. The nonuniform Fischer inequality was further generalized by Ray-Chaudhuri and Wilson [13] and Frankl and Wilson [7], who obtained tight bounds for L-intersecting set systems, both uniform and nonuniform. They showed that an L-intersecting family on $[n]$ can have at most $\binom{n}{s}$ sets if it is uniform, and at most $\sum_{i=0}^{s}\binom{n}{i}$ sets if it is nonuniform. Frankl and Wilson also proved modular versions of these results. For p prime, they showed that the same bounds hold if the intersection sizes belong to

*Received by the editors September 18, 2003; accepted for publication (in revised form) August 13, 2004; published electronically April 22, 2005.
http://www.siam.org/journals/sidma/18-4/43463.html
†Department of Mathematics, Caltech, Pasadena, CA 91125 (keevash@caltech.edu).
‡Department of Mathematics, Princeton University, Princeton, NJ 08544 (bsudakov@math.princeton.edu). This author's research was supported in part by NSF grants DMS-0355497 and DMS-0106589, and by an Alfred P. Sloan fellowship.

L mod p and the sizes of the sets in $\mathcal{A}$ do not belong to L mod p. For an excellent account of this subject and its applications we refer the reader to [2].

In this paper, we consider the following extension of these problems. Call the set systems $\mathcal{A}_1, \dots, \mathcal{A}_k$ *L-cross-intersecting* if for every pair of *distinct* sets A, B with $A \in \mathcal{A}_i$ and $B \in \mathcal{A}_j$ for some $i \neq j$ we have $|A \cap B| \in L$. We consider the problem of finding L-cross-intersecting systems with total size as large as possible, for each k. This can be thought of as a multicolored version of the Frankl–Ray-Chaudhuri–Wilson theorem in the following sense. We can reformulate the property of being L-intersecting as a forbidden configuration condition: we forbid any pair of sets with intersection size not lying in L. Now suppose we are given a list of set systems $\mathcal{A}_1, \dots, \mathcal{A}_k$, which we think of as colors. We call another set system $\mathcal{F}$ multicolored if for each $F \in \mathcal{F}$ we can choose a color $\mathcal{A}_i$ containing F in such a way that each $F \in \mathcal{F}$ gets a different color. Suppose we have an integer k and some forbidden configurations $\{\mathcal{F}_\gamma : \gamma \in \Gamma\}$. The multicolored extremal problem is to choose k colors $\mathcal{A}_1, \dots, \mathcal{A}_k$ with total size $|\mathcal{A}_1| + \cdots + |\mathcal{A}_k|$ as large as possible subject to containing no multicolored forbidden configuration $\mathcal{F}_\gamma$. The L-intersection problem has as forbidden configurations all pairs of sets with intersection sizes not belonging to L. The multicolored version of this is clearly equivalent to the L-cross-intersection problem defined above.

We refer the reader to [9] and [4] for recent results on other multicolored extremal problems and to [14] and [6] for other results on cross-intersecting families.

There are two natural examples of large *L-cross-intersecting* systems that are uniform. One is to take all of the $\mathcal{A}_i$ equal to some fixed maximum uniform L-intersecting set system, which in the case $L = \{0, 1, \dots, s-1\}$ can have as many as $\binom{n}{s}$ sets. Another is to take one $\mathcal{A}_i$ to be as large as possible, i.e., of size $\binom{n}{\lfloor n/2 \rfloor}$, and then all the other set systems have to be empty. The following theorem shows that one of these constructions is always optimal.

THEOREM 1.1. *Let L be a set of s nonnegative integers, $n > 100s^2 \log(s+1)$, and let $\mathcal{A}_1, \dots, \mathcal{A}_k$ be uniform set systems on $[n]$ that are L-cross-intersecting. Then*

$$\sum_{i=1}^{k} |\mathcal{A}_i| \leq \max\left\{ k\binom{n}{s}, \binom{n}{\lfloor n/2 \rfloor} \right\}.$$

We get a similar picture in the nonuniform case. Again we have the example where all of the $\mathcal{A}_i$ are equal to some fixed maximum L-intersecting set system, which can have as many as $\sum_{i=0}^{s} \binom{n}{i}$ sets when $L = \{0, 1, \dots, s-1\}$. Alternatively, if we take one $\mathcal{A}_i$ to be as large as possible, i.e., to contain all 2^n subsets of $[n]$, then the other $\mathcal{A}_i$ can contain only sets whose sizes belong to L (and are also L-cross-intersecting). In the case $L = \{0, 1, \dots, s-1\}$ we could take one $\mathcal{A}_i$ to contain all sets and take all the other set systems to consist of the subsets of size at most $s-1$. Again we prove that one of these two possibilities is optimal.

THEOREM 1.2. *Let L be a set of s nonnegative integers, $n > 100s^2 \log s$, and let $\mathcal{A}_1, \dots, \mathcal{A}_k$ be set systems on $[n]$ that are L-cross-intersecting. Then*

$$\sum_{i=1}^{k} |\mathcal{A}_i| \leq \max\left\{ k\sum_{i=0}^{s} \binom{n}{i}, (k-1)\sum_{i=0}^{s-1} \binom{n}{i} + 2^n \right\}.$$

One can ask similar questions in a modular setting. For a prime p, we say that a set system $\mathcal{A}$ is L-intersecting mod p if the sizes of all pairwise intersections of

sets belong to L mod p. We define L-cross-intersecting mod p in a similar fashion. The uniform modular Frankl–Ray-Chaudhuri–Wilson theorem states that if $\mathcal{A}$ is an r-uniform set system that is L-intersecting mod p and $r \notin L \bmod p$, then $|\mathcal{A}| \le \binom{n}{s}$. The nonuniform modular version is that if $\mathcal{A}$ is L-intersecting mod p and no set in $\mathcal{A}$ has size belonging to L mod p, then $|\mathcal{A}| \le \sum_{i=0}^{s} \binom{n}{i}$. We can show the following cross-intersecting versions of these results.

THEOREM 1.3. *Suppose p is prime, L is a set of $s < p$ residues modulo p, and $\mathcal{A}_1, \dots, \mathcal{A}_k$ are set systems on $[n]$ that are L-cross-intersecting* mod p *such that every set $A \in \bigcup_{i=1}^{k} \mathcal{A}_i$ has $|A| = r$ for some $r \notin L \bmod p$. Let m be chosen so that $m \notin L \bmod p$ and $|n/2 - m|$ is as small as possible. Then for $n > n(s)$ sufficiently large*

$$\sum_{i=1}^{k} |\mathcal{A}_i| \le \max\left\{ k\binom{n}{s}, \binom{n}{m} \right\}.$$

THEOREM 1.4. *Suppose p is prime, L is a set of $s < p$ residues modulo p, and $\mathcal{A}_1, \dots, \mathcal{A}_k$ are set systems on $[n]$ that are L-cross-intersecting* mod p *such that every set $A \in \bigcup_{i=1}^{k} \mathcal{A}_i$ has $|A| \notin L \bmod p$. Then for $n > n(s)$ sufficiently large*

$$\sum_{i=1}^{k} |\mathcal{A}_i| \le \max\left\{ k\sum_{i=0}^{s} \binom{n}{i}, \sum_{i \notin L \bmod p} \binom{n}{i} \right\}.$$

Our proofs use two tools from linear algebra that are often useful in problems concerning set systems with restricted intersections: the original inclusion matrix method of Ray-Chaudhuri and Wilson [13] and the polynomial method as used by Alon, Babai, and Suzuki [1]. The *s^*-inclusion matrix* of a set system $\mathcal{A}$ on $[n]$ is a matrix M with rows indexed by $\mathcal{A}$ and columns by the subsets of $[n]$ of size at most s, where if $A \in \mathcal{A}$ and $B \subset [n]$ with $|B| \le s$, we define M_{AB} to be 1 if $B \subset A$ and 0 otherwise. A key ingredient of our proofs is a tight lower bound on the rank of M, which is interesting in its own right.

For $s \in \mathbb{N}_0 = \mathbb{N} \cup \{0\}$, we define functions $f_s : \mathbb{N}_0 \to \mathbb{N}_0$ as follows. For any s we let $f_s(0) = 0$. For any $a > 0$ we let $f_0(a) = 1$. Given $s, a > 0$, write $a = 2^t + c$, where $0 \le c < 2^t$. We define $f_s(a) = \sum_{i=0}^{s} \binom{t}{i} + f_{s-1}(c)$. (Here we let $\binom{t}{i}$ be equal to $\frac{t(t-1)\cdots(t-i+1)}{i!}$ for $t \ge i \ge 1$; for $t \ge 0$ we let $\binom{t}{0} = 1$ and for other values of t and i we set $\binom{t}{i} = 0$.) The following theorem shows that these functions give a tight lower bound for the rank of the s^*-inclusion matrix over any field.

THEOREM 1.5. *If $|\mathcal{A}| = a$ and M is the s^*-inclusion matrix of $\mathcal{A}$, then* $\operatorname{rank}(M) \ge f_s(a)$. *Furthermore, there is a set system $\mathcal{A}$ for which* $\operatorname{rank}(M) = f_s(a)$.

We say that a set system $\mathcal{A}$ is *s^*-independent* if the rows of its s^*-inclusion matrix are linearly independent. It is well known (see, e.g., [2]) that if $|\mathcal{A}| < 2^{s+1}$, then $\mathcal{A}$ is s^*-independent. In a combinatorial setting this fact was proved by Frankl and Pach [5] in the study of null t-designs; it can also be viewed as determining the minimum distance of the Reed–Muller codes (see [10] for background information on codes). One can deduce this statement immediately from the above theorem together with the observation that $f_s(a) = a$ for $a < 2^{s+1}$. This observation can be proved by induction as follows. As before, write $a = 2^t + c$, where $0 \le c < 2^t$. Since $t \le s$, we have $\sum_{i=0}^{s} \binom{t}{i} = 2^t$. Then as $c < 2^s$ we have $f_{s-1}(c) = c$ (by induction), and so $f_s(a) = 2^t + c = a$, as required.

The rest of this paper is organized as follows. In the next section we prove cross-intersecting versions of the oddtown theorem and the nonuniform Fischer inequality. These are special cases of our main theorems, but have the advantage that we can prove them for all n. We set up the linear algebra machinery in section 3 and prove Theorem 1.5. Section 4 contains the proofs of Theorems 1.1 and 1.2. In section 5 we sketch how the proofs may be adapted to give the modular Theorems 1.3 and 1.4. The final section contains some concluding remarks.

We use the following notation throughout the paper. Write $[n] = \{1, \dots, n\}$. The subsets of $[n]$ of size s are denoted by $[n]^{(s)}$, and those of size at most s are denoted by $[n]^{(\leq s)}$.

2. Warm-up. In this section we will prove a couple of special cases of our main results, both for illustrative purposes and because in these cases we do not need to impose the condition that n has to be sufficiently large. We recall the oddtown theorem of Berlekamp [3] (see also [2]), which is a special case of the modular Frankl–Ray-Chaudhuri–Wilson theorem. It states that if we have a collection of odd subsets of $[n]$ such that every pairwise intersection has even size, then we can have at most n sets in total. Equality can be achieved by the collection of all singleton sets, for example. We will prove the following cross-intersecting version.

THEOREM 2.1. *Suppose $\mathcal{A}_1, \dots, \mathcal{A}_k$ are set systems on $[n]$ each consisting of odd sets so that every pair of distinct sets A, B with $A \in \mathcal{A}_i$ and $B \in \mathcal{A}_j$ for some $i \neq j$ has intersection of even size. Then $\sum_{i=1}^k |\mathcal{A}_i| \leq \max\{kn, 2^{n-1}\}$.*

Proof. Let $\mathcal{A}$ be the subsets of $[n]$ that belong to at least two of the $\mathcal{A}_i$ and let $\mathcal{B}$ be those sets that belong to exactly one of the $\mathcal{A}_i$. Then for any $A \in \mathcal{A}$ and $B \in \mathcal{A} \cup \mathcal{B}$ with $A \neq B$ we have $|A \cap B|$ even. We use boldface letters to indicate the incidence vectors in $\mathbb{F}_2^n$ corresponding to subsets of $[n]$; i.e., if $A \subset [n]$, then $\mathbf{A}$ denotes the vector whose ith coordinate is 1 if $i \in A$ and 0 otherwise. Let $\mathbf{1}$ denote the vector with all coordinates equal to 1. Any $A \in \mathcal{A} \cup \mathcal{B}$ has odd size, i.e., $\mathbf{A} \cdot \mathbf{A} = 1$, and for any $A \in \mathcal{A}$ and $B \in \mathcal{A} \cup \mathcal{B}$ with $A \neq B$ we have $\mathbf{A} \cdot \mathbf{B} = 0$. The sets in $\mathcal{A}$ are linearly independent as vectors, for if $\sum_{A \in \mathcal{A}} c_A \mathbf{A} = 0$, then taking the inner product with $\mathbf{A}$ for any $A \in \mathcal{A}$, we get $c_A = 0$. (In particular $|\mathcal{A}| \leq n$.) The sets in $\mathcal{B}$ therefore satisfy $|\mathcal{A}|$ independent homogeneous linear constraints of the form $\mathbf{A} \cdot \mathbf{B} = 0$, as well as the inhomogeneous constraint $\mathbf{1} \cdot \mathbf{B} = 1$ (because they have odd size). If $|\mathcal{A}| = n$, then these constraints are inconsistent. Then $\mathcal{B}$ is empty and we have $\sum_{i=1}^k |\mathcal{A}_i| \leq k|\mathcal{A}| \leq kn$, so we are done. Otherwise the sets in $\mathcal{B}$ belong to an affine subspace of dimension $n - |\mathcal{A}| - 1$, so $|\mathcal{B}| \leq 2^{n-|\mathcal{A}|-1}$ and then $\sum_{i=1}^k |\mathcal{A}_i| \leq k|\mathcal{A}| + 2^{n-|\mathcal{A}|-1}$. It is easy to see that $k|\mathcal{A}| + 2^{n-|\mathcal{A}|-1}$ is a convex function of $|\mathcal{A}|$ (e.g., by differentiating twice), so as $0 \leq |\mathcal{A}| \leq n-1$, it is maximized at either $|\mathcal{A}| = 0$ or $|\mathcal{A}| = n-1$. Either way we have $\sum_{i=1}^k |\mathcal{A}_i| \leq \max\{kn, 2^{n-1}\}$, as required. □

It is clear from the proof that equality can occur only when either $\mathcal{A}$ or $\mathcal{B}$ is empty. In the first case every odd set appears in exactly one $\mathcal{A}_i$. In fact, one of the $\mathcal{A}_i$ contains all the odd sets, and the other $\mathcal{A}_j$ are empty (assuming that $n \geq 3$). To see this, note that the graph on the odd sets defined by joining sets with odd intersection is connected, so if there are two of the $\mathcal{A}_j$ that are nonempty, we would find an edge of the graph going from one to the other, which is impossible. In the second case $\mathcal{A}$ must be a system of n odd sets with all pairwise intersections of even size, and $\mathcal{A}_1 = \cdots = \mathcal{A}_k = \mathcal{A}$.

We will also prove the following cross-intersecting version of the nonuniform Fischer inequality.

THEOREM 2.2. *Suppose $\mathcal{A}_1, \ldots, \mathcal{A}_k$ are set systems on $[n]$ and there is some t so that for every pair of distinct sets A, B with $A \in \mathcal{A}_i$ and $B \in \mathcal{A}_j$ for some $i \neq j$, we have $|A \cap B| = t$. Then $\sum_{i=1}^k |\mathcal{A}_i| \leq \max\{k(n+1), k-1+2^n\}$. Moreover, if $t \neq 0$, then we have $\sum_{i=1}^k |\mathcal{A}_i| \leq \max\{kn, 2^n\}$.*

Proof. Let $\mathcal{A}$ be the subsets of $[n]$ that belong to at least two of the $\mathcal{A}_i$ and let $\mathcal{B}$ be those sets that belong to exactly one of the $\mathcal{A}_i$. Then for any $A \in \mathcal{A}$ and $B \in \mathcal{A} \cup \mathcal{B}$ with $A \neq B$ we have $|A \cap B| = t$.

We first consider the case when there is no set in $\mathcal{A}$ of size t. As in the previous proof we use boldface to denote incidence vectors of sets, which we now think of as belonging to $\mathbb{R}^n$. One can show that the vectors $\{\mathbf{A} : A \in \mathcal{A}\}$ are linearly independent. (This follows from the proof of the nonuniform Fischer inequality given in [2], which we briefly sketch. Let M be the matrix with rows equal to the vectors $\{\mathbf{A} : A \in \mathcal{A}\}$. Then MM^T is the $|\mathcal{A}|$ by $|\mathcal{A}|$ intersection matrix, which has each off-diagonal entry equal to t and each diagonal entry larger than t. It is not hard to show that any such matrix is nonsingular, and therefore M has rank $|\mathcal{A}|$, as required.) It also follows that $|\mathcal{A}| \leq n$.

Now for each $A \in \mathcal{A}$ we consider the linear form $f_A(x) = \mathbf{A} \cdot x - t$ in the variables $x = (x_1, \ldots, x_n)$. Then f_A vanishes on all the incidence vectors of members of $\mathcal{A} \cup \mathcal{B}$, except $\mathbf{A}$ itself. Since the incidence vectors of sets $B \in \mathcal{B}$ satisfy $|\mathcal{A}|$ independent constraints $f_A(\mathbf{B}) = 0$, they lie in the intersection of an affine space of dimension $n - |\mathcal{A}|$ with the cube $\{0,1\}^n$. It follows that $|\mathcal{B}| \leq 2^{n-|\mathcal{A}|}$. To see this, pick any $B_0 \in \mathcal{B}$ and consider the vectors $\{\mathbf{B} - \mathbf{B_0} \bmod 2 : B \in \mathcal{B}\}$ in $\mathbb{F}_2^n$. If there are more than $2^{n-|\mathcal{A}|}$ such vectors, then they span an $\mathbb{F}_2$-vector space of dimension at least $n - |\mathcal{A}| + 1$. It follows that the real vectors $\{\mathbf{B} - \mathbf{B_0} : B \in \mathcal{B}\}$ span a real vector space of dimension at least $n - |\mathcal{A}| + 1$ and satisfy $|\mathcal{A}|$ independent constraints, which is impossible. Therefore $\sum_{i=1}^k |\mathcal{A}_i| \leq k|\mathcal{A}| + 2^{n-|\mathcal{A}|} \leq \max\{kn, 2^n\}$ by convexity. This proves both parts of the theorem under the assumption that there is no set in $\mathcal{A}$ of size t.

Now suppose there is some $A_0 \in \mathcal{A}$ with $|A_0| = t$. Then all sets in $\mathcal{A} \cup \mathcal{B}$ contain A_0. Repeating the above argument, we see that the vectors $\{\mathbf{A} : A \in \mathcal{A} \backslash A_0\}$ are linearly independent, so $|\mathcal{A} \backslash A_0| \leq n$ and $|\mathcal{B}| \leq 2^{n-|\mathcal{A}\backslash A_0|}$. If $|\mathcal{A} \backslash A_0| = n$, then $\mathcal{B}$ must be empty. For if there is $B \in \mathcal{B}$, then $\mathcal{A} \cup B$ contains $n+2$ sets with all pairwise intersections having size t, which is impossible. In this case we have $\sum_{i=1}^k |\mathcal{A}_i| \leq k|\mathcal{A}| \leq k(n+1)$. In the case $\mathcal{A} = \{A_0\}$ we have $|\mathcal{B}| \leq 2^n - 1$ (since $A_0 \notin \mathcal{B}$) so $\sum_{i=1}^k |\mathcal{A}_i| \leq k|\mathcal{A}| + |\mathcal{B}| \leq k + 2^n - 1$. Otherwise we have $2 \leq |\mathcal{A}| \leq n$ and

$$\sum_{i=1}^k |\mathcal{A}_i| \leq k|\mathcal{A}| + 2^{n-|\mathcal{A}|+1} \leq \max\{2k + 2^{n-1}, kn + 2\}$$

by convexity. Now $kn + 2 \leq k(n+1)$ for $k \geq 2$, and if $2k + 2^{n-1} > k(n+1)$, we have $k < 2^{n-1}/(n-1)$, so $(k - 1 + 2^n) - (2k + 2^{n-1}) = 2^{n-1} - (k+1) \geq 0$. (We are ignoring the case $n = 1$, for which the theorem is trivially true.) We deduce that $\sum_{i=1}^k |\mathcal{A}_i| \leq \max\{k(n+1), k-1+2^n\}$, which is the first part of the theorem.

To get the improvement when $t \neq 0$, consider the set systems $\mathcal{A}_i' = \{A \backslash A_0 : A \in \mathcal{A}_i\}$. These are defined on a set of size $n - t$, and for every pair of distinct sets A, B

with $A \in \mathcal{A}'_i$ and $B \in \mathcal{A}'_j$ for some $i \neq j$ we have $|A \cap B| = 0$. By the first part of the theorem we have

$$\sum_{i=1}^{k} |\mathcal{A}_i| = \sum_{i=1}^{k} |\mathcal{A}'_i| \leq \max\{k(n-t+1), k-1+2^{n-t}\} \leq \max\{kn, 2^n\}$$

and we are done. □

3. Tools from linear algebra. This section contains the linear algebra components of our argument, which are a tight lower bound on the rank of the s^*-inclusion matrix and the polynomial method.

3.1. The rank of the inclusion matrix. For a set system $\mathcal{A}$ on $[n]$, the *s^*-inclusion matrix* M has rows indexed by $\mathcal{A}$ and columns indexed by the subsets of $[n]$ of size at most s (including the empty set), where if $A \in \mathcal{A}$ and $B \subset [n]$ with $|B| \leq s$ we define M_{AB} to be 1 if $B \subset A$ and 0 otherwise. In this subsection we will prove a tight lower bound for the rank of this matrix, which is of interest in its own right.

Let $V = \mathbb{F}^{\sum_{i=0}^{s} \binom{n}{i}}$, where $\mathbb{F}$ is some field, and denote its standard basis by e_Z, where Z ranges over subsets of $[n]$ of size at most s. Given a set $A \in \mathcal{A}$, we define the *s^*-inclusion vector*

$$v^s_A = \sum_{Z \subset A, |Z| \leq s} e_Z.$$

These are the row vectors of the s^*-inclusion matrix. We define $V^s_{\mathcal{A}}$ to be the row space, i.e., the subspace of V spanned by the vectors $\{v^s_A : A \in \mathcal{A}\}$. Note that the rank of the s^*-inclusion matrix is equal to the dimension of $V^s_{\mathcal{A}}$.

Throughout we adopt the following standard convention for binomial coefficients. We let $\binom{t}{i}$ be equal to $\frac{t(t-1)\cdots(t-i+1)}{i!}$ for $t \geq i \geq 1$; for $t \geq 0$ we let $\binom{t}{0} = 1$ and for other values of t and i we set $\binom{t}{i} = 0$. We will use the following well-known identities, which follow easily from the fact that $\binom{t+1}{s} = \binom{t}{s} + \binom{t}{s-1}$:

$$\sum_{i=0}^{s} \binom{t+1}{i} = \sum_{i=0}^{s} \binom{t}{i} + \sum_{i=0}^{s-1} \binom{t}{i}, \tag{1}$$

$$\binom{t+1}{s} = \sum_{i=0}^{s} \binom{t-i}{s-i}. \tag{2}$$

For $s \in \mathbb{N}_0 = \mathbb{N} \cup \{0\}$, we define functions $f_s : \mathbb{N}_0 \to \mathbb{N}_0$ as follows. For any s we let $f_s(0) = 0$. For any $a > 0$ we let $f_0(a) = 1$. Given $s, a > 0$, write $a = 2^t + c$, where $0 \leq c < 2^t$. We define $f_s(a) = \sum_{i=0}^{s} \binom{t}{i} + f_{s-1}(c)$. We will show that if $|\mathcal{A}| = a$, then $\dim V^s_{\mathcal{A}} \geq f_s(a)$. First we need some inequalities for the functions f_s.

LEMMA 3.1. *If $a < 2^t$, then $f_s(a) - f_{s-1}(a) \leq \binom{t}{s}$.*

Proof. Write $a = 2^{t_1} + 2^{t_2} + \cdots$, where $t > t_1 > t_2 > \cdots$. Then $t_i \leq t - i$. We have

$$f_s(a) - f_{s-1}(a) = \sum_{i \geq 1} \binom{t_i}{s+1-i} \leq \sum_{i \geq 1} \binom{t-i}{s+1-i} = \binom{t}{s},$$

where we use (2). □

LEMMA 3.2. *If $a \geq b$, then $f_s(a+b) \leq f_s(a) + f_{s-1}(b)$ for $s \geq 1$.*

Proof. We argue by induction on $a+b$ and s. Write $a = 2^t + c$, where $0 \leq c < 2^t$. First we check the base cases of the induction. The statement is trivial when $b = 0$, so we can suppose $b > 0$. When $s = 1$ we have two cases. First suppose that $c = 0$. Then $f_1(a) = t+1$. Since $0 < b \leq a = 2^t$ we have $2^t < a+b \leq 2^{t+1}$, so $f_1(a+b) = t+2$. Since $f_0(b) = 1$ we have $f_1(a+b) = f_1(a) + f_0(b)$. Next suppose that $c > 0$. Then $f_1(a) = t+2$. Since $b \leq a < 2^{t+1}$ we have $a+b < 2^{t+2}$, so $f_1(a+b) \leq t+3 = f_1(a) + f_0(b)$, as required.

Now suppose that $s > 1$ and that the statement is true with s replaced by $s' < s$ and also for the same s applied to a pair a', b' with $a'+b' < a+b$. Note in particular that for any $s' < s$ and any x, y we have $f_{s'}(x+y) \leq f_{s'}(x) + f_{s'}(y)$. For we may suppose that $x \geq y$, and then by induction $f_{s'}(x+y) \leq f_{s'}(x) + f_{s'-1}(y) \leq f_{s'}(x) + f_{s'}(y)$.

Consider first the case that $b < 2^t - c$. Then $a+b < 2^{t+1}$. We have $f_s(a) = \sum_{i=0}^{s} \binom{t}{i} + f_{s-1}(c)$ and $f_s(a+b) = \sum_{i=0}^{s} \binom{t}{i} + f_{s-1}(b+c)$, so $f_s(a) + f_{s-1}(b) - f_s(a+b) = f_{s-1}(b) + f_{s-1}(c) - f_{s-1}(b+c) \geq 0$, by the observation in the previous paragraph.

Next we consider the case that $b \geq 2^t$, say $b = 2^t + d$, where $0 \leq d \leq c < 2^t$. Then $f_{s-1}(b) = \sum_{i=0}^{s-1} \binom{t}{i} + f_{s-2}(d)$. Since $2^{t+1} \leq a+b < 2^{t+2}$ we have $f_s(a+b) = \sum_{i=0}^{s} \binom{t+1}{i} + f_{s-1}(c+d)$. Using (1) we get $f_s(a) + f_{s-1}(b) - f_s(a+b) = f_{s-1}(c) + f_{s-2}(d) - f_{s-1}(c+d) \geq 0$ by induction (since $c \geq d$).

Finally, we are left with the case $2^t - c \leq b < 2^t$. We have $2^{t+1} \leq a+b < 2^{t+2}$, so $f_s(a+b) = \sum_{i=0}^{s} \binom{t+1}{i} + f_{s-1}(b+c-2^t)$. Since $2^t \leq b+c < 2^{t+1}$ we have $f_s(b+c) = \sum_{i=0}^{s} \binom{t}{i} + f_{s-1}(b+c-2^t)$, so $f_s(a+b) - f_s(b+c) = \sum_{i=0}^{s} \binom{t+1}{i} - \sum_{i=0}^{s} \binom{t}{i} = \sum_{i=0}^{s-1} \binom{t}{i}$. Then $f_s(a) + f_{s-1}(b) - f_s(a+b) = \sum_{i=0}^{s} \binom{t}{i} + f_{s-1}(c) + f_{s-1}(b) - f_s(b+c) - \sum_{i=0}^{s-1} \binom{t}{i} = f_{s-1}(b) + f_{s-1}(c) - f_s(b+c) + \binom{t}{s}$. If $b \geq c$, then by Lemma 3.1 we have $f_{s-1}(b) + \binom{t}{s} \geq f_s(b)$, so $f_s(a) + f_{s-1}(b) - f_s(a+b) \geq f_s(b) + f_{s-1}(c) - f_s(b+c) \geq 0$ by induction (since $b+c < a+b$). Similarly, if $c \geq b$ we have $f_s(a) + f_{s-1}(b) - f_s(a+b) \geq f_s(c) + f_{s-1}(b) - f_s(b+c) \geq 0$. In all cases we are done. ∎

Proof of Theorem 1.5. We argue by induction on a and s. The result is trivial if $a = 0$, $a = 1$, or $s = 0$, so we suppose that $a \geq 2$ and $s \geq 1$. Let $\mathcal{A}$ be a set system on a set X with $|\mathcal{A}| = a$. Pick $x \in X$ and let $\mathcal{A}_x = \{A \in \mathcal{A} : x \in A\}$, $\mathcal{A}_{\overline{x}} = \mathcal{A} \backslash \mathcal{A}_x$. Write $a_x = |\mathcal{A}_x|$ and $a_{\overline{x}} = |\mathcal{A}_{\overline{x}}|$. We can choose x so that $0 < a_x, a_{\overline{x}} < a$.

Let M be the matrix whose rows consist of the s^*-inclusion vectors of sets in $\mathcal{A}$, with the following order of rows and columns. The rows are ordered in such a way that those corresponding to sets in $\mathcal{A}_{\overline{x}}$ precede those in $\mathcal{A}_x$. The columns are ordered into three groups; the first group is those columns given by entries in the s^*-inclusion vectors corresponding to sets in $X^{(\leq s-1)}$ not containing x, the second group is those corresponding to sets in $X^{(s)}$ not containing x, and the third group is those corresponding to sets in $X^{(\leq s)}$ that contain x; each of the three groups is ordered lexicographically. Thus M has the structure

$$\begin{pmatrix} M_1 & M_2 & 0 \\ M_3 & M_4 & M_3 \end{pmatrix}$$

for some matrices M_1, M_2, M_3, M_4. Note that $rk(M) = \dim V_{\mathcal{A}}^s$.

Consider the system $\mathcal{A}' = \{A \Delta \{x\} : A \in \mathcal{A}\}$, where Δ denotes symmetric difference. Since $\mathcal{A}'_x = \{A \cup \{x\} : A \in \mathcal{A}_{\overline{x}}\}$ and $\mathcal{A}'_{\overline{x}} = \{A \backslash \{x\} : A \in \mathcal{A}_x\}$, the matrix corresponding to $\mathcal{A}'$ (with respect to the same order on rows and columns) is

$$M' = \begin{pmatrix} M_3 & M_4 & 0 \\ M_1 & M_2 & M_1 \end{pmatrix}.$$

Note that M' can be obtained from M by row and column operations. In terms of the block structure, we swap the two rows, subtract the first column from the third column, then multiply the third column by -1. This shows that $rk(M') = rk(M)$, i.e., $\dim V^s_{\mathcal{A}'} = \dim V^s_{\mathcal{A}}$. Therefore, we can suppose without loss of generality that $a_{\overline{x}} \geq a_x$.

Now note that

$$\dim V^s_{\mathcal{A}} = rk(M) \geq rk\Big(M_1 \ M_2 \Big) + rk(M_3) = \dim V^s_{\mathcal{A}_{\overline{x}}} + \dim V^{s-1}_{\mathcal{A}_x}.$$

Since $0 < a_x, a_{\overline{x}} < a$ we can apply induction to get $\dim V^s_{\mathcal{A}_{\overline{x}}} \geq f_s(a_{\overline{x}})$ and $\dim V^{s-1}_{\mathcal{A}_x} \geq f_{s-1}(a_x)$. Since $a_{\overline{x}} \geq a_x$ and $a_{\overline{x}} + a_x = a$, by Lemma 3.2 we have $\dim V^s_{\mathcal{A}} \geq f_s(a_{\overline{x}}) + f_{s-1}(a_x) \geq f_s(a)$. This proves the first part of the theorem.

Finally we note that the bound on dimension is tight. To show this, we prove by induction on a and s that if $a = 2^t + c$, with $c < 2^t$, then there is a set system $\mathcal{A}$ on $[t+1]$ with $|\mathcal{A}| = a$ and $\dim V^s_{\mathcal{A}} = f_s(a)$. This is clear if $a = 1$ or if $s = 0$, so we suppose $a \geq 2$ and $s \geq 1$. By induction we can find a set system $\mathcal{B}$ on $[t]$ with c sets so that $\dim V^{s-1}_{\mathcal{B}} = f_{s-1}(c)$. Let $\mathcal{B}' = \big\{B \cup \{t+1\} \,:\, B \in \mathcal{B}\big\}$ and $\mathcal{A} = \mathcal{P}([t]) \cup \mathcal{B}'$, i.e., $\mathcal{A}$ consists of all subsets of $[t]$ together with each set of $\mathcal{B}$ with the element $t+1$ added. In the s^*-inclusion matrix for $\mathcal{A}$, the block with rows corresponding to $\mathcal{P}([t])$ and columns corresponding to $[t]^{(\leq s)}$ has full rank $\sum_{i=0}^s \binom{t}{i}$. Any extra rank in the matrix can come only from the block with rows corresponding to $\mathcal{B}'$ and columns corresponding to $\big\{X \cup \{t+1\} \,:\, X \in [t]^{(\leq s-1)}\big\}$, and this has rank $\dim V^{s-1}_{\mathcal{B}} = f_{s-1}(c)$. Therefore $\dim V^s_{\mathcal{A}} = \sum_{i=0}^s \binom{t}{i} + f_{s-1}(c) = f_s(a)$, as required. □

We note the following properties of the function $f_s(a)$ for future reference:

$$f_s(a) \geq \sum_{i=0}^{s} \binom{\lfloor \log_2 a \rfloor}{i}; \tag{3}$$

$$\text{If } 2^n - 2^{n-s} < a \leq 2^n, \text{ then } f_s(a) = f_s(2^n) = \sum_{i=0}^{s} \binom{n}{i}. \tag{4}$$

To see the second property note that we can write $a = 2^{n-1} + 2^{n-2} + \cdots + 2^{n-s} + b$, with $0 < b < 2^{n-s}$, and so $f_s(a) = \sum_{j=0}^s \sum_{i \geq 0} \binom{n-1-j}{s-j-i} = \sum_{i \geq 0} \binom{n}{s-i}$, where we use (2).

3.2. The polynomial method. In this subsection we summarize the particular application of the polynomial method that we need in the following lemma.

LEMMA 3.3. (i) *Suppose $\mathcal{A}$ is an L-intersecting family of sets and that $|A| \notin L$ for all $A \in \mathcal{A}$. Then the s^*-inclusion vectors $\{v^s_A : A \in \mathcal{A}\}$ are linearly independent over $\mathbb{R}$.*

(ii) *Suppose also that $\mathcal{B}$ is a set system such that $|A \cap B| \in L$ for any $A \in \mathcal{A}$ and $B \in \mathcal{B}$. Then no vector v^s_B with $B \in \mathcal{B}$ lies in $V^s_{\mathcal{A}}$.*

Proof. We use boldface to denote the incidence vector corresponding to a subset of $[n]$. For a set A we define the polynomial $f_A(x) = \prod_{\ell \in L}(x \cdot \mathbf{A} - \ell)$. We will restrict $x = (x_1, \ldots, x_n)$ to range over $\{0,1\}$-vectors, so by repeatedly replacing any occurrence of x_i^2 by x_i we can represent $f_A(x)$ by a multilinear polynomial $\sum_{X \in [n]^{(\leq s)}} c_{A,X} \prod_{i \in X} x_i$. Let $w_A = \sum_{X \in [n]^{(\leq s)}} c_{A,X} e_X$, where we recall that $\{e_X : X \in [n]^{\leq s}\}$ denotes the standard basis of $V = \mathbb{R}^{\sum_{i=0}^s \binom{n}{i}}$. Then by definition we have $f_A(\mathbf{B}) = w_A \cdot v^s_B$.

Note that $f_A(\mathbf{B}) = 0$ if and only if $|A \cap B| \in L$, so for $A, B \in \mathcal{A}$ we have $w_A \cdot v^s_B = f_A(\mathbf{B}) = 0$ if and only if $A \neq B$. Now if $\sum_{A \in \mathcal{A}} t_A v^s_A = 0$, then taking the

inner product of this identity with w_A for each $A \in \mathcal{A}$ we obtain that $t_A = 0$ for every A, which proves part (i) of the lemma. Also, if $B \in \mathcal{B}$ and $v_B^s = \sum_{A\in\mathcal{A}} t_A v_A^s$, then taking the inner product with w_A, we again see that $t_A = 0$ for each $A \in \mathcal{A}$. This gives $v_B^s = \mathbf{0}$, a contradiction that proves part (ii) of the lemma. □

The same proof shows that this result holds with $\mathbb{R}$ replaced by the field with p elements (for some prime p) provided that $|A| \notin L \bmod p$ for all $A \in \mathcal{A}$.

4. Proofs of the main theorems. We start by proving Theorem 1.1, which we recall states that if $n > 100s^2 \log(s+1)$ and $\mathcal{A}_1, \dots, \mathcal{A}_k$ are uniform set systems on $[n]$ that are L-cross-intersecting, then $\sum_{i=1}^k |\mathcal{A}_i| \le \max\{k\binom{n}{s}, \binom{n}{\lfloor n/2\rfloor}\}$. First we need the following estimate on the middle binomial coefficients.

LEMMA 4.1. $\frac{2^n}{2\sqrt{n}} \le \binom{n}{\lfloor n/2\rfloor} \le \frac{2^n}{\sqrt{n}}$.

Proof. Let $g(n) = 2^{-n}\sqrt{n}\binom{n}{\lfloor n/2\rfloor}$. We want to prove that $1/2 \le g(n) \le 1$. This is easily verified for $n = 1$ and $n = 2$. We see that $g(n+2) > g(n)$: for even n we have $\frac{g(2m)}{g(2m-2)} = \left(1 - \frac{1}{2m}\right)\sqrt{\frac{m}{m-1}} > 1$, as $\left(1 - \frac{1}{2m}\right)^2 - \frac{m-1}{m} = \frac{1}{4m^2} > 0$, and for odd n we have $\frac{g(2m+1)}{g(2m-1)} = (1-\frac{1}{2(m+1)})\sqrt{\frac{2m+1}{2m-1}} > 1$, as $(1-\frac{1}{2(m+1)})^2 - \frac{2m-1}{2m+1} > \frac{1}{4(m+1)^2} > 0$. Now $g(n) \ge 1/2$ follows for all n by induction. For the upper bound we use the Stirling approximation $n! \sim \sqrt{2\pi n}\, n^n e^{-n}$, from which it follows that $g(n) \to \sqrt{2/\pi}$ as $n \to \infty$. Since $g(2m)$ and $g(2m+1)$ are increasing sequences we have $g(n) \le \sqrt{2/\pi} < 1$. □

Proof of Theorem 1.1. Let $k_c = \lfloor \binom{n}{\lfloor n/2\rfloor} / \binom{n}{s} \rfloor$. Then for $k \le k_c$ we want to show that $\sum_{i=1}^k |\mathcal{A}_i| \le \binom{n}{\lfloor n/2\rfloor}$ and for $k > k_c$ we want to show that $\sum_{i=1}^k |\mathcal{A}_i| \le k\binom{n}{s}$. Note that it suffices to prove these two statements in the specific cases $k = k_c$ and $k = k_c+1$. Then the case $k = k_c$ clearly implies that for $k \le k_c$ we have $\sum_{i=1}^k |\mathcal{A}_i| \le \binom{n}{\lfloor n/2\rfloor}$. Also the case $k > k_c + 1$ follows by induction. If we ignore the smallest $\mathcal{A}_i$ we are left with $k-1$ L-cross-intersecting set systems, which have total size at most $(k-1)\binom{n}{s}$, so the total size of all k systems is at most $\frac{k}{k-1} \cdot (k-1)\binom{n}{s} = k\binom{n}{s}$.

By the above remark we can assume that $k = k_c$ or $k = k_c + 1$. Suppose that $\mathcal{A}_1, \dots, \mathcal{A}_k$ are L-cross-intersecting r-uniform set systems with $\sum_{i=1}^k |\mathcal{A}_i| \ge \max\{k\binom{n}{s}, \binom{n}{\lfloor n/2\rfloor}\}$. Note that we can assume $r \notin L$. Let $\mathcal{A}$ be the subsets of $[n]$ that belong to at least two of the $\mathcal{A}_i$ and let $\mathcal{B}$ be those subsets that belong to exactly one of the $\mathcal{A}_i$. Since the $\mathcal{A}_i$ are L-cross-intersecting, for any $A \in \mathcal{A}$ and $B \in \mathcal{A}\cup\mathcal{B}$ we have $|A \cap B| \in L$. It follows from the Ray-Chaudhuri–Wilson theorem that $|\mathcal{A}| \le \binom{n}{s}$, and if $\mathcal{B} \ne \emptyset$, then $|\mathcal{A}| < \binom{n}{s}$ (as we can add one set from $\mathcal{B}$ to $\mathcal{A}$ and still have an L-intersecting family). From Lemma 3.3 we know that the s^*-inclusion vectors $\{v_A^s : A \in \mathcal{A}\}$ are linearly independent over $\mathbb{R}$, i.e., they form a basis of $V_{\mathcal{A}}^s$, and we also see that no vector v_B^s with $B \in \mathcal{B}$ lies in $V_{\mathcal{A}}^s$. We conclude that

$$|\mathcal{A}| + \dim V_{\mathcal{B}}^s \le \sum_{i=0}^{s} \binom{n}{i}. \tag{5}$$

Note that we can assume that both $\mathcal{A}$ and $\mathcal{B}$ are nonempty. For if $\mathcal{A} = \emptyset$ we have $\sum_{i=1}^k |\mathcal{A}_i| \le |\mathcal{B}| \le \binom{n}{r} \le \binom{n}{\lfloor n/2\rfloor}$ and if $\mathcal{B} = \emptyset$ we have $\sum_{i=1}^k |\mathcal{A}_i| \le k|\mathcal{A}| \le k\binom{n}{s}$; in either case we are done. Thus we cannot have $|\mathcal{A}| = \binom{n}{s}$ (for then $\mathcal{B} = \emptyset$), so we have $|\mathcal{A}| \le \binom{n}{s} - 1$. Since

$$k\binom{n}{s} \le \sum_{i=1}^{k} |\mathcal{A}_i| \le k|\mathcal{A}| + |\mathcal{B}| \le k\left(\binom{n}{s} - 1\right) + |\mathcal{B}|,$$

we have $|\mathcal{B}| \geq k > \left(\binom{n}{\lfloor n/2 \rfloor}/\binom{n}{s}\right) - 1$. By Lemma 4.1 we have $|\mathcal{B}| > 2^n/n^{s+1}$, so by Theorem 1.5 and (3)

$$\dim V^s_{\mathcal{B}} \geq f_s(|\mathcal{B}|) \geq \sum_{i=0}^{s} \binom{\lfloor \log_2 |\mathcal{B}| \rfloor}{i} \geq \sum_{i=0}^{s} \binom{\lfloor n - (s+1)\log_2 n \rfloor}{i}.$$

Now from (5) we get

$$\begin{aligned} |\mathcal{A}| &\leq \sum_{i=0}^{s} \binom{n}{i} - \dim V^s_{\mathcal{B}} \leq \sum_{i=0}^{s} \left(\binom{n}{i} - \binom{n - \lceil (s+1)\log_2 n \rceil}{i} \right) \\ &\leq \lceil (s+1)\log_2 n \rceil \sum_{i=0}^{s-1} \binom{n-1}{i}, \end{aligned}$$

where we use the inequality $\binom{n}{i} - \binom{n-t}{i} = \sum_{j=1}^{t} \left(\binom{n+1-j}{i} - \binom{n-j}{i} \right) = \sum_{j=1}^{t} \binom{n-j}{i-1} \leq t\binom{n-1}{i-1}$. Therefore

$$\begin{aligned} |\mathcal{B}| &\geq \sum_{i=1}^{k} |\mathcal{A}_i| - k|\mathcal{A}| \geq \binom{n}{\lfloor n/2 \rfloor} - \left(\frac{\binom{n}{\lfloor n/2 \rfloor}}{\binom{n}{s}} + 1 \right) \lceil (s+1)\log_2 n \rceil \sum_{i=0}^{s-1} \binom{n-1}{i} \\ &> \left(1 - \frac{3s(s+1)\log_2 n}{2n} \right) \binom{n}{\lfloor n/2 \rfloor}. \end{aligned} \tag{6}$$

In particular we easily see that $|\mathcal{B}| > \binom{n}{\lfloor n/3 \rfloor}$, so $n/3 < r < 2n/3$. Recalling that $\mathcal{A} \neq \emptyset$, we now consider any $A \in \mathcal{A}$. For any $B \in \mathcal{B}$ the size of its intersection with A belongs to L, so we get

$$\begin{aligned} |\mathcal{B}| &\leq \sum_{\ell \in L} \binom{r}{\ell}\binom{n-r}{r-\ell} \leq s \binom{r}{\lfloor r/2 \rfloor}\binom{n-r}{\lfloor (n-r)/2 \rfloor} \\ &< s \cdot \frac{2^r}{\sqrt{r}} \cdot \frac{2^{n-r}}{\sqrt{n-r}} < \frac{2^n s}{n/3} < \frac{6s}{\sqrt{n}} \binom{n}{\lfloor n/2 \rfloor}, \end{aligned}$$

where we use Lemma 4.1. Comparing with (6) we get

$$\frac{6s}{\sqrt{n}} > |\mathcal{B}| / \binom{n}{\lfloor n/2 \rfloor} > 1 - \frac{3s(s+1)\log_2 n}{2n}.$$

Since $n > 100s^2 \log(s+1)$, this gives the required contradiction. □

It is clear from the proof that equality can occur only when either $\mathcal{A}$ or $\mathcal{B}$ is empty. In the first case every set of size $\lfloor n/2 \rfloor$ appears in exactly one $\mathcal{A}_i$. In fact, one of the $\mathcal{A}_i$ contains all the sets of size $\lfloor n/2 \rfloor$, and the other $\mathcal{A}_j$ are empty (which can be proved as in the remark after Theorem 2.1). In the second case $\mathcal{A}$ must be a maximum uniform L-intersecting family, and $\mathcal{A}_1 = \cdots = \mathcal{A}_k = \mathcal{A}$.

Next we prove Theorem 1.2, which we recall states that if $n > 100s^2 \log s$ and $\mathcal{A}_1, \ldots, \mathcal{A}_k$ are set systems on $[n]$ that are L-cross-intersecting, then $\sum_{i=1}^{k} |\mathcal{A}_i| \leq \max\{k\sum_{i=0}^{s}\binom{n}{i}, (k-1)\sum_{i=0}^{s-1}\binom{n}{i} + 2^n\}$.

Proof of Theorem 1.2. We will assume that $s > 1$, as the case $s = 1$ is covered by Theorem 2.2. Suppose that $\mathcal{A}_1, \ldots, \mathcal{A}_k$ are L-cross-intersecting set systems with

$$\sum_{i=1}^{k} |\mathcal{A}_i| \geq \max\left\{ k\sum_{i=0}^{s}\binom{n}{i}, (k-1)\sum_{i=0}^{s-1}\binom{n}{i} + 2^n \right\}.$$

Let $\mathcal{A}$ be the sets that belong to at least two of the $\mathcal{A}_i$ and let $\mathcal{B}$ be those sets that belong to exactly one of the $\mathcal{A}_i$.

Write $k_c = \lfloor \frac{2^n - \sum_{i=0}^{s-1}\binom{n}{i}}{\binom{n}{s}} \rfloor$. Then for $k \le k_c$ we want to show that $\sum_{i=1}^k |\mathcal{A}_i| \le (k-1)\sum_{i=0}^{s-1}\binom{n}{i} + 2^n$ and for $k > k_c$ we want to show that $\sum_{i=1}^k |\mathcal{A}_i| \le k\sum_{i=0}^{s}\binom{n}{i}$. Note that it suffices to prove these two statements in the specific cases $k = k_c$ and $k = k_c + 1$. As for Theorem 1.1, the case $k > k_c + 1$ follows by induction. We can prove the case $k < k_c$ by induction on k (decreasing from k_c) with the following argument. Suppose for the sake of contradiction that we have $\sum_{i=1}^k |\mathcal{A}_i| > (k-1)\sum_{i=0}^{s-1}\binom{n}{i} + 2^n$. Then clearly $|\mathcal{A}| > \sum_{i=0}^{s-1}\binom{n}{i}$. Let $\mathcal{A}_{k+1} = \mathcal{A}$. Then $\mathcal{A}_1, \dots, \mathcal{A}_{k+1}$ are L-cross-intersecting and $\sum_{i=1}^{k+1} |\mathcal{A}_i| > k\sum_{i=0}^{s-1}\binom{n}{i} + 2^n$, which contradicts our induction hypothesis. Therefore we can assume that $k = k_c$ or $k = k_c + 1$.

Since $\mathcal{A}$ is L-intersecting we have $|\mathcal{A}| \le \sum_{i=0}^{s}\binom{n}{i}$ by the Frankl–Wilson theorem, and if $\mathcal{B} \ne \emptyset$, then $|\mathcal{A}| < \sum_{i=0}^{s}\binom{n}{i}$ (similar to the previous theorem). Let $\mathcal{A}_L = \{A \in \mathcal{A} : |A| \in L\}$ and $\mathcal{A}_{\overline{L}} = \{A \in \mathcal{A} : |A| \notin L\}$. Let ℓ be the largest element of L. Then $\mathcal{A}_L$ is $(L\backslash\ell)$-intersecting, so $|\mathcal{A}_L| \le \sum_{i=0}^{s-1}\binom{n}{i}$. Exactly as in the proof of Theorem 1.1 we see that the s^*-inclusion vectors $\{v_A^s : A \in \mathcal{A}_{\overline{L}}\}$ form a basis of $V^s_{\mathcal{A}_{\overline{L}}}$ and no vector v_B^s with $B \in \mathcal{B}$ lies in $V^s_{\mathcal{A}_{\overline{L}}}$. This shows that $|\mathcal{A}_{\overline{L}}| + \dim V^s_{\mathcal{B}} \le \sum_{i=0}^{s}\binom{n}{i}$.

We can assume that $\mathcal{B}$ is nonempty, for otherwise $\sum_{i=1}^k |\mathcal{A}_i| \le k|\mathcal{A}| \le k\sum_{i=0}^{s}\binom{n}{i}$, and we are done. We can also assume that $\mathcal{A}_{\overline{L}}$ is nonempty, for otherwise we have $|\mathcal{A}| = |\mathcal{A}_{\mathcal{L}}| \le \sum_{i=0}^{s-1}\binom{n}{i}$ and so

$$\sum_{i=1}^{k} |\mathcal{A}_i| \le k|\mathcal{A}| + |\mathcal{B}| \le k|\mathcal{A}| + (2^n - |\mathcal{A}|) \le (k-1)\sum_{i=0}^{s-1}\binom{n}{i} + 2^n,$$

and again we are done. We cannot have $|\mathcal{A}| = \sum_{i=0}^{s}\binom{n}{i}$ (for then $\mathcal{B} = \emptyset$) so we have $|\mathcal{A}| \le \sum_{i=0}^{s}\binom{n}{i} - 1$. It follows that $|\mathcal{B}| \ge k > \frac{2^n}{\binom{n}{s}} - 2 > \frac{2^n}{n^s}$, and so by Theorem 1.5 $\dim V^s_{\mathcal{B}} > \sum_{i=0}^{s}\binom{\lfloor n - s\log_2 n\rfloor}{i}$. Now we get

$$\begin{aligned}|\mathcal{A}_{\overline{L}}| &\le \sum_{i=0}^{s}\binom{n}{i} - \dim V^s_{\mathcal{B}} < \sum_{i=0}^{s}\left(\binom{n}{i} - \binom{n - \lceil s\log_2 n\rceil}{i}\right)\\ &\le \lceil s\log_2 n\rceil \sum_{i=0}^{s}\binom{n-1}{i-1} < \frac{2s^2\log_2 n}{n}\binom{n}{s}.\end{aligned}$$

Choose an integer t so that $2^{-(t+1)} \le |\mathcal{A}_{\overline{L}}|/\binom{n}{s} \le 2^{-t}$. Since $n \ge 100s^2\log s$ and $s \ge 2$, from the above inequality we have $t \ge 2$. Also, since $\mathcal{A}_{\overline{L}}$ is nonempty we have $t \le \log_2\binom{n}{s} < s\log n$.

Since $|\mathcal{A}| = |\mathcal{A}_{\overline{L}}| + |\mathcal{A}_L| \le 2^{-t}\binom{n}{s} + \sum_{i=0}^{s-1}\binom{n}{i}$ we see that

$$\begin{aligned}|\mathcal{B}| &\ge \sum_{i=1}^{k}|\mathcal{A}_i| - k|\mathcal{A}| > (k-1)\sum_{i=0}^{s-1}\binom{n}{i} + 2^n - k\left(2^{-t}\binom{n}{s} + \sum_{i=0}^{s-1}\binom{n}{i}\right)\\ &> 2^n - \sum_{i=0}^{s-1}\binom{n}{i} - \left(\frac{2^n}{\binom{n}{s}} + 1\right)2^{-t}\binom{n}{s} > 2^n - 2^{n-t+1},\end{aligned}$$

where for the last inequality we use the upper bound on t. We cannot have $t \ge s+1$, for then (4) gives $\dim V^s_{\mathcal{B}} = \sum_{i=0}^{s}\binom{n}{i}$ and then $\mathcal{A}_{\overline{L}}$ must be empty, which is a

contradiction. We deduce that $t \le s$. Now by Theorem 1.5 and (2) we have

$$\begin{aligned}\dim V^s_{\mathcal{B}} > f_s(2^n - 2^{n-t+1}) \;&=\; f_s(2^{n-1} + 2^{n-2} + \cdots + 2^{n-t+1}) \\ &= \sum_{j=0}^{t-2} \sum_{i \ge 0} \binom{n-1-j}{s-j-i} \\ &= \sum_{i \ge 0} \left(\sum_{j \ge 0} \binom{n-1-j}{s-i-j} - \sum_{j \ge 0} \binom{n-t-j}{s-i-t+1-j} \right) \\ &= \sum_{i \ge 0} \left(\binom{n}{s-i} - \binom{n-t+1}{s-i-t+1} \right).\end{aligned}$$

Therefore

$$2^{-(t+1)} \binom{n}{s} \le |\mathcal{A}_{\overline{L}}| \le \sum_{i=0}^{s} \binom{n}{i} - \dim V^s_{\mathcal{B}} \le \sum_{i \ge 0} \binom{n-t+1}{s-i-t+1} \le 2 \binom{n-t+1}{s-t+1}.$$

We deduce that $2^{t+2} \ge \binom{n}{s} / \binom{n-t+1}{s-t+1} \ge (n/s)^{t-1}$, and so $n/s \le 2^{1+3/(t-1)} \le 16$, which gives the required contradiction. □

From the proof we see that equality can occur only when either $\mathcal{B}$ or $\mathcal{A}_{\overline{L}}$ is empty. In the first case we have $\mathcal{A}_i = \mathcal{A}$ equal to an L-intersecting family of size $\sum_{i=0}^{s} \binom{n}{i}$. It was shown by Qian and Ray-Chaudhuri [12] that this is only possible when $L = \{0, 1, \ldots, s-1\}$ and $\mathcal{A} = [n]^{(\le s)}$. In the second case $|\mathcal{B}| = 2^n$ and $\mathcal{A} = \mathcal{A}_L$ must have size $\sum_{i=0}^{s-1} \binom{n}{i}$ and be $(L \backslash \ell)$-intersecting (where ℓ is the largest element of L), so again using the result of [12] we must have $L \backslash \ell = \{0, 1, \ldots, s-2\}$ and $\mathcal{A} = [n]^{(\le s-1)}$. Therefore one of the $\mathcal{A}_i$ contains all subsets of $[n]$, and the others are all equal to $[n]^{(\le s-1)}$.

5. The modular versions. The modular versions of the theorems proved in the last section have very similar proofs. The main ideas are the same, but the computations are significantly different and more involved, so we feel obliged to present them separately. We will be brief on those points of similarity to avoid excessive repetition, and we make no effort to obtain a bound on the smallest n for which the results hold. This section may be omitted on a first reading of this paper.

First we recall the statement of Theorem 1.3. Suppose p is prime, let L be a set of $s < p$ residues modulo p, and let $\mathcal{A}_1, \ldots, \mathcal{A}_k$ be set systems on $[n]$ that are L-cross-intersecting mod p such that every set $A \in \bigcup_{i=1}^{k} \mathcal{A}_i$ has $|A| = r$, for some $r \notin L \bmod p$. Let m be chosen so that $m \notin L \bmod p$ and $|n/2 - m|$ is as small as possible. The theorem claims that for $n > n(s)$ sufficiently large $\sum_{i=1}^{k} |\mathcal{A}_i| \le \max\left\{k\binom{n}{s}, \binom{n}{m}\right\}$.

We define all vectors and polynomials over $\mathbb{F}_p$ (the field with p elements) instead of $\mathbb{R}$.

Proof of Theorem 1.3. Let $k_c = \left\lfloor \binom{n}{m} / \binom{n}{s} \right\rfloor$. We can assume that $k = k_c$ or $k = k_c + 1$. Suppose that $\mathcal{A}_1, \ldots, \mathcal{A}_k$ are set systems that are L-cross-intersecting mod p such that every set $A \in \bigcup_{i=1}^{k} \mathcal{A}_i$ has $|A| = r$, for some $r \notin L \bmod p$, and suppose that $\sum_{i=1}^{k} |\mathcal{A}_i| \ge \max\left\{k\binom{n}{s}, \binom{n}{m}\right\}$. Let $\mathcal{A}$ be the subsets of $[n]$ that belong to at least two of the $\mathcal{A}_i$ and let $\mathcal{B}$ be those sets that belong to exactly one of the $\mathcal{A}_i$.

Since the $\mathcal{A}_i$ are L-cross-intersecting mod p, for any $A \in \mathcal{A}$ and $B \in \mathcal{A} \cup \mathcal{B}$ we have $|A \cap B| \in L \bmod p$. It follows from the modular Frankl–Wilson theorem that $|\mathcal{A}| \le \binom{n}{s}$, and if $\mathcal{B} \neq \emptyset$, then $|\mathcal{A}| < \binom{n}{s}$. From the remark after Lemma 3.3 we know that the s^*-inclusion vectors $\{v_A^s : A \in \mathcal{A}\}$ form a basis of the $\mathbb{F}_p$-vector space $V_{\mathcal{A}}^s$, and we also see that no vector v_B^s with $B \in \mathcal{B}$ lies in $V_{\mathcal{A}}^s$. We conclude that $|\mathcal{A}| + \dim V_{\mathcal{B}}^s \le \sum_{i=0}^s \binom{n}{i}$.

We can assume that both $\mathcal{A}$ and $\mathcal{B}$ are nonempty. Then $|\mathcal{A}| \le \binom{n}{s} - 1$, so $|\mathcal{B}| \ge k > \big(\binom{n}{m}/\binom{n}{s}\big) - 1$. By definition of m we have $|m - n/2| \le s$ so $\binom{n}{m} = (1 + o(1))\binom{n}{\lfloor n/2 \rfloor}$ and then by Lemma 4.1 we have $|\mathcal{B}| > 2^n/n^{s+1}$. Following the proof of Theorem 1.1 we get the inequalities $\dim V_{\mathcal{B}}^s \ge \sum_{i=0}^s \binom{\lfloor n-(s+1)\log_2 n \rfloor}{i}$ and $|\mathcal{A}| \le \lceil (s+1)\log_2 n \rceil \sum_{i=0}^{s-1} \binom{n-1}{i}$. Then

$$|\mathcal{B}| \ge \sum_{i=1}^k |\mathcal{A}_i| - k|\mathcal{A}| \ge \binom{n}{m} - \left(\frac{\binom{n}{m}}{\binom{n}{s}} + 1\right) \lceil (s+1)\log_2 n \rceil \sum_{i=0}^{s-1} \binom{n-1}{i}$$

$$> \left(1 - \frac{3s(s+1)\log_2 n}{2n}\right)\binom{n}{m}. \tag{7}$$

In particular $|\mathcal{B}| = (1 + o(1))\binom{n}{\lfloor n/2 \rfloor}$ so $|r - n/2| = o(\sqrt{n})$. Recalling that $\mathcal{A} \neq \emptyset$, we now consider any $A \in \mathcal{A}$. For any $B \in \mathcal{B}$ the size of its intersection with A belongs to $L \bmod p$. We can choose x so that $|x - r/2| = o(\sqrt{n})$ and $x \notin L \bmod p$. Any set of size r which intersects A in x points cannot belong to $\mathcal{B}$, and there are at least $\binom{r}{x}\binom{n-r}{r-x}$ of these. Now $\binom{r}{x} = (1 + o(1))\binom{r}{\lfloor r/2 \rfloor}$ and $r - x = (n - r)/2 + o(\sqrt{n})$, so $\binom{n-r}{r-x} = (1 + o(1))\binom{n-r}{\lfloor (n-r)/2 \rfloor}$. Therefore we can choose n large enough that $\binom{r}{x} > 2^r/3\sqrt{r}$ and $\binom{n-r}{r-x} > 2^{n-r}/3\sqrt{n-r}$. We deduce that $|\mathcal{B}| < \binom{n}{m} - 2^n/9n < (1 - 1/10\sqrt{n})\binom{n}{m}$. For $n > n(s)$ sufficiently large this contradicts (7), which completes the proof. □

Next we recall the statement of Theorem 1.4. Suppose p is prime, L is a set of $s < p$ nonnegative integers, and $\mathcal{A}_1, \dots, \mathcal{A}_k$ are set systems on $[n]$ that are L-cross-intersecting mod p such that every set $A \in \bigcup_{i=1}^k \mathcal{A}_i$ has $|A| \notin L \bmod p$. The theorem claims that for $n > n(s)$ sufficiently large

$$\sum_{i=1}^k |\mathcal{A}_i| \le \max\left\{ k\sum_{i=0}^s \binom{n}{i}, \sum_{i \notin L \bmod p} \binom{n}{i} \right\}.$$

First we need the following lemma.

LEMMA 5.1. *Suppose $|L| = s$ and $x > x(s)$ is sufficiently large. Then*

$$\sum_{i \notin L \bmod p} \binom{x}{i} > \frac{2^x}{3s}.$$

Proof. We will restrict attention to those i that lie in the interval $I = [x/2 - x^{2/3}, x/2 + x^{2/3}]$, as the sum of $\binom{x}{i}$ for i outside this interval is $o(2^x)$ by the Chernoff bound. By slightly altering I if necessary we may suppose that $|I|$ is divisible by $s + 1$, and we partition it into subintervals $\{J_\phi : \phi \in \Phi\}$ with $|J_\phi| = s + 1$ for every $\phi \in \Phi$. Note that any J_ϕ contains at least one $i \notin L \bmod p$. (Since $p \ge s + 1$, each element of J_ϕ gives a distinct residue mod p, so not every element of J_ϕ can belong to L mod p.) It is easy to see that $\binom{x}{j_1} = (1 + o(1))\binom{x}{j_2}$ for any $j_1, j_2 \in J_\phi$,

so $\sum_{i\in J_\phi,\, i\notin L \bmod p}\binom{x}{i} > \frac{1+o(1)}{s+1}\sum_{i\in J_\phi}\binom{x}{i}$. Therefore

$$\sum_{i\notin L \bmod p}\binom{x}{i} \geq \sum_{J_\phi}\sum_{i\in J_\phi,\, i\notin L \bmod p}\binom{x}{i} > \frac{1+o(1)}{s+1}\sum_{J_\phi}\sum_{i\in J_\phi}\binom{x}{i}$$
$$= \frac{1+o(1)}{s+1}\sum_{i\in I}\binom{x}{i} > \frac{2^x}{3s},$$

as required. □

Proof of Theorem 1.4. Write $k_c = \lfloor \frac{\sum_{i\notin L \bmod p}\binom{n}{i}}{\sum_{i=0}^{s}\binom{n}{i}} \rfloor$. We can assume that $k = k_c$ or $k = k_c+1$. Suppose that $\mathcal{A}_1,\dots,\mathcal{A}_k$ are set systems that are L-cross-intersecting mod p such that every set $A \in \bigcup_{i=1}^k \mathcal{A}_i$ has $|A| \notin L \bmod p$, and suppose that $\sum_{i=1}^k |\mathcal{A}_i| \geq \max\{k\sum_{i=0}^{s}\binom{n}{i}, \sum_{i\notin L \bmod p}\binom{n}{i}\}$. Let $\mathcal{A}$ be the sets that belong to at least two of the $\mathcal{A}_i$ and let $\mathcal{B}$ be those sets that belong to exactly one of the $\mathcal{A}_i$. Since $\mathcal{A}$ is L-intersecting mod p we have $|\mathcal{A}| \leq \sum_{i=0}^{s}\binom{n}{i}$ by the Frankl–Wilson theorem, and if $\mathcal{B} \neq \emptyset$, then $|\mathcal{A}| < \sum_{i=0}^{s}\binom{n}{i}$. The s^*-inclusion vectors $\{v_A^s : A \in \mathcal{A}\}$ form a basis of $V_{\mathcal{A}}^s$ over $\mathbb{F}_p$ and no vector v_B^s with $B \in \mathcal{B}$ lies in $V_{\mathcal{A}}^s$. This shows that $|\mathcal{A}| + \dim V_{\mathcal{B}}^s \leq \sum_{i=0}^{s}\binom{n}{i}$.

We can assume that both $\mathcal{A}$ and $\mathcal{B}$ are nonempty. Then $|\mathcal{A}| \leq \sum_{i=0}^{s}\binom{n}{i} - 1$, so $|\mathcal{B}| \geq k > (\sum_{i\notin L \bmod p}\binom{n}{i}/\sum_{i=0}^{s}\binom{n}{i}) - 1 > 2^n/n^{s+1}$, by Lemma 5.1. Again we get the inequalities $\dim V_{\mathcal{B}}^s \geq \sum_{i=0}^{s}\binom{\lfloor n-(s+1)\log_2 n\rfloor}{i}$ and $|\mathcal{A}| \leq \lceil (s+1)\log_2 n\rceil \sum_{i=0}^{s-1}\binom{n-1}{i}$. Then

$$|\mathcal{B}| \geq \sum_{i=1}^{k}|\mathcal{A}_i| - k|\mathcal{A}|$$
$$\geq \sum_{i\notin L \bmod p}\binom{n}{i} - \left(\frac{\sum_{i\notin L \bmod p}\binom{n}{i}}{\sum_{i=0}^{s}\binom{n}{i}} + 1\right)\lceil (s+1)\log_2 n\rceil \sum_{i=0}^{s-1}\binom{n-1}{i}$$
$$> \left(1 - \frac{3s(s+1)\log_2 n}{2n}\right)\sum_{i\notin L \bmod p}\binom{n}{i}. \tag{8}$$

Recalling that $\mathcal{A} \neq \emptyset$, we now consider any $A \in \mathcal{A}$. For any $B \in \mathcal{B}$ the size of its intersection with A belongs to $L \bmod p$. Let $\mathcal{C} = \{C \subset [n] : |C| \notin L \bmod p \text{ and } |A \cap C| \notin L \bmod p\}$. Then we have $|\mathcal{B}| \leq \sum_{i\notin L \bmod p}\binom{n}{i} - |\mathcal{C}|$. Fix a number $m > 10s$ so that Lemma 5.1 holds for all $x \geq m$.

Suppose first that $|A| < m$. Note that $\mathcal{C}$ contains all sets of the form $C = A \cup D$, where $D \cap A = \emptyset$ and $|D| \notin L' = \{\ell - |A| \bmod p : \ell \in L\}$. By applying Lemma 5.1 to L' there are at least $2^{n-m}/3s > 2^{n-2m}$ such sets D, so $|\mathcal{C}| \geq 2^{n-2m}$. Next suppose that $|A| > n-m$. Since $\mathcal{C}$ contains all sets C such that $C \subset A$ and $|C| \notin L \bmod p$, we again have $|\mathcal{C}| > 2^{n-m}/3s > 2^{n-2m}$. Finally suppose that $m \leq |A| \leq n-m$. There are at least $2^{|A|}/3s$ sets $D \subset A$ such that $|D| \notin L \bmod p$. For each such D there are at least $2^{n-|A|}/3s$ sets $E \subset [n]\backslash A$ such that $|E| \notin \{\ell - |D| \bmod p : \ell \in L\}$. We obtain at least $2^n/9s^2 > 2^{n-2m}$ sets $D \cup E \in \mathcal{C}$. In all cases we see that $|\mathcal{C}| \geq 2^{n-2m}$. Therefore $|\mathcal{B}| \leq \sum_{i\notin L \bmod p}\binom{n}{i} - 2^{n-2m} < (1-2^{-2m})\sum_{i\notin L \bmod p}\binom{n}{i}$. For $n > n(s)$ sufficiently large this contradicts (8), which completes the proof. □

6. Concluding remarks.

- It would be interesting to determine the minimum value of n for which our results hold.
- The bounds that we give are tight when $L = \{0, 1, \ldots, s-1\}$, but one could consider a variant of this problem in which the set L is fixed to be some different set. It seems plausible that the following should be true.
  (1) To maximize the total size of uniform L-cross-intersecting systems $\mathcal{A}_1, \ldots, \mathcal{A}_k$ on $[n]$ one should either take all $\mathcal{A}_i$ equal to a maximum uniform L-intersecting system or take one $\mathcal{A}_i$ equal to all sets of size $\lfloor n/2 \rfloor$ and the others empty.
  (2) To maximize the total size of nonuniform L-cross-intersecting systems $\mathcal{A}_1, \ldots, \mathcal{A}_k$ on $[n]$ one should either take all $\mathcal{A}_i$ equal to a maximum nonuniform L-intersecting system or take one $\mathcal{A}_i$ to consist of all subsets of $[n]$ and the others equal to a maximum L-intersecting system in which the sizes of all sets also belong to L.

Acknowledgment. We would like to thank an anonymous referee for some useful remarks.

REFERENCES

[1] N. Alon, L. Babai, and H. Suzuki, *Multilinear polynomials and Frankl–Ray-Chaudhuri–Wilson type intersection theorems*, J. Combin. Theory Ser. A, 58 (1991), pp. 165–180.
[2] L. Babai and P. Frankl, *Algebra Methods in Combinatorics*, Department of Computer Science, University of Chicago, Chicago, preliminary version, 1992.
[3] E. R. Berlekamp, *On subsets with intersections of even cardinality*, Canad. Math. Bull., 12 (1969), pp. 363–366.
[4] B. Bollobás, P. Keevash, and B. Sudakov, *Multicoloured extremal problems*, J. Combin. Theory Ser. A, 107 (2004), pp. 295–312.
[5] P. Frankl and J. Pach, *On the number of sets in a null t-design*, European J. Combin., 4 (1983), pp. 21–23.
[6] P. Frankl and N. Tokushige, *Some inequalities concerning cross-intersecting families*, Combin. Probab. Comput., 7 (1998), pp. 247–260.
[7] P. Frankl and R. M. Wilson, *Intersection theorems with geometric consequences*, Combinatorica, 1 (1981), pp. 357–368.
[8] J. R. Isbell, *An inequality for incidence matrices*, Proc. Amer. Math. Soc., 10 (1959), pp. 216–218.
[9] P. Keevash, M. Saks, B. Sudakov, and J. Verstraëte, *Multicolour Turán problems*, Adv. in Appl. Math., 33 (2004), pp. 238–262.
[10] J. H. van Lint, *Introduction to Coding Theory*, 2nd ed., Grad. Texts in Math. 86, Springer-Verlag, Berlin, 1992.
[11] K. N. Majumdar, *On some theorems in combinatorics relating to incomplete block designs*, Ann. Math. Statistics, 24 (1953), pp. 377–389.
[12] J. Qian and D. K. Ray-Chaudhuri, *Extremal case of Frankl–Ray-Chaudhuri–Wilson inequality*, Special issue on design combinatorics: In honor of S. S. Shrikhande, J. Statist. Plann. Inference, 95 (2001), pp. 293–306.
[13] D. K. Ray-Chaudhuri and R. M. Wilson, *On t-designs*, Osaka J. Math, 12 (1975), pp. 735–744.
[14] J. Sgall, *Bounds on pairs of families with restricted intersections*, Combinatorica, 19 (1999), pp. 555–566.

SIAM J. DISCRETE MATH.
Vol. 18, No. 4, pp. 728–740

THE COVER TIME OF RANDOM REGULAR GRAPHS*

COLIN COOPER† AND ALAN FRIEZE‡

Abstract. Let $r \geq 3$ be constant, and let $\mathcal{G}_r$ denote the set of r-regular graphs with vertex set $V = \{1, 2, \ldots, n\}$. Let G be chosen randomly from $\mathcal{G}_r$. We prove that with high probability (w.h.p.) the cover time of a random walk on G is asymptotic to $\frac{r-1}{r-2}\, n \log n$.

Key words. cover time, random graphs

AMS subject classifications. 05C80, 60C05

DOI. 10.1137/S0895480103428478

1. Introduction. Let $G = (V, E)$ be a connected graph, let $|V| = n$, and let $|E| = m$. A *random walk* $\mathcal{W}_u$, $u \in V$, on the undirected graph $G = (V, E)$ is a Markov chain $X_0 = u, X_1, \ldots, X_t, \ldots \in V$ associated to a particle that moves from vertex to vertex according to the following rule: the probability of a transition from vertex i, of degree d_i, to vertex j is $1/d_i$ if $\{i, j\} \in E$, and 0 otherwise. For $u \in V$ let C_u be the expected time taken for $\mathcal{W}_u$ to visit every vertex of G. The *cover time* C_G of G is defined as $C_G = \max_{u \in V} C_u$. The cover time of connected graphs has been extensively studied. It is a classic result of Aleliunas et al. [2] that $C_G \leq 2m(n-1)$. It was shown by Feige [8], [9] that for any connected graph G

$$(1 - o(1))n \log n \leq C_G \leq (1 + o(1))\frac{4}{27}n^3.$$

The lower bound is achieved by (for example) the complete graph K_n, whose cover time is determined by the Coupon Collector problem.

In a previous paper [7] we studied the cover time of random graphs $G_{n,p}$ when $np = c \log n$, where $c = O(1)$ and $(c-1)\log n \to \infty$. This extended a result of Jonasson, who proved in [12] that when the expected average degree $(n-1)p$ grows faster than $\log n$, with high probability (w.h.p.) a random graph has the same cover time (asymptotically) as the complete graph K_n, whereas when $np = \Omega(\log n)$, this is not the case. (A sequence of events $\mathcal{E}_n, n \geq 0$, is said to occur w.h.p. if $\lim_{n\to\infty} \mathbf{Pr}(\mathcal{E}_n) = 1$.)

THEOREM 1 (see [7]). *Suppose that* $np = c \log n = \log n + \omega$, *where* $\omega = (c-1)\log n \to \infty$ *and* $c \geq 1$. *If* $G \in G_{n,p}$, *then w.h.p.*

$$C_G \sim c \log\left(\frac{c}{c-1}\right) n \log n.$$

The notation $A_n \sim B_n$ means that $\lim_{n\to\infty} A_n/B_n = 1$.

The main new result of the paper concerns the cover time of random regular graphs.

*Received by the editors May 28, 2003; accepted for publication (in revised form) August 13, 2004; published electronically April 22, 2005.

http://www.siam.org/journals/sidma/18-4/42847.html

†Department of Computer Science, King's College, University of London, London WC2R 2LS, UK (ccooper@dcs.kcl.ac.uk).

‡Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213 (alan@random.math.cmu.edu). This author's research was supported in part by NSF grant CCR-0200945.

THEOREM 2. *Let $r \geq 3$ be constant. Let $\mathcal{G}_r$ denote the set of r-regular graphs with vertex set $V = \{1, 2, \ldots, n\}$. If G is chosen randomly from $\mathcal{G}_r$, then w.h.p.*

$$C_G \sim \frac{r-1}{r-2}\, n \log n.$$

Using a similar argument we can consider how many steps are needed for the walk to get within distance k of every vertex. Let us call this $C_G^{(k)}$. Cover time corresponds to $k = 0$. We prove the following theorem.

THEOREM 3. *Let $r \geq 3$, $k \geq 0$ be constants. Let $\mathcal{G}_r$ denote the set of r-regular graphs with vertex set $V = \{1, 2, \ldots, n\}$. If G is chosen randomly from $\mathcal{G}_r$, then w.h.p.*

$$C_G^{(k)} \sim \frac{1}{(r-2)(r-1)^{k-1}}\, n \log n.$$

The next section contains the heart of the proof of our theorems. In it we establish a good estimate of the probability that the first visit of $\mathcal{W}$ to a vertex v takes place at a time t. Once this is done, we can proceed to the proof of Theorem 2 in section 3 and the proof of Theorem 3 in section 6.

2. The first visit time lemma.

2.1. Convergence of the random walk. In this section G denotes a fixed connected graph, and u is some arbitrary vertex from which a walk $\mathcal{W}_u$ is started. Let $\mathcal{W}_u(t)$ be the vertex reached at step t, let P be the matrix of transition probabilities of the walk, and let $P_u^{(t)}(v) = \mathbf{Pr}(\mathcal{W}_u(t) = v)$. Let $\pi_v = \frac{d_v}{2m}$ for $v \in V$. Let $\lambda_{\max} > 0$ be the second largest absolute value of an eigenvalue of P. Assume that $\lambda_{\max} < 1$. Then,

$$|P_u^{(t)}(x) - \pi_x| \leq (\pi_x/\pi_u)^{1/2} \lambda_{\max}^t \leq n^{1/2} \lambda_{\max}^t. \tag{2.1}$$

See, for example, [11]. (Note that connectivity and $\lambda_{\max} < 1$ imply ergodocity.)

2.2. Generating function formulation. For the results of this section, we do not require that G be regular.

Fix two vertices u, v. Let h_t be the probability $\mathbf{Pr}(\mathcal{W}_u(t) = v) = P_u^{(t)}(v)$ that the walk $\mathcal{W}_u$ visits v at step t. Let $H(s)$ be the generating function for the sequence $h_t, t \geq 0$.

Similarly, considering the walk $\mathcal{W}_v$, starting at v, let r_t be the probability that this walk returns to v at step $t = 0, 1, \ldots$. Let $R(s)$ be the generating function for the sequence $r_t, t \geq 0$. We note that $r_0 = 1$.

Let $f_t(u \to v)$ be the probability that the first visit of the walk $\mathcal{W}_u$ to v occurs at step t. If $u \neq v$, then $f_0(u \to v) = 0$. Let $F(s)$ generate $f_t(u \to v)$. Thus

$$H(s) = F(s)R(s). \tag{2.2}$$

Let

$$T = \frac{4 \log n}{\log 1/\lambda_{\max}}. \tag{2.3}$$

We note that (2.1) gives

$$\max_{x \in V} |P_u^{(t)}(x) - \pi_x| \leq n^{-3} \qquad \text{for } t \geq T. \tag{2.4}$$

For $R(s)$ let

$$R_T(s) = \sum_{j=0}^{T-1} r_j s^j. \tag{2.5}$$

Thus $R_T(s)$ generates the probability of a return to v during steps $0, \ldots, T-1$ of a walk starting at v. Similarly for $H(s)$, let

$$H_T(s) = \sum_{j=0}^{T-1} h_j s^j. \tag{2.6}$$

2.3. First visit time: Single vertex v. The following lemma should be viewed in the context that G is an n vertex graph which is part of a sequence of graphs with n growing to infinity. We prove it in greater generality than is needed for the proof of Theorem 2.

In what follows $c_1, c_2, \ldots$ are positive constants independent of n.

LEMMA 4. *Let T be as defined in* (2.3). *Suppose that*

(a) $H_T(1) \le (1 - c_1)R_T(1)$.

(b) $\max_{|s|=1} \frac{|R_T(s) - R_T(1)|}{R_T(1)} \le 1 - c_2$.

(c) $T\pi_v = o(1)$, $T\pi_v = \Omega(n^{-2})$.

(d) $\lambda_{\max} \le c_3 < 1$.

Let

$$\lambda = \frac{c_2}{100T}. \tag{2.7}$$

Let

$$p_v = \frac{\pi_v}{R_T(1)(1 + O(T\pi_v))}, \tag{2.8}$$

$$c_{u,v} = 1 - \frac{H_T(1)}{R_T(1)(1 + O(T\pi_v))}, \tag{2.9}$$

where the values of the $1 + O(T\pi_v)$ terms are given implicitly in (2.16) *and* (2.19), *respectively. Then*

$$f_t(u \to v) = c_{u,v} \frac{p_v}{(1 + p_v)^{t+1}} + O(e^{-\lambda t/2}) \qquad \textit{for all } t \ge T. \tag{2.10}$$

Proof. Write

$$R(s) = R_T(s) + \widehat{R}_T(s) + \frac{\pi_v s^T}{1 - s}, \tag{2.11}$$

$$A(s) = (1 - s)R(s) = \pi_v s^T + (1 - s)(R_T(s) + \widehat{R}_T(s)), \tag{2.12}$$

where $R_T(s)$ is given by (2.5) and

$$\widehat{R}_T(s) = \sum_{t \ge T} (r_t - \pi_v) s^t$$

generates the error in using the stationary distribution π_v for r_t when $t \ge T$.

Note that while (2.11) is valid only for $|s| < 1$, the fact that $|r_t - \pi_v| \le n^{1/2} c_3^t$ means that the expansion (2.12) is valid for $|s| < c_3^{-1}$.

Similarly, let

$$H(s) = H_T(s) + \widehat{H}_T(s) + \pi_v \frac{s^T}{1-s}, \tag{2.13}$$

$$B(s) = (1-s)H(s) = \pi_v s^T + (1-s)(H_T(s) + \widehat{H}_T(s)). \tag{2.14}$$

Using (2.11), (2.13) we rewrite $F(s) = H(s)/R(s)$ from (2.2) as $F(s) = B(s)/A(s)$.

If $|s| \le \lambda_{\max}^{-1/3}$, then (2.1) implies that for $Z = H, R$,

$$|\widehat{Z}(s)| \le n^{1/2} \sum_{t \ge T} (s\lambda_{\max})^t = o(n^{-2}). \tag{2.15}$$

For real $s \ge 1$ and $Z = H, R$, we have

$$Z_T(1) \le Z_T(s) \le Z_T(1)s^T.$$

Let $s = 1 + \beta\pi_v$, where $\beta > 0$ is constant. Since $T\pi_v = o(1)$ we have

$$Z_T(s) = Z_T(1)(1 + O(T\pi_v)).$$

$T\pi_v = o(1)$ implies that $|s| \le \lambda_{\max}^{-1/3}$ and (2.15) applies. As $T\pi_v = \Omega(n^{-2})$ and $R_T(1) \ge 1 + r_2 > 1 + \frac{1}{n}$, this implies that

$$A(s) = \pi_v(1 - \beta R_T(1)(1 + O(T\pi_v))).$$

It follows that $A(s)$ has a real zero at s_0, where

$$s_0 = 1 + \frac{\pi_v}{R_T(1)(1 + O(T\pi_v))} = 1 + p_v, \tag{2.16}$$

say. We also see that

$$A'(s_0) = -R_T(1)(1 + O(T\pi_v)) \ne 0 \tag{2.17}$$

and thus s_0 is a simple zero (see, e.g., [6, p. 193]). The value of $B(s)$ at s_0 is

$$B(s_0) = \pi_v \left(1 - \tfrac{H_T(1)}{R_T(1)(1+O(T\pi_v))} + O(T\pi_v)\right) \ne 0. \tag{2.18}$$

Thus, from (2.8), (2.9)

$$\frac{B(s_0)}{A'(s_0)} = -p_v c_{u,v}. \tag{2.19}$$

Thus the residue of $F(s)$ at s_0 is $B(s_0)/A'(s_0)$ (see, e.g., [6, p. 195]), and the principal part of the Laurent expansion of $F(s)$ at s_0 is

$$f(s) = \frac{B(s_0)/A'(s_0)}{s - s_0}. \tag{2.20}$$

To approximate the coefficients of the generating function $F(s)$, we now use a standard technique for the asymptotic expansion of power series (see, e.g., [13, Thm. 5.2.1]).

We prove (below) that s_0 is the only zero of $A(s)$ inside the circle $C_\lambda = \{s = (1+\lambda)e^{i\theta}\}$. Thus $F(s) = f(s) + g(s)$, where $g(s)$ is analytic in C_λ. Let $M = \max_{s \in C_\lambda} |g(s)|$. Thus $M \le \max|f(s)| + \max|F(s)|$.

As $F(s) = B(s)/A(s)$ on C_λ, we have that

$$|F(s)| \le \frac{H_T(1)(1+\lambda)^T + O(T\pi_v)}{|R_T(s)| - O(T\pi_v)}.$$

Let $\widetilde{s} = s/(1+\lambda)$. We note that $|R_T(s) - R_T(\widetilde{s})| \le ((1+\lambda)^T - 1)R_T(1)$ and also that Lemma 4(b) implies that $|R_T(\widetilde{s})| \ge c_2 R_T(1)$, which implies that

$$|R_T(s)| \ge R_T(1)\left(c_2 + 1 - e^{c_2/100}\right),$$

and hence $M = O(1)$.

Let $a_t = [s^t]g(s)$; then (see, e.g., [6, p. 143]), $a_t = g^{(t)}(0)/t!$. By the Cauchy inequality (see, e.g., [6, p. 130]), we have that $|g^{(t)}(0)| \le Mt!/(1+\lambda)^t$ and thus

$$|a_t| \le \frac{M}{(1+\lambda)^t} = O(e^{-t\lambda/2}).$$

As $[s^t]F(s) = [s^t]f(s) + [s^t]g(s)$ and $[s^t]1/(s - s_0) = -1/(s_0)^{t+1}$, we have

$$[s^t]F(s) = \frac{-B(s_0)/A'(s_0)}{s_0^{t+1}} + O(e^{-t\lambda/2}). \tag{2.21}$$

Thus, we obtain

$$[s^t]F(s) = c_{u,v}\frac{p_v}{(1+p_v)^{t+1}} + O(e^{-t\lambda/2}),$$

which completes the proof of (2.10).

We now prove that s_0 is the only zero of $A(s)$ inside the circle C_λ. We use Rouché's theorem (see, e.g., [6]), which states: *Let two functions $f(z)$ and $g(z)$ be analytic inside and on a simple closed contour C. Suppose that $|f(z)| > |g(z)|$ at each point of C; then $f(z)$ and $f(z) + g(z)$ have the same number of zeros, counting multiplicities, inside C.*

Let the functions $f(s)$, $g(s)$ be given by $f(s) = (1-s)R_T(1)$ and $g(s) = \pi_v s^T + (1-s)(R_T(s) - R_T(1) + \widehat{R}_T(s))$. For $s \in C_\lambda$, let $\widetilde{s} = s/(1+\lambda)$,

$$\begin{aligned}
|g(s)|/|f(s)| &\le \frac{\pi_v(1+\lambda)^T}{\lambda R_T(1)} + \frac{|R_T(s) - R_T(\widetilde{s})|}{R_T(1)} + \frac{|R_T(\widetilde{s}) - R_T(1)|}{R_T(1)} + o(n^{-2}) \\
&\le 100e^{c_2/100}\pi_v T + \left(e^{c_2/100} - 1\right) + (1 - c_2) + o(n^{-2}) \\
&< 1.
\end{aligned}$$

As $f(s) + g(s) = A(s)$, we conclude that $A(s)$ has only one zero inside the circle C_λ. This is the simple zero at s_0. □

COROLLARY 5. *Let $\boldsymbol{A}_t(v)$ be the event that $\mathcal{W}_u$ has not visited v by step t. Then for $t \ge T$,*

$$\mathbf{Pr}(\boldsymbol{A}_t(v)) = \frac{c_{u,v}}{(1+p_v)^t} + O(\lambda^{-1}e^{-\lambda t/2}).$$

Proof. We use Lemma 4 and $\mathbf{Pr}(\boldsymbol{A}_t(v)) = \sum_{\tau > t} f_\tau(u \to v)$. □

3. Random regular graphs are nice. Our task now is to show that a typical r-regular graph satisfies conditions (a)–(d) of Lemma 4 and to compute $R_T(1)$.

We start with some typical properties of a random regular graph. Let

$$\sigma = \lfloor \log\log\log n \rfloor.$$

Say a cycle C is *small* if $|C| \le \sigma$.

An r-regular graph G is *nice* if

(P1) G is connected.

(P2) The second eigenvalue of the adjacency matrix of G is at most $2\sqrt{r-1} + \epsilon$, where $\epsilon > 0$ is arbitrarily small ($\epsilon = 1/10$ is small enough).

(P3) There are at most $r^{2\sigma}$ vertices on small cycles.

(P4) No pair of small cycles are within distance 3σ of each other.

THEOREM 6. *Let $r \ge 3$ be a constant and let G be chosen uniformly from the set $\mathcal{G}_r$ of r-regular graphs with vertex set $[n]$. Then G is nice w.h.p.*

Proof. (P1): That a random r-regular graph is r-connected w.h.p., for $r \ge 3$, was proved in [4].

(P2): That the second eigenvalue of a random r-regualr graph is this small w.h.p. was proved by Friedman [10].

For (P3), (P4) we use the configuration model as elaborated in [5]. Let $W = [n] \times [r]$ ($W_v = v \times [r]$ represents r *half-edges* incident with vertex $v \in [n]$.) A *configuration* F is a partition of W into $rn/2$ 2-element subsets, and Ω denotes the set of possible configurations. We associate with F a multigraph $\mu(F) = ([n], E(F))$, where, as a multiset,

$$E(F) = \{(v,w) : \{(v,i),(w,j)\} \in F \text{ for some } 1 \le i,\ j \le r\}.$$

(Note that $v = w$ is possible here.)

We say that F is *simple* if the multigraph $\mu(F)$ has no loops or multiple edges. Let Ω_0 denote the set of simple configurations. It is known that if F is chosen uniformly from Ω, then

(a) Each $G \in \mathcal{G}(n,r)$ is the image (under μ) of exactly $(r!)^n$ simple configurations.

(b) $\mathbf{Pr}(F \in \Omega_0) \approx e^{-(r^2-1)/4}$.

It follows from this that any property of almost every $\mu(F)$ is a property of almost every member of $\mathcal{G}_r$.

(P3): The expected number of small cycles in $\mu(F)$, F chosen randomly from Ω, is bounded by

$$\sum_{k=3}^{\sigma} \binom{n}{k} \frac{(k-1)!}{2} \frac{(r(r-1))^k M_{rn-2k}}{M_{rn}} \le \sum_{k=3}^{\sigma} \binom{n}{k} \frac{(k-1)!}{2} \left(\frac{r}{n}\right)^k \le r^{\sigma},$$

where $M_{2m} = \frac{(2m)!}{2^m m!}$ and $|\Omega| = M_{rn}$.

The almost sure occurrence of property (P3) now follows from the Markov inequality.

(P4): Similarly, the expected number of pairs of small cycles that are close to each other is bounded by

$$\sum_{a=3}^{\sigma}\sum_{b=3}^{\sigma} \binom{n}{a}\binom{n}{b-1} \frac{(a-1)!}{2}\frac{(b-1)!}{2}\left(\frac{r}{n}\right)^{a+b} + \sum_{a=3}^{\sigma}\sum_{b=3}^{\sigma}\sum_{c=1}^{\sigma} \binom{n}{a}\binom{n}{b}\binom{n}{c} \frac{(a-1)!}{2}\frac{(b-1)!}{2} ab \left(\frac{r}{n}\right)^{a+b+c+1} = o(1). \qquad \square$$

Remark 1. Although the main subject of the paper is random regular graphs, it is worth mentioning *Ramanujan graphs*. An n-vertex r-regular graph is Ramanujan if $\lambda_{\max} \le \frac{2\sqrt{r-1}}{r}$. It is known that such graphs have girth $\Omega(\log n)$ and so they are nice; see Alon [3]. Consequently, their cover time $\sim \frac{r-1}{r-2} n \log n$.

Remark 2. Aldous [1] considered the cover time of Cayley graphs and obtained a similar expression for the cover time. By relaxing the assumptions in Lemma 4 it is possible to obtain some of his results, e.g., the hypercube and toroidal grids in three or more dimensions.

4. Nice graphs. Assume from now on that G is a nice regular graph. For $v \in V$ and $k \ge 0$, let $N_k(v) = \{w : \mathrm{dist}(v,w) = k\}$ be the set of vertices at distance k from v. Let $M_l(v) = \cup_{j=0}^{l} N_l(v)$, and let $G_l(v)$ be the subgraph of G induced by $M_l(v)$. Also let us replace the notations $R_T(1), H_T(1)$ by R_v, H_v, reflecting their dependence on v.

DEFINITION 7. *We say v is locally tree-like if $G_\sigma(v)$ is a tree.*

LEMMA 8. *If v is locally tree-like, then*

$$R_T(1) = \frac{r-1}{r-2} + o(\sigma^{-1}).$$

Proof. Let T_r be the infinite r-regular tree, rooted at v. Let $\mathcal{X}$ be a random walk on T_r starting at v. Let ρ_i be the probability that $\mathcal{X}$ is at v at step i. Now we can project the walk $\mathcal{X}$ onto a walk $\mathcal{Y}$ on $\{0,1,2,\dots,\}$, where the particle moves right with probability $q = \frac{r-1}{r}$ and left with probability $p = \frac{1}{r}$, except, of course, at the origin, where it must move right. Let E_i be the expected number of visits to 0 for $\mathcal{Y}$ starting at i. Then

$$E_0 = 1 + E_1 = 1 + E_0 p/q.$$

This is because E_1 is E_0 times the expected number of visits to 0 between right moves from 1. Solving gives

$$\sum_{i=0}^{\infty} \rho_i = E_0 = \frac{r-1}{r-2}. \tag{4.1}$$

Note next that for $i \ge 0$ we have $\rho_{2i+1} = 0$ and we will argue that

$$\rho_{2i} \le \binom{2i}{i} \frac{(r-1)^i}{r^{2i}}, \tag{4.2}$$

and then

$$\sum_{i=\sigma+1}^{\infty} \rho_i \le \sum_{j=\sigma/2} \binom{2j}{j} \frac{(r-1)^i}{r^{2i}} = o(\sigma^{-1}). \tag{4.3}$$

We compare this with $R_T(1)$. First observe that $r_i = \rho_i$ for $i \le \sigma$. Then from (2.1) we see that

$$\sum_{i=\sigma+1}^{T} r_i \le \sum_{i=\sigma+1}^{T} (\pi_v + \lambda_{\max}^i) = o(\sigma^{-1}).$$

Let us now prove (4.2). First observe that the right-hand side of (4.2) is the probability that a walk $\mathcal{Y}_1$ is at the origin after $2i$ steps. Here $\mathcal{Y}_1$ is the walk on $\{0, \pm 1, \pm 2, \dots, \}$

where the particle moves right with probability $q = \frac{r-1}{r}$ and left with probability $p = \frac{1}{r}$; i.e., there is no barrier at the origin. We can couple $\mathcal{Y}, \mathcal{Y}_1$ so that $\mathcal{Y}(t) \geq |\mathcal{Y}_1(t)|$. When $\mathcal{Y}_1(t) > 0$ we can move them in the same direction and when $\mathcal{Y}_1 < 0$ then we can move $\mathcal{Y}$ further from the origin whenever $\mathcal{Y}_1$ moves further from the origin.

The lemma now follows from (4.1) and (4.3). □

Remark 3. Because there are very few non-tree-like vertices and because they are far apart, we will find that we do not need to estimate $R_T(1)$ for such vertices. It is relatively easy to show that for non-tree-like vertices $R_T(1) = 1 + O(r^{-1})$ as $r \to \infty$; thus the only difficulty is with small r.

LEMMA 9. *If v is locally tree-like, then for $|s| = 1$, $\frac{|R_T(s) - R_T(1)|}{R_T(1)} \leq \frac{5}{6}$.*

Proof. For any s,

$$|R_T(s) - R_T(1)| \leq \sum_{j=1}^{T} r_j |s^j - 1|.$$

As $|s| = 1$, we have that

$$\sum_{j=1}^{T} r_j |s^j - 1| \leq 2 \sum_{j=1}^{T} r_j. \tag{4.4}$$

We prove the lemma for $r \geq 4$ by observing that Lemma 8 implies

$$2 \sum_{j=1}^{T} r_j = 2(R_T(1) - 1) = (1 + o(1)) \frac{2}{r-2} \leq (1 + o(1)) \frac{2}{3} \cdot \frac{r-1}{r-2} = (1 + o(1)) \frac{2}{3} R_T(1). \tag{4.5}$$

When $r = 3$ we improve on (4.4) using ad hoc arguments. First observe that $\pi_v = 1/n$ for $v \in V$ and that (2.1) implies that

$$S_0 = \sum_{i=\sigma}^{T} r_j |s^j - 1| \leq 2 \sum_{i=\sigma}^{T} r_i \leq 2 \sum_{i=\sigma}^{T} (\lambda_{\max}^i + \pi_v) = o(1). \tag{4.6}$$

Now consider $j < \sigma$. For a locally tree-like vertex, $r_j = 0$ if j is odd, and $r_j > 0$ if j is even. Fix $0 \leq \theta < 2\pi$ and let $s = e^{i\theta}$; then for $j = 2k$

$$|s^j - 1| = (2(1 - \cos j\theta))^{1/2} = 2|\sin k\theta|.$$

Thus

$$S_1 = \sum_{j=1}^{\sigma-1} r_j |s^j - 1| = 2 \sum_{k=1}^{\lfloor (\sigma-1)/2 \rfloor} r_{2k} |\sin k\theta|.$$

Note now that $r_2 = \frac{1}{3}$ and $r_4 = \frac{5}{27}$. Suppose first that $\theta \notin I = [\frac{3\pi}{16}, \frac{5\pi}{16}] \cup [\frac{11\pi}{16}, \frac{13\pi}{16}]$. Then $|\sin 2\theta| \leq \sin \frac{3\pi}{8}$ and so

$$S_1 \leq 2 \sum_{j=1}^{\sigma-1} r_j - \frac{2}{3} \left(1 - \sin \frac{3\pi}{8}\right). \tag{4.7}$$

On the other hand, if $\theta \in I$, then $|\sin 4\theta| \le \sin\frac{\pi}{4}$ and then

$$S_1 \le 2\sum_{j=1}^{\sigma-1} r_j - \frac{10}{27}\left(1-\sin\frac{\pi}{4}\right). \tag{4.8}$$

Equations (4.6), (4.7), (4.8) imply that $S_0 + S_1 \le 2(R_T(1)-1) - 1/3$. The lemma follows, since $R_T(1) \sim 2$ for $r = 3$. □

Finally, we note the following lemma.

LEMMA 10. *For nice graphs,* $\frac{H_T(1)}{R_T(1)} \le \frac{9}{10}$.

Proof. Let f'_t be the probability that $\mathcal{W}_u$ has a first visit to v at time t. As $H(s) = F(s)R(s)$, we have

$$\begin{aligned} H_T(1) &\le \mathbf{Pr}(\mathcal{W}_u \text{ visits } v \text{ by time } T-1)R_T(1) \\ &= R_T(1)\sum_{t=1}^{T-1} f'_t. \end{aligned}$$

Now (2.1) implies that if $\tau_0 = \lfloor 2\log\lambda_{\max}^{-1}\log\log n\rfloor$, then

$$\sum_{t=\tau_0}^{T-1} f'_t \le \sum_{t=\tau_0}^{T-1}(\pi_v + \lambda_{\max}^t) = o(1).$$

We now estimate $\sum_{t=0}^{\tau_0} f'_t$, the probability that $\mathcal{W}_u$ visits v by time τ_0. Let $v_1, v_2, \ldots, v_r$ be the neighbors of v and let w be the first neighbor of v visited by $\mathcal{W}_u$. Then

$$\begin{aligned} \mathbf{Pr}(\mathcal{W}_u \text{ visits } v \text{ by time } \tau_0) &= \sum_{i=1}^{r} \mathbf{Pr}(\mathcal{W}_u \text{ visits } v \text{ by time } \tau_0 \mid w = v_i)\mathbf{Pr}(w = v_i) \\ &\le \sum_{i=1}^{r} \mathbf{Pr}(\mathcal{W}_{v_i} \text{ visits } v \text{ by the time } \tau_0)\mathbf{Pr}(w = v_i). \end{aligned}$$

So it suffices to prove the lemma when u is a neighbor of v. If $G_l(u)$ is a tree, then we can argue as in Lemma 8. Let ψ be the probability that a particle at the root of T_r ever returns to the root. The expected number of visits is

$$\frac{r-1}{r-2} = \sum_{k=1}^{\infty} k\psi^{k-1}(1-\psi) = \frac{1}{1-\psi}.$$

So $\psi = \frac{1}{r-1}$, and

$$\mathbf{Pr}(\mathcal{W}_u \text{ does not visit } v \text{ by time } \tau_0) \ge \frac{r-1}{r}(1-\psi-o(1)) = \frac{r-2}{r} - o(1).$$

If $G_l(u)$ contains a cycle C, then let $e = (\xi, \eta)$ be an edge of C not incident with u and let T_u be the tree $G_l(u) - e$. Let $N'(u) = \{u_1, u_2, \ldots, u_s\}$, $s \in \{r-2, r-1\}$ be the neighbors of u which are not on a shortest path from ξ or η to u in T_u. $|N'(u) \setminus \{v\}| \ge r-3$, and so

$$\mathbf{Pr}(\mathcal{W}_u \text{ does not visit } v \text{ by time } \tau_0) \ge \frac{r-3}{r}(1-\psi-o(1)) = \frac{(r-2)(r-3)}{r(r-1)} - o(1).$$

This leaves the case $r = 3$ and $N'(u) = \{v\}$. With probability $\frac{2}{3}$ we have $\mathcal{W}_u(1) \neq v$. If ξ or η is reached (possibly $N(u) = \{v, \xi, \eta\}$), then with probability $\frac{1}{3}$ the next move is away from u and $1 - \psi - o(1)$ bounds the probability that there is no return to ξ or η. Hence

$$\mathbf{Pr}(\mathcal{W}_u \text{ does not visit } v \text{ by time } \tau_0) \geq \frac{2}{9}(1 - \psi - o(1)),$$

completing the proof of the lemma. □

5. Cover time of nice graphs. We now prove that

$$C_G \sim \frac{r-1}{r-2} n \log n.$$

Assume that $u, v \in V$ and that v is tree-like. Section 3 establishes that the conditions of Lemma 4 hold and gives values for the parameters c_{uv}, p_v given by (2.8), (2.9). To summarize, we have

$$R_T(1) = \frac{r-1}{r-2} + o(1), \qquad \frac{H_T(1)}{R_T(1)} \leq \frac{9}{10}, \qquad \lambda_{\max} \leq \frac{2\sqrt{r-1} + .1}{r},$$
$$\pi_v = \frac{1}{n}, \qquad T = O(\log n), \qquad \lambda = \Omega(1/\log n).$$

Hence, the probability that $\mathcal{W}_u$ has not visited v by some step $t \geq T$ (see Corollary 5) is given by

$$\mathbf{Pr}(\boldsymbol{A}_t(v)) = (1 + o(1)) c_{uv} e^{-t p_v} + O(\lambda^{-1} e^{-\lambda t/2}).$$

Here $c_{uv} < 1$ and

$$p_v = \frac{r-2}{(r-1)n}(1 + o(\sigma^{-1})).$$

5.1. Upper bound on cover time. Let $t_0 = \lceil (1 + \sigma^{-1}) \frac{r-1}{r-2} n \log n \rceil$. We prove that for nice graphs, for any vertex $u \in V$,

$$C_u \leq t_0 + o(t_0). \tag{5.1}$$

Let $T_G(u)$ be the time taken to visit every vertex of G by the random walk $\mathcal{W}_u$. Let U_t be the number of vertices of G which have not been visited by $\mathcal{W}_u$ at step t. We note the following:

$$C_u = \mathbf{E}\, T_G(u) = \sum_{t>0} \mathbf{Pr}(T_G(u) \geq t), \tag{5.2}$$

$$\mathbf{Pr}(T_G(u) > t) = \mathbf{Pr}(U_t > 0) \leq \min\{1, \mathbf{E}\, U_t\}. \tag{5.3}$$

It follows from (5.2), (5.3) that for all t

$$C_u \leq t + \sum_{s \geq t} \mathbf{E}\, U_s = t + \sum_{v \in V} \sum_{s \geq t} \mathbf{Pr}(\boldsymbol{A}_s(v)). \tag{5.4}$$

Let V_1 be the set of locally tree-like vertices and let $V_2 = V - V_1$. If G is nice, then $|V_2| \leq r^{3\sigma}$ for there are at most r^σ vertices within distance σ of a particular vertex in a small cycle, and at most $r^{2\sigma}$ vertices on small cycles.

For $v \in V_1$ we have

$$\begin{aligned}\sum_{s\ge t_0} \mathbf{Pr}(\boldsymbol{A}_s(v)) &\le (1+o(1))e^{-t_0 p_v} \sum_{s\ge t_0} e^{-(s-t_0)p_v} + O(\lambda^{-2}e^{-\lambda t_0/2}) \\ &\le 2p_v^{-1}e^{-t_0 p_v} \\ &\le 3\frac{r-1}{r-2}.\end{aligned}$$

Furthermore, we see that in particular,

$$\mathbf{Pr}(\boldsymbol{A}_{5n}(v)) \le 2e^{-1}. \tag{5.5}$$

Suppose next that $v \in V_2$. We can find $w \in V_1$ such that $\mathrm{dist}(v,w) \le \sigma$. So from (5.5), with $\nu = 5n+\sigma$, we have

$$\mathbf{Pr}(\boldsymbol{A}_\nu(v)) \le 1 - (1-2e^{-1})r^{-\sigma}$$

since if our walk visits w, it will with probability at least $r^{-\sigma}$ visit v within the next σ steps. Thus if $\gamma = (1-2e^{-1})r^{-\sigma}$,

$$\begin{aligned}\sum_{s\ge t_0} \mathbf{Pr}(\boldsymbol{A}_s(v)) &\le \sum_{s\ge t_0} (1-\gamma)^{\lfloor s/\nu \rfloor} \qquad (5.6)\\ &\le \sum_{s\ge t_0} (1-\gamma)^{s/(2\nu)} \\ &= \frac{(1-\gamma)^{t_0/(2\nu)}}{1-(1-\gamma)^{1/(2\nu)}} \\ &\le 3\nu\gamma^{-1}. \qquad (5.7)\end{aligned}$$

Thus, for all $u \in V$,

$$\begin{aligned}C_u &\le t_0 + 3\frac{r-1}{r-2}|V_1| + 3|V_2|\nu\gamma^{-1} \\ &= t_0 + O(r^{4\sigma}n) \\ &= t_0 + o(t_0),\end{aligned}$$

as $\sigma = \lfloor \log\log\log n \rfloor$.

5.2. Lower bound on cover time. For any vertex u, we can find a set of vertices S such that at time $t_1 = t_0(1-\epsilon)$, $\epsilon \to 0$, the probability the set S is covered by the walk $\mathcal{W}_u$ tends to zero. Hence $T_G(u) > t_1$ w.h.p., which implies that $C_G \ge t_0 - o(t_0)$.

We construct S as follows. Let $S \subseteq V_1$ be some maximal set of locally tree-like vertices all of which are at least distance $2\sigma+1$ apart. Thus $|S| \ge (n - r^{3\sigma})r^{-(2\sigma+1)}$.

Let $S(t)$ denote the subset of S which has not been visited by $\mathcal{W}_u$ after step t. Now, provided $t \ge T$

$$\mathbf{E}\,|S(t)| \ge (1-o(1)) \sum_{v\in S} \left(\frac{c_{u,v}}{(1+p_v)^t} + o(n^{-2}) \right).$$

Let u be a fixed vertex of S. Let $v \in S$ and let $H_T(1)$ be given by (2.6); then (2.1) implies that

$$H_T(1) \le \sum_{t=\sigma}^{T-1} (\pi_v + \lambda_{\max}^t) = o(1). \tag{5.8}$$

Thus $c_{uv} = 1 - o(1)$. Setting $t = t_1 = (1-\epsilon)t_0$, where $\epsilon = 2\sigma^{-1}$, we have

$$\begin{aligned} \mathbf{E}\,|S(t_1)| &= (1+o(1))|S|e^{-(1-\epsilon)t_0 p_v} \\ &\geq n^{1/\sigma}. \end{aligned} \tag{5.9}$$

Let $Y_{v,t}$ be the indicator for the event that $\mathcal{W}_u$ has not visited vertex v at time t. Let $Z = \{v, w\} \subset S$. We will show (below) that for $v, w \in S$

$$\mathbf{E}\,(Y_{v,t_1}Y_{w,t_1}) = \frac{c_{u,Z}}{(1+p_Z)^{t+2}} + o(n^{-2}), \tag{5.10}$$

where $c_{u,Z} \sim 1$ and $p_Z \sim 2(r-2)/(n(r-1))$. Thus

$$\mathbf{E}\,(Y_{v,t_1}Y_{w,t_1}) = (1+o(1))\mathbf{E}\,(Y_{v,t_1})\mathbf{E}\,(Y_{w,t_1}). \tag{5.11}$$

It follows from (5.9) and (5.11) that

$$\mathbf{Pr}(S(t_1) \neq 0) \geq \frac{(\mathbf{E}\,|S(t_1)|)^2}{\mathbf{E}\,|S(t_1)|^2} = \frac{1}{\frac{\mathbf{E}|S_{t_1}|(|S_{t_1}|-1)}{(\mathbf{E}|S(t_1)|)^2} + (\mathbf{E}\,|S_{t_1}|)^{-1}} = 1 - o(1).$$

Proof of (5.10). Let Γ be obtained from G by merging v, w into a single node Z. This node has degree $2r$ and every other node has degree r.

There is a natural measure preserving mapping from the set of walks in G which start at u and do not visit v or w to the corresponding set of walks in Γ which do not visit Z. Thus the probability that $\mathcal{W}_u$ does not visit v or w in the first t steps is equal to the probability that a random walk $\widehat{\mathcal{W}}_u$ in Γ which also starts at u does not visit Z in the first t steps.

We apply Lemma 4 to Γ. That $\pi_Z = \frac{2}{n}$ is clear, and $c_{u,Z} = 1 - o(1)$ is argued as in (5.8). The derivation of $R_T(1)$ in Lemma 8 is also valid. The vertex Z is tree-like up to distance σ in Γ. The fact that the root vertex of the corresponding infinite tree has degree $2r$ does not affect the calculation of $R_T(1)$. □

6. Looking ahead. We now consider Theorem 3. Fix $u \in V$ and let $C_u^{(k)}$ be the expected time for $\mathcal{W}_u$ to have been within distance k of every vertex. In analogy to (5.4) we have

$$C_u^{(k)} \leq t + \sum_{v \in V} \sum_{s \geq t} \mathbf{Pr}(\boldsymbol{A}_s^{(k)}(v)), \tag{6.1}$$

where $\boldsymbol{A}_s^{(k)}(v)$ is the event that $\mathcal{W}_u$ has not been within distance k by time s.

Now fix v with $\mathrm{dist}(u, v) > k$. Assume that v is tree-like. Define Γ_0 by contracting $M_k(v)$ to a single vertex Z and deleting any loops created (M_k is defined in section 4). There is a natural measure preserving mapping from the set of walks in G which start at u and do not get within distance k of v to the corresponding set of walks in Γ_0 which do not visit Z. Thus the probability that $\mathcal{W}_u$ does not get within distance k in the first t steps is equal to the probability that a random walk $\widehat{\mathcal{W}}_u$ in Γ_0 which also starts at u does not visit Z in the first t steps; i.e., $\mathbf{Pr}(\boldsymbol{A}_t(Z)) = \mathbf{Pr}(\boldsymbol{A}_s^{(k)}(v))$.

We apply Lemma 4 to Γ. $\pi_Z = \frac{|N_k(v)|}{rn - O(1)} = \frac{(r-1)^k}{n - O(1)}$, $R_Z \sim \frac{r-1}{r-2}$, and $H_Z/R_Z \leq 9/10$. So now if $t_0 = \lceil \frac{1+\sigma^{-1}}{(r-2)(r-1)^{k-1}} n \log n \rceil$, then $\sum_{t \geq t_0} \mathbf{Pr}(\boldsymbol{A}_t(Z)) = O(1)$. Thus

$$\sum_{v \in V_1} \sum_{t \geq t_0} \mathbf{Pr}(\boldsymbol{A}_t^{(k)}(v)) = O(n). \tag{6.2}$$

Now $\boldsymbol{A}_t^{(k)}(v) \subseteq \boldsymbol{A}_t(v)$ and (5.7) holds, even with the smaller value of t_0. Thus

$$\sum_{v \in V_2} \sum_{t \geq t_0} \mathbf{Pr}(\boldsymbol{A}_t^{(k)}(v)) = o(n), \tag{6.3}$$

and an upper bound of $t_0 + o(t_0)$ for $C_u^{(k)}$ follows from (6.1), (6.2), and (6.3).

The lower bound is obtained by taking a set S of $n^{1-o(1)}$ tree-like vertices at distance at least 3σ apart and using the Chebyshev inequality as we did in section 5.2. Choose $u \in S$ and then for each pair of vertices $v_1, v_2 \in S \setminus \{u\}$ we form Γ_1 by contracting $M_k(v_1) \cup M_k(v_2)$ into a single vertex, removing loops and then arguing as we did before. □

REFERENCES

[1] D. J. Aldous, *On the time taken by random walks on finite groups to visit every state*, Z. Wahrsch. Verw. Gebeite, 62 (1983), pp. 361–374.

[2] R. Aleliunas, R. M. Karp, R. J. Lipton, L. Lovász, and C. Rackoff, *Random walks, universal traversal sequences, and the complexity of maze problems*, in Proceedings of the 20th Annual IEEE Symposium on Foundations of Computer Science (San Juan, Puerto Rico), IEEE, New York, 1979, pp. 218–223.

[3] N. Alon, Tools from higher algebra, *in Handbook of Combinatorics*, R. L. Graham, M. Grötschel, and L. Lovász, eds., Elsevier, Amsterdam, 1995, pp. 1749–1783.

[4] B. Bollobás, *Random graphs*, in Combinatorics, London Math. Soc. Lecture Note Ser. 52, H. N. V. Temperley, ed., Cambridge University Press, Cambridge, UK, 1981, pp. 80–102.

[5] B. Bollobás, *A probabilistic proof of an asymptotic formula for the number of labelled regular graphs*, European J. Combin., 1 (1980), pp. 311–316.

[6] J. Brown and R. Churchill, *Complex Variables and Applications*, 6th ed., McGraw–Hill, New York, 1996.

[7] C. Cooper and A. M. Frieze, *The cover time of sparse random graphs*, Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2003, pp. 140–147.

[8] U. Feige, *A tight upper bound for the cover time of random walks on graphs*, Random Structures Algorithms, 6 (1995), pp. 51–54.

[9] U. Feige, *A tight lower bound for the cover time of random walks on graphs*, Random Structures Algorithms, 6 (1995), pp. 433–438.

[10] J. Friedman, *A proof of Alon's second eigenvalue conjecture*, Mem. Amer. Math. Soc., to appear.

[11] M. Jerrum and A. Sinclair, *The Markov chain Monte Carlo method: An approach to approximate counting and integration*, in Approximation Algorithms for NP-hard Problems, D. Hochbaum, ed., PWS, Boston, 1996, pp. 482–520.

[12] J. Jonasson, *On the cover time of random walks on random graphs*, Combin., Probab. Comput., 7 (1998), pp. 265–279.

[13] H. Wilf, *generatingfunctionology*, Academic Press, Boston, 1990.

[14] L. Lovász, *Combinatorial Problems and Exercises*, 2nd ed., North–Holland, Amsterdam, 1993.

SIAM J. DISCRETE MATH.
Vol. 18, No. 4, pp. 741–748

A RECURSIVE CONSTRUCTION FOR REGULAR DIFFERENCE TRIANGLE SETS*

WENSONG CHU†, CHARLES J. COLBOURN†, AND SOLOMON W. GOLOMB‡

Abstract. A difference triangle set (DΔS) is a collection of sets of integers having the property that every integer can be written in at most one way as the difference of two elements within a set of the collection. The standard objective is to minimize the largest difference represented, given a specified size of the collection and sizes of the sets that it contains. In order to construct DΔSs, we present a new type of combinatorial design, *monotonic directed* (v, k, λ)*-designs (MDDs)*. Using MDDs, we give a general recursive construction for difference triangle sets (DΔSs). Several instances of this main construction are derived. One of these, the perfect construction, leads to an infinite family of regular (optimal) DΔSs if the existence of a single regular DΔS is known.

Key words. difference triangle set, directed design, directed packing, Golomb ruler, spanning ruler set

AMS subject classification. 05B05

DOI. 10.1137/S0895480103436761

1. Introduction.

DEFINITION 1.1. *An* (I, J)*-difference triangle set (D*Δ*S) is a set*

$$\Delta = \{\Delta_1, \Delta_2, \ldots, \Delta_I\},$$

where, for each $1 \le i \le I$, $\Delta_i = \{a_{ij} | 0 \le j \le J\}$ *is a set of nonnegative integers such that*

$$0 = a_{i0} < a_{i1} < \cdots < a_{iJ}$$

and all the integers $a_{ij} - a_{ij'}$ *with* $1 \le i \le I$ *and* $0 \le j' < j \le J$ *are distinct. Let* $m(\Delta) = \max\{a_{iJ} | 1 \le i \le I\}$. *An* (I, J)*-D*Δ*S is* regular *if* $m(\Delta) = \frac{IJ(J+1)}{2}$. *Let* $M(I, J) = \min\{m(\Delta) | \Delta$ *is an* $(I, J) - D\Delta S\}$. *If* $m(\Delta) = M(I, J)$, *then* Δ *is* optimal.

By the definition, any regular DΔS is an optimal one.

Difference triangle sets have applications in the design of missile guidance codes, radio systems to reduce intermodulation interference, convolutional self-orthogonal codes, optical orthogonal codes, and in numerous other areas [6, 13]. The fundamental problem (a rather difficult one) is to determine the value of $M(I, J)$ for a given pair (I, J). The special case of determining $M(1, J)$ is the well-studied problem of finding Golomb rulers (see, for example, [9]) which has continued to resist many attacks. Because of these difficulties, the establishment of good lower and upper bounds for $M(I, J)$ is of interest. For a general discussion and tables, see [6, 12, 13], and for progress on DΔSs and a summary of all known optimal families of DΔSs, see [16].

*Received by the editors October 25, 2003; accepted for publication (in revised form) March 25, 2004; published electronically April 22, 2005.

http://www.siam.org/journals/sidma/18-4/43676.html

†Department of Computer Science and Engineering, Arizona State University, Tempe AZ 85287-8809 (Wensong.Chu@asu.edu, Charles.Colbourn@asu.edu). The research of these authors was supported by ARO grant DAAD 19-01-1-0406.

‡The Communication Sciences Institute, Electrical Engineering Systems, University of Southern California, Los Angeles, CA 90089 (milly@usc.edu).

In this paper, we present a general recursive construction for DΔSs, the main construction. Based on this, we derive a "perfect construction"; the adjective "perfect" is chosen because it leads to perfect coverage of differences. Indeed it leads to an infinite family of regular DΔSs, given the existence of a single regular DΔS. We also develop a ruler construction, which in general produces further useful DΔSs. We begin with the main construction next.

2. Main construction.

DEFINITION 2.1. *Let X be a v-set and $\mathcal{B}$ be a collection of k-subsets of X called* blocks. *A pair $(X, \mathcal{B})$ is a* (v, k, λ)-design *(respectively,* packing*) if any pair of different elements of X is contained in exactly (respectively, at most) λ blocks.*

If we define a special order within each block of a design, then we get a directed system, formally defined next.

DEFINITION 2.2. *Let X be a v-set. A transitively ordered k-subset $B = (a_1, a_2, \ldots, a_k)$ of X consists of $\frac{k(k-1)}{2}$ ordered pairs of form (a_i, a_j) with $1 \le i < j \le k$. Let $\mathcal{B}$ be a collection of transitively ordered k-subsets of X; these are* directed blocks. *A pair $(X, \mathcal{B})$ is a* directed (v, k, λ)-design *(respectively,* packing*) if any ordered pair of different elements from X is contained in exactly (respectively, at most) λ directed blocks.*

The existence of directed (v, k, λ)-designs is completely settled when $\lambda = 1$ and $k \in \{3, 4, 5, 6\}$.

THEOREM 2.3.

1. [11] *A directed $(v, 3, 1)$-design exists if and only if $v \equiv 0, 1 \pmod 3$.*
2. [18] *A directed $(v, 4, 1)$-design exists if and only if $v \equiv 1 \pmod 3$ and $v \ge 4$.*
3. [19] *A directed $(v, 5, 1)$-design exists if and only if $v \equiv 1, 5 \pmod{10}$, $v \ge 5$ and $v \ne 15$.*
4. [2] *A directed $(v, 6, 1)$-design exists if and only if $v \equiv 1, 6 \pmod{15}$, $v \ge 6$ and $v \ne 21$.*

Known necessary conditions for the existence of a directed $(v, k, 1)$-design are that $2(v-1) \equiv 0 \pmod{k-1}$, $2v(v-1) \equiv 0 \pmod{k(k-1)}$ and $v \ge k$. However, we need some further conditions.

DEFINITION 2.4. *Let $B = (a_1, a_2, \ldots, a_k) \in \mathcal{B}$ be a directed block. Let $pos(a_i)$ denote the position of a_i in the ordered v-set $(a_1, a_1 + 1, \ldots, v-1, 0, 1, \ldots, a_1 - 1)$. Then B is a* monotonic *directed block if $pos(a_1) < pos(a_2) < \cdots < pos(a_k)$ in the real number system.*

A monotonic directed (v, k, λ)-design *(respectively,* packing*), denoted by (v, k, λ)-MDD (respectively, (v, k, λ)-MDP) is a pair $(X, \mathcal{B})$ satisfying two conditions:*

1. *$(X, \mathcal{B})$ is a directed (v, k, λ)-design (respectively, (v, k, λ)-packing).*
2. *Each directed block in $\mathcal{B}$ is monotonic.*

If $\lambda = 1$, then a $(v, k, 1)$-MDP with t directed blocks is denoted by t-MDP(v, k). Correspondingly, a $(v, k, 1)$-MDD is denoted by MDD(v, k) or $\frac{2v(v-1)}{k(k-1)}$-MDP(v, k).

The following directed blocks form an MDD$(7, 1)$ or a 7-MDP$(7, 4)$:

$$\begin{array}{cccc} (0 & 1 & 4 & 6) \\ (1 & 2 & 5 & 0) \\ (2 & 3 & 6 & 1) \\ (3 & 4 & 0 & 2) \\ (4 & 5 & 1 & 3) \\ (5 & 6 & 2 & 4) \\ (6 & 0 & 3 & 5) \end{array}$$

This MDD can be obtained by developing $(0+i, 1+i, 4+i, 6+i)$ modulo 7.

We employ certain MDD(v,k) and t-MDP(v,k). Another object we require in our recursive construction is *cyclic difference families*.

DEFINITION 2.5. *Let $\mathcal{F} = \{B_1, B_2, \ldots, B_t\}$ be a family of k-subsets, called* base blocks*, of $\mathbb{Z}_v$. $\mathcal{F}$ is a* cyclic difference family *(respectively,* cyclic difference packing*), denoted by $(v,k,1)$-DF (respectively, $(v,k,1)$-DP), if any nonzero element of $\mathbb{Z}_v$ can be represented in a unique way (respectively, at most one way) as a difference of two elements lying in some member of $\mathcal{F}$.*

The abbreviated notation DF(v,k) is used for a $(v,k,1)$-DF. We also use the notation t-DP(v,k) to indicate that the difference packing consists of t base blocks.

A DF(v,k) can be denoted as a $\frac{v-1}{k(k-1)}$-DP(v,k).

THEOREM 2.6 (main construction). *If there exists a t-DP(v,k) and an s-MDP(k,h), then there exists an $(st, h-1)$-DΔS Δ with $m(\Delta) \le v-1$.*

Proof. Let $\{B_1, B_2, \ldots, B_t\}$ be the t-DP(v,k). For each

$$B_i = \{b_{i,0}, b_{i,1}, \ldots, b_{i,k-1}\}$$

with $1 \le i \le t$, the collection of differences

$$\mathcal{D}(B_i) = \{b_{i,j_1} - b_{i,j_2} | 0 \le j_1 \ne j_2 \le k-1\}$$

contains no repeated elements modulo v. $\mathcal{D}(B_i)$ also can be viewed as follows:

$$\{(j_1, j_2) | 0 \le j_1 \ne j_2 \le k-1\}.$$

This is a subset of all possible ordered pairs of $\mathbb{Z}_k$.

Let $(X, \mathcal{C})$ be the s-MDP(k,h) with $\mathcal{C} = \{C_1, C_2, \ldots, C_s\}$. For every $C_j = (c_{j1}, c_{j2}, \ldots, c_{jh})$, $1 \le j \le s$, and each B_i from the t-DP(v,k), construct the following st sequences:

$$\Delta_i = \{(b_{i,c_{j1}}, b_{i,c_{j2}}, \ldots, b_{i,c_{jh}}) | 1 \le j \le s\},\ 1 \le i \le t.$$

Since $(X, \mathcal{C})$ is an MDP(k,h), every ordered pair (ℓ_1, ℓ_2) of $\mathbb{Z}_k$ is contained in at most one of the directed blocks of $\mathcal{C}$. Since each ordered pair represents a distinct difference, every difference $b_{i,j_1} - b_{i,j_2}$ in $\mathcal{D}(B_i)$ is contained in at most one of the sequences in Δ_i.

For each sequence $(b_{i,c_{j1}}, b_{i,c_{j2}}, \ldots, b_{i,c_{jh}})$ in Δ_i, there are two possible cases. The first is that

$$b_{i,c_{j1}} < b_{i,c_{j2}} < \cdots < b_{i,c_{jh}}$$

as integers. Then it is in the correct form as a sequence of a DΔS.

The second case is that it consists of two increasing subsequences as follows:

$$b_{i,c_{j1}} < b_{i,c_{j2}} < \cdots < b_{i,c_{jw}}, \qquad b_{i,c_{j_{w+1}}} < b_{i,c_{j_{w+2}}} < \cdots < b_{i,c_{jh}}$$

with $b_{i,c_{jw}} > b_{i,c_{j_{w+1}}}$ and $b_{i,c_{jh}} < b_{i,c_{j1}}$ as integers. Then we transform the sequence into an increasing sequence,

$$b_{i,c_{j1}} < b_{i,c_{j2}} < \cdots < b_{i,c_{jw}} < b_{i,c_{j_{w+1}}} + v < b_{i,c_{j_{w+2}}} + v < \cdots < b_{i,c_{jh}} + v.$$

The monotonicity of the s-MDP(k,h) means that any sequence in Δ_i consists of at most two increasing subsequences with the properties listed above. Hence it can be transformed into an increasing sequence with maximal difference less than v.

Take the normalized form for each sequence in Δ_i, properly transformed if necessary, with $1 \le i \le h$. Then $\cup_{i=1}^{t} \overline{\Delta_i}$ is the desired difference triangle set, where $\overline{\Delta_i}$ is the normalized form of Δ_i with possible transformations. The total number of sequences is st, and the monotonic property ensures that in each sequence the maximal element is no more than $v-1$. □

In the following sections, we derive several variations of the main construction.

3. Perfect constructions.

THEOREM 3.1 (perfect construction). *If there exists a DF(v,k) and an MDD(k,h), then there exists a regular $(\frac{2(v-1)}{h(h-1)}, h-1)$-D$\Delta$S Δ with $m(\Delta) = v-1$.*

Proof. This is a simple application of the main construction. □

The existence of cyclic difference families has been extensively studied. However research on the existence of MDDs is almost nonexistent. We only have some examples for MDD(v,k) with k small. They all come from some regular DΔSs.

LEMMA 3.2. *If there exists an (I,J)-DΔS Δ with $m(\Delta) \le m$, then there exists a t-MDP$(m+1, J+1)$ with $t = (m+1)I$.*

Proof. For each sequence in Δ, say $S = (a_0, a_1, \ldots, a_J)$, develop $(a_0 + i, a_1 + i, \ldots, a_J + i)$ modulo $m+1$. We claim that the resulting design is a t-MDP$(m+1, J+1)$, with $t = (m+1)I$. We show that each ordered pair of $\mathbb{Z}_{m+1}$ is contained in at most one ordered block. Let (a,b) be an ordered pair of $\mathbb{Z}_{m+1}$. If the difference $d = b - a \pmod{m+1}$ is contained in one sequence of Δ, say $a_j - a_i = d$ with $1 \le i < j \le J$, then (a,b) is contained in one of the ordered blocks $(a_0 + i, a_1 + i, \ldots, a_J + i)$. Since every S is in increasing order, each $(a_0 + i, a_1 + i, \ldots, a_J + i)$ is monotonic. Thus the resulting design is a t-MDP$(m+1, J+1)$ with $t = (m+1)I$. □

COROLLARY 3.3. *If there exists a regular (I,J)-DΔS, then there exists an*

$$MDD\left(\frac{IJ(J+1)}{2} + 1, J+1\right).$$

Proof. A simple counting verifies the claim. □

For $J \ge 3$, the only known regular (I,J)-DΔSs are the following.

LEMMA 3.4.

1. [10, 15] *For $I \le 50$, there exists a regular $(I,3)$-DΔS for $4 \le I \le 23$ and for $I \in \{1, 25, 28, 30, 31, 36, 41\}$.*
2. [15] *If there exists a regular $(I,3)$-DΔS, then there exists a regular $(\mathcal{I},3)$-DΔS with $\mathcal{I} = 5I+1, 7I+1, 13I+1, 19I+4, 23I+5, 15I+5, 15I+6$.*
3. [15] *There exists a regular $(I,4)$-DΔS for $I \in \{6, 8, 10\}$.*

Using fundamental results on difference families over finite fields due to Wilson, we can construct an infinite family of regular difference triangle sets from a single MDD.

THEOREM 3.5 (see [20]). *Suppose p is a prime, and $p - 1 \equiv 0 \pmod{k(k-1)}$. Then there exists a DF$(p,k)$ if*

$$p > \binom{k}{2}^{k(k-1)}.$$

We also employ the well-known Dirichlet theorem on primes in an arithmetic progression, that there exist infinitely many primes of the form $p \equiv 1 \pmod{k(k-1)}$.

THEOREM 3.6.

1. *If there exists an MDD(k,J), then for any prime $p \equiv 1 \pmod{k(k-1)}$ and $p > \binom{k}{2}^{k(k-1)}$, there exists a regular $(I, J-1)$-DΔS with $I = \frac{2(p-1)}{J(J-1)}$.*

2. *If there exists a regular (I, J)-DΔS, then for any prime $p \equiv 1 \pmod{k(k-1)}$ and $p > \binom{k}{2}^{k(k-1)}$ with $k = \frac{IJ(J+1)}{2} + 1$, there exists a regular $(\mathcal{I}, J)$-DΔS with $\mathcal{I} = \frac{2(p-1)}{J(J+1)}$.*

Proof. To obtain the proof, apply the perfect construction using Corollary 3.3 and Theorem 3.5. □

Simply stated, the existence of any single regular (I, J)-DΔS implies the existence of an infinite family of regular DΔSs with the same J.

When k is relatively small, we have much better existence results available for cyclic difference families. We illustrate this when $k = 7$, using the current existence result.

THEOREM 3.7 (see [5]). *Let $p \equiv 1 \pmod{42}$, $p \neq 43$, be a prime. Then there exists a cyclic DF$(p, 7)$ if one of the following conditions is satisfied:*

1. *$p < 261239791$ with $p \neq 127$, $p \neq 211$.*
2. *$p > 1.236597 \times 10^{13}$.*

There exists an MDD$(7, 4)$ from a regular $(1, 3)$-DΔS. Then there also exists a regular $(I, 3)$-DΔS for $I = \frac{p-1}{6}$ with $p \in \{337, 379, 421, 463, 547, 631, 673, 757, 883, 967\}$ for $p < 1000$. Since 7 is a prime, there is a recursive construction available for cyclic DF$(v, 7)$ as well.

THEOREM 3.8 (see [1]). *Let k be a prime or prime power, and $v \equiv 1 \pmod{k(k-1)}$. If there exists a cyclic DF(v, k) and a cyclic DF(u, k), then there exists a cyclic DF(uv, k).*

In other words, for any primes p and q, if a DF$(p, 7)$ and a DF$(q, 7)$ both exist, there exists a DF$(pq, 7)$. This yields many more regular $(I, 3)$-DΔSs.

4. The ruler construction. Difference triangle sets have different names. An $(1, J)$-DΔS is a *spanning ruler*. The shortest spanning ruler with $J + 1$ marks is a *Golomb ruler*; this nomenclature has been widely adopted since [7]. An (I, J)-DΔS is then a *spanning ruler set*. A cyclic t-DP(v, k) is correspondingly a set of *circular rulers*, and a cyclic DF(v, k) is a set of *perfect* circular rulers. The following well-known cyclic difference set is a perfect circular ruler, according to the "ruler" terminology.

THEOREM 4.1 (see [17]). *For any prime or prime power q, there exists a cyclic DF$(q^2 + q + 1, q + 1)$.*

The following example, which first appeared in [8], inspired our work. Let $q = 7$. Then there exists a DF-(57,8), a perfect ruler of length 57 with 8 marks:

$$\mathcal{C} = \{0, 1, 5, 7, 17, 35, 38, 49\}.$$

There exists a Golomb ruler of length 6 with 4 marks:

$$\mathcal{S} = \{0, 1, 4, 6\}.$$

Think of $m(\mathcal{S}) = 8$ instead of 6 and apply Lemma 3.2 to produce an 8-MDP$(8, 4)$. Thus, we can construct an $(8, 4)$-DΔS Δ with $m(\Delta) \leq 57-1 = 56$ (in fact, by carrying out the construction one finds that $m(\Delta) = 52$). It is not optimal, since there exists a regular $(8, 4)$-DΔS. However, it produces good sets of spanning rulers in general.

THEOREM 4.2 (ruler construction). *Let Δ be an (I, J)-DΔS with $m(\Delta) = m$. Let q be the smallest prime power such that $m \leq q$. Then there exists an $(I(q+1), J)$-DΔS Σ with $m(\Sigma) \leq q^2 + q$.*

Proof. From the (I, J)-DΔS, we can construct a $(q + 1)I$-MDP$(q + 1, J + 1)$ according to Lemma 3.2. Since q is a prime or prime power, there then exists a

$\mathrm{DF}(q^2+q+1, q+1)$ via Theorem 4.1. The conclusion then follows from the main construction. □

The ruler construction basically asserts that a set of perfect circular rulers and a set of spanning rulers can produce a larger set of spanning rulers with the same number of marks.

5. Relative difference family constructions. The main construction suggests how to make good sets of spanning rulers or DΔSs. The basic idea is to try to pack many differences from the set of circular rulers into a set of spanning rulers. Two approaches should be used simultaneously. One is to try to construct optimal MDPs (i.e., an MDP with the maximal possible number of directed blocks), and another one is to try to find a better n-DP(v,k), in particular one with large k. Unfortunately, knowledge is limited about the existence of optimal MDPs and of cyclic difference packings with large block size.

However, we do have some other options. Let us begin with an example.

There exists a regular $(6,4)$-DΔS Δ with $m(\Delta)=60$. Since 61 is a prime, by Theorem 5.1, there exists a 1-DP$(3720,1)$, where $3720=61^2-1$ with the missing differences:

$$\mathcal{M}=\{0, 62, 2\cdot 62, 3\cdot 62, \ldots, 59\cdot 62\}.$$

From Δ, we can construct an MDD$(61,5)$ with $61\times 6=366$ directed blocks. By the main construction, we can construct a $(366,4)$-DΔS Σ with $m(\Sigma)=3720$. But we can add more, by adding six more spanning rulers as follows:

$$\{62\cdot a_{i,0}, 62\cdot a_{i,1}, \ldots, 62\cdot a_{i,4}\}, \qquad 1\le i\le 6,$$

where $\{a_{i,0}, a_{i,1}, \ldots, a_{i,4}\}\in\Delta$. We have just constructed a *regular* $(372,4)$-DΔS.

The 1-DP$(3720,1)$ is an instance of a more general result due to Bose.

THEOREM 5.1 (see [3]). *For any prime power q, there exists a 1-DP(q^2-1,q) with the following missing differences:*

$$\{0, q+1, 2(q+1), \ldots, (q-2)(q+1)\}.$$

Following the example given, we obtain the following general construction.

THEOREM 5.2. *Let q be a prime power. If there exist an (I,J)-DΔS Δ with $m(\Delta)\le q-1$ and an $s-MDP(q,J+1)$, then there exists an $(sI+I,J)$-DΔS Σ with $m(\Sigma)=q^2-1$.*

Proof. The main construction yields an (sI,J)-DΔS from an $s-MDP(q,J+1)$ and an (I,J)-DΔS. Using a $1-DP(q^2-1,q)$ from Theorem 5.1, we add I more spanning rulers as follows:

$$\{(q+1)\cdot a_{i,0}, (q+1)\cdot a_{i,1}, \ldots (q+1)\cdot a_{i,J}\}, \qquad 1\le i\le I,$$

where $\{a_{i,0}, a_{i,1}, \ldots, a_{i,J}\}\in\Delta$. The differences appearing in the I spanning rulers are multiples of $(q+1)$ that are not contained in the $1-DP(q^2-1,q)$. □

COROLLARY 5.3. *Let q be a prime power. If there exists an (I,J)-DΔS Δ with $m(\Delta)\le q-1$, then there exists an $(sI+I,J)$-DΔS Σ with $m(\Sigma)\le q^2-1$ and $s=(m(\Delta)+1)I$.*

Proof. From Lemma 3.2, an (I,J)-DΔS yields an s-MDP$(m(\Delta)+1,J)$ with $s=(m(\Delta)+1)I$. □

If everything happens perfectly, we get another perfect construction.

COROLLARY 5.4. *Let q be a prime power. If there exists a regular (I, J)-DΔS Δ with $\frac{I(J+1)J}{2} = q - 1$, then there exists a regular $(qI + I, J)$-DΔS.*

Proof. All the differences in the 1-DP$(q^2 - 1, q)$ are contained in the resulting DΔS. ∎

Even further, we can get an infinite family of regular DΔSs from a single regular DΔS if $m(\Delta) + 1$ happens to be a prime power.

THEOREM 5.5. *Let q be a prime power. If there exists a regular (I, J)-DΔS Δ with $\frac{I(J+1)J}{2} = q - 1$, then there exists a regular $(\frac{(q^{2^t}-1)I}{q-1}, J)$-D$\Delta$S for any $t \geq 0$.*

Proof. If $\frac{I(J+1)J}{2} = q - 1$, then

$$(q+1)I\frac{(J+1)J}{2} + 1 = (q-1)(q+1) + 1 = q^2$$

which is still a prime power. Then Corollary 5.4 can be applied to the resulting regular DΔS, and the conclusion follows from a simple calculation. ∎

The 1-DP(q^2-1, q) can be viewed as a relative difference family, defined as follows.

DEFINITION 5.6 (see [4]). *Let N be a subgroup of $\mathbb{Z}_v$ with n elements. Let $\mathcal{F} = \{B_1, B_2, \ldots, B_t\}$ be a family of k-subsets, called* base blocks*, of $\mathbb{Z}_v$. $\mathcal{F}$ is a* cyclic relative difference family*, denoted by t-RDF(v, n, k), if every nonzero element of $\mathbb{Z}_v - N$ can be represented in a unique way as a difference of two elements lying in some member of $\mathcal{F}$.*

Finally, we generalize Theorem 5.1 to a relative difference family construction.

THEOREM 5.7 (RDF construction). *If there exists a t-RDF(v, n, k), an s-MDP$(k, J + 1)$ and an (I, J)-DΔS Δ with $m(\Delta) \leq n$, then there exists an $(st + I, J)$-DΔS Σ with $m(\Sigma) = v - 1$.*

Proof. The construction is similar to the one of Theorem 5.2. ∎

6. Conclusions. We have presented a new type of combinatorial design, the monotonic directed design. Although the existence of directed designs has been extensively studied, few families of MDDs are known currently. In order to use the perfect construction, the existence of MDDs is crucial for constructions of DΔSs. An example of MDD(v, k) with $k \geq 4$, not constructed from a regular DΔS, would be of great interest. However, we should not expect any MDD(v, k) with $k \geq 6$.

THEOREM 6.1 (see [14]). *$M(I, J) \geq \frac{IJ(J+1)}{2} + 1$ for $J > 4$.*

COROLLARY 6.2. *There does not exist an MDD(v, k) with $k \geq 6$.*

The use of monotonic directed designs and packings suggests a new direction for the construction of difference triangle sets, and hence their further study is of definite value.

Acknowledgment. The authors thank Herbert Taylor for many suggestions that inspired this research.

REFERENCES

[1] R. J. R. ABEL, *Difference families*, in CRC Handbook of Combinatorial Designs, C. J. Colbourn and J. H. Dinitz, eds., CRC Press, Boca Raton, Fl, 1996, pp. 270–287.
[2] F. E. BENNETT, R. WEI, J. YIN, AND A. MAHMOODI, *Existence of DBIBDs with block size six*, Util. Math., 43 (1993), pp. 205–217.
[3] R. C. BOSE, *An affine analogue of Singer's theorem*, J. Indian Math. Soc., 6 (1942), pp. 1–15.
[4] M. BURATTI, *Recursive constructions for difference matrices and relative difference families*, J. Combin. Des., 6 (1998), pp. 165–182.

[5] K. Chen, R. Wei, and L. Zhu, *Existence of* $(q, 7, 1)$ *difference families with* q *a prime power*, J. Combin. Des., 10 (2002), pp. 126–138.
[6] C. J. Colbourn, *Difference triangle sets*, in CRC Handbook of Combinatorial Designs, C. J. Colbourn and J. H. Dinitz, eds., CRC Press, Boca Raton, FL, 1996, pp. 312–317.
[7] M. Gardner, *Wheels, Life and Other Mathematical Amusements*, W. H. Freeman and Co., New York, 1983.
[8] S. W. Golomb and H. Taylor, *Cyclic projective planes, perfect circular rulers and good spanning rulers*, in Sequences and Their Applications, T. Helleseth, P. V. Kumar, and K. Yang, eds., Springer-Verlag, New York, 2001, pp. 166–181.
[9] B. Hayes, *Collective wisdom*, American Scientist, 86 (1998), pp. 118–122.
[10] J.-H. Huang and S. S. Skiena, *Graceful labelling prisms*, Ars Combinatoria, 38 (1994), pp. 225–242.
[11] S. H. Y. Hung and N. S. Mendelsohn, *Directed triple systems*, J. Combinatorial Theory Ser. A, 14 (1973), pp. 310–318.
[12] T. Kløve, *Bounds on the size of optimal difference triangle sets*, IEEE Trans. Inform. Theory, 34 (1988), pp. 355–361.
[13] T. Kløve, *Bounds and construction for difference triangle sets*, IEEE Trans. Inform. Theory, 35 (1989), pp. 879–886.
[14] A. Kotzig and J. Turgeon, *Regular perfect systems of difference sets*, Discrete Math., 20 (1977/78), pp. 249–254.
[15] R. Mathon, *Constructions for cyclic Steiner* 2*-designs*, Ann. Discrete Math., 34 (1987), pp. 353–362.
[16] J. B. Shearer, *Difference Triangle Sets*, http://www.research.ibm.com/people/s/shearer/dts.html.
[17] J. Singer, *A theorem in finite projective geometry and some applications to number theory*, Trans. Amer. Math. Soc., 43 (1938), pp. 377–385.
[18] D. J. Street and J. R. Seberry, *All DBIBDs with block size four exist*, Util. Math., 18 (1980), pp. 27–34.
[19] D. J. Street and W. H. Wilson, *On directed balanced incomplete block designs with block size five*, Util. Math., 18 (1980), pp. 161–174.
[20] R. M. Wilson, *Cyclotomy and difference families in elementary Abelian groups*, J. Number Theory, 4 (1972), pp. 17–47.

SIAM J. DISCRETE MATH.
Vol. 18, No. 4, pp. 749–768

ON IDENTIFYING MAXIMAL COVERS*

S. MUÑOZ†

Abstract. In this paper we present an efficient procedure for identifying all maximal covers from the set of covers implied by a 0-1 knapsack constraint. It requires tight bounds for the cardinality of certain minimal covers and an ordering of the covers implied by the knapsack constraint. This type of maximal cover can be very useful for tightening 0-1 models. We also present a modification of the procedure to identify only maximal covers whose induced inequalities are violated by a given fractional solution. Some computational experiments are reported for randomly generated 0-1 knapsack constraints and for single source capacitated plant location problem instances drawn from the literature.

Key words. maximal covers, tighter formulations, knapsack constraints, dominated inequalities

AMS subject classifications. 90C10, 90C05

DOI. 10.1137/S0895480103409584

1. Introduction.

Consider the 0-1 linear programming (LP) problem

$$\begin{aligned} &\text{Min or Max} && \sum_{j\in J} c_j x_j \\ &\text{subject to} && \sum_{j\in J} a_{ij} x_j \sim b_i \qquad \forall i \in I, \\ & && x_j \in \{0,1\} \qquad \forall j \in J, \end{aligned} \tag{1.1}$$

where $I = \{1,\dots,m\}$, $J = \{1,\dots,n\}$, $\{a_{ij}\}_{i\in I,\, j\in J}$, $\{b_i\}_{i\in I}$, $\{c_j\}_{j\in J}$ are rational numbers and $\sim$ is the sense of each constraint ($\leq$, $\geq$, $=$).

The *LP relaxation* of (P) is the same problem as (1.1), where each variable x_j is allowed to take any value in the interval $[0,1]$.

We say that two constraint systems $\boldsymbol{Ax} \sim \boldsymbol{b}$ and $\boldsymbol{A'x} \sim \boldsymbol{b'}$ are *equivalent* if $\{\boldsymbol{x} \in \{0,1\}^n \mid \boldsymbol{Ax} \sim \boldsymbol{b}\} = \{\boldsymbol{x} \in \{0,1\}^n \mid \boldsymbol{A'x} \sim \boldsymbol{b'}\}$. The system $\boldsymbol{A'x} \sim \boldsymbol{b'}$ is said to be as *tight as* the system $\boldsymbol{Ax} \sim \boldsymbol{b}$ if it is equivalent to $\boldsymbol{Ax} \sim \boldsymbol{b}$ and $\{\boldsymbol{x} \in [0,1]^n \mid \boldsymbol{A'x} \sim \boldsymbol{b'}\} \subseteq \{\boldsymbol{x} \in [0,1]^n \mid \boldsymbol{Ax} \sim \boldsymbol{b}\}$. The system $\boldsymbol{A'x} \sim \boldsymbol{b'}$ is said to be *tighter* than the system $\boldsymbol{Ax} \sim \boldsymbol{b}$ if it is equivalent to $\boldsymbol{Ax} \sim \boldsymbol{b}$ and $\{\boldsymbol{x} \in [0,1]^n \mid \boldsymbol{A'x} \sim \boldsymbol{b'}\} \subset \{\boldsymbol{x} \in [0,1]^n \mid \boldsymbol{Ax} \sim \boldsymbol{b}\}$.

A constraint $\sum_{j=1}^n a_j x_j \sim b$ is said to be *valid* for a set $R \subseteq \mathbb{R}^n$ if it is satisfied by any vector $(x_1,\dots,x_n) \in R$.

The tighter a 0-1 model, the smaller could be the gap between the optimal objective function value of the related 0-1 problem and the optimal objective function value of its LP relaxation, and, probably, less computational effort could be required to solve the problem. Therefore, we are interested in tightening the initial formulation of (1.1). It is well known that this can be done by means of valid

*Received by the editors December 23, 2003; accepted for publication (in revised form) July 27, 2004; published electronically April 22, 2005. This research was supported in part by grant BFM2002-00281 from CICYT, Spain.

http://www.siam.org/journals/sidma/18-4/40958.html.

†Departamento de Estadística e Investigación Operativa I, Facultad de Ciencias Matemáticas, Universidad Complutense de Madrid, Ciudad Universitaria, 28040 Madrid, Spain (smunoz@estad.ucm.es).

constraints for its feasible region, e.g., inequalities induced by maximal covers from the set of covers implied by any valid constraint for the feasible region of (1.1); see [4, 6, 8, 9, 10, 12, 15, 18, 20, 21, 22, 24, 28, 31] among many others.

In easy terms, a cover can be considered as a subset of indices of 0-1 variables where at most k of such variables can take the value 1. In particular, we are interested in the so-called maximal covers from the set of covers implied by a 0-1 knapsack constraint, i.e., covers derived from a 0-1 knapsack constraint such that the inequalities induced by any other covers that can be derived from the knapsack constraint are not tighter than their induced inequalities. We present an algorithm for identifying such maximal covers.

This paper is structured as follows. Section 2 reviews classical types of covers and states some related results. Section 3 introduces a simple method for computing a lower and an upper bound for the cardinality of a minimal cover with respect to a knapsack constraint. Section 4 presents a procedure for identifying all maximal covers from the set of covers implied by a knapsack constraint. Section 5 presents a modification of this procedure to identify only maximal covers whose induced inequalities are violated by a given fractional solution. Section 6 reports some computational experiments. Finally, section 7 draws some conclusions from this work.

2. Covers. Basic concepts and results. In this section we review some types of covers given in the literature; see [1, 2, 7, 14, 20, 24, 25, 30, 31, 32] among others. We also state some results concerning these types of covers; their proofs can be found in [11, 20].

Given a set of variables $\{x_1, \dots, x_n\}$ and a set $F \subseteq \{1, \dots, n\}$, let $X(F) = \sum_{j \in F} x_j$.

DEFINITION 2.1. *Given a set of variables $\{x_1, \dots, x_n\}$ and an inequality of the form $X(C^+) - X(C^-) \leq k_C - |C^-|$, where $C^+, C^- \subseteq \{1, \dots, n\}$, $C^+ \cap C^- = \emptyset$, and k_C is an integer such that $1 \leq k_C \leq |C^+ \cup C^-|$, the set $C = C^+ \cup C^-$ is said to be a* cover*, and the inequality $X(C^+) - X(C^-) \leq k_C - |C^-|$ is said to be* induced *by C.*

DEFINITION 2.2. *A* trivial cover *is a cover C such that $k_C = |C|$.*

DEFINITION 2.3. *A cover C is said to be* implied *by the constraint $\sum_{j=1}^n a_j x_j \leq b$ if its induced inequality is valid for the set $\{(x_1, \dots, x_n) \in \{0,1\}^n \mid \sum_{j=1}^n a_j x_j \leq b\}$.*

DEFINITION 2.4. *The inequality $\sum_{j=1}^n a_j x_j \leq b$ is said to be* dominated *by the inequality $\sum_{j=1}^n a'_j x_j \leq b'$ if $\{(x_1, \dots, x_n) \in [0,1]^n \mid \sum_{j=1}^n a'_j x_j \leq b'\} \subseteq \{(x_1, \dots, x_n) \in [0,1]^n \mid \sum_{j=1}^n a_j x_j \leq b\}$.*

DEFINITION 2.5. *Given a set of covers $\mathcal{C}$, $C \in \mathcal{C}$ is a* maximal cover *from $\mathcal{C}$ if its induced inequality is not dominated by the inequality induced by C' $\forall C' \in \mathcal{C}$ such that ${C'}^+ \neq C^+$ or ${C'}^- \neq C^-$ or $k_{C'} \neq k_C$.*

There are several ways to tighten the formulation of problem (1.1) by means of maximal covers whose induced inequalities are valid for its feasible region. Their induced inequalities can be appended to the constraint system of (1.1), e.g., by using them as cutting planes in a branch-and-cut framework (see [15, 18, 31], among many others). They can also be used to increase or reduce the coefficients of some constraints (see, e.g., [6, 8, 9, 10, 20]) and to detect constraint redundancy and fix variables (see, e.g., [21, 22]). In [11, 20] it was shown that the extensions of the strong covers defined in [1] are the maximal covers from the set of covers implied by the related 0-1 knapsack constraint. So, their induced inequalities can be lifted and used to obtain facets of the knapsack polytope (see [1, 2, 4, 8, 14, 24, 25, 29, 30, 31, 32], among others).

We consider valid knapsack constraints for the feasible region of problem (1.1) of

the form

$$\sum_{j\in J_0} a_j x_j \le b, \tag{2.1}$$

where $0 < a_j \le b\ \forall j \in J_0$ and $a_j \le a_{j'}\ \forall j, j' \in J_0$ such that $j < j'$. (Note that any constraint of (1.1) can be put in this form if one previously decomposes each equality into two inequalities.) Let us assume that $\sum_{j\in J_0} a_j > b$, since, otherwise, constraint (2.1) would be redundant.

Given a nonempty set $C \subseteq J_0$, let $m_l(C)$ denote the set of the l smallest indices of C, where l is an integer such that $1 \le l \le |C|$. (Note that the set $m_l(C)$ contains the indices of the l variables in $\{x_j\}_{j\in C}$ with smallest coefficients in constraint (2.1).)

PROPOSITION 2.6. *Let C be a maximal cover from the set of covers implied by constraint* (2.1). *Then C is a nontrivial cover, $C \subseteq J_0$, and its induced inequality is $X(C) \le max\{l \mid \sum_{j\in m_l(C)} a_j \le b\}$.*

Proof. See [11, 20]. □

DEFINITION 2.7. *A nontrivial cover C implied by constraint* (2.1) *with induced inequality $X(C) \le k_C$ is said to be* minimal *with respect to constraint* (2.1) *if $\sum_{j\in C\setminus\{k\}} a_j \le b\ \forall k \in C$.*

Given a nonempty set $C \subseteq J_0$, let $\underline{\gamma}(C) = \min\{j \mid j \in C\}$ and $\overline{\gamma}(C) = \max\{j \mid j \in C\}$.

PROPOSITION 2.8. *A cover C is minimal with respect to constraint* (2.1) *if and only if $C \subseteq J_0$, $\sum_{j\in C} a_j > b$, $\sum_{j\in C\setminus\{\underline{\gamma}(C)\}} a_j \le b$ and its induced inequality is $X(C) \le |C| - 1$.*

Proof. See [11, 20]. □

DEFINITION 2.9. *Let C be a minimal cover with respect to constraint* (2.1). *The* extension *of C is the set $E(C) = C \cup \{j \in J_0 \mid j > \overline{\gamma}(C)\}$.*

PROPOSITION 2.10. *If C is a minimal cover with respect to constraint* (2.1), *then*

(1) *$E(C)$ is a nontrivial cover implied by constraint* (2.1), *and the inequality $X(E(C)) \le |C| - 1$ is induced by $E(C)$.*

(2) *The inequality induced by C is dominated by the inequality $X(E(C)) \le |C|-1$.*

Proof. See [11, 20], among others. □

THEOREM 2.11. *If C is a maximal cover from the set of covers implied by constraint* (2.1), *then there exists a unique minimal cover with respect to constraint* (2.1), *say, C', such that $E(C') = C$.*

Proof. See [11, 20]. □

THEOREM 2.12. *Let C be a minimal cover with respect to constraint* (2.1) *and let $X(E(C)) \le |C|-1$ be the inequality induced by $E(C)$. Then $E(C)$ is a maximal cover from the set of covers implied by constraint* (2.1) *if and only if one of the following conditions is satisfied:*

(1) *$E(C) \subset J_0$ and $\sum_{j\in C\setminus\{\overline{\gamma}(C)\}} a_j + a_{\overline{\gamma}(J_0\setminus E(C))} \le b$.*

(2) *$E(C) = J_0$.*

Proof. See [11, 20]. □

Theorem 2.11 establishes that every maximal cover from the set of covers implied by constraint (2.1) is the extension of a unique minimal cover with respect to (2.1). Therefore, to identify these maximal covers it suffices to determine the minimal covers with respect to (2.1) by using an enumerative procedure based on Proposition 2.8 and to apply Theorem 2.12.

In the next section we present easily computable lower and upper bounds for the cardinality of the minimal covers with respect to constraint (2.1).

3. Obtaining bounds for the cardinality of a minimal cover. For simplicity, from now on it will be assumed that $J_0 = \{1, \dots, n_0\}$.

Let $\underline{k} = \max\{k \in \{1, \dots, n_0 - 1\} \mid \sum_{j=n_0-(k-1)}^{n_0} a_j \leq b\}$.

LEMMA 3.1. *Let C be a minimal cover with respect to constraint* (2.1). *Then* $|C| \geq \underline{k} + 1$.

Proof. Suppose that $|C| \leq \underline{k}$. In this case $\sum_{j\in C} a_j \leq \sum_{j=n_0-(\underline{k}-1)}^{n_0} a_j \leq b$, which contradicts Proposition 2.8. □

$$\text{Let } \underline{j} = \begin{cases} \min\{k \in J_0 \mid \sum_{j\in J_0\setminus\{k\}} a_j \leq b\} & \text{if } \sum_{j=1}^{n_0-1} a_j \leq b, \\ n_0 + 1 & \text{otherwise.} \end{cases}$$

LEMMA 3.2. *Let C be a minimal cover with respect to constraint* (2.1). *Then* $\{\underline{j}, \dots, n_0\} \subseteq C$.

Proof. Suppose that $\exists\, k \in \{\underline{j}, \dots, n_0\} \setminus C$. In this case, $C \subseteq J_0 \setminus \{k\}$, hence $\sum_{j\in C} a_j \leq \sum_{j\in J_0\setminus\{k\}} a_j \leq b$, which contradicts Proposition 2.8. □

LEMMA 3.3. *The following statements hold:*

(1) $\sum_{j=\underline{j}}^{n_0} a_j > b$ *if and only if* $\underline{j} = n_0 - \underline{k}$.

(2) $\sum_{j=\underline{j}}^{n_0} a_j \leq b$ *if and only if* $\underline{j} \geq n_0 - \underline{k} + 1$.

Proof. (1) If $\sum_{j=\underline{j}}^{n_0} a_j > b$, then $\underline{k} = n_0 - \underline{j}$, since $\sum_{j=\underline{j}+1}^{n_0} a_j \leq \sum_{j\in J_0\setminus\{\underline{j}\}} a_j \leq b$. Conversely, if $\underline{j} = n_0 - \underline{k}$, by the definition of $\underline{k}$ we have that $\sum_{j=\underline{j}}^{n_0} a_j > b$.

(2) The proof follows from the definition of $\underline{k}$. □

$$\text{Let } \overline{k} = \begin{cases} \underline{k} & \text{if } \sum_{j=\underline{j}}^{n_0} a_j > b, \\ n_0 - \underline{j} + \max\{k \in \{1, \dots, \underline{j} - 1\} \mid \sum_{j=2}^{k} a_j + \sum_{j=\underline{j}}^{n_0} a_j \leq b\} & \text{otherwise.} \end{cases}$$

LEMMA 3.4. *Let C be a minimal cover with respect to constraint* (2.1). *Then* $|C| \leq \overline{k} + 1$.

Proof. By Lemma 3.2 we have that $\{\underline{j}, \dots, n_0\} \subseteq C$.

If $C = \{\underline{j}, \dots, n_0\}$, then $|C| = n_0 - \underline{j} + 1$. On the other hand, $\sum_{j=\underline{j}}^{n_0} a_j > b$ by Proposition 2.8. Accordingly, by claim (1) of Lemma 3.3 it follows that $|C| = \overline{k} + 1$.

If $\{\underline{j}, \dots, n_0\} \subset C$, then $\{\underline{j}, \dots, n_0\} \subseteq C \setminus \{\underline{\gamma}(C)\}$, and, by Proposition 2.8, $\overline{k} = n_0 - \underline{j} + \max\{k \in \{1, \dots, \underline{j} - 1\} \mid \sum_{j=2}^{k} a_j + \sum_{j=\underline{j}}^{n_0} a_j \leq b\}$. Now, if $\max\{k \in \{1, \dots, \underline{j} - 1\} \mid \sum_{j=2}^{k} a_j + \sum_{j=\underline{j}}^{n_0} a_j \leq b\} = \underline{j} - 1$, by Proposition 2.8 we must have $C = J_0$, hence $|C| = \overline{k} + 1$; otherwise, we have $\sum_{j=2}^{\overline{k}-n_0+\underline{j}+1} a_j + \sum_{j=\underline{j}}^{n_0} a_j > b$ and, considering that $C \setminus \{\underline{\gamma}(C)\} \subseteq \{2, \dots, n_0\}$, by Proposition 2.8 it follows that $|C| \leq |\{1, \dots, \overline{k} - n_0 + \underline{j}\} \cup \{\underline{j}, \dots, n_0\}| = \overline{k} + 1$. □

COROLLARY 3.5. *Let C be a minimal cover with respect to constraint* (2.1) *with induced inequality $X(C) \leq k_C$. Then $\underline{k} + 1 \leq |C| \leq \overline{k} + 1$ and, equivalently, $\underline{k} \leq k_C \leq \overline{k}$.*

Proof. The proof follows from Lemmas 3.1 and 3.4 and from Proposition 2.8. □

Corollary 3.5 states that $\underline{k}$ and $\overline{k}$ are, respectively, a lower and an upper bound for the right-hand side of the inequality induced by any minimal cover with respect to constraint (2.1), but it is not guaranteed that they are the tightest ones. Thus,

it would be interesting to analyze whether these bounds are reachable, that is, to determine whether there exist minimal covers with respect to constraint (2.1) with induced inequality $X(C) \leq \underline{k}$ or $X(C) \leq \overline{k}$. Lemma 3.6 shows that the lower bound $\underline{k}$ is always reachable. Nevertheless, the upper bound $\overline{k}$ is not always reachable.

LEMMA 3.6. *Let $C = \{n_0 - \underline{k}, \dots, n_0\}$ be a cover with induced inequality $X(C) \leq \underline{k}$. Then C is a minimal cover with respect to constraint* (2.1).

Proof. The proof follows from Proposition 2.8. □

Example 1 illustrates how to identify all maximal covers from the set of covers implied by a knapsack constraint by using the procedure referred to at the end of section 2.

Example 1. Consider the knapsack constraint

$$x_1 + 4x_2 + 5x_3 + 5x_4 + 7x_5 \leq 13, \tag{3.1}$$

where $x_j \in \{0,1\}\ \forall j \in \{1, \dots, 5\}$.

By Corollary 3.5 and Proposition 2.8, the minimal covers with respect to constraint (3.1) are the sets $C \subset \{1,2,3,4,5\}$ such that $|C| = 3$, $\sum_{j \in C} a_j > 13$ and $\sum_{j \in C \setminus \{\underline{\gamma}(C)\}} a_j \leq 13$, since $\underline{k} = \overline{k} = 2$. Consequently, the minimal covers with respect to (3.1) are $C_1 = \{2,3,4\}$, $C_2 = \{2,3,5\}$, $C_3 = \{2,4,5\}$, and $C_4 = \{3,4,5\}$.

For each $k \in \{1,2,3,4\}$, let $X(E(C_k)) \leq 2$ be the inequality induced by the cover $E(C_k)$. By Theorems 2.11 and 2.12, $E(C_1)$ is the unique maximal cover from the set of covers implied by constraint (3.1). Notice that $\sum_{j \in C_1 \setminus \{\overline{\gamma}(C_1)\}} a_j + a_{\overline{\gamma}(J_0 \setminus E(C_1))} = a_1 + a_2 + a_3 = 10 < 13$ and $\sum_{j \in C_k \setminus \{\overline{\gamma}(C_k)\}} a_j + a_{\overline{\gamma}(J_0 \setminus E(C_k))} = a_2 + a_3 + a_4 = 14 > 13\ \forall k \in \{2,3,4\}$.

In the following section we present some results that lead to a new procedure based on Proposition 2.8. It only obtains the minimal covers with respect to constraint (2.1) whose extensions are maximal covers from the set of covers implied by constraint (2.1). Therefore, it will not be necessary to apply either the enumerative procedure referred to at the end of section 2 or in Theorem 2.12.

4. Identification of all maximal covers from the set of covers implied by a knapsack constraint. From now on let us assume that $j_1 < \cdots < j_{|C|}$ for any cover $C = \{j_1, \dots, j_{|C|}\}$.

DEFINITION 4.1. *Let $C = \{j_1, \dots, j_{|C|}\} \subseteq J_0$ and $C' = \{j'_1, \dots, j'_{|C|}\} \subseteq J_0$ be two distinct covers with the same cardinality such that $\sum_{j \in C} a_j > b$ and $\sum_{j \in C'} a_j > b$, and let $k_0 = min\{k \in \{1, \dots, |C|\} \mid j_k \neq j'_k\}$. If $j_{k_0} < j'_{k_0}$, C is said to be* previous *to C' and C' is said to be* subsequent *to C.*

Given $k_C \in \{\underline{k}, \dots, \overline{k}\}$, let $A^0_{k_C} = \{j^0_1, \dots, j^0_{k_C+1}\}$, where $j^0_k = \min\{j \in J_0 \mid j > j^0_{k-1},\ \sum_{l=1}^{k-1} a_{j^0_l} + a_j + \sum_{l=n_0-(k_C-k)}^{n_0} a_l > b\}\ \forall k \in \{1, \dots, k_C+1\}$ and $j^0_0 = 0$.

LEMMA 4.2. *Let $k_C \in \{\underline{k}, \dots, \overline{k}\}$. Then, every cover $C \subset J_0$ such that $|C| = k_C + 1, C \neq A^0_{k_C}$ and $\sum_{j \in C} a_j > b$, is subsequent to $A^0_{k_C}$.*

Proof. The proof follows from the definition of $A^0_{k_C}$. □

DEFINITION 4.3. *A* consecutive cover *is a cover $C = \{j_1, \dots, j_{|C|}\}$ such that $j_{k+1} = j_k + 1\ \forall k \in \{1, \dots, |C| - 1\}$.*

Given $C = \{j_1, \dots, j_{|C|}\} \subset J_0$ a nonconsecutive cover such that $\sum_{j \in C} a_j > b$, let $k^*(C) = \max\{k \in \{1, \dots, |C| - 1\} \mid j_k + 1 < j_{k+1}\}$ and $A^*(C) = \{j^*_1, \dots, j^*_{|C|}\}$, where $j^*_k = j_k\ \forall k \in \{1, \dots, k^*(C) - 1\}$, $j^*_{k^*(C)} = j_{k^*(C)} + 1$ and $j^*_k = \min\{j \in J_0 \mid j > j^*_{k-1},\ \sum_{l=1}^{k-1} a_{j^*_l} + a_j + \sum_{l=n_0-(|C|-k-1)}^{n_0} a_l > b\}\ \forall k \in \{k^*(C)+1, \dots, |C|\}$. (Note that $A^*(C)$ is subsequent to C.)

LEMMA 4.4. *Let $C \subset J_0$ be a nonconsecutive cover such that $\sum_{j\in C} a_j > b$. Then every cover $C' \subset J_0$ with $|C'| = |C|$, $m_{k^*(C)}(C') = m_{k^*(C)}(A^*(C))$, $C' \neq A^*(C)$, and $\sum_{j\in C'} a_j > b$, is subsequent to $A^*(C)$.*

Proof. The proof follows from the definition of $A^*(C)$. □

Propositions 4.5 and 4.7 give some necessary conditions for the extension of a minimal cover with respect to constraint (2.1) to be a maximal cover from the set of covers implied by (2.1).

PROPOSITION 4.5. *Let $C \subset J_0$ be a nonconsecutive cover such that $\sum_{j\in C} a_j > b$ and let C' be a minimal cover with respect to constraint* (2.1) *subsequent to C with $m_{k^*(C)}(C') = m_{k^*(C)}(C)$. Then $E(C')$ is not a maximal cover from the set of covers implied by constraint* (2.1).

Proof. Let $C = \{j_1, \dots, j_{|C|}\}$ and $C' = \{j'_1, \dots, j'_{|C|}\}$. Considering that $m_{k^*(C)}(C') = m_{k^*(C)}(C)$ and C' is subsequent to C, it follows that $j'_k = j_k \ \forall k \in \{1, \dots, k^*(C)\}$ and $j'_{k^*(C)+1} \geq j_{k^*(C)+1}$. Moreover, by the definition of $k^*(C)$ we have that $j_k + 1 = j_{k+1} \ \forall k \in \{k^*(C) + 1, \dots, |C| - 1\}$; hence $\sum_{k=k^*(C)+1}^{|C|} a_{j'_k} \geq \sum_{k=k^*(C)+1}^{|C|} a_{j_k}$. Therefore $\sum_{j\in C\setminus\{\underline{\gamma}(C)\}} a_j \leq \sum_{j\in C'\setminus\{\underline{\gamma}(C')\}} a_j \leq b$, since C' is a minimal cover with respect to constraint (2.1). So, taking $X(C) \leq |C| - 1$ as the inequality induced by C, we can conclude from Proposition 2.8 that C is a minimal cover with respect to constraint (2.1). Thus, by claim (1) of Proposition 2.10, $E(C)$ is a cover implied by (2.1) and the inequality $X(E(C)) \leq |C| - 1$ is induced by it.

Suppose that $E(C')$ is a maximal cover from the set of covers implied by constraint (2.1). Then, by Propositions 2.6 and 2.8, the inequality induced by $E(C')$ is $X(E(C')) \leq |C| - 1$ and, since $E(C') \subset \{j_1, \dots, j_{k^*(C)}\} \cup \{j \in J_0 \mid j \geq j_{k^*(C)+1}\} = E(C)$, the inequality $X(E(C')) \leq |C| - 1$ is dominated by $X(E(C)) \leq |C| - 1$, which is a contradiction. Consequently, $E(C')$ is not a maximal cover from the set of covers implied by constraint (2.1). □

COROLLARY 4.6. *Let $C \subset J_0$ be a nonconsecutive cover such that $\sum_{j\in C} a_j > b$ and let C' be a minimal cover with respect to constraint* (2.1) *subsequent to C and previous to $A^*(C)$. Then $E(C')$ is not a maximal cover from the set of covers implied by constraint* (2.1).

PROPOSITION 4.7. *Let $C \subset J_0$ be a consecutive cover such that $\sum_{j\in C} a_j > b$ and let C' be a minimal cover with respect to constraint* (2.1) *subsequent to C. Then $E(C')$ is not a maximal cover from the set of covers implied by constraint* (2.1).

Proof. The proof is similar to the proof of Proposition 4.5. □

Given $k_C \in \{\underline{k}, \dots, \overline{k}\}$, let $A^l_{k_C} = A^*(A^{l-1}_{k_C}) \ \forall l \in \mathbb{N}$ such that $A^{l-1}_{k_C}$ is a nonconsecutive cover. Since the sets $A^l_{k_C}$ are distinct, there will be a finite number of indices l for which the set $A^l_{k_C}$ is defined. Let p_{k_C} be the biggest of those indices. The sets $\{A^l_{k_C}\}_{k_C\in\{\underline{k},\dots,\overline{k}\},\, l\in\{0,\dots,p_{k_C}\}}$ will lead to a characterization for the maximal covers from the set of covers implied by constraint (2.1); see Corollary 4.11.

Example 2. Consider the knapsack constraint

$$2x_1 + 3x_2 + 5x_3 + 5x_4 + 7x_5 + 8x_6 + 9x_7 \leq 19,$$

where $x_j \in \{0, 1\} \ \forall j \in \{1, \dots, 7\}$.

It can easily be verified that for this constraint $\underline{k} = 2$ and $\overline{k} = 3$.

For $k_C = 2$ we have $A^0_2 = \{2, 6, 7\}$, $A^1_2 = \{3, 5, 6\}$, $A^2_2 = \{4, 5, 6\}$, and $p_2 = 2$ (see Figure 4.1).

For $k_C = 3$ we have $A^0_3 = \{1, 2, 5, 6\}$, $A^1_3 = \{1, 3, 4, 6\}$, $A^2_3 = \{1, 3, 5, 6\}$, $A^3_3 = \{1, 4, 5, 6\}$, $A^4_3 = \{2, 3, 4, 5\}$, and $p_3 = 4$ (see Figure 4.2).

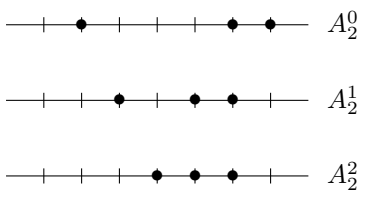


FIG. 4.1. *Graphic representation of the family* $\{A_2^l\}_{l\in\{0,1,2\}}$.

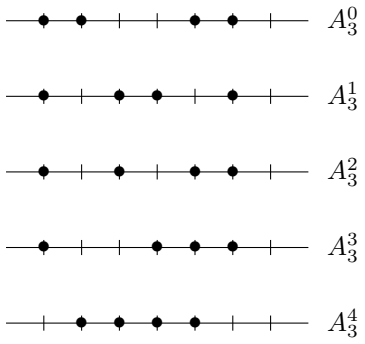


FIG. 4.2. *Graphic representation of the family* $\{A_3^l\}_{l\in\{0,1,2,3,4\}}$.

Let $C = \{j_1, \dots, j_{|C|}\} \subset J_0$ be a cover such that $\sum_{j\in C} a_j > b$. If $\underline{\gamma}(C) > 1$ or C is a nonconsecutive cover, we define $\bar{A}(C) = (C \setminus \{\overline{\gamma}(C)\}) \cup \{j_{k^*(C)+1} - 1\}$, where $k^*(C) = 0$ if C is a consecutive cover. (Note that $\bar{A}(C)$ is previous to C provided that $\sum_{j\in\bar{A}(C)} a_j > b$.)

PROPOSITION 4.8. *Let* $k_C \in \{\underline{k}, \dots, \overline{k}\}$. *If* $\underline{\gamma}(A_{k_C}^0) > 1$ *or* $A_{k_C}^0$ *is a nonconsecutive cover, then* $\sum_{j\in\bar{A}(A_{k_C}^l)} a_j \le b\ \forall l \in \{0, \dots, p_{k_C}\}$.

Proof. See Appendix A. □

COROLLARY 4.9. *Let* $k_C \in \{\underline{k}, \dots, \overline{k}\}$. *Then every consecutive cover* $C \subseteq J_0$ *such that* $|C| = k_C + 1$ *and* $\sum_{j\in C} a_j > b$ *is not previous to* $A_{k_C}^{p_{k_C}}$.

Proof. Let $C \subseteq J_0$ be a consecutive cover such that $|C| = k_C+1$ and $\sum_{j\in C} a_j > b$, and suppose that C is previous to $A_{k_C}^{p_{k_C}}$. In this case, $\underline{\gamma}(A_{k_C}^{p_{k_C}}) > \underline{\gamma}(C) \ge 1$, and, considering that $\bar{A}(A_{k_C}^{p_{k_C}}) = \{\underline{\gamma}(A_{k_C}^{p_{k_C}}) - 1, \dots, \underline{\gamma}(A_{k_C}^{p_{k_C}}) + k_C - 1\}$, by Proposition 4.8 it follows that $\sum_{j\in C} a_j \le b$, which is a contradiction. □

THEOREM 4.10. *Let* C *be a minimal cover with respect to constraint* (2.1) *and let* $X(E(C)) \le |C| - 1$ *be the inequality induced by* $E(C)$. *Then* $E(C)$ *is a maximal cover from the set of covers implied by constraint* (2.1) *if and only if* $\exists\, l \in \{0, \dots, p_{|C|-1}\}$ *such that* $C = A_{|C|-1}^l$.

Proof. ($\Rightarrow$) The proof follows from Lemma 4.2, Corollary 4.6, and Proposition 4.7.

($\Leftarrow$) If $E(C) = J_0$, by Theorem 2.12 we have that $E(C)$ is a maximal cover from the set of covers implied by constraint (2.1).

If $E(C) \subset J_0$, then $\underline{\gamma}(C) > 1$ or C is a nonconsecutive cover. Therefore $\sum_{j\in\bar{A}(C)} a_j \leq b$ by Proposition 4.8. Let $C = \{j_1, \ldots, j_{|C|}\}$. Since $\overline{\gamma}\,(J_0 \setminus E(C)) = j_{k^*(C)+1} - 1$, it follows that $\bar{A}(C) = (C \setminus \{\overline{\gamma}(C)\}) \cup \{\overline{\gamma}(J_0 \setminus E(C))\}$, from which $\sum_{j\in C\setminus\{\overline{\gamma}(C)\}} a_j + a_{\,\overline{\gamma}(J_0\setminus E(C))} \leq b$. Accordingly, by Theorem 2.12, $E(C)$ is a maximal cover from the set of covers implied by constraint (2.1). □

COROLLARY 4.11. *The set of maximal covers from the set of covers implied by constraint* (2.1) *is* $\{E(A^l_{k_C}) \mid k_C \in \{\underline{k}, \ldots, \overline{k}\},\ l \in \{0, \ldots, p_{k_C}\}, \sum_{j\in A^l_{k_C}\setminus\{\underline{\gamma}(A^l_{k_C})\}} a_j \leq b\}$.

Proof. The proof follows from Proposition 2.8, Corollary 3.5, and Theorems 2.11 and 4.10. □

LEMMA 4.12. *Let* $k_C \in \{\underline{k}, \ldots, \overline{k}\}$ *and* $l \in \{0, \ldots, p_{k_C}\}$. *Then* $\{\underline{j}, \ldots, n_0\} \subseteq A^l_{k_C}$.

Proof. Suppose that $\exists\, k \in \{\underline{j}, \ldots, n_0\} \setminus A^l_{k_C}$. In this case $A^l_{k_C} \subseteq J_0 \setminus \{k\}$, and hence $\sum_{j\in A^l_{k_C}} a_j \leq \sum_{j\in J_0\setminus\{k\}} a_j \leq b$, which is a contradiction. □

The algorithm below identifies all maximal covers from the set of covers implied by constraint (2.1) by using a procedure based on Corollary 4.11 and Lemma 4.12. In [20] it is shown that the nondominated extensions considered in [7] (see also [19]) are maximal covers from the set of covers implied by a knapsack constraint, but, in general, the procedures proposed in [7] only obtain a small fraction of the whole set of maximal covers.

ALGORITHM 1.

STEP 1. Compute $\underline{k}$, $\underline{j}$, and $\overline{k}$. Set $h = 0$, $j_0 = 0$, and $k_C = \underline{k}$.

STEP 2. Set $k_0 = 1$ and $j_k = k - k_C + n_0 - 1\ \forall k \in \{k_C - n_0 + \underline{j} + 1, \ldots, k_C + 1\}$.

STEP 3. Set $j_k = \min\{j \in J_0 \mid j > j_{k-1}, \sum_{l=1}^{k-1} a_{j_l} + a_j + \sum_{l=n_0-(k_C-k)}^{n_0} a_l > b\}\ \forall k \in \{k_0, \ldots, k_C - n_0 + \underline{j}\}$ and $C = \{j_1, \ldots, j_{k_C+1}\}$.

STEP 4. If $\sum_{k=2}^{k_C+1} a_{j_k} \leq b$, set $h = h + 1$ and $C_h = E(C)$.

STEP 5. If $j_{k_C+1} - j_1 = k_C$, go to Step 8.

STEP 6. If $\sum_{k=2}^{k_C+1} a_{j_k} > b$ and $j_{k_C+1} - j_1 = k_C + 1$, go to Step 8 (there is not any minimal cover with respect to constraint (2.1) subsequent to C).

STEP 7. Set $k^*(C) = \max\{k \in \{1, \ldots, k_C\} \mid j_k + 1 < j_{k+1}\}$ and $j_{k^*(C)} = j_{k^*(C)} + 1$. If $j_{k^*(C)} + 1 = j_{k^*(C)+1}$, set $C = \{j_1, \ldots, j_{k_C+1}\}$ and go to Step 4. Otherwise, set $k_0 = k^*(C) + 1$ and go to Step 3.

STEP 8. If $k_C < \overline{k}$, set $k_C = k_C + 1$ and go to Step 2. Otherwise, STOP (all maximal covers from the set of covers implied by constraint (2.1) have been identified).

Note. For $k_C = \underline{k}$, all of the covers $\{j_1, \ldots, j_{k_C+1}\}$ identified by Algorithm 1 are such that $\sum_{k=2}^{k_C+1} a_{j_k} \leq \sum_{j=n_0-(\underline{k}-1)}^{n_0} a_j \leq b$.

The problem of identifying all maximal covers from the set of covers implied by a 0-1 knapsack constraint is intractable in the sense that the number of this type of maximal covers cannot be bounded by a polynomial function of the number of variables in the knapsack constraint; see [13]. Therefore, every algorithm that solves this problem (in particular, Algorithm 1) will have exponential time complexity.

Example 3. Consider the knapsack constraint

$$x_1 + 2x_2 + 2x_3 + 3x_4 + 5x_5 + 7x_6 + 7x_7 \leq 15, \tag{4.1}$$

where $x_j \in \{0, 1\}\ \forall j \in \{1, \ldots, 7\}$.

Appendix B gives the steps of Algorithm 1 to identify the maximal covers from the set of covers implied by constraint (4.1). These maximal covers are $C_1 = \{2, 6, 7\}, C_2 = \{3, 6, 7\}$, $C_3 = \{4, 6, 7\}$, $C_4 = \{5, 6, 7\}$, $C_5 = \{1, 4, 5, 6, 7\}$, $C_6 = \{2, 3, 5, 6, 7\}, C_7 = \{2, 4, 5, 6, 7\}$, and $C_8 = \{3, 4, 5, 6, 7\}$. Their induced inequalities are

$$
\begin{array}{rrrrrrrrrrrrrrr}
 & x_2 & & & & & & & + & x_6 & + & x_7 & \leq & 2, \\
 & & & x_3 & & & & & + & x_6 & + & x_7 & \leq & 2, \\
 & & & & & x_4 & & & + & x_6 & + & x_7 & \leq & 2, \\
 & & & & & & & x_5 & + & x_6 & + & x_7 & \leq & 2, \\
x_1 & & & & + & x_4 & + & x_5 & + & x_6 & + & x_7 & \leq & 3, \\
 & x_2 & + & x_3 & & & + & x_5 & + & x_6 & + & x_7 & \leq & 3, \\
 & x_2 & & & + & x_4 & + & x_5 & + & x_6 & + & x_7 & \leq & 3, \\
 & & & x_3 & + & x_4 & + & x_5 & + & x_6 & + & x_7 & \leq & 3.
\end{array}
$$

If the inequality induced by a maximal cover from the set of covers implied by constraint (2.1) is dominated by the same (2.1), then it will not be necessary to append it to the constraint system of problem (1.1), since the resulting formulation will not be tighter than the current one. Nevertheless, in certain cases this type of maximal cover can help to reformulate some constraints of (1.1); see [10, 20]. Propositions 4.13 and 4.14 lead to a characterization of the maximal covers from the set of covers implied by constraint (2.1) whose induced inequalities are dominated by (2.1). This characterization is stated in Corollary 4.15.

PROPOSITION 4.13. *Let C be a maximal cover from the set of covers implied by constraint* (2.1) *whose induced inequality is dominated by constraint* (2.1). *Then*

(1) $C = J_0$.

(2) $a_j = \frac{b}{\overline{k}}\ \forall j \in \{1, \ldots, \overline{k} + 1\}$.

(3) $a_j = \frac{b}{\overline{k}}\ \forall j \in J_0$ *if* $\underline{k} = \overline{k}$.

Proof. See Appendix A. □

PROPOSITION 4.14. *Let $X(J_0) \leq \overline{k}$ be the inequality induced by J_0. If $a_j = \frac{b}{\overline{k}}\ \forall j \in \{1, \ldots, \overline{k} + 1\}$, then J_0 is a maximal cover from the set of covers implied by constraint* (2.1)*, and its induced inequality is dominated by constraint* (2.1).

Proof. If $a_j = \frac{b}{\overline{k}}\ \forall j \in \{1, \ldots, \overline{k} + 1\}$, by Proposition 2.8 and Theorem 2.12 it follows that J_0 is a maximal cover from the set of covers implied by constraint (2.1).

Let $(x_j)_{j \in J} \in [0, 1]^n$ be such that $\sum_{j \in J_0} a_j x_j \leq b$. By considering that $\frac{\overline{k}}{b}\ a_j \geq 1\ \forall j \in J_0$, we get that $\sum_{j \in J_0} x_j \leq \frac{\overline{k}}{b} \sum_{j \in J_0} a_j x_j \leq \frac{\overline{k}}{b}\ b = \overline{k}$. Consequently, the inequality induced by J_0 is dominated by constraint (2.1). □

COROLLARY 4.15. *Let C be a maximal cover from the set of covers implied by constraint* (2.1). *Then the inequality induced by C is dominated by constraint* (2.1) *if and only if $C = J_0$ and $a_j = \frac{b}{\overline{k}}\ \forall j \in \{1, \ldots, \overline{k} + 1\}$.*

Proof. The proof follows from Propositions 2.6, 4.13, and 4.14. □

Note. If constraint (2.1) is such that $a_j = a_{j'}\ \forall j, j' \in J_0$, then by using Corollary 4.11 it is easy to see that the unique maximal cover from the set of covers implied by (2.1) is J_0. Its induced inequality coincides with constraint (2.1) after applying the Euclidean reduction procedure (see [15, 20], among others).

In section 6.1 it will be shown that the number of maximal covers from the set of covers implied by a knapsack constraint can be very large. Therefore, it may be advisable to introduce some modifications into Algorithm 1 to restrict the maximal covers that it is allowed to identify. For example, if problem (1.1) is solved within a branch-and-cut framework, at each node one can opt for identifying only maximal covers whose induced inequalities are violated by the optimal solution to the current

LP subproblem, provided that this optimal solution is fractional. In the next section we present a modification of Algorithm 1 to identify only certain maximal covers whose induced inequalities are violated by a given fractional solution.

5. Identification of maximal covers from the set of covers implied by a knapsack constraint whose induced inequalities are violated by a given fractional solution. Let $(\overline{x}_j)_{j\in J}$ be a fractional feasible solution for the LP relaxation of problem (1.1). We are interested in identifying maximal covers from the set of covers implied by constraint (2.1), say, $E(C)$, whose induced inequalities are violated by $(\overline{x}_j)_{j\in J}$, that is, $\sum_{j\in E(C)} \overline{x}_j > |C|-1$. Since the computational effort required for obtaining all of these maximal covers can be excessive, it may be preferable to identify only maximal covers such that the violation of their induced inequalities reaches a certain threshold. Now, considering that $\sum_{j\in E(C)} \overline{x}_j \le \sum_{j\in J_0} \overline{x}_j$, we find that a possible choice is to identify maximal covers such that $\sum_{j\in E(C)} \overline{x}_j > \max\{|C|-1, \alpha \sum_{j\in J_0} \overline{x}_j\}$, where $\alpha \in [0,1)$. (Note that all maximal covers from the set of covers implied by (2.1) whose induced inequalities are violated by $(\overline{x}_j)_{j\in J}$ will be obtained if we take $\alpha = 0$.)

Let $\overline{k}' = \min\{\overline{k}, [\sum_{j\in J_0} \overline{x}_j]\}$ and let us assume that $\underline{k} \le \overline{k}'$, since, otherwise, there would not be any maximal covers from the set of covers implied by constraint (2.1) whose induced inequalities are violated by $(\overline{x}_j)_{j\in J}$.

The algorithm below identifies all maximal covers from the set of covers implied by constraint (2.1) such that $\sum_{j\in E(C)} \overline{x}_j > \max\{|C|-1, \alpha \sum_{j\in J_0} \overline{x}_j\}$ by using a procedure based on Algorithm 1.

ALGORITHM 2.

STEP 1. Compute $\underline{k}$, $\underline{j}$, and $\overline{k}'$. If $\underline{j} > n_0 - \underline{k}$, set $h = 0$, $j_0 = 0$, $k_C = \underline{k}$, and go to Step 2. If $\sum_{j=\underline{j}}^{n_0} \overline{x}_j > \max\{\underline{k}, \alpha \sum_{j\in J_0} \overline{x}_j\}$, set $C_1 = \{\underline{j}, \dots, n_0\}$. STOP.

STEP 2. Set $j_k = k - k_C + n_0 - 1$ $\forall k \in \{k_C - n_0 + \underline{j} + 1, \dots, k_C + 1\}$, $k = 1$ and $k_\alpha = \max\{k_C, \alpha \sum_{j\in J_0} \overline{x}_j\}$.

STEP 3. Set $j_k = \min\{j \in J_0 \mid j > j_{k-1}, \sum_{l=1}^{k-1} a_{j_l} + a_j + \sum_{l=n_0-(k_C-k)}^{n_0} a_l > b\}$. If $k = k_C + 1$, go to Step 6.

STEP 4. If $k = k_C - n_0 + \underline{j}$, go to Step 7.

STEP 5. If $\sum_{l=1}^{k-1} \overline{x}_{j_l} + \sum_{l=j_k}^{n_0} \overline{x}_l > k_\alpha$, set $k = k+1$ and go to Step 3. Otherwise, go to Step 8.

STEP 6. If $\sum_{l=1}^{k_C} \overline{x}_{j_l} + \sum_{l=j_k}^{n_0} \overline{x}_l > k_\alpha$, set $C = \{j_1, \dots, j_{k_C+1}\}$ and go to Step 10. Otherwise, go to Step 9.

STEP 7. If $\sum_{l=1}^{k} \overline{x}_{j_l} + \sum_{l=\underline{j}}^{n_0} \overline{x}_l > k_\alpha$, set $C = \{j_1, \dots, j_{k_C+1}\}$ and go to Step 10. If $j_k < \underline{j} - 1$, set $j_k = j_k + 1$ and repeat Step 7.

STEP 8. If $k = 1$, go to Step 16.

STEP 9. Set $k = k-1$, $j_k = j_k + 1$ and go to Step 5.

STEP 10. If $\sum_{l=2}^{k_C+1} a_{j_l} \le b$, set $h = h+1$ and $C_h = E(C)$.

STEP 11. If $j_{k_C+1} - j_1 = k_C$, go to Step 16.

STEP 12. If $\sum_{l=2}^{k_C+1} a_{j_l} > b$ and $j_{k_C+1} - j_1 = k_C + 1$, go to Step 16.

STEP 13. Set $k^*(C) = \max\{l \in \{1, \dots, k_C\} \mid j_l + 1 < j_{l+1}\}$.

STEP 14. Set $j_{k^*(C)} = j_{k^*(C)} + 1$. If $j_{k^*(C)} + 1 < j_{k^*(C)+1}$, set $k = k^*(C)$ and go to Step 4.

STEP 15. If $\sum_{l=1}^{k^*(C)-1} \overline{x}_{j_l} + \sum_{l=j_{k^*(C)}}^{n_0} \overline{x}_l > k_\alpha$, set $C = \{j_1, \dots, j_{k_C+1}\}$ and go to Step 10. If $k^*(C) > 1$, set $k^*(C) = k^*(C) - 1$ and go to Step 14.

STEP 16. If $k_C < \overline{k}'$, set $k_C = k_C + 1$ and go to Step 2. Otherwise, STOP.

Example 4. Consider the 0-1 LP problem

$$\begin{array}{lll} & \text{Max} & 2x_1 + 11x_2 + x_3 + 5x_4 + 3x_5 + 12x_6 + 2x_7 \\ (5.1) & \text{subject to} & x_1 + 2x_2 + 2x_3 + 3x_4 + 5x_5 + 7x_6 + 7x_7 \leq 15, \\ & & x_j \in \{0,1\}\ \forall j \in \{1,\dots,7\}. \end{array}$$

(Note that the feasible region of problem (5.1) is defined by constraint (4.1); see Example 3.)

Let us apply a branch-and-cut procedure to problem (5.1). Let $(\overline{x}_1,\dots,\overline{x}_7)$ be the optimal solution to the LP relaxation of (5.1). It can be shown that $\overline{x}_1 = \overline{x}_2 = \overline{x}_4 = \overline{x}_6 = 1$, $\overline{x}_5 = \frac{2}{5}$, and $\overline{x}_3 = \overline{x}_7 = 0$.

Appendix C gives the steps of Algorithm 2 to identify the maximal covers from the set of covers implied by constraint (4.1) whose induced inequalities are violated by $(\overline{x}_1,\dots,\overline{x}_7)$ (for this purpose, we have taken $\alpha = 0$). These maximal covers are $C_1 = \{1,4,5,6,7\}$ and $C_2 = \{2,4,5,6,7\}$. Their induced inequalities are

$$\begin{array}{lllllllllll} x_1 & & + & x_4 & + & x_5 & + & x_6 & + & x_7 & \leq & 3, \\ & x_2 & + & x_4 & + & x_5 & + & x_6 & + & x_7 & \leq & 3. \end{array}$$

It can be shown that the optimal solution to the LP relaxation of the problem that results from appending these inequalities to the formulation of problem (5.1) is $x_1^* = x_2^* = x_3^* = x_4^* = x_6^* = 1$ and $x_5^* = x_7^* = 0$. Therefore, $(x_1^*,\cdots,x_7^*)$ is an optimal solution to (5.1) as well.

6. Computational experience.

6.1. Results on randomly generated knapsack constraints. In this section we report some computational experiments where Algorithm 1 has been applied to 0-1 knapsack constraints of the form $\sum_{j=1}^{n_0} a_j x_j \leq b$ with $a_1 \leq \cdots \leq a_{n_0}$. For each fixed value of n_0, the coefficients $a_1,\dots,a_{n_0}$ have been randomly generated from the set $\{1,\dots,1000\}$ (the Quicksort procedure has been used for sorting them in nondecreasing order; see subroutines "sort" and "indexx" in sections 8.2 and 8.4 of [27], respectively), and several right-hand sides $b \in \{a_{n_0},\dots,\sum_{j=1}^{n_0} a_j - 1\}$ have been considered. Let ρ denote the relative position of b within its variation range, i.e., $\rho = \frac{b-a_{n_0}}{\sum_{j=1}^{n_0-1} a_j - 1}$ (note that $\rho \in [0,1]$ provided that $\sum_{j=1}^{n_0-1} a_j \neq 1$). Let m_0 denote the number of maximal covers from the set of covers implied by the constraint $\sum_{j=1}^{n_0} a_j x_j \leq b$ and t the CPU time (expressed in seconds) required by Algorithm 1 to identify all of them.

The implementation platform is Microsoft FORTRAN PowerStation v4.0 and Pentium III, 1000 Mhz, 256 Mb RAM.

Tables 6.1–6.4 show the values of b, ρ, m_0, and t for different values of n_0.

We can observe in Table 6.1 that the computational effort required by Algorithm 1 is practically null and the number of maximal covers is small. Therefore, it is advisable to identify all of them.

In Table 6.2 the computational effort remains practically null. However, for some values of b, the number of maximal covers becomes too large to keep all of them stored in a cover pool.

In Tables 6.3 and 6.4 it is worth noting the large number of maximal covers that can be obtained from knapsack constraints with relatively few variables. This fact reveals the need for imposing an upper bound on the number of maximal covers that Algorithm 1 is allowed to identify.

TABLE 6.1
Computational results for $n_0 = 10$.

b	ρ	m_0	t
966	0.0000	8	0.00
1500	0.1001	18	0.00
1850	0.2039	24	0.00
2300	0.3077	32	0.00
2700	0.4000	33	0.00
3150	0.5038	30	0.00
3600	0.6076	22	0.00
4010	0.7022	18	0.00
4450	0.8037	10	0.00
5100	0.9536	1	0.00

TABLE 6.2
Computational results for $n_0 = 20$.

b	ρ	m_0	t
945	0.0000	100	0.00
1800	0.1022	831	0.00
2700	0.2098	4 215	0.00
3500	0.3054	10 318	0.00
4300	0.4010	16 586	0.00
5200	0.5085	17 218	0.00
6000	0.6042	11 352	0.00
6850	0.7057	4 321	0.00
7700	0.8073	834	0.00
8900	0.9508	12	0.00

TABLE 6.3
Computational results for $n_0 = 30$.

b	ρ	m_0	t
956	0.0000	2 913	0.00
2100	0.1085	126 729	0.11
3100	0.2033	995 051	0.87
4200	0.3077	4 078 936	2.74
5200	0.4025	7 835 604	5.11
6300	0.5068	8 247 850	5.71
7300	0.6017	4 642 029	3.73
8400	0.7060	1 178 565	1.43
9400	0.8008	150 130	0.22
11000	0.9526	320	0.00

TABLE 6.4
Computational results for $n_0 = 40$.

b	ρ	m_0	t
999	0.0000	154	0.00
3500	0.1057	334 805	0.17
5800	0.2028	28 614 341	6.54
8200	0.3043	575 346 532	109.63
10500	0.4014	2 907 873 285	517.18
13000	0.5071	4 710 541 502	807.02
15200	0.6000	2 443 519 127	417.38
17600	0.7014	349 382 467	64.38
20000	0.8028	10 718 552	3.07
23500	0.9507	621	0.00

We have carried out an extensive computational experience on randomly generated knapsack constraints with up to 10,000 variables allowing a maximum of 1000 identified maximal covers. In all cases, the CPU time required by Algorithm 1 has been almost null.

6.2. Results on single source capacitated plant location problem instances. Given a set of potential plant locations with known capacities and a set of customers with known demands, the *single source capacitated plant location problem (SSCPLP)* is to minimize the total cost for opening a plant at each selected location and assigning the customers to those plants in such a way that the demand of each customer is served from a single plant and the total demand supplied from each plant does not exceed its capacity.

Let n_l denote the number of potential locations to open the plants, n_c the number of customers, f_j the cost of opening a plant at location j, c_{ij} the cost of assigning customer i to the plant located at j, d_i the demand of customer i, and b_j the capacity of the plant located at j.

Defining the variables

$$y_j = \begin{cases} 1 \text{ if a plant is opened at location } j \\ 0 \text{ otherwise} \end{cases} \quad \forall j \in \{1, \dots, n_l\} \text{ and}$$

$$x_{ij} = \begin{cases} 1 \text{ if customer } i \text{ is assigned to the plant located at } j \\ 0 \text{ otherwise} \end{cases} \quad \forall i \in \{1, \dots, n_c\}, \forall j \in \{1, \dots, n_l\},$$

the SSCPLP can be formulated as follows:

$$\begin{aligned} \text{Min} \quad & \sum_{j=1}^{n_l} f_j y_j + \sum_{i=1}^{n_c} \sum_{j=1}^{n_l} c_{ij} x_{ij} \\ \text{subject to} \quad & \sum_{j=1}^{n_l} x_{ij} = 1 && \forall i \in \{1, \dots, n_c\}, \\ & \sum_{i=1}^{n_c} d_i x_{ij} \le b_j y_j && \forall j \in \{1, \dots, n_l\}, \\ & y_j \in \{0, 1\} && \forall j \in \{1, \dots, n_l\}, \\ & x_{ij} \in \{0, 1\} && \forall i \in \{1, \dots, n_c\}, \forall j \in \{1, \dots, n_l\}. \end{aligned} \tag{6.1}$$

Exact algorithms for the SSCPLP are based on column generation procedures and Lagrangian relaxations; see [3, 5, 16, 23], among others.

TABLE 6.5
Problem dimensions.

Problems	n_l	n_c
$p1$-$p6$	10	20
$p7$-$p17$	15	30
$p18$-$p25$	20	40
$p26$-$p33$	20	50

In this section we present the computational results obtained when applying three different branch-and-cut algorithms to several SSCPLP instances. These instances can be downloaded from website http://www-eio.upc.es/~elena/sscplp/index.html. Table 6.5 contains their dimensions.

The purpose of these computational experiments is not to find a branch-and-cut algorithm that uses maximal covers and can compete against the current exact algorithms for the SSCPLP, but to show that maximal cover identification can improve the performance of the traditional branch-and-cut method.

The implementation platform is Microsoft Visual C++ v5.0, CPLEX v7.0 (see [17]), and Pentium III, 667 Mhz, 320 Mb RAM.

We have run the CPLEX mixed integer optimizer by using the default rules, except that generation of any type of cover cuts has not been allowed, a higher priority has been assigned to variables $y_1, \ldots, y_{n_l}$ in the branching process, the relative and absolute optimality tolerances have been set to zero, and a time limit of 30 minutes has been imposed.

We have followed three different strategies to apply the CPLEX branch-and-cut algorithm to the SSCPLP instances.

The first strategy is to run CPLEX alone, without identifying maximal covers. The related computational results are given in column "CPLEX alone" of Table 6.6.

TABLE 6.6
Comparative computational results.

	CPLEX alone			CPLEX + Alg. 1				CPLEX + Alg. 2			
Prob.	Nodes	Time	Best Val.	Nodes	Time	Best Val.	M. Cov.	Nodes	Time	Best Val.	M. Cov.
$p1$	482322	405.13	2014	326	1.98	2014	132	857	3.73	2014	117
$p2$	250771	224.98	4251	580	3.68	4251	214	1022	4.94	4251	189
$p3$	5539	4.07	6051	212	1.16	6051	98	219	1.43	6051	92
$p4$	349132	244.36	7168	672	2.20	7168	138	723	3.46	7168	214
$p5$	311654	230.91	4551	4397	15.33	4551	283	1316	5.87	4551	223
$p6$	1533	1.54	2269	174	1.26	2269	89	168	0.99	2269	56
$p7$	1273177	>1800	• 4552	165489	>1800	• 4407	173	5136	76.79	4366	815
$p8$	1253406	>1800	• 8455	108733	>1800	• 8457	448	7831	161.26	7926	976
$p9$	1143537	>1800	• 2509	10890	63.38	2480	410	4808	44.22	2480	456
$p10$	837850	1053.86	23112	47188	359.54	23112	373	1586	12.08	23112	276
$p11$	748924	980.64	3447	14905	97.55	3447	436	5551	80.46	3447	686
$p12$	80999	95.29	3711	14607	43.94	3711	197	1352	13.51	3711	391
$p13$	69235	78.71	3760	4876	14.00	3760	165	2223	17.52	3760	414
$p14$	577505	589.62	5965	52276	188.12	5965	436	12001	199.10	5965	1448
$p15$	1174835	>1800	• 7827	7273	51.57	7816	371	8490	91.73	7816	685
$p16$	1146998	>1800	11543	3029	24.66	11543	301	5259	65.19	11543	649
$p17$	187097	223.77	9884	77856	286.22	9884	174	4509	41.68	9884	564
$p18$	830052	>1800	• 16799	95851	>1800	• 16902	578	7162	195.53	15607	1067
$p19$	126844	223.28	18683	39961	132.70	18683	117	519	5.82	18683	162
$p20$	831332	>1800	• 28382	87011	>1800	• 28499	686	23584	>1800	• 26724	2694
$p21$	839309	>1800	• 8013	55011	>1800	• 7848	943	24909	>1800	• 7379	1698
$p22$	654600	1402.64	3271	21251	114.74	3271	230	879	13.18	3271	315
$p23$	29186	56.19	6036	3964	14.78	6036	117	2467	37.18	6036	511
$p24$	6174	12.14	6327	1726	5.99	6327	79	214	2.53	6327	86
$p25$	337	1.64	8947	58	0.55	8947	15	239	3.79	8947	64
$p26$	646726	>1800	• 4738	159662	>1800	• 4855	200	25392	>1800	• 4684	2640
$p27$	42795	89.36	10921	4278	13.18	10921	50	2422	21.42	10921	39
$p28$	456869	1028.10	11117	73131	203.83	11117	36	20368	416.83	11117	1088
$p29$	648881	1465.69	9832	110364	>1800	• 9925	261	25006	676.07	9832	1097
$p30$	680870	>1800	• 11757	66112	>1800	• 11314	811	19189	>1800	• 11162	2523
$p31$	620110	>1800	• 4725	101200	>1800	• 4664	268	11330	334.77	4466	1235
$p32$	2250	5.93	9881	3392	10.88	9881	26	4829	66.51	9881	322
$p33$	705161	>1800	• 41796	72696	>1800	• 43170	1073	20221	>1800	• 41343	3222

The second strategy is to generate a maximal cover pool whose induced inequalities will be used as cutting planes. This pool is generated before starting the branch-and-cut method. Since the 0-1 knapsack constraints $\sum_{i=1}^{n_c} d_i x_{ij} \leq b_j \ \forall j \in \{1, \ldots, n_l\}$ are valid for the feasible region of problem (6.1), Algorithm 1 is applied to these constraints imposing an upper bound on the number of maximal covers that is allowed to identify for the pool in such a way that the number of maximal covers obtained from each constraint will be approximately the same. (The Quicksort procedure has been used for sorting the coefficients $d_1, \ldots, d_{n_c}$ in nondecreasing order; see routines "sort" and "indexx" in sections 8.2 and 8.4 of [26], respectively.) At each branch-and-cut node, CPLEX appends to the formulation the inequalities induced by the maximal covers from the pool that are violated by the optimal solution to the current LP subproblem. For the instances under consideration, the best general computational results have been obtained allowing a maximum of $6n_l n_c$ identified maximal covers for the pool. These computational results are given in column "CPLEX + Alg. 1" of Table 6.6.

Finally, the third strategy is to apply Algorithm 2 at each branch-and-cut node to the 0-1 knapsack constraints $\sum_{i=1}^{n_c} d_i x_{ij} \leq b_j \ \forall j \in \{1, \ldots, n_l\}$ to identify maximal covers from the set of covers implied by these constraints whose induced inequalities are violated by the optimal solution to the current LP subproblem, say, $\big((\overline{y}_j)_{j\in\{1,\ldots,n_l\}}, (\overline{x}_{ij})_{i\in\{1,\ldots,n_c\},\, j\in\{1,\ldots,n_l\}}\big)$. Whenever a maximal cover is identified, CPLEX appends to the formulation its induced inequality. For the instances under consideration, the best general computational results have been obtained taking $\alpha = 0.95$, applying Algorithm 2 only to the constraints such that $\sum_{i=1}^{n_c} d_i \overline{x}_{ij} = b_j \overline{y}_j$ and $\overline{y}_j = 1$, allowing at most one identified maximal cover from each constraint at each branch-and-cut node, and imposing a limit of 400 iterations per constraint for setting each index j_k. These computational results are given in column "CPLEX + Alg. 2" of Table 6.6.

The headings in Table 6.6 are as follows: Nodes is the number of branch-and-cut nodes evaluated; Time is the CPU time expressed in seconds; Best Val. is the value of the objective function at the incumbent solution (a value preceded by • indicates that the incumbent solution is not proved to be optimal); and M. Cov. is the number of maximal covers whose induced inequalities have been appended to the original formulation.

We can observe from Table 6.6 that for most of the instances under consideration, the second and third strategies outperform the first strategy. In fact, there is only one instance ($p32$) in which the first strategy is the fastest one. For those instances in which the second strategy is the fastest ($p1, p2, p3, p4, p13, p14, p15, p16, p23, p25, p27$, $p28$), the difference between the CPU times required by the second and third strategies is, in general, quite small. However, for those instances where the third strategy is the fastest ($p5, p6, p7, p8, p9, p10, p11, p12, p17, p18, p19, p22, p24, p29, p31$), this difference is much bigger. Moreover, for those instances that cannot be solved to optimality by any of the three strategies ($p20, p21, p26, p30, p33$), the best incumbent solution is given by the third strategy. Thus, we propose to use the third strategy.

In the above computational experimentation, the inequalities induced by the identified maximal covers have only been used as cutting planes in a branch-and-cut framework. It is likely that if they were also used in other ways for tightening the original models (see section 2), the efficiency of our maximal covers would increase.

On the other hand, it can be expected that if the inequalities induced by our maximal covers are treated in a similar fashion as the constraints generated in [3],

they will also lead to a tightening of the bounds given by Lagrangian-relaxation-based methods. If so, maximal covers could help to improve the performance of the current exact algorithms for the SSCPLP.

7. Conclusions. In this paper a new procedure for identifying all maximal covers from the set of covers implied by a 0-1 knapsack constraint has been presented. It does not require one to check explicitly whether a characterization for this type of covers is satisfied; it only requires one to check whether the initial covers that it determines are minimal with respect to the knapsack constraint under consideration.

It is well known that cover inequalities have numerous applications in 0-1 model tightening. Therefore, our procedure can be very useful for this purpose.

A modification of this procedure to identify only maximal covers whose induced inequalities are violated by a given fractional solution has been also presented.

Some computational experiments with randomly generated knapsack constraints have been reported. They show that, in general, due to its large cardinality, the whole set of maximal covers from the set of covers implied by a knapsack constraint can only be stored when the number of variables is small enough. Consequently, we propose to impose a limit on the number of covers that our procedure is allowed to identify (this limit will be problem dependent). In this way, we manage to get a considerable quantity of maximal covers with practically null computational effort.

Some computational experiments with single source capacitated plant location problem instances have been also reported. The optimization engine CPLEX v7.0 has been used to check the efficiency of treating the inequalities induced by certain maximal covers as cutting planes in a branch-and-cut framework. We believe that the results show promise in solving 0-1 linear programming problems.

The utilization of our maximal covers for constraint tightening, redundancy, and infeasibility detection and variable fixing is an area of future research.

Appendix A. Proofs of Propositions 4.8 and 4.13.

Proof of Proposition 4.8. Suppose that $\sum_{j\in\bar{A}(A^0_{k_C})} a_j > b$. In this case the cover $\bar{A}(A^0_{k_C})$ is previous to $A^0_{k_C}$, which contradicts Lemma 4.2. Accordingly, we must have $\sum_{j\in\bar{A}(A^0_{k_C})} a_j \le b$.

Let $l_0 \in \{0,\dots,p_{k_C}-1\}$ and suppose that $\sum_{j\in\bar{A}(A^{l_0}_{k_C})} a_j \le b$. If it can be shown that $\sum_{j\in\bar{A}(A^{l_0+1}_{k_C})} a_j \le b$, then by induction it will follow that $\sum_{j\in\bar{A}(A^{l}_{k_C})} a_j \le b\ \forall l \in \{0,\dots,p_{k_C}\}$.

Let $A^{l_0}_{k_C} = \{j_1,\dots,j_{k_C+1}\}$ and $A^{l_0+1}_{k_C} = \{j^*_1,\dots,j^*_{k_C+1}\}$. By the definition of $A^{l_0+1}_{k_C}$ we have that $j^*_k = j_k\ \forall k \in \{1,\dots,k^*(A^{l_0}_{k_C})-1\}$ and $j^*_{k^*(A^{l_0}_{k_C})} = j_{k^*(A^{l_0}_{k_C})}+1$.

- If $j^*_k+1 = j^*_{k+1}\ \forall k \in \{k^*(A^{l_0}_{k_C}),\dots,k_C\}$, then $k^*(A^{l_0+1}_{k_C}) = k^*(A^{l_0}_{k_C})-1$ and $j^*_k \le j_k\ \forall k \in \{k^*(A^{l_0}_{k_C})+1,\dots,k_C+1\}$. Therefore $\sum_{k=1}^{k^*(A^{l_0}_{k_C})-1} a_{j^*_k} = \sum_{k=1}^{k^*(A^{l_0}_{k_C})-1} a_{j_k}$, $a_{j^*_{k^*(A^{l_0}_{k_C})}} = a_{j_{k^*(A^{l_0}_{k_C})+1}} \le a_{j_{k^*(A^{l_0}_{k_C})+1}-1}$, $\sum_{k=k^*(A^{l_0}_{k_C})+1}^{k_C} a_{j^*_k} \le \sum_{k=k^*(A^{l_0}_{k_C})+1}^{k_C} a_{j_k}$ and $a_{j^*_{k^*(A^{l_0+1}_{k_C})+1}-1} = a_{j^*_{k^*(A^{l_0}_{k_C})}-1} = a_{j_{k^*(A^{l_0}_{k_C})}}$. Hence, by the definition of $\bar{A}(A^{l_0+1}_{k_C})$, we have $\sum_{j\in\bar{A}(A^{l_0+1}_{k_C})} a_j \le \sum_{k=1}^{k^*(A^{l_0}_{k_C})-1} a_{j_k} + a_{j_{k^*(A^{l_0}_{k_C})+1}-1} + \sum_{k=k^*(A^{l_0}_{k_C})+1}^{k_C} a_{j_k} + a_{j_{k^*(A^{l_0}_{k_C})}} = \sum_{j\in\bar{A}(A^{l_0}_{k_C})} a_j \le b$.

- If $\exists\, k \in \{k^*(A^{l_0}_{k_C}), \dots, k_C\}$ with $j^*_k + 1 < j^*_{k+1}$, then $j^*_{k^*(A^{l_0+1}_{k_C})+1} - 1 > j^*_{k^*(A^{l_0}_{k_C})}$, from which $m_{k^*(A^{l_0}_{k_C})}(\bar{A}(A^{l_0+1}_{k_C})) = m_{k^*(A^{l_0}_{k_C})}(A^{l_0+1}_{k_C})$. Consequently, taking $C = A^{l_0}_{k_C}$ and $C' = \bar{A}(A^{l_0+1}_{k_C})$ in Lemma 4.4, we can conclude that $\sum_{j\in\bar{A}(A^{l_0+1}_{k_C})} a_j \le b$. □

Proof of Proposition 4.13. By Theorem 2.11 and Propositions 2.6 and 2.8, the inequality induced by C is $X(C) \le |C'| - 1$, where C' is the unique minimal cover with respect to constraint (2.1) such that $E(C') = C$.

(1) Suppose that $\exists\, k \in C'$ with $\sum_{j\in C'\setminus\{k\}} a_j < b$. In this case, taking $x_j = 1\ \forall j \in C' \setminus \{k\}$, $x_j = 0\ \forall j \in J \setminus C'$ and $x_k \in \big(0, \frac{b-\sum_{j\in C'\setminus\{k\}} a_j}{a_k}\big]$ we obtain a solution in $[0,1]^n$ that satisfies constraint (2.1) but not the inequality induced by C, which is a contradiction. Consequently, we must have $\sum_{j\in C'\setminus\{k\}} a_j = b\ \forall k \in C'$, hence $\sum_{j\in C'\setminus\{\overline{\gamma}(C')\}} a_j + a_k > b\ \forall k \in J_0 \setminus C$ and, by Theorem 2.12, it follows that $C = J_0$.

(2) Considering that $E(C') = C = J_0$, we get that $C' = \{1, \dots, |C'|\}$. On the other hand, by the proof of claim (1) above we have that $\sum_{j\in C'\setminus\{k\}} a_j = b\ \forall k \in C'$, from which $a_j = \frac{b}{|C'|-1}\ \forall j \in C'$.

If $C' \subset J_0$, then $\sum_{j=1}^{n_0-1} a_j \ge \sum_{j\in C'} a_j > b$; therefore, $\underline{j} = n_0 + 1$ and $\overline{k} = -1 + \max\{k \in J_0 \mid \sum_{j=2}^{k} a_j \le b\} = |C'| - 1$. If $C' = J_0$, then $\underline{j} = 1$ and $\overline{k} = \max\{k \in \{1, \dots, n_0 - 1\} \mid \sum_{j=n_0-(k-1)}^{n_0} a_j \le b\} = |C'| - 1$. Accordingly, it follows that $a_j = \frac{b}{\overline{k}}\ \forall j \in \{1, \dots, \overline{k} + 1\}$.

(3) If $\underline{k} = \overline{k}$, then $\sum_{j=n_0-(\overline{k}-1)}^{n_0} a_j \le b$ and, since $a_j = \frac{b}{\overline{k}}\ \forall j \in \{1, \dots, \overline{k} + 1\}$, we have that $a_j \ge \frac{b}{\overline{k}}\ \forall j \in J_0$, hence $\sum_{j=n_0-(\overline{k}-1)}^{n_0} a_j \ge \overline{k}\ \frac{b}{\overline{k}} = b$. Therefore, $\sum_{j=n_0-(\overline{k}-1)}^{n_0} a_j = b$, and, thus, $a_j = \frac{b}{\overline{k}}\ \forall j \in \{n_0 - (\overline{k} - 1), \dots, n_0\}$, from which $a_j = \frac{b}{\overline{k}}\ \forall j \in J_0$. □

Appendix B. Steps of Algorithm 1 in Example 3.

Algorithm 1 proceeds as follows when it is applied to constraint (4.1):

Step 1. $\underline{k} = 2$, $\underline{j} = 8$, $\overline{k} = 4$, $h = 0$, $j_0 = 0$, $k_C = 2$.
Step 2. $k_0 = 1$.
Step 3. $j_1 = 2$, $j_2 = 6$, $j_3 = 7$, $C = \{2, 6, 7\}$.
Step 4. $h = 1$, $C_1 = \{2, 6, 7\}$.
Step 7. $k^*(C) = 1$, $j_1 = 3$, $k_0 = 2$.
Step 3. $j_2 = 6$, $j_3 = 7$, $C = \{3, 6, 7\}$.
Step 4. $h = 2$, $C_2 = \{3, 6, 7\}$.
Step 7. $k^*(C) = 1$, $j_1 = 4$, $k_0 = 2$.
Step 3. $j_2 = 6$, $j_3 = 7$, $C = \{4, 6, 7\}$.
Step 4. $h = 3$, $C_3 = \{4, 6, 7\}$.
Step 7. $k^*(C) = 1$, $j_1 = 5$, $C = \{5, 6, 7\}$.
Step 4. $h = 4$, $C_4 = \{5, 6, 7\}$.
Step 8. $k_C = 3$.
Step 2. $k_0 = 1$.
Step 3. $j_1 = 1$, $j_2 = 2$, $j_3 = 6$, $j_4 = 7$, $C = \{1, 2, 6, 7\}$.
Step 7. $k^*(C) = 2$, $j_2 = 3$, $k_0 = 3$.
Step 3. $j_3 = 6$, $j_4 = 7$, $C = \{1, 3, 6, 7\}$.
Step 7. $k^*(C) = 2$, $j_2 = 4$, $k_0 = 3$.
Step 3. $j_3 = 5$, $j_4 = 6$, $C = \{1, 4, 5, 6\}$.

Step 4. $h = 5$, $C_5 = \{1, 4, 5, 6, 7\}$.
Step 7. $k^*(C) = 1$, $j_1 = 2$, $k_0 = 2$.
Step 3. $j_2 = 3$, $j_3 = 5$, $j_4 = 6$, $C = \{2, 3, 5, 6\}$.
Step 4. $h = 6$, $C_6 = \{2, 3, 5, 6, 7\}$.
Step 7. $k^*(C) = 2$, $j_2 = 4$, $C = \{2, 4, 5, 6\}$.
Step 4. $h = 7$, $C_7 = \{2, 4, 5, 6, 7\}$.
Step 7. $k^*(C) = 1$, $j_1 = 3$, $C = \{3, 4, 5, 6\}$.
Step 4. $h = 8$, $C_8 = \{3, 4, 5, 6, 7\}$.
Step 8. $k_C = 4$.
Step 2. $k_0 = 1$.
Step 3. $j_1 = 1$, $j_2 = 2$, $j_3 = 3$, $j_4 = 5$, $j_5 = 6$, $C = \{1, 2, 3, 5, 6\}$.

Appendix C. Steps of Algorithm 2 in Example 4.

Algorithm 2 proceeds as follows when it is applied to constraint (4.1) taking $\overline{x}_1 = \overline{x}_2 = 1$, $\overline{x}_3 = 0$, $\overline{x}_4 = 1$, $\overline{x}_5 = \frac{2}{5}$, $\overline{x}_6 = 1$, $\overline{x}_7 = 0$, and $\alpha = 0$:

Step 1. $\underline{k} = 2$, $\underline{j} = 8$, $\overline{k}' = 4$, $h = 0$, $j_0 = 0$, $k_C = 2$.
Step 2. $k = 1$, $\overline{k}_\alpha = 2$.
Step 3. $j_1 = 2$.
Step 5. $k = 2$.
Step 3. $j_2 = 6$.
Step 9. $k = 1$, $j_1 = 3$.
Step 5. $k = 2$.
Step 3. $j_2 = 6$.
Step 9. $k = 1$, $j_1 = 4$.
Step 5. $k = 2$.
Step 3. $j_2 = 6$.
Step 9. $k = 1$, $j_1 = 5$.
Step 16. $k_C = 3$.
Step 2. $k = 1$, $k_\alpha = 3$.
Step 3. $j_1 = 1$.
Step 5. $k = 2$.
Step 3. $j_2 = 2$.
Step 5. $k = 3$.
Step 3. $j_3 = 6$.
Step 9. $k = 2$, $j_2 = 3$.
Step 5. $k = 3$.
Step 3. $j_3 = 6$.
Step 9. $k = 2$, $j_2 = 4$.
Step 5. $k = 3$.
Step 3. $j_3 = 5$.
Step 5. $k = 4$.
Step 3. $j_4 = 6$.
Step 6. $C = \{1, 4, 5, 6\}$.
Step 10. $h = 1$, $C_1 = \{1, 4, 5, 6, 7\}$.
Step 13. $k^*(C) = 1$.
Step 14. $j_1 = 2$, $k = 1$.
Step 5. $k = 2$.
Step 3. $j_2 = 3$.
Step 5. $k = 3$.
Step 3. $j_3 = 5$.

Step 9. $k = 2,\ j_2 = 4.$
Step 5. $k = 3.$
Step 3. $j_3 = 5.$
Step 5. $k = 4.$
Step 3. $j_4 = 6.$
Step 6. $C = \{2, 4, 5, 6\}.$
Step 10. $h = 2,\ C_2 = \{2, 4, 5, 6, 7\}.$
Step 13. $k^*(C) = 1.$
Step 14. $j_1 = 3.$
Step 16. $k_C = 4.$
Step 2. $k = 1,\ k_\alpha = 4.$
Step 3. $j_1 = 1.$
Step 5. $k = 2.$
Step 3. $j_2 = 2.$
Step 5. $k = 3.$
Step 3. $j_3 = 3.$
Step 5. $k = 4.$
Step 3. $j_4 = 5.$
Step 9. $k = 3,\ j_3 = 4.$
Step 5. $k = 4.$
Step 3. $j_4 = 5.$
Step 5. $k = 5.$
Step 3. $j_5 = 6.$
Step 6. $C = \{1, 2, 4, 5, 6\}.$

Acknowledgments. The author is grateful to two anonymous referees, whose comments on the first version of this paper led to significant improvements.

REFERENCES

[1] E. Balas, *Facets of the knapsack polytope*, Math. Program., 8 (1975), pp. 146–164.
[2] E. Balas and E. Zemel, *Facets of the knapsack polytope from minimal covers*, SIAM J. Appl. Math., 34 (1978), pp. 119–148.
[3] J. Barceló, Å. Hallefjord, E. Fernández, and K. Jörnsten, *Lagrangean relaxation and constraint generation procedures for capacitated plant location problems with single sourcing*, OR Spektrum, 12 (1990), pp. 79–88.
[4] H. Crowder, E. L. Johnson, and M. Padberg, *Solving large-scale zero-one linear programming problems*, Oper. Res., 31 (1983), pp. 803–834.
[5] J. A. Díaz and E. Fernández, *A branch-and-price algorithm for the single source capacitated plant location problem*, J. Oper. Res. Soc., 53 (2002), pp. 728–740.
[6] B. L. Dietrich, L. F. Escudero, and F. Chance, *Efficient reformulation for* 0-1 *programs—methods and computational results*, Discrete Appl. Math., 42 (1993), pp. 147–175.
[7] B. L. Dietrich, L. F. Escudero, A. Garín, and G. Pérez, $O(n)$ *procedures for identifying maximal cliques and non-dominated extensions of consecutive minimal covers and alternates*, Top, 1 (1993), pp. 139–160.
[8] L. F. Escudero, A. Garín, and G. Pérez, $O(n\ log n)$ *procedures for tightening cover inequalities*, European J. Oper. Res., 113 (1999), pp. 676–687.
[9] L. F. Escudero, S. Martello, and P. Toth, *On tightening* 0-1 *programs based on extensions of pure* 0-1 *knapsack and subset-sum problems*, Ann. Oper. Res., 81 (1998), pp. 379–404.
[10] L. F. Escudero and S. Muñoz, *On characterizing tighter formulations for* 0-1 *programs*, European J. Oper. Res., 106 (1998), pp. 172–176.
[11] L. F. Escudero and S. Muñoz, *On characterizing maximal covers*, Investigación Oper., 23 (2002), pp. 136–149.
[12] L. F. Escudero and S. Muñoz, *On identifying dominant cliques*, European J. Oper. Res., 149 (2003), pp. 65–76.

[13] M. R. Garey and D. S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, 1979.
[14] P. L. Hammer, E. L. Johnson, and U. N. Peled, *Facets of regular* 0-1 *polytopes*, Math. Program., 8 (1975), pp. 179–206.
[15] K. L. Hoffman and M. W. Padberg, *Improving LP-representations of zero-one linear programs for branch-and-cut*, ORSA J. Comput., 3 (1991), pp. 121–134.
[16] K. Holmberg, M. Rönnqvist, and D. Yuan, *An exact algorithm for the capacitated facility location problems with single sourcing*, Eur. J. Oper. Res., 113 (1999), pp. 544–559.
[17] ILOG, *CPLEX* 7.0, *User's Manual*, Gentilly, France, 2000.
[18] E. L. Johnson, G. L. Nemhauser, and M. W. P. Savelsbergh, *Progress in linear programming-based algorithms for integer programming: An exposition*, INFORMS J. Comput., 12 (2000), pp. 2–23.
[19] S. Muñoz, *A correction of the justification of the Dietrich–Escudero–Garín–Pérez* $O(n)$ *procedures for identifying maximal cliques and non-dominated extensions of consecutive minimal covers and alternates*, Top, 3 (1995), pp. 161–165.
[20] S. Muñoz, *Reforzamiento de Modelos en Programación Lineal* 0-1, Ph.D. thesis, Departamento de Estadística e Investigación Operativa I, Universidad Complutense de Madrid, Madrid, 1999.
[21] S. Muñoz, *Detecting constraint redundancy in* 0-1 *linear programming problems*, Revista de Matemática: Teoría y Aplicaciones, 8 (2001), pp. 1–12.
[22] S. Muñoz, *Detecting infeasibility and fixing variables in* 0-1 *linear programming problems*, Investigación Oper., 23 (2002), pp. 91–103.
[23] A. W. Neebe and M. R. Rao, *An algorithm for the fixed-charge assigning users to sources problem*, J. Oper. Res. Soc., 34 (1983), pp. 1107–1113.
[24] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley, New York, 1988.
[25] M. W. Padberg, *A note on zero-one programming*, Oper. Res., 23 (1975), pp. 833–837.
[26] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C. The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 1992.
[27] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in FORTRAN. The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 1992.
[28] M. W. P. Savelsbergh, *Preprocessing and probing techniques for mixed integer programming problems*, ORSA J. Comput., 6 (1994), pp. 445–454.
[29] R. Weismantel, *On the* 0/1 *knapsack polytope*, Math. Program., 77 (1997), pp. 49–68.
[30] L. A. Wolsey, *Faces for a linear inequality in* 0-1 *variables*, Math. Program., 8 (1975), pp. 165–178.
[31] L. A. Wolsey, *Integer Programming*, John Wiley, New York, 1998.
[32] E. Zemel, *Easily computable facets of the knapsack polytope*, Math. Oper. Res., 14 (1989), pp. 760–764.

SIAM J. DISCRETE MATH.
Vol. 18, No. 4, pp. 769–777

MINIMAL-TIME k-LINE BROADCASTING*

I. GABER†

Abstract. Broadcasting refers to the process of sending a message from a source to an entire communication network. Line broadcasting (defined by Farley [*Networks*, 10 (1980), pp. 59–70]) assumes that members may "switch-through" any number of calls during a time unit. Namely, two members of the network may communicate with each other through a path as long as no link is involved in more than one call at the same time unit. Two paths may intersect, during a given time unit, only in vertices. A generalization of that model is the k-line-broadcasting model, which has the same properties as the line-broadcasting model with the additional constraint that the distance between two communicating members is at most k. The parameters to the problem are the network, the originator, and k.

In this paper we generalize Farley's algorithm into the k-line-broadcasting model. An algorithm is presented which produces a k-line-broadcasting scheme for any given tree and source vertex which consumes $O(\frac{D}{k} + \log_2 n)$ time units, where n is the number of vertices and D is the distance of the furthest vertex from the originator. This asymptotically achieves the lower bound provided.

Key words. broadcasting, k-line broadcasting

AMS subject classifications. 68W05, 94A40

DOI. 10.1137/S0895480101386620

1. Introduction.

The general synchronous *broadcasting* problem refers to the process of message dissemination in a communication network. One member of the network, the *originator*, initiates a message that should be transmitted to all members of the network. In *local broadcasting* a member can send a message (make a call) only to an adjacent member. In *line broadcasting* a member that is already informed of the message may call any other member using the communication links of any simple path between the two with the restriction that no link is used in more than one call at a given time unit.

The solution to the broadcasting problem is given in the form of a broadcasting scheme. A *broadcasting scheme* for a network is a specification of which calls are made during each time unit and which communication paths are used to make the calls.

When no member is involved in more than one communication call in any given time unit and each call is completed during one time unit, the number of informed members can at most double during each time unit. Therefore, at least $\lceil \log_2 n \rceil$ time units are needed for broadcasting in a network of n members. In the local broadcasting model it is not always possible to inform all members of a network in a time of $\lceil \log_2 n \rceil$ (called *minimum time*). However, Farley [3] showed that there is a minimum-time line-broadcasting scheme for any originator in any connected network.

In this paper a modification to the above question is raised, namely, *How fast can an originator inform a given network under the line communication model while limiting the length (number of occupied edges) of a call by some given k?*

This new model was proposed by Fujita and Farley [5]. In their paper they constructed minimal broadcast graphs in terms of Δ (the maximum degree) and k.

*Received by the editors March 16, 2001; accepted for publication (in revised form) July 13, 2004; published electronically April 22, 2005.

http://www.siam.org/journals/sidma/18-4/38662.html

†Department of Computer Science, School of Computer Sciences, Tel Aviv University, Tel Aviv 69978, Israel, and Department of Computer Science, The Academic College of Tel-Aviv-Yaffo, Tel-Aviv 61161, Israel (gaber@mta.ac.il).

This call-length constraint is reasonable from an implementation standpoint; long-distance calls utilize more network resources, can be more difficult to complete, are more likely to fail, and can cause bottlenecks for other competing communication processes.

The input to the problem is the graph, G, and k. It is natural to consider the given graph in terms of n, the number of members, and $d(G)$, its diameter (the maximum distance between any pair of members) to make the upper and lower bounds tight. It is obvious that minimum-time k-line broadcasting is not always possible. It suffices to look at a path on n members and to put k to be 2. This paper presents an algorithm that accomplishes the task in minimal time given the topology of the network and k.

Define the *cost* of a call to be the number of lines on the path between the caller and the receiver. The *cumulative cost* of a line-broadcasting scheme is the sum of the costs of all calls involved in the scheme. The problem of finding a minimum-time line-broadcasting scheme was first introduced by Farley [3], who studied general trees. He presented a minimum-time line-broadcasting scheme on any given tree on n members. The cumulative cost of his scheme is at most $(n-1)\lceil \log_2 n \rceil$. Following his work, other researchers focused on specific graphs whose structure was known in advance and presented minimum-time schemes that minimize the cumulative cost. Specifically, Kane and Peters [6] studied the cycle on n members. Fujita and Farley [4] discussed minimum-cost line broadcast in paths. Averbuch, Roditty, and Shoham [2] obtained efficient line-broadcast algorithms in a d-dimensional grid, which produce a linear cumulative cost in n, and Averbuch, Gaber, and Roditty [1] showed minimum-time line-broadcast schemes for complete binary trees, for which the cumulative cost is less than $2n$.

2. Preliminaries. Let $G = (V, E)$ be a connected undirected graph with n vertices, where V and E represent the set of vertices and the set of edges of G, respectively. For any $u, v \in V$, let $d(u, v)$ denote the length of the shortest path connecting u and v in the graph G, where $d(u, v)$ is called the *distance* between u and v in G. Denote by $d(G)$ the diameter of the graph, i.e., the maximal distance between any pair of vertices. Formally,

$$d(G) = \max\{d(u, v) \mid u, v \in G\}.$$

For other graph theoretical concepts see [8]. All trees in this paper are assumed to have $n \geq 2$ vertices.

DEFINITION 2.1. *Let k be a positive integer (possibly a function of n). A k-line communication is a communication model defined as follows:*

1. *All communications act synchronously according to a global clock.*
2. *At any given time, each vertex can call at most one other vertex at distance no more than k.*
3. *A call succeeds if it shares no edges with another call placed at that time unit.*

Note that one-line communication is equivalent to local broadcasting in which a call can be made only between adjacent vertices and that $(n-1)$-line communication is equivalent to the general line-broadcasting model in which calls may be made between nonadjacent vertices of any distance.

DEFINITION 2.2. *A broadcast scheme in G on n vertices is said to be a minimum-time k-line-broadcast scheme if it requires $\lceil \log_2 n \rceil$ time units under the k-line communication model.*

This paper deals with given trees and designated originators. For the upper bound for a connected graph this is sufficient, since any connected graph contains a

spanning tree.

The following simple lemma states that a minimum-time k-line-broadcast scheme in a tree is not always possible for a given originator.

LEMMA 2.3. *Let T be a tree on n vertices. Under the k-line communication model,*

1. *a minimum-time broadcast scheme is always possible when $d(T) \leq k$;*
2. *there always exists an originator such that a minimum-time broadcast scheme is not possible for $d(T) \geq 2k+1$.*

Proof. The first claim is trivial. If calls at distance k are allowed, then any minimum-time line-broadcast scheme is a minimum-time k-line-broadcast scheme, and such a scheme always exists according to Farley [3]. For the second claim, let T be any tree on $n = 2^r$ vertices with $d(T) \geq 2k+1$. Suppose that there exists a minimum-time broadcast scheme. Put $d = d(T)$. Since n is a power of 2, every vertex upon receiving the message must transmit it at all time units until the last one, r. Let $P = u_0, u_1, \ldots, u_d$ be a path of $d+1$ vertices (such P always exists). The edge $(u_{\lfloor \frac{d}{2} \rfloor}, u_{\lfloor \frac{d}{2} \rfloor + 1})$ cuts the original tree into two subtrees. Denote the subtrees obtained from T by deleting the edge $(u_{\lfloor \frac{d}{2} \rfloor}, u_{\lfloor \frac{d}{2} \rfloor + 1})$ $T_{\lfloor \frac{d}{2} \rfloor}$ and $T_{\lfloor \frac{d}{2} \rfloor + 1}$, respectively. Without loss of generality, assume that $T_{\lfloor \frac{d}{2} \rfloor}$ is larger in terms of number of vertices and make u_d the originator. (Otherwise, make u_0 the originator). Clearly, after the first time unit, there are two informed vertices, both on the smaller part of the tree. Since the edge $(u_{\lfloor \frac{d}{2} \rfloor}, u_{\lfloor \frac{d}{2} \rfloor + 1})$ can be used only once every time unit, by induction it is easy to see that the number of informed vertices at the subtree rooted at u_0 is always at least two less than the number of informed vertices at the subtree rooted at u_d, and hence minimum-time line broadcast cannot be achieved. ∎

In this paper a general graph, the originator, and the parameter $k \geq 2$ are an input to the problem. For $k = 1$ an optimal broadcast algorithm for trees is presented in [7]. An algorithm is given in the next section producing a k-line-broadcast scheme that runs in time $O(\frac{d}{k} + \log_2 n)$. This asymptotically achieves the lower bound, which is given in the following lemma.

LEMMA 2.4. *There exist networks G for which no k-line-broadcast scheme can end before $\Omega(\frac{d(G)}{k} + \log_2 n)$ time units.*

Proof. Put $d = d(G)$ and consider a graph which is a path of length $d-1$, $v_0, \ldots, v_{d-1}$, where the end-vertex v_{d-1} is a center of a star with $n-d$ end-vertices. Let the originator be v_0. Clearly, it takes $\Omega(\frac{d}{k})$ time units for the message to reach v_{d-1}; after that, informing the vertices of the star takes $\lceil \log_2(n-d) \rceil$ more time units. ∎

Note that if $k = d$ we get $\Omega(\log_2 n)$, as proved by Farley [3].

3. k-line-broadcast scheme.

3.1. Algorithm outline. Given a tree on n vertices, a designated originator, and a parameter $k > 1$, we present a k-line-broadcast scheme.

The intuition behind the algorithm is the following. The message proceeds "fast" to "distant" parts of the network. Once there is a copy of the message "near" each vertex, broadcast can be completed in a minimum time, since every informed vertex is part of a group with a diameter at most k.

DEFINITION 3.1. *Given a network and an originator, let D be the distance of the furthest vertex from the originator.*

The algorithm for finding the scheme starts by partitioning the tree into levels between 0 and D, denoted $l_0, \ldots, l_D$, using the breadth first search (BFS) algorithm.

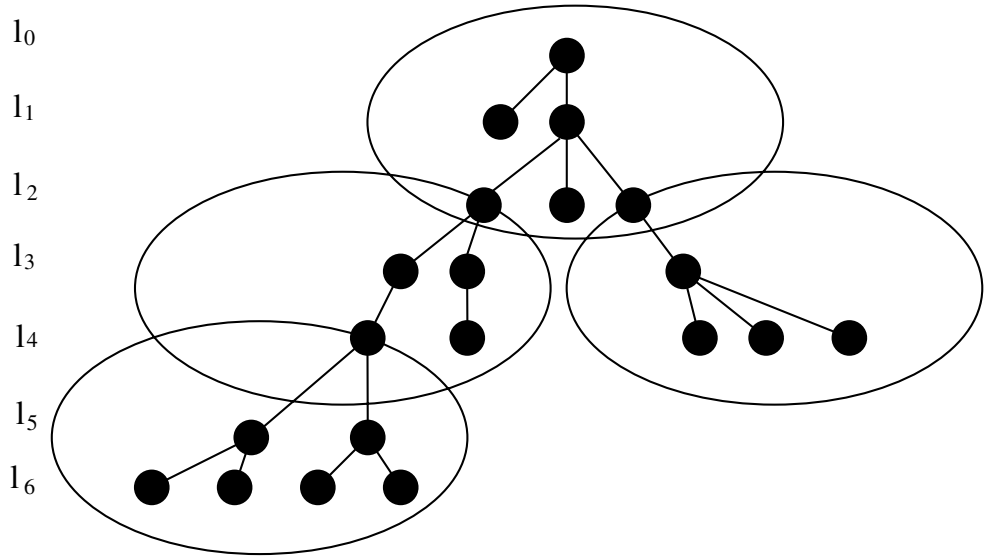


FIG. 1. *An example of three superlevels and four clusters.*

Level 0 contains only the originator. Then, the levels are divided into groups of $\lfloor \frac{k}{2} \rfloor + 1$ successive BFS levels each, called *superlevels*, such that the first superlevel contains levels $l_0, \ldots l_{\lfloor \frac{k}{2} \rfloor}$, the second contains levels $l_{\lfloor \frac{k}{2} \rfloor}, \ldots, l_{2\lfloor \frac{k}{2} \rfloor}$, etc., except, perhaps, for the last group, which may not be full. Note that there is an intersection between the highest level of a superlevel and the lowest level of its successive superlevel (see Figure 1). The number of the superlevels is $\lceil \frac{D}{\lfloor \frac{k}{2} \rfloor} \rceil$. Formally, for $0 \leq i \leq \lfloor \frac{D}{\lfloor \frac{k}{2} \rfloor} \rfloor$,

$$SL_i = l_{i\lfloor \frac{k}{2} \rfloor} \ldots l_{(i+1)\lfloor \frac{k}{2} \rfloor}.$$

Within each superlevel, there are *clusters*, which are defined as follows.

For each vertex u at the lowest level of a superlevel SL_i, add all the neighbors of u at the next levels within the superlevel; i.e., a cluster is a subtree rooted at u, containing all its descendants (see Figure 2).

Formally, for every $u \in l_{i \cdot \lfloor \frac{k}{2} \rfloor}$ define $S_u^{(i)}$ as follows:

$$S_u^{(i)} \stackrel{\text{def}}{=} \left\{ v \mid v \in l_j \wedge \operatorname{dist}(u,v) = j - \left(i \cdot \left\lfloor \frac{k}{2} \right\rfloor \right); \left(i \cdot \left\lfloor \frac{k}{2} \right\rfloor \right) + 1 < j \leq (i+1) \cdot \left\lfloor \frac{k}{2} \right\rfloor \right\}.$$

Another way to view this definition is to consider a superlevel as a directed graph, in which the edges are all directed from lower to higher levels. In such a setting, a cluster at a superlevel SL_i, $S_u^{(i)}$, includes all the vertices reachable from u.

This construction guarantees that each vertex at the highest level has an ancestor at the lowest level, such that the path between them is contained within the cluster.

After building the clusters in all the superlevels, the result is a *directed* tree of clusters: $\mathcal{T}$ $(\mathcal{V},\mathcal{E})$. The vertices of the tree, $\mathcal{V}$, are the clusters, and an edge $(S_u^{(i)}, S_v^{(i+1)}) \in \mathcal{E}$ exists if the root of cluster $S_v^{(i+1)}$ is a leaf of $S_u^{(i)}$. The cluster $S_u^{(i)}$ is called the *parent* of $S_v^{(i+1)}$, and $S_v^{(i+1)}$ is called the *child* of $S_u^{(i)}$.

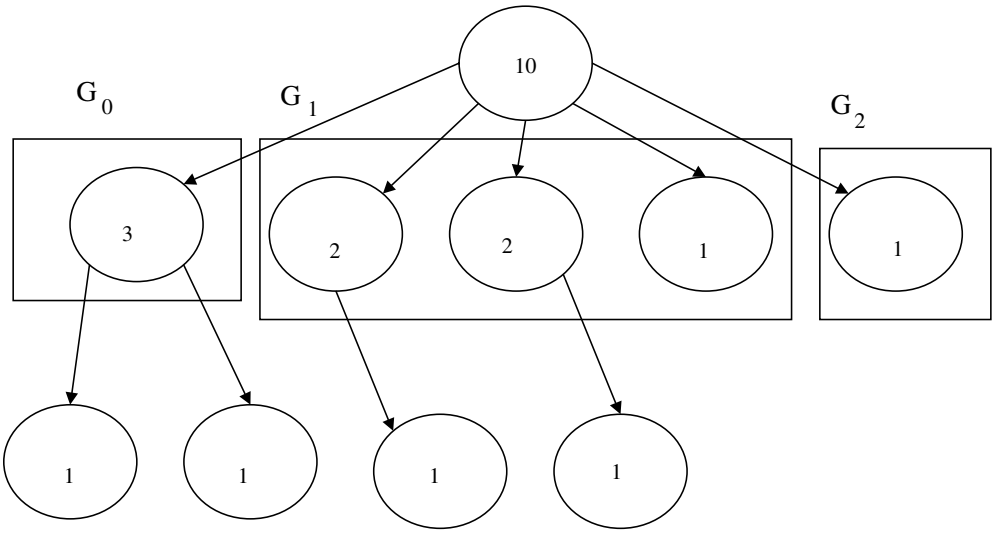


FIG. 2. *An example of a weighted tree.*

The next stage is giving each cluster a *weight*. The weight of a cluster is the number of clusters in its subtree, i.e., the number of its descendants including the node itself.

DEFINITION 3.2. *Let $c \in \mathcal{V}$ be a cluster in $\mathcal{T}$. Define T_c to be the subtree rooted at c produced by deleting the edge connecting c and its parent. Define W(c) to be the number of vertices in T_c, namely, $W(c) =\mid T_c \mid$.*

This weight function acts as a priority function: according to the weights, the cluster chooses which group of its children gets the message first, which gets it second, and so on. This is done using the *informing-children* procedure described below. The procedure produces a scheme for informing a group of x children in $\lceil \log_2(x+1) \rceil$ time units.

Once there is at least one informed vertex at each cluster, broadcast can be done in parallel using the minimum-time line-broadcast scheme presented by Farley [3]. Note that each cluster's diameter is at most k; therefore, any broadcast scheme is a k-line-broadcast scheme inside a cluster. Also, since the clusters are edge-disjoint, no line-use conflicts occur. The broadcast scheme for each cluster's root is composed of two processes: passing the message to its children according to their weights and informing the rest of the cluster's vertices. The next subsections explain the different stages of the algorithm. Section 3.2 specifies the scheme of informing a cluster's children, and section 3.3 gives the scheme in detail.

3.2. Informing-children procedure. This procedure receives a tree T, with a known diameter which is at most k, such that any line broadcast scheme is a k-line broadcast scheme on it. The root of the tree is the originator that knows the message. Given a group of leaves of size x, the goal is to inform all the leaves in minimum time, i.e., $\lceil \log_2(x+1) \rceil$ time units.

Assume, for simplicity, that the number of involved vertices, the root and the group of leaves, is a power of 2, i.e., $x + 1 = 2^r, r \geq 0$. (This is also the case in the proposed broadcast scheme.) The procedure is composed of r steps, such that in each step the number of informed vertices is doubled.

DEFINITION 3.3. *A step of the scheme is the collection of calls taking place in a single time unit. An active vertex in a given step of the scheme is a vertex that participates in the calls that take place during that time unit. An idle vertex is a nonactive vertex.*

The algorithm for producing the scheme starts at the last step, r. The algorithm repeatedly applies the *coupling* procedure, which is described in detail below, each time reducing the number of active vertices by half, until there is one remaining vertex in T, the root. Every instance of the coupling procedure divides the active vertices into couples while avoiding line-use conflicts; i.e., no edge may belong to more than one call at the same time unit. Each of the couples is designated for a call at the next time unit of the scheme. After each step, the algorithm arbitrarily picks one of each couple to be active (the root must always be active), and the other to be idle, and calls the procedure again.

The actual scheme begins at step 1. A *call* at step i between a couple chosen by the coupling procedure is actually a call between an active vertex of step $i-1$ and an idle vertex of step $i-1$, meaning the active one already has received the message and is informing the other vertex at time unit i.

3.2.1. The coupling procedure.
input : tree T containing active and idle vertices
output : A list of pairs of active vertices

make an empty list
while (T not empty)

1. If there is an idle leaf—prune it from the tree.
2. If there is an even-size group of active leaves sharing the same parent with no nonleaf siblings—
   - Divide the leaves arbitrarily into couples.
   - Add the couples to the list.
   - Prune the leaves from the tree.
3. If there is an odd-size group of active leaves sharing the same parent with no nonleaf siblings—
   - Divide the leaves arbitrarily into couples until there is one leaf with no pair.
   - Add the couples to the list.
   - Prune the couples from the tree.
   - If the parent is an active vertex—Couple the remaining active leaf with the parent, add them to the list and prune them both from the tree. Otherwise—*exchange* the leaf with its parent and remove the edge that connects them.

3.2.2. The informing-children scheme. Mark x leaves and the root in T as active, and the rest of the vertices as idle. For i:= r to 1 do

1. list[i]=Coupling (T).
2. Schedule all couples in list[i] to make calls during the ith time unit.
3. Pick one from each couple and mark it idle.

The next sections deal with the correctness of the scheme.

3.2.3. Proof of correctness.

LEMMA 3.4. *The coupling procedure ends and outputs all active vertices divided into couples, with no line-use conflicts.*

Proof. It is easily seen that in each move of the coupling procedure we can always find at least one of the cases described above. Moreover, in every stage of the algorithm we prune vertices until we remain with an empty tree and the procedure ends.

Note that whenever we prune two leaves that share the same parent (cases 2 and 3), the two edges that connect them to the parent are used only for the communication between the two members. Also, in the case in which we match a leaf with its active parent (case 3), the edge between them is used only for the call made between them.

Finally, when we switch an active leaf with its idle parent (case 3), the edge that connects them is not needed for any other pair; therefore, once we find a match for the leaf, this edge will be used by that pair only. In summing, no line-use conflicts occur in the scheduling scheme. □

LEMMA 3.5. *The main procedure produces a legal k-line-broadcast scheme that terminates after all vertices of the tree know the message.*

Proof. The algorithm starts with one active vertex, the root. At every stage, the number of active vertices is doubled. Therefore, after time unit i ($1 \leq i \leq r$) there are indeed 2^i informed vertices. Recall as well that the procedure is applied to a cluster with a known diameter, which is at most k; therefore any line-broadcast scheme is a k-line-broadcast scheme on it. □

Remark. In the case where $2^{r-1} < x + 1 < 2^r$ vertices, we add fictitious vertices $(x+2), \ldots, 2^r$ arbitrarily to one of the leaves. The first step of the scheme may couple fictitious vertices with regular ones. In case a vertex is coupled with a fictitious vertex it does nothing during this time unit. Note that starting from step 2 the number of active vertices is a power of 2, namely, 2^{r-1}, so the algorithm continues normally. The number of time units needed is therefore $\lceil \log_2(x+1) \rceil$.

3.3. The k-line-broadcast scheme. Given a tree T, the preprocessing of it produces a tree, $\mathcal{T}$, of weighted clusters, each with a diameter of at most k.

Every cluster starts by sorting its children from the child with the maximal weight to the child with the minimal weight, $W(c_1) \geq W(c_2) \geq W(c_3)$, etc. Then, it divides the children into groups. The "heaviest" child, namely c_1, belongs to G_0. If there are more children, the next 3 children, c_2, c_3, c_4, belong to G_1, G_2 contains the next 15 children, and so on. In general, $|G_i| = 2^{2^i} - 1$, except, of course, for the last group, which may not be full.

The groups receive the message one after the other using the informing-children scheme, which is applied as many times as the number of groups, which can be no more than $\log_2 \log_2 n$.

Summing up, the scheme for a *root* of a cluster contains two stages upon receiving the message:

1. Inform the children, from group G_0 on, using the scheme of the informing-children procedure.
2. Inform the other vertices of the cluster using minimum-time line-broadcast scheme (see [3]).

A vertex which is not a root of a cluster, upon receiving the message, acts according to the minimum-time line-broadcast scheme within its cluster.

3.4. Computing the total time of the broadcast scheme. In this section we prove our main result.

THEOREM 3.6. *The proposed k-line-broadcast scheme ends within at most $O(\frac{D}{k} + \log_2 n)$ time units.*

The proof is divided into two parts. Every cluster is shown to receive the message after $O(\frac{D}{k}+\log_2 n)$ time units, and once the root of a cluster has received the message, the cluster can complete broadcast within $O(\log_2 n)$ more time units.

The first part is proved by induction on the height of the tree, $\lceil \frac{D}{\lfloor \frac{k}{2} \rfloor} \rceil$. Let m be the number of clusters. Since $m < n$, the final result uses n as the parameter. The next lemma refers to the clusters tree assumed to have m vertices.

LEMMA 3.7. *The proposed broadcast scheme applied on a tree of m vertices and height h ends after at most $4(h + \lceil \log_2 m \rceil)$ time units.*

Proof. For the base of the induction consider a tree on m vertices and height 1. This is a tree with one root and $m-1$ children, a star.

Using the scheme, the children are divided into groups such that $|G_i| = 2^{2^i} - 1$ (except, perhaps, for the last group, which may be not full), and the number of groups is therefore less than or equal to $\lceil \log_2 \log_2 m \rceil$.

The informing-children scheme applied on a group of x children ends after $\lceil \log_2(x+1) \rceil$ time units.

Thus, informing G_i takes at most 2^i time units. Summing the time units needed to inform all children, we get

$$\sum_i Time(G_i) \leq \sum_{i=0}^{\log_2 \log_2 m} 2^i < 2 \log_2 m < 4(1 + \lceil \log_2 m \rceil).$$

Assume the claim is true for some height $h-1$ and consider a tree on m vertices and height h. Let c be the one child of the root that needs the largest number of time units to finish broadcast in its subtree. That is, the last of the leaves to receive the message is a descendant of c. Let $c \in G_i$ for some i.

If $i = 0$ (c is the heaviest child, c_1), according to the induction assumption the child, which is a tree of height $h-1$ and $m-1$ children finishes broadcast after at most $4(h-1) + 4\lceil \log_2(m-1) \rceil$ time units, adding the time unit it takes for the root to pass the message to c the total time is less than $4h + 4\lceil \log_2 m \rceil$ time units and the claim holds.

Otherwise $i \geq 1$. The number of children that receives the message *before* c (and obviously their weight is greater than that of c) is therefore at least $\sum_{j=0}^{i-1} |G_j| = \sum_{j=0}^{i-1} (2^{2^j} - 1) > 2^{2^{i-1}} - 1$, meaning the number of vertices in c's subtree is at most $\frac{m}{(2^{2^{i-1}}-1)}$.

Given the induction assumption, the total time for c to finish is therefore $4(h-1) + 4\lceil \log_2(\frac{m}{(2^{2^{i-1}}-1)}) \rceil \leq 4h - 4 + 4\lceil \log_2 m \rceil - 2^{i+1} + 4 = 4h + 4\lceil \log_2 m \rceil - 2^{i+1}$.

Now, since $c \in G_i$, the worst case is that it gets the message last of its group, and certainly after the previous groups. Therefore the time units it takes until the message reaches c is at most

$$\sum_{j=0}^{i} Time(G_j) = \sum_{j=0}^{i} 2^j < 2^{i+1}.$$

Adding both results, the last vertex to receive the message gets it at most by time

$$4h + 4\lceil \log_2 m \rceil - 2^{i+1} + 2^{i+1} = 4h + 4\lceil \log_2 m \rceil. \qquad \square$$

Proof of Theorem 3.6. Lemma 3.7 proves that the root of every cluster receives the message after at most $4(h + \lceil \log_2 m \rceil)$ time units. Since $h = \lceil \frac{D}{\lfloor \frac{k}{2} \rfloor} \rceil$, it means

that a cluster receives the message after $O(\frac{D}{k} + \log_2 n)$ time units. Now, since every cluster contains at most n vertices (the worst case is that there is one cluster in the tree), using the minimum-time line broadcast, broadcast has be shown to take $\lceil \log_2 n \rceil$ time units (see [3]). Once every cluster receives the message, all clusters' roots can broadcast in parallel within the clusters with no line-use conflicts. The total time of the scheme is therefore $O(\frac{D}{k} + \log_2 n)$ time units. □

Acknowledgments. I would like to thank my advisors, Professor Amir Averbuch and Professor Yehuda Roditty, for their great help in writing this paper.

REFERENCES

[1] A. Averbuch, I. Gaber, and Y. Roditty, *Low cost minimum-time line broadcasting in complete binary trees*, Networks, 38 (2001), pp. 189–193.
[2] A. Averbuch, Y. Roditty, and B. Shoham, *Efficient line broadcast in a d-dimensional grid*, Discrete Appl. Math., 113 (2001), pp. 129–141.
[3] A. M. Farley, *Minimum-time line broadcast networks*, Networks, 10 (1980), pp. 59–70.
[4] S. Fujita and A. Farley, *Minimum-cost line broadcasting in paths*, Discrete Appl. Math., 75 (1997), pp. 255–268.
[5] S. Fujita and A. Farley, *Sparse hypercubes—A minimal k-line broadcast graph*, Discrete Appl. Math., 127 (2003), pp. 431–446.
[6] J. O. Kane and J. G. Peters, *Line broadcasting in cycles*, Discrete Appl. Math., 83 (1998), pp. 207–228.
[7] P. J. Slater, E. J. Cockayne, and S. T. Hedetniemi, *Information dissemination in trees*, SIAM J. Comput., 10 (1981), pp. 692–701.
[8] D. West, *Introduction to Graph Theory*, Simon and Schuster, Upper Saddle River, NJ, 1996.

SIAM J. DISCRETE MATH.
Vol. 18, No. 4, pp. 778–786

HYPERCUBES AS DIRECT PRODUCTS*

BOŠTJAN BREŠAR†, WILFRIED IMRICH‡, SANDI KLAVŽAR§, AND BLAŽ ZMAZEK¶

Abstract. Let G be a connected bipartite graph. An involution α of G that preserves the bipartition of G is called bipartite. Let G^α be the graph obtained from G by adding to G the natural perfect matching induced by α. We show that the k-cube Q_k is isomorphic to the direct product $G \times H$ if and only if G is isomorphic to Q_{k-1}^α for some bipartite involution α of Q_{k-1} and $H = K_2$.

Key words. direct product, hypercube, automorphism, involution

AMS subject classifications. 05C70, 05C25

DOI. 10.1137/S0895480103438358

1. Introduction.

This paper is concerned with hypercubes and the direct product of graphs. The main result is the characterization of all graphs G for which $G \times K_2$ is a hypercube and the proof of the fact that there are no other factorizations of the hypercube with respect to the direct product.

The paper was motivated by the problem of representing median graphs—that is, retracts of hypercubes—as direct products [2]. In this context the first question pertains to the possibility of decomposing the hypercube itself. The original proof of the result was unwieldy and long but could be considerably simplified by the application of ideas connected with the density of subgraphs of sparse graphs, together with the concept of the Cartesian skeleton [10, 11], which was introduced for the investigation of the direct product.

The paper illustrates the importance and applicability of Graham's density lemma and adds to the numerous interesting properties of the hypercube, which plays a prominent role in many areas of mathematics and computer science; see, e.g., the papers [8, 16, 20] on networks, routings, and flows, respectively. It may also shed some light on the decomposition of bipartite graphs with respect to the direct product.

The direct product, together with the Cartesian, the strong, and the lexicographic product, is one of the four standard graph products [11]. It is the natural product in the category of graphs [7] and harbors intriguing and challenging problems. Foremost of all is Hedetniemi's conjecture, which asserts that the chromatic number of the direct product is the minimum of the chromatic numbers of its factors. It is the big open problem in the area and has led to many different approaches and new concepts; cf. surveys [17, 21]. More generally, the direct product is a widely used tool in the area of graph colorings; see, for instance, [6, 22, 23]. It is also replete with interesting

*Received by the editors December 10, 2003; accepted for publication (in revised form) September 14, 2004; published electronically May 13, 2005.
http://www.siam.org/journals/sidma/18-4/43835.html

†University of Maribor, FEECS, Smetanova 17, SI-2000 Maribor, Slovenia (bostjan.bresar@uni-mb.si). The research of this author was supported by the Ministry of Education, Science and Sport of Slovenia, grant Z1-3073-0101-01.

‡Montanuniversität Leoben, A-8700 Leoben, Austria (imrich@unileoben.ac.at).

§Department of Mathematics and Computer Science, University of Maribor, PeF, Koroška cesta 160, SI-2000 Maribor, Slovenia (sandi.klavzar@uni-mb.si). The research of this author was supported by the Ministry of Education, Science and Sport of Slovenia, grant 0101-P1-0297.

¶University of Maribor, FME, Smetanova 17, SI-2000 Maribor, Slovenia (blaz.zmazek@uni-mb.si). The research of this author was supported by the Ministry of Education, Science and Sport of Slovenia, grant 0101-P1-0297.

ideas and concepts relating to other areas of graph theory, for example to matching theory [1, 9] and stability in graphs [3, 13].

This product has been introduced and studied from several points of view and is known under many different names, for instance as the cardinal product, the Kronecker product, and the categorical product. Moreover, it is universal in the sense that every graph is an induced subgraph of a suitable direct product of complete graphs [15].

In 1971 McKenzie [14] proved that finite, nonbipartite, connected graphs have unique prime factor decomposition (UPFD) with respect to the direct product in the class of undirected graphs with loops. Many years later, in 1998, this result was extended in [10] by showing that the UPFD can be found in polynomial time. For disconnected graphs or bipartite graphs the prime factorizations need not be unique. It is also not unique for finite nonbipartite graphs in the class of simple graphs without loops.

Despite the extensive and deep investigations of the direct product, factorizations of bipartite graphs have rarely been investigated. If a bipartite graph is a direct product of two graphs, one factor must be bipartite, but not the other. (The direct product of two connected bipartite graphs consists of two connected (bipartite) components [19].) This also holds for the hypercube, and we cannot directly apply the above results to our problem. Nevertheless, the concept of the Cartesian skeleton that proved useful in the nonbipartite case can be fruitfully applied here, too. In the nonbipartite case the Cartesian skeleton is connected, but not in the bipartite one, and this accounts for the nonuniqueness of the factorizations.

In the remainder of the section we fix terminology and notation. All graphs considered here are undirected, finite graphs that may contain loops.

The *direct product* $G \times H$ of two graphs G and H is defined on the Cartesian product $V(G) \times V(H)$ of the vertex sets of the factors. Its edge set is the set of all pairs of vertices $(a, x), (b, y) \in V(G) \times V(H)$, where $ab \in E(G)$ and $xy \in E(H)$. It is commutative and associative, and the one-vertex graph with a loop is a unit.

The *Cartesian product* $G\Box H$ has the same vertex set as the direct product. Its edge set consists of all pairs (a, x), (b, y) with $ab \in E(G)$ and $x = y$, or $a = b$ and $xy \in E(H)$. It is also commutative and associative. Its unit is K_1.

The subgraph of $G\Box H$ induced by the vertices (a, x), $x \in V(H)$, is called an *H-layer* of $G\Box H$ and denoted by $H^{(a,x)}$. Note that any H-layer is isomorphic to H. Analogously one defines G-layers. The d-dimensional *hypercube* or *d-cube* Q_d is the Cartesian product of d copies of the complete graph K_2 on two vertices. So $Q_1 = K_2$ and we also set $Q_0 = K_1$. Let $Q_{d-1}\Box K_2$ be an arbitrary factorization of Q_d. The edges between the two Q_{d-1}-layers are said to be of the same *color* or *parallel* in Q_d.

Let $V(Q_d) = X + Y$ be the bipartition of Q_d. Then the *halved cube* Q'_d is the graph with $V(Q'_d) = X$, where u is adjacent to v in Q'_d if u and v have a common neighbor in Q_d. A subgraph H of G is called *spanning* if $V(H) = V(G)$.

The concept of layers is defined analogously for the direct product. In the case of the direct product the layer $H^{(a,x)}$ is isomorphic to H only if a carries a loop (in G); otherwise the edge-set of $H^{(a,x)}$ is empty.

2. Graham's density lemma for hypercubes. At a first glance the hypercube looks simple, and from many points of view this is true. Nevertheless, it has a rich subgraph structure. For example, if one subdivides every edge of a given graph G on n vertices into a path of length two and adds a vertex that is adjacent to the original n vertices of G, then the resulting graph can be isometrically embedded into

Q_n; see [12]. This ambivalence between simplicity and structure definitely adds to its attractiveness.

As the number $|Q_k|$ of vertices of Q_k is 2^k and the number of edges $k2^k/2$, the *density* of Q_k; that is, the quotient $|E(Q_k)|/|Q_k|$ is $k/2$. This is rather small if one considers the fact that the complete graph on the same number of vertices as Q_k has density $(2^k-1)/2$. More important, this sparseness is inherited by the subgraphs of the hypercube.

Before formulating the lemma, we note that the statement $|E(Q_k)| = \frac{1}{2}|Q_k| \cdot \log_2 |Q_k|$ is equivalent to the assertion that the density of Q_k is $k/2$.

We are now ready for Graham's density lemma from [5]. We include a proof (modelled after the proof given in [11]) because its main idea appears again in the proof of Lemma 2.

LEMMA 1 (density lemma). *Let G be a subgraph of a hypercube. Then*

$$|E(G)| \le \frac{|G|}{2} \log_2 |G|, \tag{2.1}$$

with equality holding if and only if G is a hypercube.

Proof. The proof is similar to that of [11, Proposition 1.24]. Let G be a subgraph of Q_k. We proceed by induction on k. The assertions of the lemma are true for $k = 1$ and 2. Suppose they are true for $k \ge 2$ and that G is a subgraph of $Q_{k+1} = Q_k \Box K_2$. If G meets only one Q_k-layer, then the assertion is true by the induction hypothesis. Thus both intersections G_1 and G_2 of G with the Q_k-layers are nonempty. Let the notation be chosen such that $x = |G_1| \ge |G_2| = y \ge 1$. Again by the induction hypothesis $|E(G_1)| \le \frac{x}{2} \log_2 x$ and $|E(G_2)| \le \frac{y}{2} \log_2 y$. Since every vertex of G_2 has at most one neighbor in G_1 the number z of edges between G_1 and G_2 is at most y. We thus have

$$|E(G)| \le \frac{x}{2} \log_2 x + z + \frac{y}{2} \log_2 y. \tag{2.2}$$

Since $z \le y$ and $\frac{1}{2}(x+y)\log_2(x+y) = \frac{1}{2}|G| \log_2 |G|$ it suffices to show that

$$\frac{x}{2} \log_2 x + y + \frac{y}{2} \log_2 y \le \frac{x+y}{2} \log_2(x+y) \tag{2.3}$$

and that equality holds in (2.1) if and only if G is a hypercube.

We show the validity of inequality (2.3) first. It is clearly true for $x = y$; in this case the equality sign holds. We now fix y and increase x. Comparing the partial derivatives with respect to x on both sides of (2.3) we arrive at the inequality

$$\frac{1}{2} \log_2 x + \frac{1}{2} \log_2 e < \frac{1}{2} \log_2(x+y) + \frac{1}{2} \log_2 e.$$

This means that the right side grows strictly faster than the left and in (2.3) equality only holds for $x = y$.

Now, suppose $|E(G)| = \frac{1}{2}|G| \log_2 |G|$. Then the equality sign must hold everywhere, $z = y$ and $x = y$. Also, $|E(G_1)|$ must be $\frac{x}{2} \log_2 x$, just as $|E(G_2)|$ must be $\frac{y}{2} \log_2 y$. By the induction hypothesis both G_1 and G_2 are hypercubes. Since $x = y$ they have the same dimension, and $z = y$ implies that G is the Cartesian product of a hypercube of dimension $\log_2 x$ by a K_2, with the layers G_1 and G_2.

This completes the proof, because equality clearly holds in (2.1) if G is a hypercube. □

This result has been generalized by Squier, Torrence, and Vogt [18] to Cartesian products of complete graphs. They prove that subgraphs G of the k-fold Cartesian product of K_p have at most $\frac{1}{2}(p-1)|G|\log_p|G|$ edges, with equality holding if and only if G is a Cartesian power of K_p.

3. Factorizations of hypercubes. We continue with the investigation of the structure of the graphs G with $Q_k = G \times K_2$. It is easy to see that every hypercube Q_k of dimension $k > 0$ can be represented as a nontrivial direct product $G \times K_2$, where G is obtained from Q_{k-1} by addition of a loop to every vertex. This is a special case of the following lemma.

LEMMA 2. *Let $Q_k = G \times K_2$. Then G has a spanning hypercube.*

Proof. Let $V(K_2) = \{b, w\}$. For convenience we color b black, w white, and assign the same colors to the vertices of Q_k that are mapped into b and w, respectively. Moreover, for $x = (g, b)$ and $y = (g, w)$ we set $x' = y$ and $y' = x$.

We proceed by induction on the size of G. It suffices to show that Q_k has a factorization $Q_{k-1}\Box K_2$ such that both Q_{k-1}-layers are mapped injectively into G. Clearly the theorem is true for Q_1. In this case G is the graph on one vertex with a loop and Q_{k-1} in the decomposition $Q_{k-1}\Box K_2$ is $Q_0 = K_1$.

Suppose it is true for all Q_i with $1 \le i < k$. Let $Q_k = G \times K_2$ be a given factorization. We consider all decompositions $Q\Box K_2$ of the given Q_k, where Q is a $(k-1)$-dimensional hypercube. Without loss of generality we can assume that Q is a subgraph of Q_k. In the rest of the proof let p_G denote the projection map onto G. If p_G projects Q injectively into G there is nothing to show. Furthermore, since the color classes in a regular bipartite graph have the same size, the numbers of black and white vertices of Q are equal.

Suppose there is a Q whose projection $p_G Q$ meets exactly half the vertices of G. By induction Q has a $(k-2)$-dimensional subcube H that is mapped injectively into $p_G Q$. Let H_b be the set of the black vertices of H and H_w be its set of white ones. Note that $H'_b \cup H'_w$ also spans a subcube of Q with dimension $k-2$. We denote it by H'; it is the other H-layer in the decomposition $H\Box K_2$.

Let $\overline{Q}$ be the second Q-layer of Q_k and F be the set of edges between Q and $\overline{Q}$; we color them blue. The blue edges induce matchings between H and $\overline{H}$ and between H' and $\overline{H'}$. With every edge uv from a vertex $u \in H$ to a vertex $v \in \overline{H}$ the pair $u'v'$ is an edge from H' to $\overline{H'}$. Hence $p_G\overline{H} = p_G\overline{H'}$. Since the union of these projections is $p_G\overline{Q}$ all three projections are equal. Thus $p_G(H \cup \overline{H}) = V(G)$ and $H \cup \overline{H}$ induces a hypercube of dimension $k-1$.

In the remaining case there is a nonempty part A of Q with $p_G A_b = p_G A_w$ and a nonempty part B that maps injectively into G. In other words, the sets $p_G B_b$ and $p_G B_w$ are disjoint and at least one of them is nonempty. Since Q has the same number of vertices as G this means that there is a further nonempty part C of Q_k with $p_G C_b = p_G C_w$. Of course this is only possible if $k \ge 3$, which we will assume henceforth. A simple calculation shows that $|A| = |C|$. The corresponding situation of this last case is schematically shown in Figure 1.

We wish to show now that A and B are hypercubes of dimension $k-2$. We introduce the notation $x = |A|$, $y = |B|$ and show first that the number of edges between A and B is at most $\min(x, y)$. By the definition of the direct product the number of edges between A and B is the same as the number of edges between A' (which is A) and B'.

For an estimate we consider $\overline{Q}$, the second layer of Q. It is spanned by the union of B' and C. This means that the number of edges between A and B'—they are part

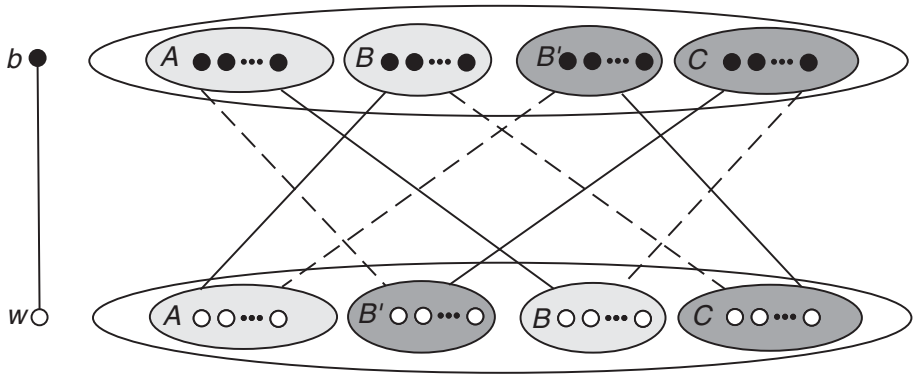


FIG. 1. *Situation from the proof;* $Q = [A \cup B]$ *is brighter and* $\overline{Q} = [B' \cup C]$ *darker.*

of the matching between Q and $\overline{Q}$—is at most $\min(x, y)$.

By the density lemma $\frac{1}{2}x \log_2 x + \min(x, y) + \frac{1}{2}y \log_2 y$ is an upper bound for the number of edges in Q, but the latter equals $\frac{1}{2}(x+y)\log_2(x+y)$ since Q is a hypercube. We thus arrive at the inequality

$$\frac{x}{2}\log_2 x + \min(x, y) + \frac{y}{2}\log_2 y \geq \frac{x+y}{2}\log_2(x+y).$$

As the proof of the density lemma shows this is only possible if both sides are equal, $x = y$, and both A and B are hypercubes.

Thus, A and B have the same size x and are hypercubes of dimension $k-2$. Moreover, there are exactly x edges between them and they form a matching. We color them red. By the matching between Q and $\overline{Q}$ they correspond to edges in $\overline{Q}$ that we also color red; cf. Figure 1, which schematically shows the matchings of red edges by unbroken lines. The edges of the matching between Q and $\overline{Q}$ we color blue. These edges have the same projections into G as the red ones and are indicated in the picture by broken lines.

By the induction hypothesis there is a color in A, call it green, whose removal decomposes A into two hypercubes that are projected injectively into G by p_G. Let us remove all edges from Q_k that are parallel to the green edges in A. The resulting graph consists of two hypercubes of dimension $k-1$. Let H^* be one of these components. We wish to show that H^* projects injectively into G. To see this, we consider $A^* = A \cap H^*$ and extend it to $B^* = B \cap H^*$ by the matching induced by the red edges and to $B'^* = B' \cap H^*$ by the matching induced by the blue ones. The matching to C^* can then be effected either from B^* by blue edges or B'^* by red ones; cf. Figure 2.

Note that $A_b \setminus A^*$ and A^*_w have the same projections into G. Since the red and blue edges also have the same projections into G one sees that $B'_b \setminus B'^*$ and B^*_b have the same projections too, from which we infer that B'^*_b and B^*_b have different ones. Continuing this way it is easily seen that H^* projects injectively into G. □

An *involution* of a graph is an automorphism of order two. For a bipartite graph G with bipartition $V(G) = X + Y$ we call an involution α *bipartite* if $\alpha(X) = X$. For a bipartite involution α we let G^α denote the graph obtained from G by addition of the perfect matching $\{uv \mid u = \alpha(v), v \in V(G)\}$.

THEOREM 3. *The hypercube* Q_k *is representable as a product of the form* $G \times K_2$ *if and only if* G *is isomorphic to* Q^α_{k-1} *for some bipartite involution* α *of* Q_{k-1}.

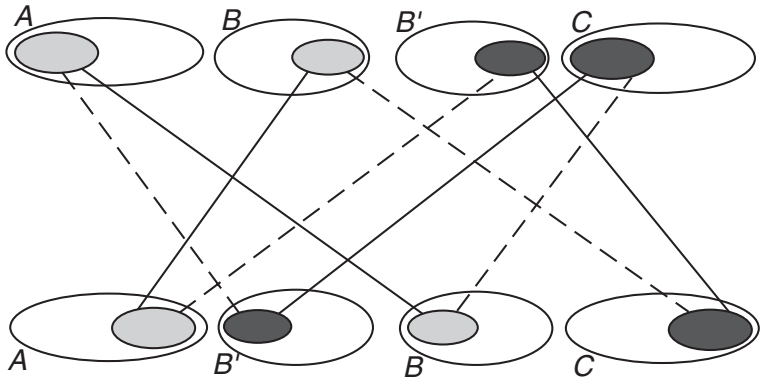


FIG. 2. *Situation from the proof; H^* is indicated brighter.*

Proof. Recall that the vertices of Q_k can also be represented as strings from $\{0,1\}^k$ and that all vertices with an even number of 1's form one of the bipartition sets of Q_k. Clearly, any two such vertices have even distance.

Suppose that $G \times K_2$ is a k-cube. By Lemma 2, G contains Q_{k-1} as a spanning subgraph; we denote it by S. Then $S \times K_2$ is a subgraph of $G \times K_2$ that consists of two disjoint hypercubes Q_{k-1}, say S_1 and S_2. As $G \times K_2$ is isomorphic to Q_k, each vertex x of S_1 is incident with an edge from $(G \times K_2) \setminus (S \times K_2)$ that connects x with a vertex y of S_2. Hence the distance in S between $p_G(x)$ and $p_G(y)$ must be even. Moreover, the edges between S_1 and S_2 induce an isomorphism between S_1 and S_2, so their projections to G induce an automorphism α of Q_{k-1} which maps each vertex v to a vertex $\alpha(v)$ with even distance from v. Also, the projections of the edges from $(G \times K_2) \setminus (S \times K_2)$ form a perfect matching of G. We conclude that α is a bipartite involution of G.

For the converse it suffices to show that every $Q_{k-1}^{\alpha} \times K_2$ is isomorphic to Q_k. □

If we are interested only in simple graphs G that factor Q_k with respect to the direct product, it suffices to restrict attention to fixed point free involutions α. We state this as a corollary.

COROLLARY 4. *The hypercube Q_k is representable as a direct product $G \times K_2$ of a simple graph G by K_2 if and only if G is isomorphic to Q_{k-1}^{α} for some fixed point free bipartite involution α of Q_{k-1}.*

4. The direct product representations of Q_k. To find all representations of Q_k as a direct product we first note that no two vertices of Q_k have the same set of neighbors. Such graphs are called *thin*; their prime factorizations with respect to the direct product are similar to the prime factorizations of graphs with respect to the Cartesian product. For any thin graph G one can show the existence of a *Cartesian skeleton* H. It is defined on the vertex set of G, is invariant under automorphisms of G, and, most important, to any decomposition $G_1 \times G_2$ of G corresponds a decomposition $H_1 \,\square\, H_2$ of H such that the vertex-sets of the G_i-layers of G are the vertex-sets of the H_i-layers of H. In particular, this means that G is prime with respect to the direct product if its Cartesian skeleton H is prime with respect to the Cartesian product.

The Cartesian skeleton was introduced in [10] (see also [11]) to investigate the decomposition properties of graphs with respect to the direct product. It led to a

polynomial algorithm for the prime factorization of nonbipartite connected graphs with respect to the direct product and to a new proof of the uniqueness of this decomposition for such graphs. It generalizes ideas of Feigenbaum and Schäffer [4], who presented a polynomial algorithm for the prime factorization of connected graphs with respect to the strong product and a new proof of its uniqueness.

For Q_k we cannot apply the result in full strength, because the Cartesian skeleton of bipartite graphs is disconnected, whereas it is connected in the nonbipartite case. However, we can use the results of [11, Lemmas 5.34 and 5.35], which hold for Cartesian skeletons in general.

In particular, we can apply the fact from [11, Lemma 5.35] that two vertices x and y are an edge of the Cartesian skeleton if the intersection $N(x,y) = N(x) \cap N(y)$ of the neighborhoods $N(x)$ of x and $N(y)$ of y is strictly maximal in the set $\mathcal{N}(x) = \{N(x,y) \mid N(x,y) \neq \emptyset\}$.

PROPOSITION 5. *The Cartesian skeleton of Q_k consists of two (isomorphic) halved cubes H_1 and H_2.*

Proof. Any two vertices x and y with intersecting neighborhoods $N(x)$ and $N(y)$ have distance two; the intersection $N(x,y) = N(x) \cap N(y)$ has exactly two elements (for $k > 1$) and $N(x,y) = N(x,z)$ if and only if $x = y$. This implies that every $N(x,y)$ is strictly maximal in the set $\mathcal{N}(x) = \{N(x,y) \mid N(x,y) \neq \emptyset\}$. Therefore xy is an edge of the Cartesian skeleton H of Q_k if and only if $d(x,y) = 2$. Thus the Cartesian skeleton H of Q_k consists of the two halved cubes H_1 and H_2. □

Clearly H is disconnected because Q_k is bipartite. Nevertheless, every factorization of Q_k with respect to the direct product induces a factorization of H with respect to the Cartesian product. This means that Q_k cannot be a product of more factors with respect to the direct product than H with respect to the Cartesian one. We therefore decompose H first.

Either any two edges ab and ac of a halved cube are in a triangle abc or there are two triangles abd and adc (with the common edge bd). This implies that every halved cube is prime with respect to the Cartesian product. Thus, the only possible factorization of H with respect to the Cartesian product is $H_1 \,\square\, D_2$, where D_2 is the graph on two vertices without edges or loops.

For Q_k this implies that it can only be decomposed into a product $G \times K$ of two factors, where K is a graph on two vertices: where $V(H_1)$ projects onto one vertex of K and $V(H_2)$ onto the other. Since no pair of vertices in either H_1 or H_2 is adjacent in G, we infer that K cannot have loops.

Moreover, both G and K must be connected because Q_k is. We thus show the following proposition.

PROPOSITION 6. *Every factorization of Q_k with respect to the direct product is of the form $G \times K_2$. All such graphs G are prime with respect to the direct product.*

Together with Theorem 3 we can summarize our findings in the following theorem.

THEOREM 7. *Every decomposition of the hypercube Q_k into a direct product has exactly two factors. One factor is always K_2 and the other one any of the graphs Q_{k-1}^{α} for a bipartite involution α of Q_{k-1}.*

It would be interesting to enumerate the bipartite involutions of Q_k as well as the factorizations of Q_k with respect to the direct product. These questions are open.

We wish to conclude with the remark that nonunique factorizations can easily be found, also for factors different from K_2. For example, the direct product of a path P_n with a triangle is isomorphic to the product of P_n by a path of length two with loops added to the endpoints; cf. Figure 3 where an isomorphism is indicated for the

case $n = 5$.

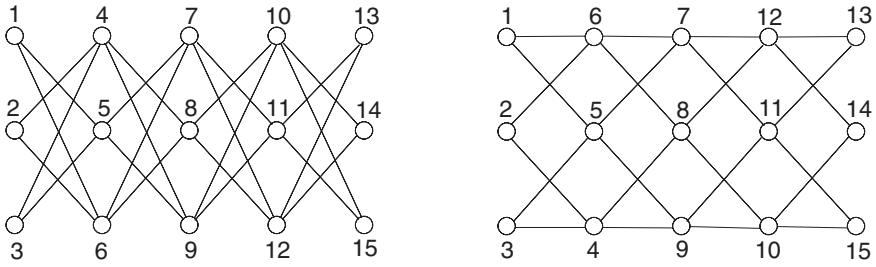


FIG. 3. *Isomorphic direct products.*

Acknowledgments. We wish to thank the referees and Doug Rall for invaluable suggestions and remarks.

REFERENCES

[1] M. H. ALBERT, D. HOLTON, AND R. J. NOWAKOWSKI, *The ultimate categorical matching in a graph*, Discrete Math., 232 (2001), pp. 1–9.
[2] B. BREŠAR, P. K. JHA, S. KLAVŽAR, AND B. ZMAZEK, *Median direct products of graphs*, Discuss. Math. Graph Theory, 25 (2005), in press.
[3] J. I. BROWN, R. J. NOWAKOWSKI, AND D. RALL, *The ultimate categorical independence ratio of a graph*, SIAM J. Discrete Math., 9 (1996), pp. 290–300.
[4] J. FEIGENBAUM AND A. A. SCHÄFFER, *Finding the prime factors of strong direct product graphs in polynomial time*, Discrete Math., 109 (1992), pp. 77–102.
[5] R. L. GRAHAM, *On primitive graphs and optimal vertex assignments*, Ann. New York Acad. Sci., 175 (1970), pp. 170–186.
[6] D. GREENWELL AND L. LOVÁSZ, *Applications of product colouring*, Acta Math. Acad. Sci. Hungar., 25 (1974), pp. 335–340.
[7] P. HELL, *An introduction to the category of graphs*, in Topics in Graph Theory (New York, 1977), Ann. New York Acad. Sci. 328, New York Acad. Sci., New York, 1979, pp. 120–136.
[8] M.-C. HEYDEMANN, J. G. PETERS, AND D. SOTTEAU, *Spanners of hypercube-derived networks*, SIAM J. Discrete Math., 9 (1996), pp. 37–54.
[9] L.-H. HSU, *A note on the ultimate categorical matching in a graph*, Discrete Math., 256 (2002), pp. 487–488.
[10] W. IMRICH, *Factoring cardinal product graphs in polynomial time*, Discrete Math., 192 (1998), pp. 119–144.
[11] W. IMRICH AND S. KLAVŽAR, *Product Graphs*, Wiley-Interscience, New York, 2000.
[12] W. IMRICH, S. KLAVŽAR, AND H. M. MULDER, *Median graphs and triangle-free graphs*, SIAM J. Discrete Math., 12 (1999), pp. 111–118.
[13] B. LAROSE AND C. TARDIF, *Projectivity and independent sets in powers of graphs*, J. Graph Theory, 40 (2002), pp. 162–171.
[14] R. MCKENZIE, *Cardinal multiplication of structures with a reflexive relation*, Fund. Math., 70 (1971), pp. 59–101.
[15] J. NEŠETŘIL, *Representations of graphs by means of products and their complexity*, in Mathematical Foundations of Computer Science, 1981 (Štrbské Pleso, 1981), Lecture Notes in Comput. Sci. 118, Springer, Berlin, New York, 1981, pp. 94–102.
[16] M. RAMRAS, *Congestion-free routings of linear complement permutations*, SIAM J. Discrete Math., 11 (1998), pp. 487–500.
[17] N. SAUER, *Hedetniemi's conjecture—A survey*, Discrete Math., 229 (2001), pp. 261–292.
[18] R. SQUIER, B. TORRENCE, AND A. VOGT, *The number of edges in a subgraph of a Hamming graph*, Appl. Math. Lett., 14 (2001), pp. 701–705.
[19] P. M. WEICHSEL, *The Kronecker product of graphs*, Proc. Amer. Math. Soc., 13 (1962), pp. 47–52.

[20] B. Yu, J. Cheriyan, and P. E. Haxell, *Hypercubes and multicommodity flows*, SIAM J. Discrete Math., 10 (1997), pp. 190–200.
[21] X. Zhu, *A survey on Hedetniemi's conjecture*, Taiwanese J. Math., 2 (1998), pp. 1–24.
[22] X. Zhu, *Construction of uniquely H-colorable graphs*, J. Graph Theory, 30 (1999), pp. 1–6.
[23] X. Zhu, *The fractional chromatic number of the direct product of graphs*, Glasg. Math. J., 44 (2002), pp. 103–115.

SIAM J. DISCRETE MATH.
Vol. 18, No. 4, pp. 787–793

SMALL CONTINGENCY TABLES WITH LARGE GAPS*

SETH SULLIVANT†

Abstract. We construct examples of contingency tables on n binary random variables where the gap between the linear programming lower/upper bound and the true integer lower/upper bounds on cell entries is exponentially large. These examples provide evidence that linear programming may not be an effective heuristic for detecting disclosures when releasing margins of multiway tables.

Key words. disclosure limitation, integer programming, Gröbner basis, contingency table

AMS subject classifications. 90C10, 62H17

DOI. 10.1137/S0895480104444090

1. Introduction.

A fundamental problem in data security is to determine what information about individual survey respondents can be inferred from the release of partial data. The particular instance of this problem we are interested in concerns the release of margins of a multidimensional contingency table. In particular, given a collection of margins of a multiway table, can individual cell entries in the table be inferred? This type of problem arises when statistical agencies like census bureaus release summary data to the public, but are required by law to maintain the privacy of individual respondents.

Many authors [1, 2, 3] have proposed that an individual cell entry is secure if, among all contingency tables with the given fixed marginal totals, the upper bound and lower bound for the cell entry are far enough apart. In general, solving the integer programs associated with finding the sharp integer upper and lower bounds on a cell entry is known to be NP-hard. A heuristic which has been suggested for approximating these upper and lower bounds is to solve the appropriate linear programming relaxations. Based on theoretical results for 2-way tables and practical experience for some small multiway tables, some authors have suggested that the linear programming bounds and other heuristics [1, 2, 5] should always constitute good approximations to the true bounds for cell values.

In this paper, we attempt to refute the claim that the linear programming bounds are, in general, good approximations to the true integer bounds. In particular, we will show the following theorem.

THEOREM 1. *There is a sequence of contingency tables on n binary random variables and a collection of margins such that the gap between the linear programming lower (upper) bounds and the integer programming lower (upper) bounds for a cell entry grows exponentially in n.*

For instance, on 10 binary random variables, our construction produces an instance where this difference is more than 100. This constitutes a significant discrepancy between the heuristic and reality, in a problem of size which is quite small from the practical standpoint. In previous work with Develin [4] we constructed a family of examples which imply that the gap grows at least linearly in the number of random variables.

*Received by the editors May 25, 2004; accepted for publication (in revised form) September 16, 2004; published electronically May 13, 2005.
http://www.siam.org/journals/sidma/18-4/44409.html
†Department of Mathematics, University of California, Berkeley, CA 94720–3840 (seths@math.berkeley.edu).

The outline of this paper is as follows. In the next section we review hierarchical models and the algebraic techniques that we will use to construct our examples. The third section is devoted to the explicit construction, and in the fourth section we discuss practical consequences of our examples.

2. Hierarchical models, Gröbner bases, and Graver bases. A hierarchical model is given by a simplicial complex Δ on the n-element set $[n] := \{1, 2, \ldots, n\}$ together with an integer vector $d = (d_1, \ldots, d_n)$. In the setting of probabilistic inference, a hierarchical model is intended to encode interactions between a collection of n discrete random variables: the number of states of the ith random variable is d_i and there is an interaction factor between the set of random variables indexed by each $F \in \Delta$ (see, for example, [9] for an introduction). From the standpoint of data security, n is the number of dimensions of a multiway contingency table, the d_i represent the number of levels in each dimension, and the elements $F \in \Delta$ index the particular margins that are released. For the rest of this paper $d_i = 2$ for all i; that is, we are considering *dichotomous* tables or *binary* random variables.

Computing the Δ-margins of a multiway table is a linear transformation. We denote by A_Δ the matrix in the standard basis that computes these margins. An explicit description of the construction of these matrices appears, for example, in [8].

Example 2. Consider the simplicial complex Γ_4 on $[3]$, $\Gamma_4 = \{\{1\}, \{2\}, \{3\}\}$. In other words, this simplicial complex consists of three isolated points; however, we could also describe this as the union of the boundary of a 1-simplex and an isolated point. The second interpretation plays a crucial role in the proof of the main theorem.

The 6×8 matrix A_{Γ_4} computes all the 1-way margins of a 3-way table. It is the following matrix:

$$A_{\Gamma_4} = \begin{array}{c} \\ \{1\},0 \\ \{1\},1 \\ \{2\},0 \\ \{2\},1 \\ \{3\},0 \\ \{3\},1 \end{array} \begin{array}{c} \begin{array}{cccccccc} 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array}.$$

Finding the minimum value for the $(0, 0, \ldots, 0) = 00\cdots0 = \mathbf{0}$ cell entry given the Δ-margins $\mathbf{b}$ amounts to solving the following integer program, which we denote IP_Δ:

$$\text{Minimize } u_{\mathbf{0}} \text{ subject to } A_\Delta \mathbf{u} = \mathbf{b}, \mathbf{u} \geq \mathbf{0}, \text{ and } \mathbf{u} \text{ integral.}$$

The linear programming relaxation drops the integrality condition. We denote it by LP_Δ:

$$\text{Minimize } u_{\mathbf{0}} \text{ subject to } A_\Delta \mathbf{u} = \mathbf{b} \text{ and } \mathbf{u} \geq \mathbf{0}.$$

The integer programming gap $gap_-(\Delta)$ is the largest difference between the optimal solution of IP_Δ and LP_Δ over all feasible marginals $\mathbf{b}$ [6]. Explicitly computing the integer programming gap is a difficult problem, even for quite small models Δ. However, using properties of Gröbner bases, it is sometimes possible to produce nontrivial lower bounds on this gap. Here is a rough definition of a Gröbner basis.

DEFINITION 3. *A reduced Gröbner basis $G_{\Delta,\mathbf{c}}$ of A_Δ with respect to the cost vector $\mathbf{c}$ is a minimal set of improving vectors that solves the integer program $IP_{\Delta,\mathbf{c}}$ for any feasible right-hand side $\mathbf{b}$.*

In the literature of discrete optimization, Gröbner bases are often called test sets. A lower bound on $gap_-(\Delta)$ is given by inspecting the coordinates of the Gröbner basis with respect to the cost vector $\mathbf{c} = \mathbf{e}_{00\cdots0}$.

THEOREM 4 (see [6, Corollary 4.3]). *The value $gap_-(\Delta)$ is greater than or equal to one less than the largest coordinate $g_{00\cdots0}$ of any element in the reduced Gröbner basis $G_{\Delta,\mathbf{c}}$ of A_Δ.*

The precise definition of the Gröbner basis with respect to a cost vector $\mathbf{c}$ can be found in [11]; however, we will restrict ourselves to a special family of models where the Gröbner basis elements we need have a simpler description. For this, we will need to recall the definition of the Graver basis. Note that any integer vector $\mathbf{u}$ can be written uniquely as $\mathbf{u} = \mathbf{u}^+ - \mathbf{u}^-$, where $\mathbf{u}^+$ and $\mathbf{u}^-$ are nonnegative with disjoint support.

DEFINITION 5. *A nonzero integer vector $\mathbf{u} \in \ker(A_\Delta)$ is called* primitive *if there does not exist an integer vector $\mathbf{v} \in \ker(A_\Delta)\backslash\{\mathbf{0}, \mathbf{u}\}$ such that $\mathbf{v}^+ \leq \mathbf{u}^+$ and $\mathbf{v}^- \leq \mathbf{u}^-$. The set of vectors $Gr(A_\Delta) = \{\mathbf{u} \in \ker(A_\Delta) | \mathbf{u}$ is primitive$\}$ is called the* Graver basis *of A_Δ.*

Given a simplicial complex Γ on $[n-1]$ there is a natural construction of a new simplicial complex $\Delta = \text{logit}(\Gamma)$ on $[n]$ which corresponds to taking the logit model with a binary response variable. The new model is defined as

$$\text{logit}(\Gamma) := \{S \cup \{n\} | S \in \Gamma\} \cup 2^{[n-1]},$$

where $2^{[n-1]}$ is the set of all subsets of $[n-1]$.

Example 6. Let $\Gamma_4 = \{\{1\}, \{2\}, \{3\}\}$ the simplicial complex on [3] from example 2. The new simplicial complex is

$$\Delta_4 = \text{logit}(\Gamma_4) = \{\{1,4\}, \{2,4\}, \{3,4\}, \{1,2,3\}\},$$

where we list only the facets in Δ_4.

After a suitable reordering of its rows and columns (see [8] or [10]), the matrix for $A_{\text{logit}(\Gamma)}$ takes the form

$$A_{\text{logit}(\Gamma)} = \begin{pmatrix} A_\Gamma & 0 \\ 0 & A_\Gamma \\ I_{2^{n-1}} & I_{2^{n-1}} \end{pmatrix},$$

where $I_{2^{n-1}}$ is the $2^{n-1} \times 2^{n-1}$ identity matrix. In particular, $A_{\text{logit}(\Gamma)}$ is the Lawrence lifting of A_Γ [10]. Note that $\ker(A_\Gamma)$ and $\ker(A_{\text{logit}(\Gamma)})$ are isomorphic as vector spaces, and there is a natural identification: $\mathbf{u} \in \ker(A_\Gamma)$ if and only if $(\mathbf{u}, -\mathbf{u}) \in \ker(A_{\text{logit}(\Gamma)})$. A fundamental fact about Lawrence liftings (and hence, logit models) is that their Gröbner bases are easy to describe in terms of the Graver basis of A_Γ, as seen in the following theorem.

THEOREM 7 (see [11, Theorem 7.1]). *Let Γ be a model and $\Delta = logit(\Gamma)$; then*

1. $Gr(A_\Delta) = \{(\mathbf{u}, -\mathbf{u}) | \mathbf{u} \in Gr(A_\Gamma)\}$,
2. $\{\mathbf{g} \in Gr(A_\Delta) | \mathbf{c} \cdot \mathbf{g} > 0\} \subseteq G_{\Delta,\mathbf{c}}$.

Note that Theorem 7 is only true when the response variable is binary. We now have all the tools in hand to construct our example.

3. The construction. Our main result is the following theorem.

THEOREM 8. *For each $n \geq 3$, there is a hierarchical model Δ_n on n-binary random variables such that*

$$gap_-(\Delta_n) \geq 2^{n-3} - 1.$$

A similar statement about exponential growth of the gap for upper bounds can be derived by an analogous argument.

Proof. Our strategy will be to construct a hierarchical model Δ_n which has a Gröbner basis element whose $\mathbf{0}$ entry is large. This will force the large gap by Theorem 4.

Let B_n be the hierarchical model on $n-2$ random variables

$$B_n = \{S | S \subset [n-2], S \neq [n-2]\}$$

and let Γ_n be the hierarchical model on $n-1$ random variables

$$\Gamma_n = B_n \cup \{\{n-1\}\}.$$

That is, Γ_n is the union of the boundary of an $n-3$ simplex together with an isolated point. Take $\Delta_n = \text{logit}(\Gamma_n)$.

To show the theorem with respect to Δ_n, it suffices to show that A_{Γ_n} has an element in its Graver basis that has a large entry in its $\mathbf{0}$ coordinate, by Theorem 7. Consider the vector

$$\mathbf{f}_n = 2^{n-3} e_{(\mathbf{0},0)} + \sum_{\mathbf{i} | \mathbf{i} \neq \mathbf{0}, \sum i_j even} e_{(\mathbf{i},1)} \; - \; (2^{n-3}-1) e_{(\mathbf{0},1)} - \sum_{\mathbf{i} | \sum i_j odd} e_{(\mathbf{i},0)}.$$

Here $e_{(\mathbf{i},k)}$ denotes the standard unit vector whose index is $(\mathbf{i},k) \in \{0,1\}^{n-1}$; that is, $e_{(\mathbf{i},k)}$ is the integral table whose only nonzero entry is a one in the $(\mathbf{i},k)$ position. Note that $\mathbf{i} \in \{0,1\}^{n-2}$ is an index on the first $n-2$ random variables.

We will now show that $\mathbf{f}_n$ is a primitive vector in $\ker(A_{\Gamma_n})$. First we must show that $\mathbf{f}_n \in \ker(A_{\Gamma_n})$. This amounts to showing that the margin of $\mathbf{f}_n$ is zero for all $F \in \Gamma_n$. Given a subset $F \subseteq [n-1]$, we denote by $\mathbf{f}_n|_F$ the F-marginal of $\mathbf{f}_n$. Given a multiway table presented as the sum of standard unit vectors the F-marginal is computed by deleting the indices in $[n-1] \setminus F$. Thus we have

$$\mathbf{f}_n|_{\{n-1\}} = 2^{n-3} e_0 + \sum_{\mathbf{i} | \mathbf{i} \neq \mathbf{0}, \sum i_j even} e_1$$

$$-(2^{n-3}-1) e_1 - \sum_{\mathbf{i} | \sum i_j odd} e_{(0)}$$

$$= 2^{n-3} e_0 + (2^{n-3}-1) e_1 - (2^{n-3}-1) e_1 - 2^{n-3} e_0 = \mathbf{0}.$$

Now we must show that $\mathbf{f}_n|_F = \mathbf{0}$ for each $F \in [n-2]$ with $F \neq [n-2]$. Since $\mathbf{f}_n$ is symmetric with respect to permuting the indices on $[n-2]$, it suffices to show that the margin $\mathbf{f}_n|_{[n-3]}$ is zero. We compute

$$\mathbf{f}_n|_{[n-3]} = 2^{n-3} e_{(\mathbf{0},0)}|_{[n-3]} + \sum_{\mathbf{i} | \mathbf{i} \neq \mathbf{0}, \sum i_j even} e_{(\mathbf{i},1)}|_{[n-3]}$$

$$-(2^{n-3}-1) e_{(\mathbf{0},1)}|_{[n-3]} - \sum_{\mathbf{i} | \sum i_j odd} e_{(\mathbf{i},0)}|_{[n-3]}$$

$$= 2^{n-3} e_{\mathbf{0}} + \sum_{\mathbf{i} \in \{0,1\}^{n-3} | \mathbf{i} \neq \mathbf{0}} e_{\mathbf{i}} \; - \; (2^{n-3}-1) e_{\mathbf{0}} - \sum_{\mathbf{i} \in \{0,1\}^{n-3}} e_{\mathbf{i}} = \mathbf{0}.$$

We have shown that $\mathbf{f}_n|_F = \mathbf{0}$ for each $F \in \Gamma_n$; hence $\mathbf{f}_n \in \ker(A_{\Gamma_n})$.

Now we must show that $\mathbf{f}_n$ is a primitive vector in $\ker(A_{\Gamma_n})$. Suppose to the contrary that there was some nontrivial integral $\mathbf{g}_n \in \ker(A_{\Gamma_n})$ such that $\mathbf{g}_n^+ \leq \mathbf{f}_n^+$ and $\mathbf{g}_n^- \leq \mathbf{f}_n^-$. First note that $\mathbf{g}_n \in \ker(A_{\Gamma_n})$ implies $\mathbf{g}_n|_{[n-2]} \in \ker(A_{B_n})$. This follows because if $F \subset [n-2]$, then $\mathbf{g}_n|_F = \mathbf{g}_n|_{[n-2]}|_F$. However, by Theorem 2.6 in [7], we know that $\ker(A_{B_n})$ is spanned by the single vector

$$\mathbf{h}_n = \sum_{\mathbf{i}|\sum i_j even} e_{\mathbf{i}} - \sum_{\mathbf{i}|\sum i_j odd} e_{\mathbf{i}},$$

so $\mathbf{g}_n|_{[n-2]}$ is an integral multiple $\mathbf{h}_n$. An arbitrary integral vector $\mathbf{u}$ whose $[n-2]$ marginal is equal to $a \cdot \mathbf{h}_n$ is a vector which can be written in the form

$$\mathbf{u} = \sum_{\mathbf{i}|\sum i_j even} (a+b_{\mathbf{i}})e_{(\mathbf{i},0)} - \sum_{\mathbf{i}|\sum i_j even} b_{\mathbf{i}} e_{(\mathbf{i},1)} - \sum_{\mathbf{i}|\sum i_j odd} (a+b_{\mathbf{i}})e_{(\mathbf{i},0)} + \sum_{\mathbf{i}|\sum i_j odd} b_{\mathbf{i}} e_{(\mathbf{i},1)}$$

for 2^{n-2} arbitrary parameters $b_{\mathbf{i}}$. In particular, $\mathbf{g}_n$ has this form. Since $\mathbf{g}_n^+ \leq \mathbf{f}_n^+$ and $\mathbf{g}_n^- \leq \mathbf{f}_n^-$, it must be the case that the support of $\mathbf{g}_n$ is contained in the support of $\mathbf{f}_n$. This allows us to deduce that the coefficient of $e_{(\mathbf{i},0)}$ is zero whenever $\sum i_j$ is even and not zero and the coefficient of $e_{(\mathbf{i},1)}$ is zero whenever $\sum i_j$ is odd. This, in turn, forces $b_{\mathbf{i}} = 0$ for all $\mathbf{i} \neq \mathbf{0}$. Thus, we can see that $\mathbf{g}_n$ has the form

$$\mathbf{g}_n = (a+b_{\mathbf{0}})e_{(\mathbf{0},0)} + \sum_{\mathbf{i}|\mathbf{i}\neq\mathbf{0},\sum i_j even} a e_{(\mathbf{i},1)} - b_{\mathbf{0}} e_{(\mathbf{0},1)} - \sum_{\mathbf{i}|\sum i_j odd} a e_{(\mathbf{i},0)}.$$

Since $\mathbf{g}_n \in \ker(A_{\Gamma_n})$ we must have $\mathbf{g}_n|_{\{n-1\}} = \mathbf{0}$. From this condition, we deduce that $b_{\mathbf{0}} = (2^{n-3}-1)a$ and, hence, that $\mathbf{g}_n = a\mathbf{f}_n$. Since $\mathbf{g}_n^+ \leq \mathbf{f}_n^+$ and $\mathbf{g}_n^- \leq \mathbf{f}_n^-$, we must have $a = 0, 1$, which contradicts our assumption about the nontriviality of $\mathbf{g}_n$ and implies that $\mathbf{f}_n$ is primitive. □

To explicitly construct an example of a set of margins $\mathbf{b}$ with respect to Δ_n where the gap between the LP and IP optima is $2^{n-3}-1$, just take

$$\mathbf{u} = (2^{n-3}-1)e_{(\mathbf{0},0,0)} + \sum_{\mathbf{i}|\mathbf{i}\neq\mathbf{0},\sum i_j even} e_{(\mathbf{i},1,0)} + (2^{n-3}-1)e_{(\mathbf{0},1,1)} + \sum_{\mathbf{i}|\sum i_j odd} e_{(\mathbf{i},0,1)},$$

and $\mathbf{b} = A_{\Delta_n}\mathbf{u}$. It follows that $\mathbf{u}$ cannot be improved to a nonnegative integer table with smaller $(\mathbf{0},0,0)$ coordinate by appealing to the Gröbner basis. However, the nonnegative rational vector

$$\mathbf{v} = \mathbf{u} - \frac{2^{n-3}-1}{2^{n-3}}(\mathbf{f}_n, -\mathbf{f}_n)$$

has the same margins $\mathbf{b}$ as $\mathbf{u}$ but has $(\mathbf{0},0,0)$ coordinate 0. Furthermore, $\mathbf{u}$ is the only nonnegative integral vector with these margins, so a heuristic which relies solely on the LP bounds would be inclined to declare the $\mathbf{0}$ cell entry safe (the LP upper and lower bounds differ by at least $2^{n-3}-1$), whereas the integer bounds show that the entry is disclosed.

Example 9. In the case $n = 5$, the 4-way array $\mathbf{f}_5$ can be represented "in the plane" as a partitioned 4×4 array:

$$\mathbf{f}_5 = \left(\begin{array}{cc|cc} 4 & -1 & -1 & 0 \\ -1 & 0 & 0 & -1 \\ \hline -3 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{array}\right).$$

The 5-way table $\mathbf{u}$ described above whose gap is equal to three is represented by two partitioned 4×4 arrays:

$$\mathbf{u} = \left(\begin{array}{cc|cc} 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{array}\right) \left(\begin{array}{cc|cc} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ \hline 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right).$$

An LP optimal table $\mathbf{v}$ for the associated marginal constraint $A_{\Delta_5}\mathbf{v} = A_{\Delta_5}\mathbf{u}$ is the 5-way table:

$$\mathbf{v} = \left(\begin{array}{cc|cc} 0 & \frac{3}{4} & \frac{3}{4} & 0 \\ \frac{3}{4} & 0 & 0 & \frac{3}{4} \\ \hline \frac{9}{4} & 0 & 0 & \frac{1}{4} \\ 0 & \frac{1}{4} & \frac{1}{4} & 0 \end{array}\right) \left(\begin{array}{cc|cc} 3 & \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{1}{4} & 0 & 0 & \frac{1}{4} \\ \hline \frac{3}{4} & 0 & 0 & \frac{3}{4} \\ 0 & \frac{3}{4} & \frac{3}{4} & 0 \end{array}\right).$$

4. Discussion. In this paper, we constructed an example to show that the gap between the linear programming lower bounds and the integer programming lower bounds for a cell entry can be exponentially large in the number of binary random variables of a hierarchical model. Previous explicit constructions of this type [4] gave gaps that were linear in the number of random variables.

There are a number of possible modifications to our result which can be made to produce examples of different flavors. For instance, small modifications of our argument can be used to produce exponential gaps between the linear programming and integer programming upper bounds for cell entries. Furthermore, by adding extra dimensions by subdividing Δ and using some of the techniques in [4], one can produce instances of purely graphical models with these exponential growth properties.

While it is not clear how often one should expect to encounter the exponentially large gaps we have demonstrated, we believe that for problems on large sparse tables, large gaps between the LP and IP solutions will be not be exceptional. This feeling is based on the observation that if any particular gap value can occur, then so can all the integer values smaller than this gap. This suggests that research needs to be done to determine better heuristics for approximating bounds on cell entries in large sparse tables.

Acknowledgment. The author is grateful to an anonymous referee whose comments greatly improved the paper.

REFERENCES

[1] L. Buzzigoli and A. Giusti, *An algorithm to calculate the lower and upper bounds of the elements of an array given its marginals*, in Statistical Data Protection Proceedings, Eurostat, Luxembourg, 1999, pp. 131–147.

[2] S. D. Chowdhury, G. T. Duncan, R. Krishnan, S. F. Roehrig, and S. Mukherjee, *Disclosure detection in multivariate categorical databases: Auditing confidentiality protection through two new matrix operators*, Management Sci., 45 (1999), pp. 1710–1723.

[3] L. Cox and J. George, *Controlled rounding for tables with subtotals*, Ann. Oper. Res., 20 (1989), pp. 141–157.

[4] M. Develin and S. Sullivant, *Markov bases of binary graph models*, Ann. Combin., 7 (2003), pp. 441–466.

[5] A. Dobra and S. E. Fienberg, *Bounds for cell entries in contingency tables induced by fixed marginal totals*, UNECE Statist. J., 18 (2001), pp. 363–371.

[6] S. Hoşten and B. Sturmfels, *Computing the integer programming gap*, Combinatorica, to appear.

[7] S. Hoşten and S. Sullivant, *Gröbner bases and polyhedral geometry of reducible and cyclic models*, J. Combin. Theory Ser. A, 100 (2002) pp. 277–301.
[8] S. Hoşten and S. Sullivant, *A finiteness theorem for Markov bases of hierarchical models*, math.CO/0401379, 2004, submitted.
[9] S. Lauritzen, *Graphical Models*, Oxford University Press, New York, 1996.
[10] F. Santos and B. Sturmfels, *Higher Lawrence configurations*, J. Combin. Theory Ser. A, 103 (2003) pp. 151–164.
[11] B. Sturmfels, *Gröbner Bases and Convex Polytopes*, AMS, Providence, RI, 1996.

SIAM J. DISCRETE MATH.
Vol. 18, No. 4, pp. 794–824

ON THE BEHAVIOR OF A FAMILY OF META-FIBONACCI SEQUENCES*

JOSEPH CALLAGHAN†, JOHN J. CHEW III†, AND STEPHEN M. TANNY†

Abstract. A family of meta-Fibonacci sequences is defined by the k-term recursion

$$T_{a,k}(n) := \sum_{i=0}^{k-1} T_{a,k}(n-i-a-T_{a,k}(n-i-1)), \quad n > a+k,\ k \geq 2,$$

with initial conditions $T_{a,k}(n) = 1$ for $1 \leq n \leq a+k$. Some partial results are obtained for $a \geq 0$ and $k > 1$. The case $a = 0$ and k odd is analyzed in detail, giving a complete characterization of its structure and behavior, marking the first time that such a parametric family of meta-Fibonacci sequences has been solved. This behavior is considerably more complex than that of the more familiar Conolly sequence ($a = 0$, $k = 2$). Various properties are derived: for example, a certain difference of summands turns out to consist of palindromic subsequences, and the mean values of the functions on these subsequences are computed. Conjectures are made concerning the still more complex behavior of $a = 0$ and even $k > 2$.

Key words. Hofstadter, iterated recursion, meta-Fibonacci, Q sequence

AMS subject classifications. 11B37, 11B39, 11B99

DOI. 10.1137/S0895480103421397

1. Introduction. In this paper, all values are integers and defining equations use the ":=" operator. For a sequence T and an integer d, we write $\Delta_d T(n) := T(n) - T(n-d)$.

Hofstadter's Q sequence $Q(n)$, illustrated in Figure 1.1, first mentioned in [7] and defined by the "self-referencing" recursion

$$Q(n) := Q(n-Q(n-1)) + Q(n-Q(n-2)), \quad n > 2, \tag{1.1}$$

and initial conditions $Q(1) := Q(2) := 1$ is the most famous example of a so-called *meta-Fibonacci sequence*. This sequence, recently renamed $U(n)$ by Hofstadter and his collaborators [9], remains the focus of ongoing investigation in [8] and elsewhere, although to date very little has been proven about its enigmatic behavior.

At the same time, various authors have examined seemingly close relatives to the above recursion, which have turned out to be far better behaved and about which a great deal can be demonstrated. In [3], Conolly introduced the following very well-behaved variant of the Q-sequence recursion, illustrated in Figure 1.2:

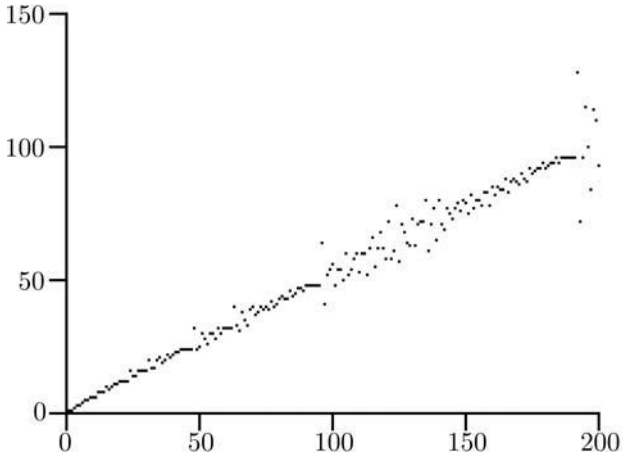
$$F(n) := F(n-F(n-1)) + F(n-1-F(n-2)), \quad n > 2, \tag{1.2}$$

with initial conditions $\{F(1) = 0,\ F(2) = 1\}$ or $\{F(1) = 1,\ F(2) = 1\}$. He notes that for $n > 2$ the recursion yields the same sequence whether $F(1) = 0$ or $F(1) = 1$.

Much can be said about this sequence. For example, Conolly shows that if $F(1) = 0$ and $n = 2^i + j$ with $i \geq 1$ and $0 \leq j < 2^i$, then

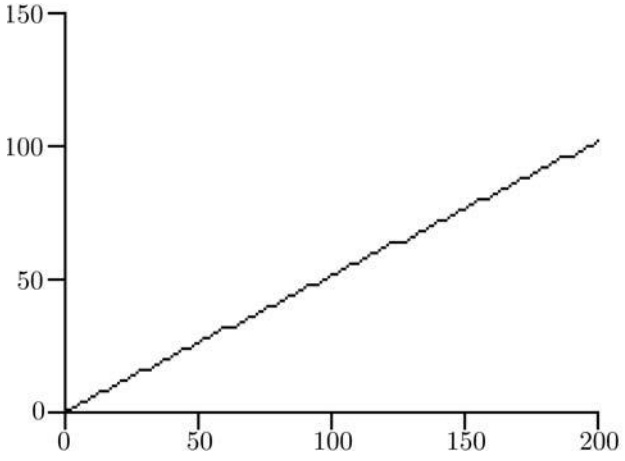
$$F(n) = 2^{i-1} + F(j+1). \tag{1.3}$$

*Received by the editors January 16, 2003; accepted for publication (in revised form) November 3, 2004; published electronically May 13, 2005.
http://www.siam.org/journals/sidma/18-4/42139.html
†Department of Mathematics, University of Toronto, Toronto, ON, M5S 3G3, Canada (calaghan@math.utoronto.ca, jjchew@math.utoronto.ca, tanny@math.utoronto.ca).



n	$Q(n)$	n	$Q(n)$
1	1	13	8
2	1	14	8
3	2	15	10
4	3	16	9
5	3	17	10
6	4	18	11
7	5	19	11
8	5	20	12
9	6	21	12
10	6	22	12
11	6	23	12
12	8	24	16

FIG. 1.1. *Hofstadter's Q sequence.*



n	$F(n)$	n	$F(n)$
1	1	13	8
2	1	14	8
3	2	15	8
4	2	16	8
5	3	17	9
6	4	18	10
7	4	19	10
8	4	20	11
9	5	21	12
10	6	22	12
11	6	23	12
12	7	24	13

FIG. 1.2. *Conolly's F sequence.*

Tanny [14] and Higham and Tanny [5, 6] developed a considerably more extensive analysis of the very similar recursion, illustrated in Figure 1.3, defined by

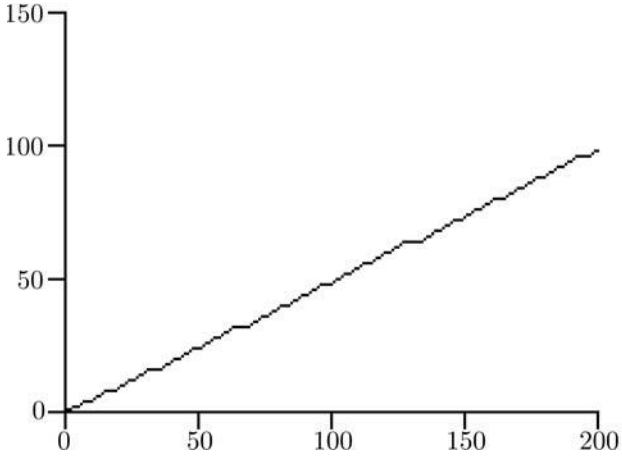
$$T(n) := T(n-1-T(n-1)) + T(n-2-T(n-2)), \quad n > 2, \tag{1.4}$$

with initial conditions $T(0) := T(1) := T(2) := 1$, which generates a sequence with highly analogous properties. Detailed results were provided concerning the structure and behavior of this sequence, and of some other sequences generated by (1.4) but with alternative choices for the initial conditions.

In [5], a k-term generalization of (1.4) was suggested:

$$T_k(n) := \sum_{i=1}^{k} T_k(n-i-T_k(n-i)), \quad n > k. \tag{1.5}$$

It was proven there that with the initial conditions $T_k(0) := T_k(1) := 1$ and $T_k(i) := i-1,\ 2 \le i \le k$, the sequence generated by (1.5) behaves in very much the



n	$T(n)$	n	$T(n)$
1	1	13	6
2	1	14	7
3	2	15	8
4	2	16	8
5	2	17	8
6	3	18	8
7	4	19	8
8	4	20	9
9	4	21	10
10	4	22	10
11	5	23	11
12	6	24	12

FIG. 1.3. *Tanny's T sequence.*

same way as the sequence generated by (1.4) with initial conditions $T(0) := T(1) := T(2) := 1$. That is, it is monotone, its consecutive terms increase by either 0 or 1, and it hits every positive integer. Further, we can determine explicit formulae that characterize its behavior.

The work cited above with (1.5) demonstrates that recursions expressible as homogeneous sums may be approachable even when the sums have many terms. Motivated by this success, we introduce an even more general formulation of the Conolly recursion (1.2) that incorporates all of the specific variants discussed above.

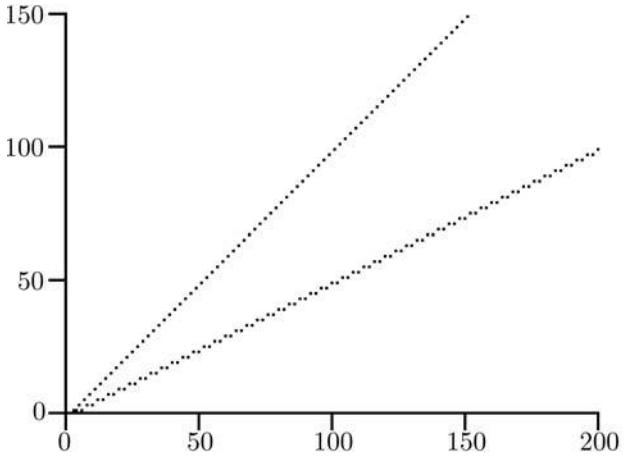
For $a \geq 0$ and $k \geq 2$ consider the family of recursions

$$T_{a,k}(n) := \sum_{i=0}^{k-1} T_{a,k}(n-i-a-T_{a,k}(n-i-1)), \quad n > a+k. \tag{1.6}$$

Equation (1.6) reduces to (1.2) when $a = 0$ and $k = 2$, to (1.4) when $a = 1$ and $k = 2$, and to (1.5) when $a = 1$. For $k = 2$ and arbitrary (even negative) a, Elzinga [4] reports that some analogues of certain results of (1.5) appear to hold, although no proof is provided; again, for $k = 2$ and $a \in \{-1, -2\}$, he describes the behavior of (1.6) for a substantial number of sets of initial conditions in an attempt to identify a behavioral classification scheme. For $k = 3$ and $a = 0$, Allenby and Smith [1] present some initial results and conjectures.

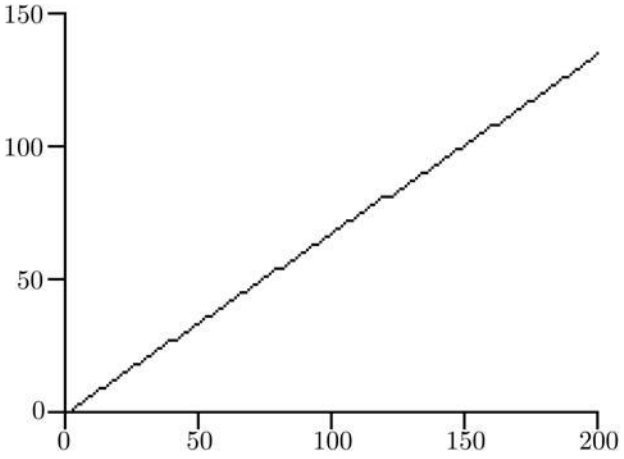
As is typical for meta-Fibonacci recursions, the behavior of the individual members of this family is highly sensitive to the choice of the parameters a and k and to the initial conditions. Some choices lead to sequences with identifiable and regular (though potentially very complex) patterns, while others generate highly chaotic sequences or even cause the sequence $T_{a,k}(n)$ to fail to be defined for some n.

For example, consider the choice of initial values for the sequence $T_{0,3}$. If we require that each of the three summands that make up the recursive definition (1.6) of $T_{0,3}(4)$ evaluates to one of the chosen initial values, then the three conditions $1 \leq 4-i-T_{0,3}(3-i) \leq 3$, $i \in \{1,2,3\}$, allow 27 possible sets of initial values ranging lexicographically from $(T_{0,3}(1), T_{0,3}(2), T_{0,3}(3)) = (-1,0,1)$ to $(1,2,3)$. Four of the 27 choices—$(-1,0,2)$, $(-1,0,3)$, $(-1,2,3)$, and $(1,0,3)$—give a $T_{0,3}(n)$ which is not well defined for some $5 \leq n \leq 7$. In the other 23 cases, we have verified empirically that $T_{0,3}(n)$ is defined at least up to $n = 10{,}000$.



n	$T(n)$	n	$T(n)$
1	0	13	12
2	0	14	6
3	2	15	14
4	2	16	8
5	4	17	16
6	2	18	8
7	6	19	18
8	4	20	10
9	8	21	20
10	4	22	10
11	10	23	22
12	6	24	12

FIG. 1.4. $T_{0,3}$ *with initial values* $(0, 0, 2)$.

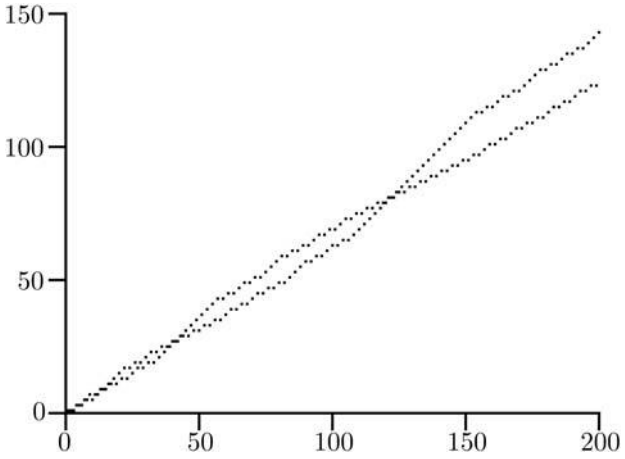


n	$T(n)$	n	$T(n)$
1	0	13	9
2	0	14	9
3	1	15	9
4	2	16	10
5	3	17	11
6	3	18	12
7	4	19	12
8	5	20	13
9	6	21	14
10	6	22	15
11	7	23	15
12	8	24	16

FIG. 1.5. $T_{0,3}$ *with initial values* $(0, 0, 1)$.

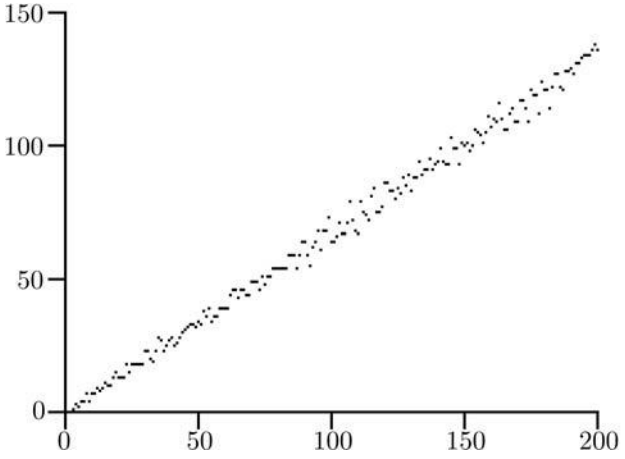
Four of the remaining 23 cases—$(0, 0, 2)$, $(0, 0, 1)$, $(1, 1, 1)$, and $(-1, 0, 1)$—are graphed and tabulated in Figures 1.4, 1.5, 1.6, and 1.7 and illustrate the wide range of possible behavior for these sequences. Observe that the first and third sequences visibly bifurcate into an even subsequence and an odd subsequence, each of which is monotonic even though the full sequence is not. The second sequence appears to be very similar to Conolly's sequence, while the fourth sequence looks completely chaotic.

It is also typical of meta-Fibonacci recursions that alternative choices for the parameters and initial conditions can lead to essentially the same sequence. In this vein, it is not difficult to show that if $T(n)$ is always well defined and positive, and some other $T^*(n)$ defined according to (1.6) satisfies $T^*(N + p) = T(p)$ for an $N \geq 0$ and $1 \leq p \leq a + k$, then $T^*(N + p) = T(p)$ for all $p \geq 1$. For instance, setting $a = 0$, $k = 3$, and $\{T(n)\}_{n=1,2,3} = (-1, 1, 2)$ or $(1, 1, 1)$ gives two sequences that are identical except that one has four extra values at its start. We also note that we may replace n by $n' + x$ in (1.6) to shift a sequence x places.



n	$T(n)$	n	$T(n)$
1	1	13	9
2	1	14	9
3	1	15	9
4	3	16	11
5	3	17	11
6	3	18	13
7	5	19	11
8	5	20	15
9	7	21	13
10	5	22	17
11	7	23	13
12	7	24	17

FIG. 1.6. $T_{0,3}$ *with initial values* $(1, 1, 1)$.



n	$T(n)$	n	$T(n)$
1	-1	13	8
2	0	14	9
3	1	15	11
4	3	16	10
5	2	17	10
6	4	18	13
7	4	19	15
8	7	20	13
9	4	21	13
10	7	22	13
11	7	23	18
12	9	24	15

FIG. 1.7. $T_{0,3}$ *with initial values* $(-1, 0, 1)$.

In this paper, we analyze the general family defined by (1.6) and initial conditions $T_{a,k}(n) = 1$ for $1 \le n \le a + k$. We choose this set of initial values because, as shown in Figure 1.6, its bifurcated, undulating behavior is complex enough to be interesting while still structured enough to be amenable to rigorous analysis.

In section 2 we prove that for general a, the sequence generated by each recursion in the family can be decomposed into k disjoint subsequences, each of which begins with 1 and increases monotonically in increments of 0 or $k - 1$. Further, for all odd k, the differences $T_{a,k}(n) - T_{a,k}(n - 2)$ are also always either 0 or $k - 1$. This finding substantially generalizes an earlier result of this type for (1.4) appearing in Higham and Tanny [5, 6].

In sections 3 and 4, we apply and extend the results of section 2 to derive a detailed characterization of the behavior of (1.6) in the special case $a = 0$, $k = 3$ and explore the many interesting properties and beautiful symmetries that it exhibits. This case illustrates well the additional complexity that occurs once k is increased beyond 2, yet remains manageable for expository purposes. In the course of our work,

TABLE 2.1
$T_{0,7}(n)$.

	n							n					
	1	2	3	4	5	6		1	2	3	4	5	6
$T(n+0)$	1	1	1	1	1	1	$T(n+60)$	49	49	49	55	55	55
$T(n+6)$	1	7	7	7	7	7	$T(n+66)$	55	55	55	61	55	61
$T(n+12)$	7	7	13	13	13	13	$T(n+72)$	61	61	61	67	61	67
$T(n+18)$	13	13	19	13	19	19	$T(n+78)$	61	67	67	73	67	73
$T(n+24)$	19	19	25	19	25	19	$T(n+84)$	67	73	67	79	73	79
$T(n+30)$	25	25	31	25	31	25	$T(n+90)$	73	79	73	85	73	85
$T(n+36)$	31	25	31	31	37	31	$T(n+96)$	79	85	79	91	79	91
$T(n+42)$	37	31	37	37	37	37	$T(n+102)$	79	91	85	97	85	97
$T(n+48)$	43	37	43	43	43	43	$T(n+108)$	85	97	85	103	91	103
$T(n+54)$	43	43	49	49	49	49	$T(n+114)$	91	103	91	109	91	109

we considerably extend the results on this case that appear in [1] and settle all the conjectures stated there.

While the case $a = 0$, $k = 3$ is interesting in its own right, what is more exciting is that the behavior in this case appears to provide a roadmap for understanding and characterizing the detailed behavior of the sequences for all odd k. This is the content of section 5, where we prove that certain key results derived for $a = 0$, $k = 3$ hold for all odd k. On this basis, we show that the detailed structure for all odd k is analogous to that shown for $k = 3$. To the best of our knowledge, this is the first instance in the area of meta-Fibonacci recursions where such broadly based behavioral results have been shown to hold for sequences with such complex behavior.

For even $k \geq 4$ the behavior of $T(n)$ is much more complicated and erratic than for odd k (the case $k = 2$, the Conolly sequence discussed earlier, is completely understood and quite straightforward). There is evidence that the same kind of approaches that we used for odd k can be adapted for even k, although we have not yet investigated this very far. In section 6, we conclude with some conjectures relating to even k.

2. A family of meta-Fibonacci sequences. In this section, we prove some results concerning the general recursion

$$T_{a,k}(n) := \sum_{i=0}^{k-1} T_{a,k}\big(n - i - a - T_{a,k}(n-i-1)\big), \quad n > a + k, \tag{1.6}$$

$$T_{a,k}(n) := 1, \quad 1 \leq n \leq k + a, \tag{2.1}$$

which for brevity's sake we will refer to simply as $T(n)$.

Table 2.1 lists some values of $T(n)$ for the case $a = 0$, $k = 7$ and lets us observe a few of its properties: $T(n) \equiv 1 \pmod{k-1}$; $T(n)$ is not monotonic but *is* composed of $k-1$ interleaved monotonic subsequences $\{T((k-1)i+j)\}_{i=0}^{\infty}$, $j = 1, \ldots, k-1$; and each of these subsequences includes every possible value subject to the modulo constraint. That is, $\Delta_{k-1}T(n) := T(n) - T(n-k+1) \in \{0, k-1\}$ for all $n \geq k$. Furthermore, in the case of odd k, we will find that $\Delta_d T(n) := T(n) - T(n-d) \in \{0, k-1\}$ for all even $d \in [0, k-1]$.

We begin our analysis of $T(n)$. In the case of an ordinary Fibonacci sequence, the recursive definition of a sequence member refers simply to immediately preceding sequence members. In the meta-Fibonacci sequence $T(n)$, the recursive summands are usually much earlier sequence members, whose distance from the current n can vary

TABLE 2.2
$U_{0,7}(n)$.

	n							n					
	1	2	3	4	5	6		1	2	3	4	5	6
$U(n+0)$	–	1	1	1	1	1	$U(n+60)$	7	7	7	13	7	7
$U(n+6)$	1	1	1	1	1	1	$U(n+66)$	7	7	7	13	7	13
$U(n+12)$	1	1	7	1	1	1	$U(n+72)$	7	7	7	13	7	13
$U(n+18)$	1	1	7	1	7	1	$U(n+78)$	7	13	7	13	7	13
$U(n+24)$	1	1	7	1	7	1	$U(n+84)$	7	13	7	19	7	13
$U(n+30)$	7	1	7	1	7	1	$U(n+90)$	7	13	7	19	7	19
$U(n+36)$	7	1	7	7	7	1	$U(n+96)$	7	13	7	19	7	19
$U(n+42)$	7	1	7	7	7	7	$U(n+102)$	7	19	7	19	7	19
$U(n+48)$	7	1	7	7	7	7	$U(n+108)$	7	19	7	25	7	19
$U(n+54)$	7	7	7	7	7	7	$U(n+114)$	7	19	7	25	7	25

considerably within the sum. We therefore need to examine these summands more closely and reword the definition (1.6) of $T_{a,k}(n)$ to label two functions of subsequent interest:

$$T_{a,k}(n) = \sum_{i=0}^{k-1} U_{a,k}(n-i), \quad n > a+k, \tag{2.2}$$

where

$$U_{a,k}(n) := T_{a,k}(R_{a,k}(n)), \quad n > a+1,$$
$$R_{a,k}(n) := n - a - T_{a,k}(n-1), \quad n > 1.$$

As with $T(n)$, we abbreviate $U_{a,k}(n)$ to $U(n)$ and $R_{a,k}(n)$ to $R(n)$ wherever the understood subscripts are clear. Returning to the case $a = 0$, $k = 7$, Table 2.2 lists the first 119 values of $U(n)$ and shows that it has the same modulo and subsequence properties that $T(n)$ has; that is, $\Delta_{k-1}U(n)$ takes on only the two values 0 and $k-1$.[1] Table 2.3 lists the first 119 values of $R(n)$, which also shares all of these properties with $T(n)$ and $U(n)$.

Note 2.1. We can make a stronger observation for $\Delta_6 U(n)$ that turns out to generalize to any k: every pair of 6's in this difference sequence appears to be separated by at least seven zeros. The following are the first 88 values of $\Delta_6 U(n)$, listed with a break before each 6: $\{\Delta_6 U(n)\}_{n=8}^{95} = \{0,0,0,0,0,0,0,$ $6,0,0,0,0,0,0,0,$ $6,0,0,0,0,0,0,0,$ $6,$ $0,0,0,0,0,0,0,$ $6,0,0,0,0,0,0,0,$ $6,0,0,0,0,0,0,0,$ $6,0,0,0,0,0,0,0,$ $6,0,0,0,0,0,0,0,$ $6,0,0,0,0,$ $0,0,0,$ $6,0,0,0,0,0,0,0,$ $6,0,0,0,0,0,0,0\}$.

Since $\Delta_{k-1}T(n) = \sum_{i=0}^{k-1} \Delta_{k-1}U(n-i)$, it follows that if we can fully describe $\Delta_{k-1}U(n)$, then we have described its double sum $T(n)$. In fact we will demonstrate that, as is illustrated in Note 2.1, $\Delta_{k-1}U(n)$ has a beautiful, highly regular structure for general k from which many properties of $\Delta_{k-1}T(n)$ and hence $T(n)$ can readily be deduced. For example, if Note 2.1 is generally true for $k = 7$, then $\Delta_6 T(n)$ also takes on only the values 0 and 6. We may thus conclude that (unlike many meta-Fibonacci sequences) $T(n)$ is well defined for all n.

We now show that this is true for general k.

PROPOSITION 2.2. *The following differences are all either* 0 *or* $k-1$*:* $\Delta_{k-1}T(n)$ *for* $n \geq k$, $\Delta_{k-1}R(n)$ *for* $n > k$, *and* $\Delta_{k-1}U(n)$ *for* $n > k+a$. *As a result, the*

[1] $\Delta_2 U_{0,3}(n)$ is what Allenby and Smith [1] call a pairing, though they neither explicitly define the sequence $\{\Delta_2 U_{0,3}(n)\}$ nor make it the focus of their analysis as we do here.

TABLE 2.3
$R_{0,7}(n)$.

	n							n					
	1	2	3	4	5	6		1	2	3	4	5	6
$R(n+0)$	-	1	2	3	4	5	$R(n+60)$	12	13	14	15	10	11
$R(n+6)$	6	7	2	3	4	5	$R(n+66)$	12	13	14	15	10	17
$R(n+12)$	6	7	8	3	4	5	$R(n+72)$	12	13	14	15	10	17
$R(n+18)$	6	7	8	3	10	5	$R(n+78)$	12	19	14	15	10	17
$R(n+24)$	6	7	8	3	10	5	$R(n+84)$	12	19	14	21	10	17
$R(n+30)$	12	7	8	3	10	5	$R(n+90)$	12	19	14	21	10	23
$R(n+36)$	12	7	14	9	10	5	$R(n+96)$	12	19	14	21	10	23
$R(n+42)$	12	7	14	9	10	11	$R(n+102)$	12	25	14	21	10	23
$R(n+48)$	12	7	14	9	10	11	$R(n+108)$	12	25	14	27	10	23
$R(n+54)$	12	13	14	9	10	11	$R(n+114)$	12	25	14	27	10	29

following sequence values are all (well) defined: $T(n)$ *for* $n > 0$, $R(n)$ *for* $n > 1$, *and* $U(n)$ *for* $n > a + 1$.

Proof. We begin an inductive proof by verifying that the proposition holds for $n \le 2k + a$.

$T(n) = 1$ for $1 \le n \le k + a$ by (2.1). It is easy to compute using (1.6) and (2.1) that $T(n) = k$ for $k + a < n \le 2k + a$. $\Delta_{k-1}T(n)$ is thus either 0 or $k - 1$ for $k \le n \le 2k + a$.

$R(n) := n - a - T(n-1)$ is $n - a - 1$ for $2 \le n \le k + a + 1$ and is $n - a - k > 0$ for $k + a + 1 < n \le 2k + a + 1$. $\Delta_{k-1}R(n) = k - 1 - \Delta_{k-1}T(n-1)$ is either 0 or $k - 1$ for $k + 1 \le n \le 2k + a + 1$.

$U(n) := T(R(n))$. When $k + a + 1 \le n \le 2k + a$, we have $1 \le R(n) \le k$. When $1 \le k \le n$, we have $T(n) = 1$. So when $k + a + 1 \le n \le 2k + a$, we have $U(n) = T(R(n)) = 1$. Thus, $\Delta_{k-1}U(n) = 0$ is for $k + a + 1 \le n \le 2k + a$.

Now let $n > 2k + a$ and proceed assuming that the proposition holds for all lesser n.

$R(n) := n - a - T(n-1)$ is defined.

$\Delta_{k-1}R(n) = k - 1 - \Delta_{k-1}T(n-1)$ is either $k - 1$ or 0, since $\Delta_{k-1}T(n-1)$ is either 0 or $k - 1$.

$U(n) := T(R(n))$. To confirm that $U(n)$ is well defined, we need to show that $R(n)$ is well defined (which we just did) and that T is well defined at $R(n)$. That is, we require that $0 < R(n) < n$. Let $m \in [k+a+2, 2k+a+1]$ with $m \equiv n \pmod{k-1}$. Then $R(n) = R(m)+\Delta_{k-1}R\big(m+k-1\big)+\Delta_{k-1}R\big(m+2(k-1)\big)+\cdots+\Delta_{k-1}R\big(n\big)$. The first summand is positive and the rest are all either 0 or $k - 1$, so $R(n)$ is positive. Likewise, $T(n) > 0$, so $R(n) := n - a - T(n-1) < n$. Therefore, we can apply induction to find that $U(n)$ is defined.

$\Delta_{k-1}U(n) = T(R(n))-T(R(n-k+1))$. We just showed that $R(n)-R(n-k+1) = \Delta_{k-1}R(n)$ is either 0 or $k-1$. If this difference is 0, then so is $\Delta_{k-1}U(n)$. If it is $k-1$, then $\Delta_{k-1}U(n) = (\Delta_{k-1}T)(R(n))$, which is 0 or $k - 1$ by the induction assumption.

From (2.2), we have $T(n) = T(n-1) + U(n) - U(n-k)$, so $T(n)$ is well defined.

$\Delta_{k-1}T(n) = \sum_{i=0}^{k-1} \Delta_{k-1}U(n-i)$ from (2.2) and the linearity of the difference operator. If all the summands are zero, then so is $\Delta_{k-1}T(n)$ and we are done. If not, let j be the largest integer in $[n - k + 1, n]$ for which $k - 1 = \Delta_{k-1}U(j) = T(R(j)) - T(R(j - k + 1))$. Then $\Delta_{k-1}R(j) = k - 1$ and $0 = k - 1 - \Delta_{k-1}R(j) = \Delta_{k-1}T(j-1) = \sum_{i=0}^{k-1} \Delta_{k-1}U(j - 1 - i)$. Therefore at most one of $\Delta_{k-1}U(n-k+1)$, $\ldots$, $\Delta_{k-1}U(n)$ can have the value $k - 1$ while the others must be 0, so their sum $\Delta_{k-1}T(n)$ is either 0 or $k - 1$. ∎

Note that other choices of initial values for T will prevent the initial conditions of the induction from being satisfied and that in such cases $\Delta_{k-1}U(n)$ and $\Delta_{k-1}T(n)$ may belong to a larger finite set (see Figure 1.4, or the case $a = 0$, $k = 3$ with initial values $\{0, 2, 1\}$) or take on an infinite number of different values (see Figure 1.7).

COROLLARY 2.3. *For $n > 2k+a+1$, if $\Delta_{k-1}T(n) = 0$, then $\Delta_{k-1}U(n-k+1) = \Delta_{k-1}U(n-k+2) = \cdots = \Delta_{k-1}U(n) = 0$.*

Proof. If $0 = \Delta_{k-1}T(n) = \sum_{j=0}^{k-1} \Delta_{k-1}U(n-j)$ and each summand is either 0 or $k-1$, each summand must be zero. □

COROLLARY 2.4. *For each j in $[1, k]$, the subsequence $\{T((k-1)i+j)\}_{i=0}^{\infty}$ begins with 1 and increases monotonically in increments of 0 or $k-1$.*

Proof. The result follows immediately from the initial conditions for T and what we have shown about the differences $\Delta_{k-1}T(n)$. □

COROLLARY 2.5. *In the sequence $\Delta_{k-1}U(n)$, $n > a+k$, there are at least k zeros between each pair of nonzero values.*

Proof. We showed in the proof of Proposition 2.2 that $\Delta_{k-1}U(n) = 0$ for $k+a < n \leq 2k+a$ and if for some $j > 2k+a$ the jth member of $\Delta_{k-1}U$ is nonzero, then it is preceded by at least k zeros. □

The preceding corollary describes the fine structure of $\Delta_{k-1}U(n)$ and helps explain why that sequence attracts our interest. We conclude this section by formally defining the blocks that make up the structure of the $\Delta_{k-1}U(n)$ sequence.

DEFINITION 2.6. *A* p-block *is a consecutive subsequence of length p in $\{\Delta_{k-1}U(n)\}$, consisting of an initial nonzero value together with all of the zeros which follow it.*

Example 2.7. The values of $\Delta_6 U(n)$ listed in Note 2.1 constitute seven zeros, followed by two 8-blocks, one 9-block, and seven more 8-blocks.

3. The structure of a three-term recursion. We move now from general observations about properties of our family of difference functions to the specific case $k = 3$; observe that this is a three-term generalization of Conolly's sequence (1.2). This case illustrates the increased structural complexity that occurs for $k > 2$. It also demonstrates concretely the behavior typical of odd k, to be elaborated in section 5.[2]

In (1.6), set $a = 0$, $k = 3$ and assume the initial conditions $T_{0,3}(n) = 1$ for $n = 1, 2, 3$. For the remainder of this section and in the next section, we refer to this sequence as $T(n)$, to $R_{0,3}(n)$ as $R(n)$, and to $U_{0,3}(n)$ as $U(n)$:

$$\begin{aligned} T(n) &:= U(n) + U(n-1) + U(n-2), \quad n > 3, \\ U(n) &:= T(R(n)), \quad n > 1, \\ R(n) &:= n - T(n-1), \quad n > 1. \end{aligned} \tag{3.1}$$

Table 3.1 lists values of $T(n)$, $U(n)$, and $R(n)$ for small n.

Note 3.1. For future reference, we list the values of $\{\Delta_2 U(n)\}_{n=4}^{97}$ = {0,0,0, 2,0,0,0,0, 2,0,0,0, 2,0,0,0, 2,0,0,0,0, 2,0,0,0,0, 2,0,0,0,0, 2,0,0,0, 2,0,0,0, 2,0,0,0, 2,0,0,0, 2,0,0,0,0, 2,0,0,0,0, 2,0,0,0,0, 2,0,0,0,0, 2,0,0,0, 2,0,0,0,0, 2,0,0,0, 2,0,0,0,0, 2,0,0,0,0}.

We completely describe the behavior of T, as well as the related sequences U and R. As is the case with many other meta-Fibonacci sequences, the T sequence can be naturally decomposed into strings of consecutive terms called *generations*. The generational structure is clearly visible in Figure 1.6, where the end of each

[2] Using very different methods, Allenby and Smith [1] analyze this recursion together with several other, less complicated, meta-Fibonacci recursion variants. In this and the next section, we extend considerably the results proven in [1] and settle all the conjectures stated therein.

TABLE 3.1
$T(n)$, $U(n)$, $R(n)$ *for* $a = 0$, $k = 3$.

n	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39
$T(n)$	1	1	3	5	7	7	9	9	11	11	13	13	15	17	17	19	19	21	23	25
$T(n+1)$	1	3	3	5	5	7	9	11	13	15	17	17	19	19	21	23	23	25	25	27
$U(n)$	–	1	1	3	3	3	3	3	3	3	3	3	5	5	5	5	5	7	7	9
$U(n+1)$	1	1	1	1	1	3	3	5	5	7	7	7	7	7	9	9	9	9	9	9
$R(n)$	–	2	2	4	4	6	6	6	6	6	6	6	8	8	10	10	10	12	12	14
$R(n+1)$	1	3	3	3	3	5	5	7	7	9	9	11	11	11	13	13	15	15	15	15

generation is marked by several points of intersection of the two curves. The lengths of the generations increase geometrically, and properties of the sequence recur from generation to generation giving a structure which might be described as *logarithmic-periodic*.

To describe completely the behavior of $T(n)$ it is sufficient to characterize fully the behavior of $\Delta_2 U(n)$, since $T(n)$ can be reconstructed as a double sum of $\Delta_2 U(n)$. Indeed, the generational structure of $T(n)$ is mirrored in a surprising way in a corresponding generational structure in the $\Delta_2 U(n)$ sequence. Even further, we will show that each of these corresponding generations in $\Delta_2 U$ is in turn made up of substrings we call *lines* and subsubstrings we call *feet*. That is, we will show how $\Delta_2 U(n)$ consists of feet, concatenations of which form lines, and how consecutive lines complete the generation. We also show that the feet in each generation in $\Delta_2 U(n)$ are determined by (and in fact are in the one-to-one correspondence shown in Figure 3.1 with) the values of $\Delta_2 T$ in the immediately preceding generation. In this way, the preceding generation of T determines new feet, which make up new lines; these new lines make up a new generation in $\Delta_2 U$, which further determines the new generation of T.

From Note 3.1, we empirically observe (and will show below in Proposition 3.3) that from $n = 7$ onward $\Delta_2 U(n)$ appears to consist only of 4-blocks and 5-blocks. We will see that the interesting values of n are those which start a 4-block, start a 5-block, or end a 5-block, and that it is logical then to look at $\Delta_2 U$ as consisting mostly of subsequences $\{2, 0, 0, 0\}$ with the occasional extra $\{0\}$.

Perhaps not surprisingly, the ancient Greeks [2, 11] had words for such patterns (feet) of one stressed beat followed by three unstressed beats (a first paeon), for a sequence of such patterns (a line), and for an extra unstressed beat at the end of a sequence (a hypercatalectic). While Greek prosodists were concerned with long and short syllables in poetry and we are dealing with 2's and 0's in an iterated recursion, the analogy is precise, and so we borrow their terminology to define our first type of subsequence.

DEFINITION 3.2 (foot). *A paeon is a sequence* {2,0,0,0} *of consecutive values of* $\Delta_2 U(n)$. *A hypercatalectic is a singleton sequence* {0}, *immediately preceded in* $\{\Delta_2 U(n)\}$ *by a paeon. A foot is either a paeon or a hypercatalectic. For convenience, we will write* {P} *interchangeably with* $\{2, 0, 0, 0\}$ *and likewise* {H} *with* {0} *when listing values of* $\Delta_2 U$. *We also define* $\varphi(n)$ *as the symbol* P *if* $\Delta_2 U(n) = 2$ *begins a paeon and* H *if* $\Delta_2 U(n) = 0$ *is a hypercatalectic, and we leave it undefined otherwise.*

So $\varphi(7) = \varphi(12) = \varphi(16) = \varphi(20) = P$, $\varphi(11) = \varphi(24) = H$ and $\varphi(n)$ is not defined for other $7 \le n \le 24$.

PROPOSITION 3.3. $\{\Delta_2 U(n)\}_{n=7}^{\infty}$ *consists only of feet.*

Proof. By Definition 3.2, this is equivalent to saying that there are no p-blocks of length $p > 5$ in $\Delta_2 U(n)$ or that if $\Delta_2 U(n) = 2$ and $\Delta_2 U(n + 4) = 0$, then $\Delta_2 U(n + 5) = 2$.

We proceed to prove the latter by induction. The enumeration in Note 3.1 shows that the proposition holds for $n = 7$. In particular, $\Delta_2 U(7) = \Delta_2 U(12) = 2$ and $\Delta_2 U(11) = 0$. Assume that the proposition is true for all n less than some $N > 7$, that $\Delta_2 U(N) = 2$, and that $\Delta_2 U(N+4) = 0$. Corollary 2.5 and induction tell us that for some nonnegative q, $\Delta_2 U(N) = \Delta_2 U(N-4) = \cdots = \Delta_2 U(N-4q) = \Delta_2 U(N-4q-5) = 2$ and $\Delta_2 U(i) = 0$ for all other i in the interval $[N-4q-5, N+4]$.

We can compute differences of T by expanding into these known values of $\Delta_2 U$: $T(N+4) - T(N+2) = \Delta_2 T(N+4) = \sum_{i=N+2}^{N+4} \Delta_2 U(i) = 0$, $T(N-4q-2) - T(N-4q-4) = \Delta_2 T(N-4q-2) = \sum_{i=N-4q-4}^{N-4q-2} \Delta_2 U(i) = 0$, and $T(N+2) - T(N-4q-2) = \sum_{i=1}^{2q+2} \Delta_2 T(N-2i+4) = \sum_{i=1}^{2q+2} \sum_{j=0}^{2} \Delta_2 U(N-2i-j+4) = 4q+4$.

So $T(N+4) = T(N-4q-4) + 4q + 4$, $R(N+5) = N + 5 - T(N+4) = 2 + N + 3 - T(N+2) = R(N+3) + 2$, $R(N+5) = N - 4q + 1 - T(N-4q-4) < N$, and $R(N+5) = 2 + N - 4q - 1 - T(N-4q-2) = R(N-4q-1) + 2$.

Let $y := R(N+5)$. Then $\Delta_2 T(y-2) = \Delta_2 U(N-4q-1) = 0$. By Corollary 2.3, $\Delta_2 U(y-2) = \Delta_2 U(y-3) = \Delta_2 U(y-4) = 0$. By induction, $\Delta_2 U(y-1)$ and $\Delta_2 U(y)$ cannot both be zero, or else $\Delta_2 U$ would have five consecutive zeros somewhere before N, at $y-4$, $y-3$, $y-2$, $y-1$, and y. Therefore, $\Delta_2 U(y) + \Delta_2 U(y-1) + \Delta_2 U(y-2) = 2 = \Delta_2 T(y) = U(N+5) - T(R(N+5)-2) = U(N+5) - U(N+3) = \Delta_2 U(N+5)$ as required. □

Note that as with Proposition 2.2, other choices of initial values for T (such as $\{T(n)\}_{n=1}^{3} := \{1, 2, 1\}$) will render false the initial conditions for the induction in the proof of Proposition 3.3. In such cases, the spacing between nonzero members of $\{\Delta_2 U(n)\}$ can grow without bound.

We now use our results so far to create Table 3.2, a tabulation of six of our functions over the course of a paeon beginning at n_0, followed possibly by a hypercatalectic. We express the values of the functions in terms of the boxed parameters t_0, t_1, r_0, r_1, and d_4. (The paeon is followed by a hypercatalectic iff $d_4 = 0$, that is, $\varphi(n_0 + 4) = H$ iff $d_4 = 0$.)

This description of the intricate local relationship among the T, R, and U sequences tells us how to easily compute the parameter values in one foot of $\Delta_2 U$ from the parameter values in the preceding foot. Further, it will be fundamental to an understanding of how the occurrence of feet in $\Delta_2 U$ relates to much earlier values of

TABLE 3.2
Function values over a paeon.

n	$T(n)$	$\Delta_2 T(n)$	$R(n)$	$\Delta_2 R(n)$	$\Delta_2 U(n)$	$\varphi(n)$
$n_0 - 3$	$t_1 - 2$	2	?	0	0	–
$n_0 - 2$	$t_0 - 2$	?	$r_0 - 2$	0	0	–
$n_0 - 1$	$t_1 - 2$	0	r_1	?	0	?
$\boxed{n_0}$	$\boxed{t_0}$	2	$\boxed{r_0}$	2	2	P
$n_0 + 1$	$\boxed{t_1}$	2	$\boxed{r_1}$	0	0	–
$n_0 + 2$	$t_0 + 2$	2	r_0	0	0	–
$n_0 + 3$	t_1	0	r_1	0	0	–
$n_0 + 4$	$t_0 + d_4 + 2$	d_4	$r_0 + 2$	2	$\boxed{d_4}$	P or H
$n_0 + 5$	$t_1 + 2$	2	$r_1 + 2 - d_4$	$2 - d_4$	$2 - d_4$	– or P
$n_0 + 6$	$t_0 + d_4 + 4$	2	$r_0 + 2$	0	0	–
$n_0 + 7$	$t_1 - d_4 + 4$	$2 - d_4$	$r_1 + 2 - d_4$	0	0	–

$\Delta_2 T$. This then constitutes the first of our three structural theorems.

THEOREM 3.4 (foot pattern theorem). *Suppose the parameters n_0, t_0, t_1, r_0, r_1, and d_4 satisfy all of the following conditions: $n_0 \geq 7$, $T(n_0) = t_0$, $T(n_0+1) = t_1$, $R(n_0) = r_0$, $R(n_0+1) = r_1$, $\Delta_2 U(n_0) = 2$, and $\Delta_2 U(n_0+4) = d_4$. Then $T(n)$, $\Delta_2 T(n)$, $R(n)$, $\Delta_2 R(n)$, $\Delta_2 U(n)$, and $\varphi(n)$ have the values shown in Table* 3.2.

Proof. Use Proposition 3.3 and Corollary 2.5 to calculate $\Delta_2 U(n)$. Apply the Δ_2 operator to (2.2) to calculate $\Delta_2 T(n)$ (and Proposition 3.3 for $\Delta_2 T(n_0 - 3)$). Use $\Delta_2 R(n) = 2 - \Delta_2 T(n-1)$ to calculate $\Delta_2 R(n)$. Use the definition of Δ_2 to calculate $R(n)$ and $T(n)$. □

Since $\Delta_2 U(n) := T(R(n)) - T(R(n-2))$, the preceding theorem shows that the reason $\Delta_2 U(n_0+1) = \Delta_2 U(n_0+2) = \Delta_2 U(n_0+3) = 0$ is not because the values of $T(n)$ at two distinct arguments of T happen to be equal, but rather because T is being evaluated at two equal values of $R(n)$, respectively, r_1, r_0, and r_1. For this reason, these three zero terms immediately following any two in $\{\Delta_2 U(n)\}$ are rather uninteresting. Note that these three zeros in each paeon of $\Delta_2 U(n)$ correspond precisely to those terms for which $\Delta_2 R(n) = 0$, or equivalently the ones for which $\varphi(n)$ is undefined.

Conversely when $\Delta_2 R(n) = 2$, that is, when $\varphi(n)$ is defined,

$$\Delta_2 U(n_0) = T(r_0) - T(r_0 - 2) = \Delta_2 T(r_0)$$
$$\Delta_2 U(n_0+4) = T(r_0+2) - T(r_0) = \Delta_2 T(r_0+2),$$

and so in both cases we obtain the key correspondence,

$$\Delta_2 U(n) = (\Delta_2 T)(R(n)) = T\big(R(n)\big) - T\big(R(n)-2\big). \tag{3.2}$$

We could also give an alternate definition: $\varphi(n) = P$ if $\Delta_2 U(n) + \Delta_2 R(n) = 4$, and $\varphi(n) = H$ if $\Delta_2 U(n) + \Delta_2 R(n) = 2$. Or also equivalently, $\varphi(n) = P$ if $\Delta_2 T(n) = 2$ and $\Delta_2 T(n-1) = 0$, and $\varphi(n) = H$ if $\Delta_2 T(n) = 0$ and $\Delta_2 T(n-1) = 0$.

Note 3.5. $\Delta_2 T(n)$, $\Delta_2 U(n)$, and $\Delta_2 R(n)$ exhibit a *recursive symmetry*, just as do many other meta-Fibonacci sequences in the literature (e.g., in [12, 10]). As a result, there is a natural partition of their domain into *generations* (finite, consecutive strings of increasing length) such that certain function values within one generation can be expressed elegantly in terms of function values in preceding generations. We show in Theorem 3.14 that (3.2) will express $\Delta_2 U$ in one generation in terms of $\Delta_2 T$ in the preceding generation.

DEFINITION 3.6 (generation). *For any $g > 0$, let $m_g := \frac{1}{2}(3^{g+1}+5) = 3 + \sum_{i=0}^{g} 3^i$ and call the interval $[m_g, m_{g+1} - 1]$ the* gth generation*, written as* $\mathrm{gen}(g)$*. We partition the gth generation into two nonconsecutive subsequences: the* gth even semigeneration $\mathrm{sg}_0(g) := \{n \in \mathrm{gen}(g) \mid n \equiv m_g \pmod 2\}$ *and the* gth odd semigeneration $\mathrm{sg}_1(g) := \{n \in \mathrm{gen}(g) \mid n \not\equiv m_g \pmod 2\}$*. For any sequence $s(n)$, we will refer to the subsequence $\{s(n) \mid n \in \mathrm{gen}(g), s(n)$ defined$\}$ as the* gth generation of s *and similarly for semigenerations. An* even (odd) *foot is one that starts in an even (odd) semigeneration.*

Note that because the length 3^{g+1} of $\mathrm{gen}(g)$ is odd, m_g and m_{g+1} always have opposite parity. The foot that follows a paeon is always of the same parity, because the paeon has even length; while the foot that follows a hypercatalectic is always of opposite parity, because the hypercatalectic has odd length.

We list for future reference the first five values of m_g: 7, 16, 43, 124, 367.

TABLE 3.3
Function values over a line.

$n - n_0$	$T(n)$	$\Delta_2 T(n)$	$R(n)$	$\Delta_2 R(n)$	$\Delta_2 U(n)$	$\varphi(n)$
-2	$t_0 - 2$	?	$r_0 - 2$	0	0	–
-1	$t_1 - 2$	0	r_1	?	0	?
0	t_0	2	r_0	2	2	P
1	t_1	2	r_1	0	0	–
⋮	⋮	⋮	⋮	⋮	⋮	–
$4q-4$	$t_0 + 4q - 4$	2	$r_0 + 2q - 2$	2	2	P
$4q-3$	$t_1 + 2q - 2$	2	r_1	0	0	–
$4q-2$	$t_0 + 4q - 2$	2	$r_0 + 2q - 2$	0	0	–
$4q-1$	$t_1 + 2q - 2$	0	r_1	0	0	–
$4q$	$t_0 + 4q - 2$	0	$r_0 + 2q$	2	0	H
$4q+1$	$t_1 + 2q$	2	$r_1 + 2$	2	2	P
$4q+2$	$t_0 + 4q$	2	$r_0 + 2q$	0	0	–

Now we refine further our description of the structure of $\Delta_2 U(n)$ within a generation. Once again we borrow from the terminology of Greek poetry, here to name sequences of feet with no internal hypercatalectics.

DEFINITION 3.7 (line). *A* line *is a maximal subsequence of consecutive paeons and hypercatalectics in a generation, none of whose paeons is preceded by one of its hypercatalectics. We write a line as an overlined sequence of P's and H: e.g., $\overline{PPP\ldots PH}$. An* even (odd) *line is one that starts in an even (odd) semigeneration, that is, one whose feet are all even (odd).*

Proposition 3.3 tells us that a line is either a nonempty sequence of paeons followed by a hypercatalectic or a sequence of paeons at the end of a generation.

Example 3.8. The first generation of $\Delta_2 U$ begins at $\Delta_2 U(7)$, ends at $\Delta_2 U(15)$, and consists of the 9 numbers $\{\Delta_2 U(n)\}_{n\in \mathrm{gen}(1)} = \{2,0,0,0,\ 0,\ 2,0,0,0\}$, the 3 feet $\{\varphi(n)\}_{n\in \mathrm{gen}(1)\cap \mathrm{Dom}\,\varphi} = \{P, H, P\}$ that form the first generation of φ, or the 2 lines $\{\overline{PH},\ \overline{P}\}$. The second generation of $\Delta_2 U$ continues on from $\Delta_2 U(16)$ and consists of the 27 numbers $\{\Delta_2 U(n)\}_{n\in \mathrm{gen}(2)} = \{2,0,0,0,\ 2,0,0,0,\ 0,\ 2,0,0,0,\ 0,\ 2,0,0,0,\ 0,\ 2,0,0,0,\ 2,0,0,0\}$ or the 4 lines $\{\overline{PPH},\ \overline{PH},\ \overline{PH},\ \overline{PP}\}$. The third generation of $\Delta_2 U$ consists of the 10 lines $\{\overline{PPPPH},\ \overline{PH},\ \overline{PH},\ \overline{PH},\ \overline{PPH},\ \overline{PPH},\ \overline{PH},\ \overline{PH},\ \overline{PH},\ \overline{PPPP}\}$.

As was the case with feet, there is an intricate relationship among the values of the T, R, and U sequences on the paeons and hypercatalectic that make up a line. This is captured in the second of our three structural theorems.

THEOREM 3.9 (line pattern theorem). *Suppose $T(n_0) = t_0$, $T(n_0+1) = t_1$, $R(n_0) = r_0$, $R(n_0+1) = r_1$, and $\Delta_2 U(n_0)$ is the beginning of a line of q paeons and a hypercatalectic, ending therefore at $\Delta_2 U(n_0 + 4q)$. Then $T(n)$, $\Delta_2 T(n)$, $R(n)$, $\Delta_2 R(n)$, $\Delta_2 U(n)$, and $\varphi(n)$ have the following values shown in Table* 3.3.

Proof. By Definition 3.7, $\Delta_2 U(n_0) = \Delta_2 U(n_0+4) = \cdots = \Delta_2 U(n_0+4q-4) = \Delta_2 U(n_0+4q+1) = 2$ and all other intervening values of $\Delta_2 U$ are zero. The first two rows of the table depend on whether or not the line begins a generation (and hence whether or not the preceding foot is a paeon or a hypercatalectic), and its values can be computed using Foot Pattern Theorem 3.4. The rest of the result follows from q applications of Foot Pattern Theorem 3.4, setting n_0 successively to the start of each of these paeons. ∎

We observe that the feet within a line all have the same parity, and the lines within a generation alternate in parity. Further, the lines within a generation are essentially symmetric about the middle of the generation; for example, in the third generation, the fifth and sixth are identical, as are the fourth and seventh, third and eighth, and second and ninth. The first and tenth lines have identical paeons, but the first line ends in a hypercatalectic while the tenth does not. In fact, it is precisely this symmetry that led us to the specification of m_g as the start of the gth generation.

We will show in Theorem 3.15, to which we are now building, that a generation consists entirely of lines, with no excess terms. Further, the proof of that theorem will demonstrate how the values of $\Delta_2 U(n)$ in the gth generation are determined by the values of $\Delta_2 T(n)$ in the $(g-1)$st generation. This provides an understanding of how successive generations interrelate. In order to get there, we require some additional background.

We mentioned in Note 3.5 that the gth generation of $\Delta_2 U$ would be expressed in terms of the $(g-1)$st generation of $\Delta_2 T$. It turns out that the natural restriction of the function R to those n in the gth generation that start a foot is one to one onto the $(g-1)$st generation; for each such n, the value of $R(n)$ is the *unique* r in the $(g-1)$st generation for which $\Delta_2 T(r) = \Delta_2 U(n)$. We can then use the inverse of this map to construct a beautiful correspondence from the $(g-1)$st generation to the gth, as we will soon show.

Example 3.10. We illustrate this inverse map of R with a simple example. Generation 2 begins at $n = 16$ and ends at $n = 42$. We know from Note 3.1 and Theorems 3.4 and 3.9 that the feet of $\Delta_2 U$ in generation 2 begin at $n = 16$, 20, 24, 25, 29, 30, 34, 35, and 39. From Table 3.1 we have $R(16) = 7$, $R(20) = 9$, $R(24) = 11$, $R(25) = 8$, $R(29) = 10$, $R(30) = 13$, $R(34) = 15$, $R(35) = 12$, and $R(39) = 14$. Observe that all of these values of $R(n)$ are distinct and that they include all of the integers from 7 to 15, which is precisely the first generation. So the inverse of R, evaluated on the first generation, gives the beginnings of all of the feet in the second generation of $\Delta_2 U$. In this way, the successive generations of T are interrelated.

We now show that the invertibility property holds.

PROPOSITION 3.11. *Let E be the set of even numbers and $D = \operatorname{Dom}\varphi$. Then $R|_{D\cap E}$ begins with 5 and increases (strictly) only by 2, while $R|_{D\setminus E}$ begins with 4 and also increases only by 2.*

Proof. By (3.1) and Note 3.1, $R(7) = 4$, $R(12) = 5$. The result follows from the values of $R(n_0 + 4)$ and $R(n_0 + 5)$ given in Foot Pattern Theorem 3.4. □

We do not know yet that $\operatorname{Ran}(R|_{\operatorname{Dom}\varphi})$ is all of $[4, \infty)$. This will be established after we show in Theorem 3.15 that each generation of $\Delta_2 U(n)$ has at least one hypercatalectic, so $\operatorname{Ran}(R|_{\operatorname{Dom}\varphi})$ flips parity infinitely often. We now name the inverse of this restriction of R.

DEFINITION 3.12. *For $n \in \operatorname{Dom}\varphi$, let $f(R(n)) = n$. That is, if $\Delta_2 R(n) = 2$, then $f(R(n)) = n$.*

In Example 3.10, we had $f(7) = 16$, $f(8) = 25$, $f(9) = 20$, and so on. Also, since f is bijective by construction, $R(f(r)) = r$ when $f(r)$ is defined. Observe that $R(f(7)) = R(16) = 7$ and $f(7) = 16$ and $R(7) = 4$ so $f(7) \equiv R(7) \not\equiv 7 \pmod 2$.

COROLLARY 3.13. *When $f(r)$ is defined, $f(r) \equiv R(r) \not\equiv r \pmod 2$. If $r \in \operatorname{Ran} R$, then $f(n)$ is defined for all $n \equiv r \pmod 2$ in $[4, r]$. $f(r)$ is the smallest member of the set $R^{-1}(r)$.*

Proof. By Proposition 3.11, $R(n)$ is even when n is odd, and vice versa. Since f

is the inverse of a restriction of R, $f(r)$ is odd when r is even, and vice versa.

Also by Proposition 3.11, the range of $R|_{D\cap E}$ is a consecutive sequence of odd numbers starting at 5 with no gaps; so if an odd $r \in \operatorname{Ran} R$, then $\operatorname{Ran} R$ also includes all lesser odd numbers down to 5; and similarly for even r.

By Theorem 3.4, $\Delta_2 R(n)$ is 2 at the beginning of a foot in $\Delta_2 U$ and 0 elsewhere. Therefore, among all the n for which $R(n)$ is equal to a particular r, it is the smallest one which belongs to $\operatorname{Dom}\varphi$. □

We are now close to demonstrating that our partition of the domains of our functions into generations is a natural one, by showing that the values of functions on one generation depend solely on the values of functions in the immediately preceding generation. As we illustrated in Example 3.10, if n belongs to the gth generation, then $R(n)$ belongs to the $(g-1)$st generation. Therefore, if r belongs to the gth generation, then $f(r)$ belongs to the $(g+1)$st generation.

It turns out though that f does much more than simply map one generation into the next. If for some r in the gth generation $\Delta_2 T(r) = 2$, then $f(r)$ is the beginning of a paeon in the $(g+1)$st generation of $\Delta_2 U$: that is, $\varphi(f(r)) = P$. Conversely if $\Delta_2 T(r) = 0$, then $f(r)$ is the beginning of a hypercatalectic of $\Delta_2 U$ and $\varphi(f(r)) = H$. Thus f establishes a correspondence between each member of one generation and each *foot* of $\Delta_2 U$ in the next generation, incidentally accounting for how the successive generations grow.

Figure 3.1 displays this correspondence for the first two generations and elaborates on the consequences of the parity principle of Corollary 3.13. As shown in Example 3.8, the first generation of $\Delta_2 U$ consists of the feet $\{P, H, P\}$. According to Foot Pattern Theorem 3.4, it follows that the first generation of $\Delta_2 T$ must be $\{2, 2, 2, 0; 0; 2, 2, 2, 0\}$, where semicolons indicate the ends of the feet in $\Delta_2 U$. In Figure 3.1 we show the four entries of $\Delta_2 T$ that correspond to each paeon of $\Delta_2 U$ enclosed in a parallelogram, and the single (zero) entry of $\Delta_2 T$ that corresponds to a hypercatalectic of $\Delta_2 U$ enclosed in a triangle.

The action of f on members of the first generation determines the structure of the second generation. If we have r such that $\Delta_2 T(r) = 2$, then $f(r)$ marks the beginning of a paeon in $\Delta_2 U$: that is, $\varphi(f(r)) = P$, as indicated by the dashed lines in the figure. Likewise dashed lines connect the points at which $\Delta_2 T(r) = 0$ to the points where $\varphi(f(r)) = H$.

Successive integers in the first generation alternate in parity. By Corollary 3.13, their images under f must also alternate in parity. However, since all of the feet in a line must have the same parity, the image under f of the integers corresponding to a paeon in $\Delta_2 U$ lie in two separate lines of opposite parity. The lines in the second generation are enclosed in boxes, and arrows indicate the intertwined order in which they are to be concatenated to form the second generation.

Observe that we can now easily count the number of feet and lines in each successive generation. Since each foot in the first generation of $\Delta_2 U$ ends at a point where $\Delta_2 T$ is zero, it corresponds to a line-ending hypercatalectic in the second generation of $\Delta_2 U$. So, the number of hypercatalectics in one generation is the same as the number of feet in the preceding generation. And hence, the number of lines in one generation is one more (taking into account the last line that lacks a hypercatalectic) than the number of feet in the preceding generation.

The following theorem proves the correspondence between 0's and 2's and H's and P's and will be the foundation for much of the rest of the paper.

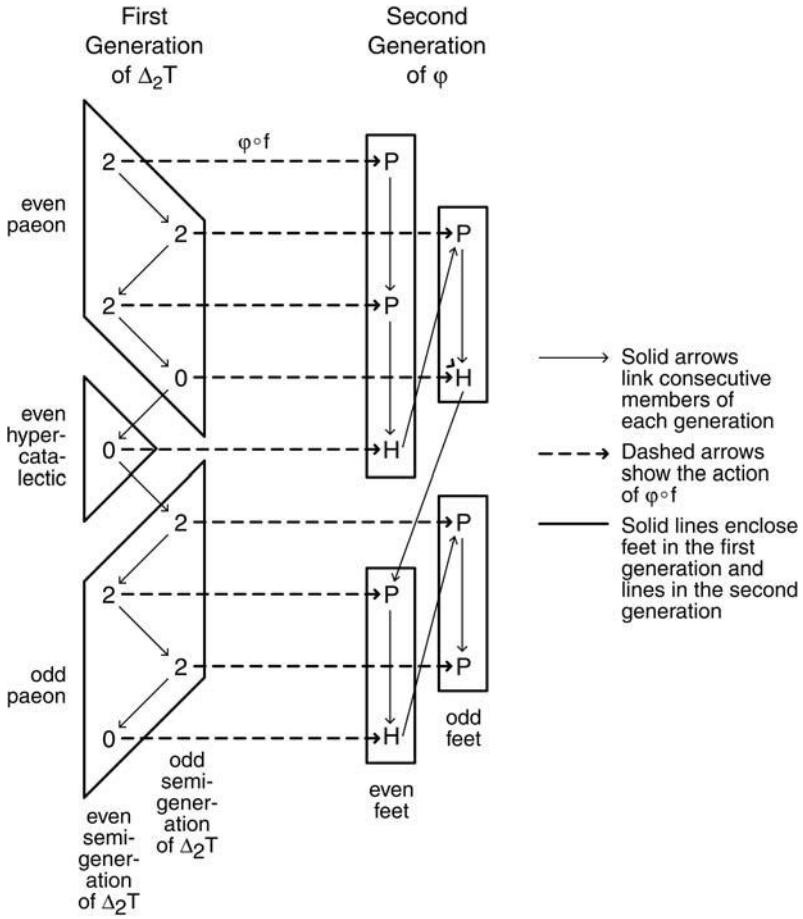


FIG. 3.1. *How successive generations correspond.*

THEOREM 3.14 (generational correspondence theorem). *The diagram*

$$\begin{array}{ccc} \operatorname{Dom} f & \underset{R}{\overset{f}{\rightleftarrows}} & \operatorname{Ran} f \\ {\scriptstyle \Delta_2 T}\downarrow & & \downarrow{\scriptstyle \varphi} \\ \{0,2\} & \underset{2\mapsto P}{\overset{0\mapsto H}{\longrightarrow}} & \{H,P\} \end{array}$$

commutes. That is, for $4 \le r \in \operatorname{Dom} f$, $\varphi(f(r)) = P$ *iff* $\Delta_2 T(r) = 2$.

Proof. If $\varphi(f(r)) = P$, then by (3.2), $2 = \Delta_2 U(f(r)) = \Delta_2 T(R(f(r))) = \Delta_2 T(r)$. If $\varphi(f(r)) = H$, then $0 = \Delta_2 U(f(r)) = \Delta_2 T(R(f(r))) = \Delta_2 T(r)$. □

We are now ready to present the last of our three structural theorems, which will rely on Theorem 3.14 to express each generation in terms of its predecessor. This will prove that the correspondence illustrated in Figure 3.1 does map each generation exactly onto the next.

THEOREM 3.15 (generation pattern theorem). *The* g*th generation of* $\Delta_2 U$ *consists entirely of* $3^{g-1}+1$ *lines. Its last line consists only of (odd) paeons, the last of which has Foot Pattern Theorem* 3.4 *parameters* $n_0 = m_{g+1} - 4$, $t_0 := T(n_0) = 3^{g+1}-2$, $t_1 := T(n_0+1) = 3^{g+1}$, $r_0 := R(n_0) = m_g - 2$, *and* $r_1 := R(n_0+1) = m_g - 1$.

TABLE 3.4
Function values near the end of a generation.

n	$T(n)$	$\Delta_2 T(n)$	$R(n)$	$\Delta_2 R(n)$	$\Delta_2 U(n)$	$\varphi(n)$
$m_g - 4$	$3^g - 2$	2	$m_{g-1} - 2$	2	2	P
$m_g - 3$	3^g	2	$m_{g-1} - 1$	0	0	–
$m_g - 2$	3^g	2	$m_{g-1} - 2$	0	0	–
$m_g - 1$	3^g	0	$m_{g-1} - 1$	0	0	–
.....						
m_g	$3^g + 2$	2	m_{g-1}	2	2	P
$m_g + 1$	$3^g + 2$	2	$m_{g-1} - 1$	0	0	–
$m_g + 2$	$3^g + 4$	2	m_{g-1}	0	0	–

Proof. The values of T and $\Delta_2 U$ necessary to verify the result for $g = 1$ and $g = 2$ have been listed above in Note 3.1. We proceed to larger g by induction, assuming that the results hold for all generations before the gth. In particular, applying Foot Pattern Theorem 3.4 to the last foot of the $(g-1)$st generation gives us the first four rows of the key function values shown in Table 3.4.

To obtain the last three of the rows, we need to show that the gth generation begins with a paeon and not a hypercatalectic. Observe that by induction the $(g-1)$st generation begins with a paeon, so the Foot Pattern Theorem 3.4 applied to $n_0 = m_{g-1} - 4$ gives us $T(m_{g-1}) = 3^{g-1} + 2$ and $\Delta_2 T(m_{g-1}) = 2$. Since $\Delta_2 U(m_g)$ begins a foot, (3.2) tells us that $\Delta_2 U(m_g) = \Delta_2 T(m_{g-1}) = 2$ and the start of the first foot in the gth generation of $\Delta_2 U$ corresponds with the first term in the $(g-1)$st generation of $\Delta_2 T$, as desired. Apply Theorem 3.4 to $n_0 = m_g$ to fill in the rest of the values.

Next, we show that f maps $\mathrm{gen}(g-1)$ *onto* $\mathrm{gen}(g) \cap \mathrm{Dom}\,\varphi$, as shown for $g = 2$ in Figure 3.1. By Corollary 3.13 and the fact that all generations have odd length, we will know then that f maps $\mathrm{sg}_i(g-1)$ onto $\mathrm{sg}_i(g) \cap \mathrm{Dom}\,\varphi$ for $i = 0, 1$.

We begin at $R(m_g) = m_{g-1}$, so $f(m_{g-1}) = m_g$, for which Theorem 3.14 with $r = m_{g-1}$ gives us a correspondence between $\Delta_2 T(m_{g-1}) = 2$ and $\varphi(f(m_{g-1})) = P$. We claim that if we continue on through all of the $(g-1)$st generation of $\Delta_2 T$, the values of f will not overrun the end of the gth generation. To check this, we have to carefully count 2's, 0's, paeons, and hypercatalectics.

We look first at the 2's and 0's in the $(g-1)$st generation of $\Delta_2 T$. In the even case, $\sum_{n \in \mathrm{sg}_0(g-1)} \Delta_2 T(n) = T(m_g - 1) - T(m_{g-1} - 2) = 3^g - 3^{g-1} = 2 \cdot 3^{g-1}$ and $(m_g - 1) - (m_{g-1} - 2) = 3^g + 1$, so $\Delta_2 T(n)$ has the value two 3^{g-1} times and zero the remaining $\frac{1}{2}(3^g + 1) - 3^{g-1} = \frac{1}{2}(3^{g-1} + 1)$ times. In the odd case, $\sum_{n \in \mathrm{sg}_1(g-1)} \Delta_2 T(n) = T(m_g - 2) - T(m_{g-1} - 1) = 3^g - 3^{g-1} = 2 \cdot 3^{g-1}$ and $(m_g - 2) - (m_{g-1} - 1) = 3^g - 1$, so $\Delta_2 T(n)$ is two 3^{g-1} times and zero the remaining $\frac{1}{2}(3^g - 1) - 3^{g-1} = \frac{1}{2}(3^{g-1} - 1)$ times.

Using the correspondence of Theorem 3.14, we conclude that in the gth generation the $\frac{1}{2}(3^{g-1}+1)$ even hypercatalectics outnumber the $\frac{1}{2}(3^{g-1}-1)$ odd hypercatalectics by one. Thus, the correspondence ends (after a total of $3^{g-1} + 1$ lines) with an even hypercatalectic followed by a run of odd paeons. There are 3^{g-1} even paeons and 3^{g-1} odd paeons. Adding together the lengths of all the paeons and hypercatalectics gives $4 \cdot 3^{g-1} + 4 \cdot 3^{g-1} + \frac{1}{2}(3^{g-1}+1) + \frac{1}{2}(3^{g-1}-1) = 3^{g+1}$, which is exactly the length of the gth generation.

Finally, we need to calculate the Foot Pattern Theorem 3.4 parameters when $n_0 = m_{g+1} - 4$ at the end of the gth generation. Let $P_0 = P_1 = 3^{g-1}$, $H_0 = \frac{1}{2}(3^{g-1} + 1)$,

and $H_1 = \frac{1}{2}(3^{g-1} - 1)$ be, respectively, the number of even paeons, odd paeons, even hypercatalectics, and odd hypercatalectics in the gth generation. Then by repeated application of Line Pattern Theorem 3.9, $r_0 = (m_{g-1} - 1) + 2H_0 + 2(P_1 - 1) = m_{g-1} - 1 + 3^{g-1} + 1 + 2 \cdot 3^{g-1} - 2 = m_{g-1} + 3^g - 2 = m_g - 2$, $r_1 = m_{g-1} + 2H_1 + 2P_1 = m_g - 1$, $t_0 = (3^g + 2) + 4P_0 + 2P_1 - 4 = 3^{g+1} - 2$, and $t_1 = (3^g + 2) + 4P_0 + 2P_1 - 2 = 3^{g+1}$. □

4. Some interesting properties of the three-term case. We continue to use T, U, and R as defined in (3.1). In this section we explore a variety of interesting properties of the T sequence, as well as of the $\Delta_2 T$ and $\Delta_2 U$ difference sequences.

Before we proceed, we need to present a short technical lemma concerning the relationship between differences in the T sequence and differences in the U sequence.

LEMMA 4.1. $\Delta_d T(n) = \sum_{i=0}^{k-1} \Delta_d U(n-i) = \sum_{i=0}^{d-1} \Delta_k U(n-i)$.

Proof. Rearrange the summation and difference operators as follows:

$$\begin{aligned}\Delta_d T(n) &= \sum_{i=0}^{k-1} \Delta_d U(n-i) = \sum_{i=0}^{k-1} U(n-i) - \sum_{i=0}^{k-1} U(n-d-i) \\ &= \sum_{i=0}^{k+d-1} U(n-i) - \sum_{i=0}^{d-1} U(n-k-i) - \sum_{i=0}^{k-1} U(n-d-i) \\ &= \sum_{i=0}^{d-1} U(n-i) - \sum_{i=0}^{d-1} U(n-k-i) = \sum_{i=0}^{d-1} \Delta_k U(n-i). \qquad \square\end{aligned}$$

Allenby and Smith [1] first observed that there is no n for which $T(n) = T(n-2) = T(n-4)$, or equivalently, for which $\Delta_2 T(n) = \Delta_2 T(n-2) = 0$. This follows directly from Foot Pattern Theorem 3.4.

PROPOSITION 4.2. *There do not exist n for which $\Delta_2 T(n) = \Delta_2 T(n-2) = 0$.*

Proof. The proof follows directly from Foot Pattern Theorem 3.4. □

In fact, Proposition 4.2 is equivalent to Proposition 3.3: if $\Delta_2 T(n) = \Delta_2 T(n-2) = 0$, then by Corollary 2.3, $\Delta_2 U(n) = \Delta_2 U(n-1) = \cdots = \Delta_2 U(n-4) = 0$; while if Proposition 4.2 holds, then for any n, $0 \neq \Delta_4 T(n) = \Delta_2 T(n) + \Delta_2 T(n-2) = \Delta_2 U(n) + \Delta_2 U(n-1) + 2\Delta_2 U(n-2) + \Delta_2 U(n-3) + \Delta_2 U(n-4)$, so $\Delta_2 U$ never has five consecutive zeros.

One of the more interesting properties of Conolly's sequence (1.2), our case $k = 2$, is that it consists of a monotonically increasing sequence of integers whose frequency counts form the Gray binary sequence (Sloane Sequence A001511 [13]), omitting its first term. We can prove the analogous property concerning the frequency counts of $T(n)$ for $k = 3$ and in so doing answer a conjecture given in Allenby and Smith [1].

THEOREM 4.3. $T(n) = T(n+1) = T(n+2)$ *iff* $n = m_g - 3$ *for some* g.

Proof (First conjectured in Allenby and Smith [1]). The Foot Pattern Theorem 3.4 parameters proven in Generation Pattern Theorem 3.15 give us the required equality when $n = m_g - 3$. We need to show then that equality does not occur elsewhere.

Suppose n^* is a minimal counterexample. That is, $n^* + 3$ is not the start of a generation, $t := T(n^*) = T(n^* + 1) = T(n^* + 2)$, and no smaller counterexamples exist. By Theorem 3.4, $\varphi(n^* - 1) = P$, $R(n^* + 1) = R(n^* - 1) = n^* - t + 1$, and $R(n^* + 2) = R(n^*) = R(n^* - 2) = n^* - t + 2$. By Lemma 4.1, $0 = \Delta_1 T(n^* + 2) = \Delta_3 U(n^* + 2) = T(n^* - t + 2) - T(n^* - t + 1) = \Delta_1 T(n^* - t + 2)$.

By Theorem 3.4, Lemma 4.1, and (3.1), we have $2 = \Delta_2 U(n^* - 1) = (\Delta_2 T)(R(n^* - 1)) = (\Delta_2 T)(R(n^* + 1)) = (\Delta_2 U)(R(n^* + 1)) + (\Delta_2 U)(R(n^* + 1) - 1) + (\Delta_2 U)(R(n^* +$

$1) - 2) = \Delta_2 U(n^* - t + 1) + \Delta_2 U(n^* - t) + \Delta_2 U(n^* - t - 1)$. Exactly one of these three terms is equal to 2: but which?

If $\Delta_2 U(n^* - t - 1) = 2$, then by Theorem 3.4, $T(n - t) = T(n - t + 2)$; and since we have already observed that $T(n - t + 1) = T(n - t + 2)$, we can invoke the minimality of n^* to conclude that there must be a g such that $m_g = n^* - t + 3$. But then $R(n^* - 1) = m_g - 2$ and $n^* - 1 = f(m_g - 2) = m_{g+1} - 4$, so $n^* = m_{g+1} - 3$, in contradiction to our hypothesis.

If $\Delta_2 U(n^* - t + 1) = 2$, then there must be a g such that $m_g = n^* - t + 1$ and similarly $n^* = m_{g+1} + 1$, which contradicts the observation in Generation Pattern Theorem 3.15 that the second and third members of any generation of T differ.

Finally, if $\Delta_2 U(n^* - t) = 2$, then there must be a g such that $m_g = n^* - t + 4$. $R(m_{g+1} - 6) = m_g - 4$ and $R(m_{g+1}) = m_g$, so by monotonicity R attains the value $m_g - 2$ at most twice, which contradicts $R(n^* - 2) = R(n^*) = R(n^* + 2) = m_g - 2$.

So none of the three choices is possible, and the original assumption of a counterexample must have been false. □

COROLLARY 4.4. *There is no n for which $T(n) = T(n+1) = T(n+2) = T(n+3)$.*

Proof. This is proven directly in Allenby and Smith [1] but follows simply from Proposition 4.3. □

In fact, it turns out that even two consecutive terms of T are rarely equal. Looking back at Figure 1.6, we can see that the points at which the two lines intersect all occur near the ends of generations.

COROLLARY 4.5. *$T(n) = T(n + 1)$ iff there is a g such that $n \in \{m_g - 5, m_g - 3, m_g - 2, m_g\}$.*

Proof. If $n \in \{m_g - 5, m_g - 3, m_g - 2, m_g\}$, then by the Generation Pattern Theorem 3.15, $T(n) = T(n + 1)$. Conversely, suppose $T(n) = T(n - 1)$ and consider where the nearest paeon starts. If $\varphi(n) = P$, then by Foot Pattern Theorem 3.4, $T(n - 3) = T(n - 2) = T(n - 1) = T(n) - 2$ and by Theorem 4.3, $n = m_g$ for some g. If $\varphi(n + 1) = P$, then similarly $n = m_g - 5$. If $\varphi(n + 2) = P$, then $n = m_g - 2$. Otherwise, $T(n) = T(n + 1) = T(n + 2)$ and $n = m_g - 3$. □

COROLLARY 4.6. *There are infinitely many n for which $T(n) = T(n + 1)$ and $T(n + 2) = T(n + 3)$.*

Proof. This was conjectured in Allenby and Smith [1] and follows from Corollary 4.5. The values of n for which this holds are precisely those that can be written as $n = m_g - 2$ for some g. It is shown in [1] that such n must satisfy $\{\Delta_1 T(n-1), \Delta_1 T(n+5)\} = \{-2, 2\}$ (in some order), which follows from the Generation Pattern Theorem 3.15. This theorem also tells us that if $T(n) = T(n+1)$, then either $T(n-2) = T(n-1)$ or $T(n + 2) = T(n + 3)$. □

COROLLARY 4.7. *There is no n for which $T(n) = T(n+1)$, $T(n+2) = T(n+3)$, and $T(n + 4) = T(n + 5)$.*

Proof. This is proven directly in Allenby and Smith [1] but follows simply from Corollary 4.5. □

Theorem 4.3 and the four corollaries that follow from it provide the characterization of the frequency counts that we seek. Because the case $k = 3$ gives a nonmonotonic sequence (unlike Conolly's $k = 2$), its repeated values can be either consecutive or nonconsecutive. We define the *consecutive repeated values* of a sequence

$$\{\overbrace{a_1, \ldots, a_1}^{n_1}, \overbrace{a_2, \ldots, a_2}^{n_2}, \ldots, \overbrace{a_m, \ldots, a_m}^{n_m}\},$$

where $a_1 \neq a_2$, $a_2 \neq a_3$, $\ldots$, $a_{m-1} \neq a_m$ in the obvious way as the sequence $\{n_1, \ldots, n_m\}$. Then we can summarize these results in the following proposition.

PROPOSITION 4.8. *The consecutive repeated values of the gth generation of T are*

$$\{2, \overbrace{1, \ldots, 1}^{3^{g+1}-7}, 2, 3\}.$$

Proof. This is a restatement of Corollary 4.5. ∎

In section 3 we showed the crucial role played by the $\Delta_2 U$ sequence that leads to our understanding of the detailed structure of the T sequence. It turns out that the $\Delta_2 U$ sequence has many fascinating properties, including beautiful symmetries both within and between generations, that make it of independent interest.

We begin by giving a more detailed description of the structure of each generation of $\Delta_2 U$. By Generation Pattern Theorem 3.15, the gth generation of $\Delta_2 U$ consists entirely of $3^{g-1}+1$ lines.

DEFINITION 4.9. *For $g > 0$ and $i \in [0, 3^{g-1}]$ let $q_{i,g}$ be the number of paeons in the ith line (numbered starting with the zeroth) of the gth generation of $\Delta_2 U$.*

In Example 3.8 we showed that $q_{0,1} = q_{1,1} = 1$; $q_{0,2} = 2$, $q_{1,2} = q_{2,2} = 1$, $q_{3,2} = 2$; $q_{0,3} = 4$, $q_{1,3} = q_{2,3} = q_{3,3} = 1$, $q_{4,3} = q_{5,3} = 2$, $q_{6,3} = q_{7,3} = q_{8,3} = 1$, and $q_{9,3} = 4$.

PROPOSITION 4.10. *The sequence consisting of the number of paeons in each even line in the $(g+1)$st generation of $\Delta_2 U$ can be expressed in terms of the number of paeons in the lines in the gth generation of $\Delta_2 U$ as follows:*

$$\{\, 2q_{0,g}, \overbrace{1, \ldots, 1}^{q_{1,g}}, 2q_{2,g}, \overbrace{1, \ldots, 1}^{q_{3,g}}, 2q_{4,g}, \overbrace{1, \ldots, 1}^{q_{5,g}}, \ldots, 2q_{3^{g-1}-1,g}, \overbrace{1, \ldots, 1}^{q_{3^{g-1},g}} \}.$$

Proof. Generational Correspondence Theorem 3.14 (see Figure 3.1), together with Generation Pattern Theorem 3.15, shows that each even (odd) paeon in a generation produces two paeons in the next even (odd) semigeneration and a paeon and a hypercatalectic in the next odd (even) semigeneration, while an even (odd) hypercatalectic simply produces an even (odd) hypercatalectic. Thus each paeon contributes a single paeon line (i.e., PH) to the next subgeneration of opposite parity and two paeons (PP) to the next subgeneration of equal parity. The number of even paeons in each even line in the gth generation therefore appears doubled in the $(g+1)$st even semigeneration, alternating with runs of single-paeon lines. ∎

Recall that a sequence is *palindromic* if it has reflective symmetry. Two basic transformations that preserve palindromicity are that of replacing every occurrence of a member of a sequence by a palindromic subsequence, and that of applying any transformation to the lengths of consecutive runs of identical members in a sequence. For example, the string "AABABAA" is palindromic and changing every "AA" to "A" and every "B" to "CDC" gives "ACDCACDCA," which remains palindromic. We use these palindromicity-preserving transformations and Foot Pattern Theorem 3.4 to show the following generational palindromicity property, which we can use to find $q_{i,g}$ for odd i.

THEOREM 4.11. *Each generation of $\Delta_2 U(n)$ consists of a palindromic sequence of feet.*

Proof. The result is true for the first generation, which consists of $\{P, H, P\}$; see Example 3.8. We proceed by induction, assuming that it is true for all generations before the gth.

By induction, the $(g-1)$st generation of $\Delta_2 U$ is a palindromic sequence of feet. Transforming that sequence using $\{P \mapsto 02220, H \mapsto 0\}$ followed by $\{000 \mapsto 00, 00 \mapsto$

0} gives us a palindromic sequence, which Foot Pattern Theorem 3.4 lets us recognize as the $(g-1)$st generation of $\Delta_2 T$, preceded by a zero. Because the length of every generation is odd, it follows that the $(g-1)$st even semigeneration of $\Delta_2 T(n)$ is the reverse of the $(g-1)$st odd semigeneration of $\Delta_2 T(n)$, followed by a zero. Then by Generational Correspondence Theorem 3.14 (see Figure 3.1), the sequence $\{q_{0,g}, q_{2,g}, q_{4,g}, \dots\}$ is the reverse of $\{q_{1,g}, q_{3,g}, q_{5,g}, \dots\}$. Thus, the entire sequence $\{q_{i,g}\}_{i=0}^{3^g}$ is palindromic and all the feet in the gth generation form a palindromic sequence. ∎

COROLLARY 4.12. *Each generation of $\Delta_2 T(n)$ consists of a palindromic sequence of 2's and 0's, followed by an extra 0.*

Proof. The proof follows from Theorem 4.11 by application of Foot Pattern Theorem 3.4. ∎

PROPOSITION 4.13. *$q_{i,g}$ is always a power of 2.*

Proof. The proposition is true for $g = 1$, since $q_{0,1} = q_{1,1} = 1$. Assume inductively that the proposition is true for all generations preceding the gth. Proposition 4.10 tells us that $q_{i,g}$ for even i is a power of 2, and Theorem 4.11 tells us that the $q_{i,g}$ for odd i have the same (power of 2) values in reverse order. ∎

We now use what we have learned about the structure of our sequences to obtain some quantitative properties of the sequences.

PROPOSITION 4.14. *For positive g and $0 \le x \le m_{g+1} - 4$, the sum $T(m_g + x) + T(m_{g+1} - 4 - x) = 3^g + 3^{g+1}$ is constant. For $0 \le y \le m_{g+1} - 6$, the sum $U(m_g + y) + U(m_{g+1} - 6 - y) = 3^{g-1} + 3^g$ is constant.*

Proof. The proof follows readily from palindromicity of $\Delta_2 T(n)$ and $\Delta_2 U(n)$ within the gth generation. ∎

PROPOSITION 4.15. *The mean value of the gth generation of $T(n)$ is $2 \cdot 3^g + 1$. The mean value of the gth generation of $U(n)$ is $2 \cdot 3^{g-1} + \frac{5}{9}$.*

Proof. The proof follows from Proposition 4.14 and the values indicated by Generation Pattern Theorem 3.15 and Foot Pattern Theorem 3.4 for the last few values in the gth generations of $T(n)$ and $U(n)$. ∎

COROLLARY 4.16. *The asymptotic value of $\frac{T(n)}{n}$ is $\frac{2}{3}$.*

Proof. The mean value of n in the gth generation is $(m_g + m_{g+1} - 1)/2 = (3^{g+1} + 5 + 3^{g+2} + 5 - 2)/4 = 3^{g+1} + 2 = 3 \cdot 3^g + 2$. By Proposition 4.15, the mean value of T in the same generation is $2 \cdot 3^g + 1$. The ratio $(2 \cdot 3^g + 1)/(3 \cdot 3^g + 2)$ approaches $\frac{2}{3}$ as g approaches infinity. ∎

PROPOSITION 4.17. *The mean value of the gth generation of $\Delta_2 T(n)$ is $\frac{4}{3}$. The mean value of the gth generation of $\Delta_2 U(n)$ is $\frac{4}{9}$.*

Proof. In the proof of the Generation Pattern Theorem 3.15, we counted $2 \cdot 3^{g-1}$ paeons in the gth generation, which has length 3^{g+1}. Recall from Foot Pattern Theorem 3.4 that at a hypercatalectic, $\Delta_2 T(n) = \Delta_2 U(n) = 0$, while each paeon contributes 6 to the sum of $\Delta_2 T$ and 2 to the sum of $\Delta_2 U$. ∎

We conclude this section with an observation whose generalization will play a key role in the next section.

PROPOSITION 4.18. $\Delta_4 U(n) \in \{0, 2\}$.

Proof. The result is easily verified for small n. Expand $\Delta_6 U(n)$ in two ways to obtain $\Delta_4 U(n) = \Delta_4 U(n-2) + (\Delta_2 U(n) - \Delta_2 U(n-4))$. By Foot Pattern Theorem 3.4, $\Delta_6 U$ has period 2 (alternating between the values 2 and 0) on any run of paeons and is 0 at a hypercatalectic. ∎

To date, we have not yet succeeded in proving that no closed form exists for $T(n)$. Nonetheless, the tools that we have created have facilitated the analysis which nor-

mally follows from the discovery of a closed form, namely, a complete characterization of the structure of the sequence, the computation of mean and asymptotic values, and the rapid calculation of large values of the sequence.

5. The structure for general odd *k*. In the preceding two sections, we examined the case $k = 3$ in great detail. In this section, we find that most of our results generalize to greater odd k. In the recursion (1.6) we again set $a = 0$ and assume the initial conditions $T_{0,k}(n) = 1$ for $1 \le n \le k$ with k odd. We will refer to this sequence as $T(n)$, to $R_{0,k}(n)$ as $R(n)$, and to $U_{0,k}(n)$ as $U(n)$:

$$\begin{aligned} T(n) &:= U(n) + U(n-1) + \cdots + U(n-k+1), \quad n > k, \\ U(n) &:= T(R(n)), \quad n > 1, \\ R(n) &:= n - T(n-1), \quad n > 1. \end{aligned} \tag{5.1}$$

When $k = 3$, the behavior of the recursion can largely be characterized in terms of the pattern of feet in the values of $\Delta_{k-1}U$, most of which have length $k+1 = 4$, with the occasional hypercatalectic. For general odd k, we find that $\Delta_{k-1}U$ follows the same pattern, consisting mostly of paeons now of length $k+1$, with a hypercatalectic of length 1 every now and then.

In order to prove that there is only one zero or hypercatalectic between runs of paeons in $\Delta_{k-1}U$—which is not the case for even k—we will need to prove an additional property of these runs of paeons. This is that paeons always come in groups of $\frac{k-1}{2}$, a property which was trivial for $k = 3$, where $\frac{k-1}{2} = 1$.

It happens that $\Delta_{k+1}U$ has repeating patterns resembling feet of length $k-1$, also with the occasional extra term resembling a hypercatalectic. We will show that the hypercatalectics in $\Delta_{k-1}U$ *always coincide* with the analogous disruptions in $\Delta_{k+1}U$. This gives the recursion an additional level of structure: paeons always come grouped in *polypaeons* of length $\operatorname{lcm}(k-1, k+1) = \frac{1}{2}(k^2-1)$. That is, a polypaeon consists of $\frac{k-1}{2}$ consecutive paeons.

We begin by generalizing our previous definitions and defining what we mean by a polypaeon.

DEFINITION 5.1. *A* paeon *is a sequence*

$$\{k-1, \overbrace{0, \ldots, 0}^{k}\}$$

of $k+1$ consecutive values of $\Delta_{k-1}U(n)$. A hypercatalectic *is a singleton sequence $\{0\}$, immediately preceded in $\{\Delta_{k-1}U(n)\}$ by a paeon. A* foot *is either a paeon or a hypercatalectic. A* polypaeon *is a sequence of $\frac{1}{2}(k-1)$ consecutive paeons. For convenience, we will write $\{P\}$ interchangeably with the paeon $\{k-1, 0, \ldots, 0\}$ and likewise $\{H\}$ with the hypercatalectic $\{0\}$ when listing values of $\Delta_{k-1}U$. We also define $\varphi(n)$ on a subset of the natural numbers as the symbol P if $\Delta_{k-1}U(n) = k-1$ begins a paeon and H if $\Delta_{k-1}U(n) = 0$ is a hypercatalectic, and we leave it undefined otherwise.*

We now generalize Proposition 2.2 to facilitate proving the polypaeon structure.

PROPOSITION 5.2. *For fixed odd $k > 1$, all even $d \in [0, k-1]$, and any $n > k$, both $|\Delta_d U(n)|$ and $\Delta_d T(n)$ belong to the set $\{0, k-1\}$.*

Proof. The case $k = 3$ was already proven in Proposition 2.2, and the case $d = 0$ is trivial. We assume therefore in what follows that $k > 3$ and $d > 0$.

As in the proof of Proposition 2.2, we proceed by induction and leave it to the reader to verify that the result holds for $n \le 2k$. Assume then that $n > 2k$ and that the result holds for lesser n and all relevant d.

$\Delta_d U(n) := T(R(n)) - T(R(n-d)) = (\Delta_x T)(R(n))$, where $x = \Delta_d R(n) = d - \Delta_d T(n-1)$. By the induction assumption, $\Delta_d T(n-1) \in \{0, k-1\}$, so $x \in \{d, d-k+1\}$. In either case, $|x|$ is even and in $[0, k-1]$ and the greater of $R(n)$ and $R(n-d)$ is strictly less than n, so we can apply induction again to find that $|(\Delta_x T)(R(n))| \in \{0, k-1\}$ and hence $|\Delta_d U(n)| \in \{0, k-1\}$.

Observe that $\Delta_d T(n) + \Delta_{k-1-d} T(n-d) = \Delta_{k-1} T(n)$. By Proposition 2.2, $\Delta_{k-1} T(n) \in \{0, k-1\}$; and by induction, $\Delta_{k-1-d} T(n-d) \in \{0, k-1\}$. Therefore, $|\Delta_d T(n)| \in \{0, k-1\}$. We need to rule out the case $\Delta_d T(n) = 1-k$. We expand $\Delta_d T(n)$ as a telescoping sum:

$$\Delta_d T(n) = \sum_{\substack{i \text{ even} \\ \in [0,d)}} \Delta_2 T(n-i). \tag{5.2}$$

By induction, when $i > 0$, $\Delta_2 T(n-i) \in \{0, k-1\}$. To prove that $\Delta_d T(n) \ge 0$, it therefore suffices to show that the first summand $\Delta_2 T(n) \ge 0$, that is, $\Delta_2 T(n) \ne 1-k$. Write

$$\Delta_2 T(n) = \Delta_{k-1} T(n) - \Delta_{k-3} T(n-2). \tag{5.3}$$

We will assume that $\Delta_2 T(n) = 1-k$ and show that a contradiction ensues. From (5.3) it follows that $0 = \Delta_{k-1} T(n)$ and $k-1 = \Delta_{k-3} T(n-2)$. But $\Delta_{k-3} T(n-2) = \Delta_{k-1} T(n-2) - \Delta_2 T(n-k+1)$, and by the same reasoning $0 = \Delta_2 T(n-k+1)$, which will be contradicted below.

By Lemma 4.1, $1-k = \Delta_2 T(n) = \Delta_k U(n) + \Delta_k U(n-1) = \Delta_{k+1} U(n) + \Delta_{k-1} U(n-1) = \Delta_2 U(n) + \Delta_{k-1} U(n-2) + \Delta_{k-1} U(n-1)$. Since by Proposition 2.2 the last two terms are nonnegative, $\Delta_2 U(n) = 1-k$. $\Delta_2 R(n) = 2 - \Delta_2 T(n-1) \in \{\, 2, 3-k \,\}$ by induction, but if it were 2, then $\Delta_2 U(n)$ would be equal to $(\Delta_2 T)(R(n)) \in \{0, k-1\}$, contradicting our earlier assumption. Thus $\Delta_2 T(n-1) = k-1$ and $\Delta_2 R(n) = 3-k$. By (5.2), $\Delta_{k-1} T(n-1) = k-1$.

So $\Delta_{k-1} T(n-1) - \Delta_{k-1} T(n) = (k-1) - 0$, and if we expand this using (5.1) and gather terms, $k-1 = \Delta_{k-1} U(n-k) - \Delta_{k-1} U(n)$, and by Proposition 2.2, $k-1 = \Delta_{k-1} U(n-k)$. By Corollary 2.5, (5.1), and the properties of the difference operator (respectively), $0 = \sum_{i=k+1}^{2k} \Delta_{k-1} U(n-i) = \Delta_{k-1} T(n-k-1) = \Delta_{k-3} T(n-k-1) + \Delta_2 T(n-2k+2)$, both of whose last terms are thus zero, while by (1.6), $k-1 \ge \Delta_{k-1} T(n-k+1) \ge \Delta_{k-1} U(n-k) = k-1$.

But then $\Delta_2 T(n-k+1) = \Delta_{k-1} T(n-k+1) - \Delta_{k-3} T(n-k-1) = k-1$, a contradiction. Thus, $\Delta_2 T(n) \in \{0, k-1\}$ and the induction is complete. □

The polypaeon structure can be viewed as arising from the difference identity

$$\Delta_{k+1} \Delta_{k-1} U(n) = \Delta_{k-1} \Delta_{k+1} U(n). \tag{5.4}$$

This elementary difference equation causes periodicity with period $k \pm 1$ in the sequences $\Delta_{k \mp 1} U(n)$ to mutually reinforce each other: $\Delta_{k-1} U(n)$ is periodic with period $k+1$ on an interval (i.e., the left-hand side of (5.4) is zero) iff $\Delta_{k+1} U(n)$ is periodic on that interval with period $k-1$ (i.e., the right-hand side of (5.4) is zero).

Recall Note 2.1, which lists values of $\Delta_6 U(n)$ when $k = 7$. We can now recognize the values beginning with $n = 15$ as forming three paeons, then a hypercatalectic, then seven more paeons. $\Delta_6 U(n)$ is therefore periodic with period 8 on the interval

from $n = 15$ to $n = 38$, and again from $n = 40$ to $n = 95$. This means that according to (5.4), $\Delta_8 U(n)$, whatever its values may be, is periodic with period 6 on those two ranges of n. The reader may verify that in fact $\Delta_8 U(n) = 0$, except at the beginning of each period, when it is equal to 6.

For general odd k, we will show that the periods in $\Delta_{k+1}U(n)$ behave similarly to the case $k = 7$: the difference sequence is 0 except at the beginning of each period of length $k - 1$, where it has the value $k - 1$. So we have two difference sequences, $\Delta_{k-1}U(n)$ and $\Delta_{k+1}U(n)$, which are mostly zero but almost periodically equal to $k - 1$ (with different periods). In what follows, we will need to discuss the phase difference between these sequences, which we now define.

DEFINITION 5.3. *Let n mark the beginning of a paeon in $\Delta_{k-1}U(n)$, that is, $\varphi(n) = P$. The* phase difference $\theta(n)$ *between the sequences $\Delta_{k\pm1}U(n)$ is the smallest nonnegative value for which $\Delta_{k+1}U(n + \theta(n)) \neq 0$.*

Since the periods in $\Delta_{k+1}U$ are shorter than the periods in $\Delta_{k-1}U$ by 2, the phase difference $\theta(n)$ decreases by 2 (modulo $k - 1$) with each successive consecutive paeon in $\Delta_{k-1}U$. We will also find that because of the way in which the periodicity in the two difference sequences is mutually reinforcing, the only time that a hypercatalectic can occur in $\Delta_{k-1}U$ is when the phase difference is about to drop from 2 to 0, $\theta(n)$ returns to zero at the beginning of each polypaeon and $\Delta_{k+1}U(n)$ is zero except where it is $k - 1$. We will use these ideas in the following proof of the polypaeon structure of $\Delta_{k-1}U(n)$, which while lengthy illustrates well the added complexity of the case of general odd k.

PROPOSITION 5.4 (polypaeon structure of $\Delta_{k-1}U(n)$).

(I) *Along each polypaeon, $\Delta_{k+1}U$ consists of $\frac{k+1}{2}$ copies of the $k - 1$ integers $(k-1, 0, \ldots, 0)$. Formally, suppose $\varphi(n) = P$ marks the start of a polypaeon, and $n \leq m < n + \frac{k^2-1}{2}$. Then $\Delta_{k+1}U(m) = k - 1$ if $m \equiv n \pmod{k-1}$, and $\Delta_{k+1}U(m) = 0$ otherwise.*

(II) *At a hypercatalectic, $\Delta_{k+1}U$ vanishes. Formally, if $\varphi(n) = H$, then we have $\Delta_{k+1}U(n) = 0$.*

(III) *Polypaeons are followed by either polypaeons or hypercatalectics. Formally, if $\varphi(n - \frac{k^2-1}{2}) = P$ marks the start of a polypaeon, then that polypaeon is followed at n by either another polypaeon or a hypercatalectic.*

(IV) *Hypercatalectics are followed by polypaeons. Formally, if $\varphi(n-1) = H$, then a polypaeon starts at n.*

Proof. It is laborious, but not difficult, to verify directly that there is a polypaeon which starts at $2k + 1$ and is followed by a hypercatalectic at $\frac{1}{2}(k^2 + 4k + 1)$, along both of which $\Delta_{k+1}U$ has the required values.

We proceed by induction, assuming that all four statements hold for all lesser n. In what follows, we will use statements (I) and (II) to compute values of $\Delta_{k+1}U$ that precede n, and use statements (III), (IV) and the above-mentioned presence of the initial polypaeon starting at $2k + 1$ and hypercatalectic at $\frac{1}{2}k^2 + 4k + 1$ to ensure the polypaeon-hypercatalectic structure that immediately precedes n. We begin by proving statement (I).

By (5.4) on the commutativity of the difference operator, $\Delta_{k+1}U(n) = \Delta_{k+1}U(n-k+1) + \Delta_{k-1}U(n) - \Delta_{k-1}U(n-k-1)$. By induction, $\Delta_{k+1}U(n-k+1) = \Delta_{k-1}U(n-k-1)$, with both being equal to zero if n is preceded by a hypercatalectic and equal to $k-1$ if n is preceded by a paeon. $\Delta_{k-1}U(n) = k - 1$ by assumption, so $\Delta_{k+1}U(n)$ is also equal to $k - 1$. The rest of statement (I) follows by induction (to obtain earlier values of $\Delta_{k+1}U$) and the periodicity that (5.4) and the repeated paeons in

the polypaeon impose on $\Delta_{k+1}U$.

Proving statement (II) simply consists of using (5.4) and induction to compute $\Delta_{k+1}U(n) = \Delta_{k+1}U(n-k+1) + \Delta_{k-1}U(n) - \Delta_{k-1}U(n-k-1) = 0+0-0 = 0$.

Statement (III) is more difficult to prove. It follows directly from Corollary 2.5 and Definition 5.1 that what follows the last paeon of a polypaeon must be either a hypercatalectic (if $\Delta_{k-1}U(n) = 0$) or a paeon (if $\Delta_{k-1}U(n) = k-1$). We assume therefore the latter case. It is not obvious at the outset that the paeon at n is the first paeon of a full polypaeon. We will prove this in fact true by showing (i) that $\theta(n) = 0$, (ii) that θ drops by 2 modulo $k-1$ with each successive paeon, and (iii) that the only time the next hypercatalectic can occur is when θ would drop from 2 to 0.

In our proof above of statement (I), we showed that $\varphi(n) = P$ implies that $\Delta_{k+1}U(n) = k-1$, and showed how to use periodicity to calculate the values of $\Delta_{k+1}U(n)$ that are induced by consecutive paeons, showing (ii). Since $\Delta_{k+1}U(n) = k-1$, we have $\theta(n) = 0$. And since the $(k-1)$'s in $\Delta_{k+1}U$ recur with period $k-1$ as long as we continue to have consecutive paeons at $n' \in \{n, n+k+1, n+2(k+1), \dots\}$, then $\theta(n')$ drops by 2 modulo $k-1$ with each successive paeon.

To obtain (iii), we need to determine whether the next foot after the paeon at some n' is a paeon or a hypercatalectic. We calculate

$$\begin{aligned}\Delta_{k-1}U(n'+k+1) &= 0+\cdots+0+\Delta_{k-1}U(n'+k+1)\\ &= \sum_{i=0}^{k-1}\Delta_{k-1}U(n'+k+1-i)\\ &= \Delta_{k-1}T(n'+k+1)\\ &= \sum_{0\le j\ even<k-1}\Delta_2T(n'+k+1-j) \quad \text{by (5.2).}\end{aligned} \tag{5.5}$$

If $\theta(n') > 2$, then $\Delta_2T(n'+\theta(n'))$ is one of these summands, which we can then compute as follows:

$$\begin{aligned}\Delta_2T(n'+\theta(n')) &= \sum_{i=0}^{k-1}\Delta_2U(n'+\theta(n')-i), \text{ which telescopes to}\\ &= \Delta_{k+1}U(n'+\theta(n')) + \Delta_{k-1}U(n'+\theta(n')-1)\\ &= (k-1)+0 = k-1.\end{aligned} \tag{5.6}$$

It is worth noting in passing that (5.6) shows that $\Delta_2T(n') = k-1$ whenever $\Delta_{k+1}U(n') = k-1$.

We now consider three cases separately: $\theta(n') > 2$, $\theta(n') = 0$, and $\theta(n') = 2$.

In the first case, if $\theta(n') > 2$, then (5.6) shows that $\Delta_{k-1}U(n'+k+1) = k-1$ starts a paeon.

In the second case, if $\theta(n') = 0$, then by the same argument used in the proof of (I) above, $\Delta_{k+1}U(n'+k-1) = k-1$. And by the same argument used in (5.6), substituting $k-1$ for $\theta(n')$, we have $\Delta_2T(n'+k-1) = k-1$. This value of Δ_2T is among the summands in the last line of (5.5), so $\Delta_{k-1}U(n'+k+1) = k-1$ begins a paeon.

It is therefore only in the third case, $\theta(n') = 2$, that a hypercatalectic can occur. This completes the proof of statement (III). All that remains now is statement (IV).

We know from Proposition 2.2 that $\Delta_{k-1}U(n)$ is either 0 or $k-1$. If $\Delta_{k-1}U(n) = k-1$, then we can use the preceding proof of statement (III) to show that it marks the beginning of a polypaeon. What we need to show then is simply that $\Delta_{k-1}U(n) \neq 0$.

To do so, we generalize the proof of Proposition 3.3, relying on the polypaeon property to ensure that the number of preceding paeons is a multiple of $\frac{k-1}{2}$, so that we can compute some key differences of T.

By induction and the presence of a first hypercatalectic at $n = \frac{1}{2}k^2 + 4k + 1$, we know that the hypercatalectic at $n-1$ is preceded by some positive number of polypaeons, the first of which is itself preceded by a hypercatalectic. For some positive $q = r\,\frac{k-1}{2}$ then, $\Delta_{k-1}U(n-k-2) = \Delta_{k-1}U(n-2(k+1)-1) = \cdots = \Delta_{k-1}U(n-q(k+1)-1) = \Delta_{k-1}U(n-(q+1)(k+1)-2) = k-1$ and $\Delta_{k-1}U(i) = 0$ for all other i in the interval $[n-(q+1)(k+1)-2, n-1]$.

Use these known values of $\Delta_{k-1}U$, the distributivity of the difference operator over the definition of T in (5.1), and the equation $\Delta_d R(n) = d - \Delta_d T(n-1)$ to compute four important differences of R:

$$\begin{aligned}\Delta_2 R(n) &= 2 - \Delta_2 T(n-1)\\ &= 2 - \Delta_{k+1}T(n-1) + \Delta_{k-1}T(n-3)\\ &= 2 - \sum_{i=0}^{k-1}\Delta_{k+1}U(n-1-i) + \sum_{i=0}^{k-1}\Delta_{k-1}U(n-3-i)\\ &= 2-(k-1)+(k-1) = 2.\end{aligned} \tag{5.7}$$

The preceding sums are evaluated by using induction to find the values of the differences of U and observing that exactly one summand in each range is nonzero. The following three equations are derived similarly, using in addition a parity argument for (5.10):

$$\Delta_{k-1}R(n) = k-1, \tag{5.8}$$

$$\Delta_{k-1}R(n-q(k+1)-2) = k-1, \tag{5.9}$$

$$\Delta_{k-1}R(n-i(k-1)-2) = 0 \text{ for } 0 \le i \le r\,\frac{k+1}{2}. \tag{5.10}$$

By definition, $\Delta_{k-1}U(n) = U(n) - U(n-k+1) = T(R(n)) - T(R(n-k+1))$. Since $\Delta_{k-1}R(n) = k-1$, we can continue the equation $T(R(n)) - T(R(n-k+1)) = T(R(n)) - T(R(n)-k+1) = (\Delta_{k-1}T)(R(n)) = \sum_{i=0}^{k-1}(\Delta_{k-1}U)(R(n)-i)$. We need to show that this last sum is equal to $k-1$. We claim that this is so because one of its first two terms is $k-1$ and the rest are all zero. By the inductive assumption that the sequence preceding n consists of feet, it suffices to show that $(\Delta_{k-1}U)(R(n)-i) = 0$ for $i \in [2, k+1]$, forcing one of the two following differences ($i = 0$ or $i = 1$) of U to be the nonzero start of a paeon:

$$\begin{aligned}\sum_{i=2}^{k+1}(\Delta_{k-1}U)(R(n)-i) &= (\Delta_{k-1}T)(R(n)-2)\\ &= (\Delta_{k-1}T)(R(n-2)) \text{ by (5.7)}\\ &= (\Delta_{k-1}T)\left(R(n-r\,\frac{k+1}{2}(k-1)-2)\right) \text{ by (5.10)}\\ &= (\Delta_{k-1}T)(R(n-q(k+1)-2))\\ &= (\Delta_{k-1}U)(n-q(k+1)-2) \text{ by (5.9)}\\ &= 0.\end{aligned}$$

TABLE 5.1
Function values over a paeon for odd k.

n	$T(n)$	$\Delta_{k-1}T(n)$	$R(n)$	$\Delta_{k-1}R(n)$	$\Delta_{k-1}U(n)$	$\varphi(n)$
n_0-k	$t_{k-2}-k+1$	$k-1$	?	0	0	–
n_0-k+1	t_0-k+1	$k-1$	r_0-k+1	0	0	–
⋮	⋮	⋮	⋮	⋮	⋮	⋮
n_0-3	$t_{k-4}-k+1$	$k-1$	?	0	0	–
n_0-2	$t_{k-3}-k+1$	?	r_0-2	0	0	–
n_0-1	$t_{k-2}-k+1$	0	r_1	?	0	?
$\boxed{n_0}$	$\boxed{t_0}$	$k-1$	$\boxed{r_0}$	$k-1$	$k-1$	P
n_0+1	$\boxed{t_1}$	$k-1$	$\boxed{r_1}$	0	0	–
⋮	⋮	⋮	⋮	⋮	⋮	⋮
n_0+k-2	$\boxed{t_{k-2}}$	$k-1$	$\boxed{r_{k-2}}$	0	0	–
n_0+k-1	t_0+k-1	$k-1$	r_0	0	0	–
n_0+k	t_1	0	r_1	0	0	–
n_0+k+1	t_2+d	d	r_2+k-1	$k-1$	$\boxed{d}$	P or H
n_0+k+2	t_3+k-1	$k-1$	$\substack{r_3+\\ k-1-d}$	$k-1-d$	$k-1-d$	– or P
n_0+k+3	t_4+k-1	$k-1$	r_4	0	0	–

All the summands are therefore zero, and so one of $(\Delta_{k-1}U)(R(n)-1)$ and $(\Delta_{k-1}U)(R(n))$ must be $k-1$ and we are done. ∎

The fundamental result that $\Delta_{k-1}U$ consists of polypaeons separated by at most one zero (constituting a hypercatalectic) now follows as a direct consequence.

COROLLARY 5.5 (a generalized version of Proposition 3.3). *$\{\Delta_{k-1}U(n)\}_{n=2k+1}^{\infty}$ consists only of feet, and each of its paeons occurs within a run of polypaeons.*

Proof. The proof follows immediately from Proposition 5.4. ∎

Since the remainder of the results in this section may be proven by straightforward generalizations of the corresponding earlier propositions, we state them here without proof.

THEOREM 5.6 (a generalized version of Foot Pattern Theorem 3.4). *Suppose the parameters n_0, $t_0,\dots,t_{k-2}$, $r_0,\dots,r_{k-2}$ and d satisfy all of the following conditions: $n_0 \geq 2k+1$, $\Delta_{k-1}U(n_0) = k-1$, $\Delta_{k-1}U(n_0+k+1) = d$, and for $0 \leq i \leq k-2$, $T(n_0+i) = t_i$, and $R(n_0+i) = r_i$. Then $T(n)$, $\Delta_{k-1}T(n)$, $R(n)$, $\Delta_{k-1}R(n)$, $\Delta_{k-1}U(n)$, and $\varphi(n)$ have the values shown in Table* 5.1.

DEFINITION 5.7 (generation). *For any $g > 0$, let $m_g := \frac{1}{k-1}(k^{g+1}+k^2-k-1) = k+\sum_{i=0}^{g}k^i$ and call the interval $[m_g, m_{g+1}-1]$ the* gth generation*, written as* $\mathrm{gen}(g)$*. We partition the gth generation into two nonconsecutive subsequences: the* gth even semigeneration $\mathrm{sg}_0(g) := \{n \in \mathrm{gen}(g) \mid n \equiv m_g \pmod 2\}$ *and the* gth odd semigeneration $\mathrm{sg}_1(g) := \{n \in \mathrm{gen}(g) \mid n \not\equiv m_g \pmod 2\}$*. For any sequence $s(n)$, we will refer to the subsequence $\{s(n) \mid n \in \mathrm{gen}(g), s(n)$ defined$\}$ as the* gth generation of s *and similarly for semigenerations. An* even (odd) *foot is one that starts in an even (odd) semigeneration. Note that because the length k^{g+1} of $\mathrm{gen}(g)$ is odd, m_g and m_{g+1} always have opposite parity.*

THEOREM 5.8 (a generalized version of Generation Pattern Theorem 3.15). *The gth generation of $\Delta_{k-1}U$ consists entirely of k^g feet, which make up $k^{g-1}+1$ lines.*

Its last line consists only of (odd) paeons, the last of which has Foot Pattern Theorem 5.6 parameters $n_0 = m_{g+1} - k - 1$, $t_0 = k^{g+1} - k + 1$, $t_1 = \cdots = t_{k-2} = k^{g+1}$, $r_0 = m_g - 2$, $r_1 = m_g - 1$ *and for* $2 \le i \le k-2$, $r_i = m_g + i - 2$.

At this point, we have generalized all of section 3 for odd k, describing the generational structure of the sequences. The remaining results in this section generalize the indicated propositions and theorems about properties of the sequences, proven for $k = 3$ in section 4.

PROPOSITION 5.9 (a generalized version of Proposition 4.10). *The sequence consisting of the number of paeons in each even line in the* $(g+1)$*st generation of* $\Delta_2 U$ *is*

$$\left\{ \frac{k+1}{2}\, q_{0,g}, \overbrace{\frac{k-1}{2}, \ldots, \frac{k-1}{2}}^{q_{1,g}}, \ldots, \frac{k+1}{2}\, q_{k^{g-1}-1,g}, \overbrace{\frac{k-1}{2}, \ldots, \frac{k-1}{2}}^{q_{k^{g-1},g}} \right\}.$$

THEOREM 5.10 (a generalized version of Theorem 4.11). *Each generation of the sequence* $\Delta_{k-1} U(n)$ *consists of a palindromic sequence of feet.*

PROPOSITION 5.11 (a generalized version of Proposition 4.13). $q_{i,g}$ *is always* $\frac{k-1}{2}$ *times a power of* $\frac{k+1}{2}$.

PROPOSITION 5.12 (a generalized version of Proposition 4.14). *For positive* g *and* $0 \le x \le m_{g+1} - k - 1$, *the sum* $T(m_g + x) + T(m_{g+1} - k - 1 - x) = k^g + k^{g+1}$ *is constant. For* $0 \le y \le m_{g+1} - 2k$, *the sum* $U(m_g + y) + U(m_{g+1} - 2k - y) = k^{g-1} + k^g$ *is constant.*

PROPOSITION 5.13 (a generalized version of Proposition 4.15). *The mean value of the* g*th generation of* $T(n)$ *is* $\frac{1}{2}(k^{g+1} + k^g + k - 1)$.

COROLLARY 5.14 (a generalized version of Corollary 4.16). *The asymptotic value of* $\frac{T(n)}{n}$ *is* $\frac{k-1}{k}$.

PROPOSITION 5.15 (a generalized version of Proposition 4.17). *The mean value of the* g*th generation of* $\Delta_2 T(n)$ *is* $\frac{2(k-1)}{k}$. *The mean value of the* g*th generation of* $\Delta_2 U(n)$ *is* $\frac{2(k-1)}{k^2}$.

We conclude this section with a few conjectures based on empirical evidence. This paper does not for the most part discuss the case of $a \neq 0$, but the following observation seems closely enough related to the palindromicity property proven in section 4 to be worth mentioning here.

CONJECTURE 5.16 (a generalized version of Proposition 4.11). *For general* a *and odd* k, $\Delta_{k-1} U$ *has a palindromic generational structure, but there are* k *extra zeros after each generation.*

We have observed the following curious property, which describes a surprising way in which the even and odd q sequences dovetail together.

CONJECTURE 5.17. *For* $a = 0$, *odd* k, *and sufficiently large* g, *the number of consecutive times that* $q_{i,g}$ *has the value* $\frac{k-1}{2}$ *is always* k.

As in the case of $k = 3$, maximal runs of identical values in the $T(n)$ sequence tend to occur at the ends of generations. This appears to be true for any odd k.

CONJECTURE 5.18 (a generalized version of Theorem 4.3). *For* $a = 0$ *and odd* k, $T(n-k) = \cdots = T(n-1)$ *iff* $n = m_g$ *for some* g.

We conclude this section with two open questions suggesting avenues for further research, and welcome correspondence concerning them.

QUESTION 5.19. *What can be said of the sequences generated by other initial values? In particular, which initial values give sequences which are well defined, and*

which ones lead to a generational structure of the sort described in this paper?

QUESTION 5.20. *Let $a = 0$ and k be odd. The palindromic symmetry property means that the values of each sequence at odd integers can be simply expressed in terms of the values at even integers, and vice versa. Is there a simple recurrence for $T(2n)$ in terms of T at other even integers?*

6. Conjectures for even *k*. Figure 6.1 shows $T_{0,4}(n)$ with the usual initial values $(1, 1, 1, 1)$. Unlike when k is odd, we do not see a bifurcation into two intertwined subsequences; rather, the sequence stays close to the expected line $T = \frac{3}{4}n$.

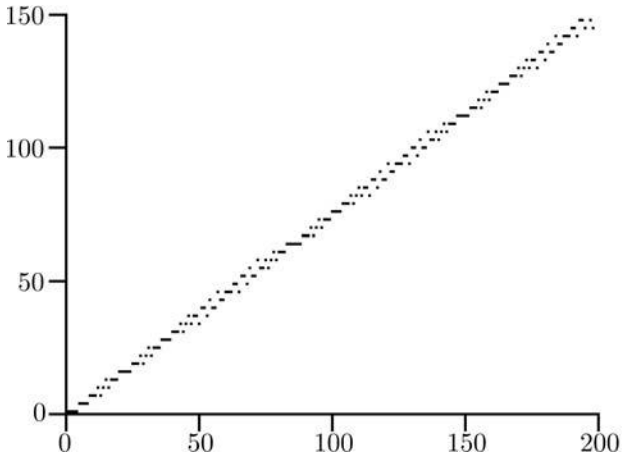
Nonetheless, because of Corollary 2.5, we know that the sequence $\Delta_3 U_{0,4}(n)$ has a block structure of runs of paeons separated by one or more zeros (resembling hypercatalectics). These zeros can appear singly or (unlike when k is odd) multiply: there are two zeros in a row at $n = 87$ and $n = 88$ following the paeon that runs from $n = 82$ to $n = 86$. Still, if we make a minor change in our definition of feet to allow for consecutive hypercatalectics, we can define feet, lines, and generations in a natural way that fits our empirical observations.

DEFINITION 6.1. *Let $a = 0$ and k be even. A* paeon *is a sequence*

$$\{k-1, \overbrace{0, \ldots, 0}^{k}\}$$

of $k+1$ consecutive values of $\Delta_{k-1}U(n)$. A hypercatalectic *is a singleton sequence $\{0\}$ in $\{\Delta_{k-1}U(n)\}$ that is not part of a paeon. A* foot *is either a paeon or a hypercatalectic. For convenience, we will write $\{\mathrm{P}\}$ interchangeably with the paeon $\{k-1, 0, \ldots, 0\}$ and likewise $\{H\}$ with the hypercatalectic $\{0\}$ when listing values of $\Delta_{k-1}U$. We also define $\varphi(n)$ as the symbol P if $\Delta_{k-1}U(n) = k-1$ begins a paeon, H if $\Delta_{k-1}U(n) = 0$ is a hypercatalectic, and leave it undefined otherwise. For any $g > 0$, let $m_g := \frac{1}{k-1}(k^{g+1} + k^2 - k - 1) = k + \sum_{i=0}^{g} k^i$ and call the interval $[m_g, m_{g+1} - 1]$ the* gth generation*, written as* gen(g)*. For any sequence $s(n)$, we will refer to the subsequence $\{s(n) \mid n \in \mathrm{gen}(g),\ s(n)$ defined$\}$ as the* gth generation of s *and similarly for semigenerations. An* even (odd) *foot is one that starts in an even (odd) semigeneration.*

We say that this definition is a natural one, because it allows some of the sequence



n	$T(n)$	n	$T(n)$
1	1	13	7
2	1	14	10
3	1	15	13
4	1	16	10
5	4	17	13
6	4	18	13
7	4	19	13
8	4	20	16
9	7	21	16
10	7	22	16
11	7	23	16
12	10	24	16

FIG. 6.1. *$T_{0,4}(n)$ with initial values $(1, 1, 1, 1)$.*

properties we observed earlier for odd k to continue to hold for even k. For example, there is always a run of equal values of $T(n)$ at the end of each generation, as can be seen in Figure 6.1 preceding $n = m_2 = 25$ and $n = m_3 = 89$.

In light of this definition, it is instructive to look back at the case $k = 2$ (Conolly's sequence), which is completely understood, in light of this definition. In its corresponding $\Delta_1 U$ sequence, beginning with the second generation at $n = m_2 = 9$, paeons always occur in pairs, and the number of consecutive hypercatalectics that occur between pairs of paeons forms the previously mentioned Gray binary sequence $1, 2, 1, 3, 1, 2, 1, 4, \ldots$, with an extra hypercatalectic at the end of each generation.

We believe that all of section 3 can be generalized to even k. We do not, however, observe palindromic symmetry or polypaeon structure, so not all of section 4 can be carried over to this case. For example, when $k = 4$, the first generation of $\Delta_2 U$ consists of the feet $\{P, P, P, H\}$; the second generation is $\{P, P, P, P, P, H, P, P, P, P, P, H, P, P, H, H\}$; the third generation begins with a copy of the second generation, includes a run of eight paeons, and ends with three consecutive hypercatalectics. None of the generations are palindromic, and consecutive paeons appear in varying and relatively prime numbers.

The following four conjectures state how we believe our results of section 3 will generalize to even k, based on empirical evidence.

CONJECTURE 6.2. *For $a = 0$, even k, and the usual initial conditions $T_{0,k}(n) = 1$ for $1 \le n \le k$, implicitly define $f(R(n)) := n$ for $n \in \operatorname{Dom} \varphi$. Then f is well defined on its domain and $f(r)$ is the least n for which $r = R(n)$.*

CONJECTURE 6.3 (a generalized version of Foot Pattern Theorem 5.6). *Let $a = 0$, k be even, and assume the usual initial conditions $T_{0,k}(n) = 1$ for $1 \le n \le k$. Suppose the parameters n_0, $t_0, \ldots, t_{k-2}$, $r_0, \ldots, r_{k-2}$ and d satisfy all of the following conditions: $n_0 \ge 2k + 1$, $\Delta_{k-1}U(n_0) = k - 1$, $\Delta_{k-1}U(n_0 + k + 1) = d$, and for $0 \le i \le k - 2$, $T(n_0 + i) = t_i$, and $R(n_0 + i) = r_i$. Then $T(n)$, $\Delta_{k-1}T(n)$, $R(n)$, $\Delta_{k-1}R(n)$, $\Delta_{k-1}U(n)$, and $\varphi(n)$ have the values shown in Table 5.1.*

CONJECTURE 6.4 (a generalized version of Generational Correspondence Theorem 3.14). *Let $a = 0$ and k be even. Then the diagram*

$$\begin{array}{ccc} \operatorname{Dom} f & \underset{R}{\overset{f}{\rightleftarrows}} & \operatorname{Ran} f \\ {\scriptstyle \Delta_{k-1}T}\big\downarrow & & \big\downarrow{\scriptstyle \varphi} \\ \{0, k-1\} & \underset{k-1 \mapsto P}{\overset{0 \mapsto H}{\longrightarrow}} & \{H, P\} \end{array}$$

commutes. That is, for $k + 1 \le r \in \operatorname{Dom} f$, $\varphi(f(r)) = P$ iff $\Delta_{k-1}T(r) = k - 1$.

CONJECTURE 6.5 (a generalized version of Generation Pattern Theorem 3.15). *Let $a = 0$ and k be even. Then the gth generation of $\Delta_{k-1}U$ consists entirely of k^g feet, which make up k^{g-1} lines. The generation ends with g hypercatalectics preceded by a paeon, and if we let $n_0 := m_{g+1} - k - 1$, then for $0 \le i \le k - 2$, $T(n_0 + i) = k^{g+1}$ and $R(n_0 + i) = m_g - k - 1 + i$.*

While we do not have palindromic symmetry when k is even, the existence of f appears to be sufficient to generalize the following sequence property observed in section 5.

CONJECTURE 6.6 (a generalized version of Proposition 5.15). *For $a = 0$ and any (not necessarily odd) k, the mean value of the gth generation of $\Delta_2 T(n)$ is $\frac{2(k-1)}{k}$. The mean value of the gth generation of $\Delta_2 U(n)$ is $\frac{2(k-1)}{k^2}$.*

As in the case of odd k described in Conjecture 5.18, it is also true for even k that maximal runs of identical values in the $T(n)$ sequence tend to occur at the ends of generations.

CONJECTURE 6.7 (a generalized version of Theorem 4.3). *For $a = 0$, even k and any g, $T(m_{g+1} - k - g) = \cdots = T(m_{g+1} - 1)$ and there are no earlier runs of $k + g$ identical consecutive values of $T(n)$.*

In a forthcoming communication, we hope to resolve the above conjectures in the broader context of the determination of the complete structure of our sequences for even k.

REFERENCES

[1] R. B. J. T. ALLENBY AND R. C. SMITH, *Some sequences resembling Hofstadter's*, J. Korean Math. Soc., 40 (2003), pp. 921–932.

[2] ARISTOTLE, *Ars Rhetorica*, Sir David Ross, ed., Oxford University Press, Oxford, 1959, pp. 158–159, Bekker index 1409a.

[3] B.W. CONOLLY, *Meta-Fibonacci sequences*, in Fibonacci & Lucas Numbers, and the Golden Section, S. Vajda, ed., Wiley, New York, 1986, pp. 127–137.

[4] C. H. ELZINGA, *private communication*, October 2001.

[5] J. HIGHAM AND S. TANNY, *More well-behaved meta-Fibonacci sequences*, Congr. Numer., 98 (1993), pp. 3–17.

[6] J. HIGHAM AND S. TANNY, *A tamely chaotic meta-Fibonacci sequence*, Congr. Numer., 99 (1994), pp. 67–94.

[7] D. HOFSTADTER, *Gödel, Escher, Bach: An Eternal Golden Braid*, Basic Books, New York, 1979.

[8] G. HUBER AND D. HOFSTADTER, *private communication*, 1999.

[9] G. HUBER, *private communication*, May 2002.

[10] T. KUBO AND R. VAKIL, *On Conway's recursive sequence*, Discrete Math., 152 (1996), pp. 225–252.

[11] I. LANCASHIRE, *Glossary of Poetic Terms (Version* 2.07*)*, Representative Poetry Online (Version 2.11), http://www.library.utoronto.ca/utel/rp/intro.html (1994).

[12] C. L. MALLOWS, *Conway's challenge sequence*, J. Amer. Math. Mon., 98 (1991), pp. 5–20.

[13] N. J. A. SLOANE, *Sloane's Encyclopedia of Integer Sequences*, http://www.research.att.com/∼njas/sequences.

[14] S. M. TANNY, *A well-behaved cousin of the Hofstadter sequence*, J. Discrete Math., 105 (1992), pp. 227–239.

SIAM J. DISCRETE MATH.
Vol. 18, No. 4, pp. 825–834

ON OPTIMAL EDGE-ROBUST AND VERTEX-ROBUST $(1, \leq l)$-IDENTIFYING CODES*

TERO LAIHONEN†

Abstract. The motivation for identifying codes comes from maintenance of multiprocessor architectures. In this paper, we give infinite families of optimal edge-robust identifying codes and vertex-robust identifying codes in binary Hamming spaces.

Key words. identifying code, multiprocessor system, optimal code, multiple covering, Hamming space

AMS subject classifications. 94C12, 94B65, 05C70

DOI. 10.1137/S0895480104440754

1. Introduction and preliminaries.

Assume that $G = (V, E)$ is an undirected and connected graph. We denote the *(graphic) distance* by $d(u, v)$; that is, $d(u, v)$ is the number of edges in any shortest path between $u \in V$ and $v \in V$. For $v \in V$, the *ball*

$$B_r(v) = \{u \in V \mid d(u, v) \leq r\}.$$

If $d(x, y) \leq 1$, we say that x and y *cover* each other. An edge $\{x, y\} \in E$ is denoted by xy, and we say that x (resp., y) is an *endvertex* of xy and that x (resp., y) is *incident* with xy.

A *code* is a nonempty subset of V, and its elements are called *codewords*.

Let C be a code. For all $X \subseteq V$, we denote

$$I_r(G, C; X) = I_r(X) := C \cap \left(\bigcup_{x \in X} B_r(x) \right).$$

This is called the *I-set* of X. Whenever convenient, we omit the arguments G and/or C from $I_r(G, C; X)$. In this paper, mostly $r = 1$, we then drop the subscript and write $I_r(X) = I(X)$. We also denote $I_r(v_1, \ldots, v_s) = I_r(\{v_1, \ldots, v_s\})$, where $v_i \in V$ for all $i = 1, \ldots, s$.

A code $C \subseteq V$ is called *$(r, \leq l)$-identifying* (in G) if the sets $I_r(G, C; X)$ are distinct for all $X \subseteq V$, where $|X| \leq l$.

Notice that if C is $(r, \leq l)$-identifying, then $I_r(X) = \emptyset$ if and only if $X = \emptyset$.

The concept of identifying codes was introduced by Karpovsky, Chakrabarty, and Levitin [11] in 1998. Next we describe an application [11] of identifying codes to fault diagnosis of multiprocessor architectures.

Suppose that each vertex of G contains a processor and that an edge is a communication link between two processors. Some of the processors (but at most l of them) can be malfunctioning, and we wish to locate them in the following way. We choose a code $C \subseteq V$, i.e., a subset of processors, and each codeword is assigned the following

*Received by the editors February 9, 2004; accepted for publication (in revised form) August 27, 2004; published electronically May 20, 2005. This research was supported by the Academy of Finland under grant 207303.

http://www.siam.org/journals/sidma/18-4/44075.html

†Department of Mathematics, University of Turku, 20014 Turku, Finland (terolai@utu.fi).

task. The codeword, say $c \in C$, checks all the processors in $B_r(c)$ and sends a single bit value "1" to the host if it detects any problems and "0" if everything is fine in the ball. Based on the bits coming from the codewords (in other words, knowing $I_r(X)$) the host must be able to determine where the faulty processors lie (i.e., determine X). This can be done if the codewords form an $(r, \leq l)$-identifying code. Of course, we would like to find a code with the smallest possible (i.e., *optimal*) cardinality.

Identification has been widely studied for such graphs G as the binary Hamming space (i.e., the binary hypercube), the square lattice, the triangular grid, the hexagonal mesh, and the king grid (see, e.g., [1, 2, 3, 5, 7, 8, 9, 10, 11, 12, 14, 15] and the references therein).

In [6], Honkala, Karpovsky, and Levitin considered codes that remain identifying although I-sets can be corrupted. They examined (for $l = 1$) the following extensions of identifying codes.

The symmetric difference $A \triangle B = (A \setminus B) \cup (B \setminus A)$. Denote by $\binom{V}{2}$ the set of unordered pairs of V.

DEFINITION 1 (see [11, 13]). *A code $C \subseteq V$ is t-edge-robust $(r, \leq l)$-identifying (in G) if C is $(r, \leq l)$-identifying in every graph $G' = (V, E')$, where $E' = E \triangle E^1$ and $E^1 \subseteq \binom{V}{2}$ has size at most t.*

In this case, some communication links can be deleted from E and some new links added to E. The total number of erased and added links must be together at most t.

Notice that the vertices covered by a vertex may change from G to G', and hence I-sets can alter.

DEFINITION 2. *A code $C \subseteq V$ is t-vertex-robust $(r, \leq l)$-identifying (in G) if for any two different sets $X, \Gamma \subseteq V$, where $|X|, |\Gamma| \leq l$, we have*

$$I_r(X) \triangle A \neq I_r(\Gamma) \triangle B$$

for any $A, B \subseteq C$ with $|A|, |B| \leq t$.

In this variant, some codewords can be missing from an I-set or some new added to it (together again at most t changes).

If $I_r(x) = \{c_1, \ldots, c_{2t}\}$ for $x \in V$, then choosing $\Gamma = \emptyset$, $A = \{c_1, \ldots, c_t\}$, and $B = \{c_{t+1}, \ldots, c_{2t}\}$, one obtains

$$I_r(x) \triangle A = \{c_{t+1}, \ldots, c_{2t}\} = I_r(\Gamma) \triangle B.$$

According to this, a t-vertex-robust $(r, \leq l)$-identifying code satisfies (if $l \geq 1$)

$$|I_r(x)| \geq 2t + 1 \tag{1.1}$$

for all $x \in V$.

Definition 2 (for $l = 1$) is slightly different from the following definition of [6] (and also different from the one in [16]): A subset $C \subseteq V$ is called a t-vertex-robust r-identifying code (in G) if $|I_r(v)| \geq t + 1$ for all $v \in V$ and if for all $u, v \in V$, $u \neq v$, and $A, B \subseteq C$ with $|A|, |B| \leq t$, we have $I_r(u) \triangle A \neq I_r(v) \triangle B$.

The advantage of Definition 2 is that we can always separate, by virtue of (1.1), the situation where there exist faulty processors from the situation where every processor is fine.

A code satisfying (only) the definition from [6] (or the definition of [16]) cannot distinguish, for example, between the cases $X = \{x\}$ (one faulty processor), where $I_r(x) = \{c_1, \ldots, c_{t+1}\}$, and $\Gamma = \emptyset$ (no faulty processors), because choosing $A = \{c_1, \ldots, c_t\}$ and $B = \{c_{t+1}\}$ gives $I_r(x) \triangle A = I_r(\emptyset) \triangle B$.

We consider in this paper exclusively binary Hamming spaces (i.e., binary hypercubes) defined as follows. Denote the binary field by $F = \{0, 1\}$. The vertex set (a vertex is also called a *word*) is the n-fold Cartesian product $F^n = F \times F \times \cdots \times F$ of F. The *Hamming distance* between $x \in F^n$ and $y \in F^n$ is the number of coordinate positions in which they differ. There exists an edge between two words if and only if the Hamming distance equals one. The set of edges is denoted by E_n. We denote the obtained graph by $G_n = (F^n, E_n)$. Notice that the Hamming distance and the graphic distance coincide in G_n. If $C \subseteq F^n$, then n is called the *length* of C.

In what follows, we will often utilize the following easy lemma. We denote $S_i(x) = \{y \in F^n \mid d(x, y) = i\}$, where the distance is *always* with respect to G_n.

LEMMA 1. *Consider the graph G_n.*

(i) *For $a, b \in F^n$ we have*

$$|B_1(a) \cap B_1(b)| = \begin{cases} n+1 & \text{if } a = b, \\ 2 & \text{if } d(a,b) = 1 \text{ or } 2, \\ 0 & \text{otherwise.} \end{cases}$$

(ii) *The intersection of three different balls of radius one consists of a unique point if the intersection is nonempty.*

(iii) *Let $x \in F^n$. If $a, b \in S_i(x)$, $a \neq b$, for some i, $0 < i < n-1$, then $B_1(a) \cap B_1(b)$, if nonempty, contains a unique point in $S_{i-1}(x)$ and in $S_{i+1}(x)$.*

(iv) *Let $x \in F^n$. If $a, b \in S_i(x)$, $a \neq b$ for some i, $0 < i \leq n$, then $|B_1(a) \cap S_{i-1}(x)| = i$. Moreover, $B_1(a) \cap B_1(b) \cap S_i(x) = \emptyset$.*

A code $C \subseteq F^n$ is called a *μ-fold r-covering* (with respect to G_n) if for every word x of F^n we have $|I_r(G_n, C; x)| \geq \mu$. For coverings consult, for instance, [4, Chapter 14].

THEOREM 1 (see [4, Theorem 14.2.4]). *A μ-fold 1-covering of length n and smallest possible cardinality $\mu \cdot 2^n/(n+1)$ exists if and only if there are integers $i \geq 0$, $\mu_0 > 0$ such that $\mu_0 \mid \mu$, $\mu \leq 2^i \mu_0$ and $n = \mu_0 2^i - 1$.*

In the next section, we give infinite sequences of optimal t-edge-robust $(1, \leq l)$-identifying codes for $l \geq 3$ and $t \geq 1$ and also for $l = 2$ and $t \geq 2$ (for the case $l = 2$ and $t = 1$, see [13]). If $l = 1$ and $t \geq 1$, no infinite optimal families are known. Interesting asymptotic behavior of the size of such codes is given in [6].

In the last section, infinite families of optimal t-vertex-robust $(1, \leq l)$-identifying codes are provided for $l \geq 2$ and $t \geq 1$ and for $l = 1$ and $t \geq 2$.

2. On edge-robust identification. We consider edge-robust identification in G_n. Theorem 2 is from [13]. Apart from this theorem the results of [13] are exclusively for $l = 2$ and $t = 1$. If $t = 1$, we can "fix" the added or deleted edge, but if $t > 1$ (as mainly in this paper), we use a different approach (see Fact 3 of Theorem 3 below).

THEOREM 2 (see [13]). *Let $l \geq 2$, $t \geq 1$, and $2l + t \leq n + 2$. Any t-edge-robust $(1, \leq l)$-identifying code $C \subseteq F^n$ is a $(2l + t - 1)$-fold 1-covering (with respect to G_n). Thus*

$$|C| \geq \left\lceil (2l + t - 1) \frac{2^n}{n+1} \right\rceil.$$

The next theorem shows that, if $l \geq 2$, then in many cases the reverse statement is also true.

THEOREM 3. *If*

$$l \geq 3 \quad \text{and} \quad t \geq 1$$

or

$$l = 2 \quad \textit{and} \quad t \geq 3,$$

then a $(2l + t - 1)$*-fold* 1*-covering* $C \subseteq F^n$ *(with respect to* G_n*) is also* t*-edge-robust* $(1, \leq l)$*-identifying (in* G_n*).*

Proof. Let $C \subseteq F^n$ be a $(2l+t-1)$-fold 1-covering. In order to see that C is also t-edge-robust $(1, \leq l)$-identifying, we need to show that $I(G; X) \neq I(G; \Gamma)$ for any two distinct sets $X, \Gamma \subseteq F^n$ of cardinality at most l each, where G is G_n or any G'_n—here G'_n is obtained from G_n by adding and/or deleting together at most t edges. Assume to the contrary that $I(G; X) = I(G; \Gamma)$.

Without loss of generality we can assume $|X| \geq |\Gamma|$. Thus we have $x \in X$ such that $x \notin \Gamma$. Let us denote the changes in edges from G_n to G as in Definition 1 by E_n^1. We shall next show that our assumption $I(G; X) = I(G; \Gamma)$ implies the following.

Fact 1. $I(G_n; x) \cap I(G_n; \gamma) \neq \emptyset$ for all $\gamma \in \Gamma$.

Fact 2. $|X| = |\Gamma| = l$, and $2l + t - 1 \leq |I(G_n; x)| \leq 2l + t$.

Fact 3. There exist at least $t - 1$ edges in E_n^1 such that each one of them has as an endvertex an element of $I(G_n; x) \setminus I(G_n; \Gamma)$. The other endvertex belongs to $\Gamma \cup \{x\}$. We denote this set of at least $t - 1$ edges by P. Since $|E_n^1| \leq t$, there is at most one edge e not having such endvertices, i.e., $e \in E_n^1 \setminus P$.

We know that $|I(G_n, C; x)| \geq 2l + t - 1$ and, trivially, $I(G; x) \subseteq I(G; X)$. With the aid of E_n any element of Γ can cover (see Lemma 1(i)) at most 2 codewords of $I(G_n; x)$. Since $|\Gamma| \leq l$, there are at least $t - 1$ codewords left in $I(G_n; x) \setminus I(G_n; \Gamma)$. However, $I(G; X) = I(G; \Gamma)$, so each codeword $c \in I(G_n; x) \setminus I(G_n; \Gamma)$ must be an endvertex of an edge in E_n^1 such that if c belongs to $I(G; \Gamma)$ (resp., does not belong to $I(G; \Gamma)$), then Γ contains the other endvertex (resp., the other endvertex equals x) and the edge is added to (resp., deleted from) E_n. This gives Fact 3.

If there is $\gamma \in \Gamma$ for which $I(G_n; x) \cap I(G_n; \gamma) = \emptyset$ or if $|\Gamma| < l$ or if $|I(G_n; x)| \geq 2l + t + 1$, then $I(G; x)$ contains at least one codeword not in $I(G; \Gamma)$. Indeed, there exist at least $t+1$ codewords in $I(G_n; x) \setminus I(G_n; \Gamma)$, but $|E_n^1| \leq t$. This gives the first two facts.

By Fact 1, we know that $\Gamma \subseteq S_1(x) \cup S_2(x)$. Recall that the notation $S_i(x)$ is with respect to G_n. We denote $S_i = S_i(x)$ and $S_{\geq k} = \cup_{i \geq k} S_i$.

If X contains an element $y \in S_{\geq 3}$, then always $I(G; X) \neq I(G; \Gamma)$. This can be seen as follows. Obviously, then $|I(G_n, C; y) \cap S_{\geq 4}| \geq 2l + t - 5$ by Lemma 1(iv). Since $\Gamma \subseteq S_1 \cup S_2$, its elements do not cover, using E_n, any of these codewords. By Fact 3 we conclude that also none of P is incident with these codewords. The possible edge $e \in E_n^1 \setminus P$ can either add at most one of the codewords to $I(G; \Gamma)$ or remove at most one from $I(G_n; y)$. Therefore, $I(G; y)$ contains at least $2l + t - 6$ codewords which do not belong to $I(G; \Gamma)$. Since the assumptions of the theorem give $2l + t - 6 \geq 1$, we can assume that $X \setminus \{x\}$ is a subset of $S_1 \cup S_2$.

Fact 2 implies the existence of an element $\alpha \in \Gamma \setminus X$ and we know $\alpha \in S_1 \cup S_2$.

(i) First, let $\alpha \in S_2$. Then $|I(G_n; \alpha) \cap S_3| \geq 2l + t - 4$ by Lemma 1(iv).

We know that $X \setminus \{x\} \subseteq S_1 \cup S_2$. If $y \in X \cap S_2$, we get by Lemma 1(iii) that it can cover in G_n at most one of the codewords of $I(G_n; \alpha) \cap S_3$ because $y \neq \alpha$. If $y \in S_1$, it cannot cover, using E_n, any such codewords. Fact 3 implies that none of the edges in P is incident with an element of $I(G_n; \alpha) \cap S_3$. Moreover, the one possible edge $e \in E_n^1 \setminus P$ can remove (resp., add) at most one of the elements from $(I(G_n; \alpha) \cap S_3) \setminus I(G_n; X)$ (resp., to $I(G; X)$). Thus there are at least $(2l + t - 4) - 1 - |X \setminus \{x\}| = l + t - 4$ codewords left in $I(G; \alpha) \cap S_3 \subseteq I(G; \Gamma)$ which are not contained in $I(G; X)$. Apart

from the case $l = 3$ and $t = 1$, the assumptions of the theorem yield $l + t - 4 \geq 1$ and we are done.

Let $l = 3$ and $t = 1$. Denote $X = \{x, y, z\}$. If $\alpha \notin C$, then $I(G_n; \alpha) \cap S_3$ contains at least four codewords, and we are done as above. Let $\alpha \in C$. We can write $I(G_n; \alpha) \cap S_3 = \{c_1, c_2, c_3\}$. Without loss of generality, we may assume that $c_1 \in I(G_n, y)$, $c_2 \in I(G_n, z)$ and that the only edge e in E_n^1 belongs to the set $\{c_3 v \mid v \in X\}$ (if it is added) or equals αc_3 (if it is deleted). However, now $y, z \in S_2$ and hence $\alpha \in I(G; \Gamma)$ do not belong to $I(G; X)$.

(ii) Let $\alpha \in S_1$. Then $I(G_n; \alpha) \cap S_2$ contains at least $2l + t - 3$ codewords (see Lemma 1(iv) again).

An element in $X \setminus \{x\} \subseteq S_1 \cup S_2$ covers, using E_n, at most one of these codewords. There can be at most one codeword in $I(G_n; \alpha) \cap S_2$ which is incident with an edge (or edges) in P. Indeed, by Fact 3, such codewords must belong to Γ and, if there are at least two of them, then Γ covers, using E_n, at most $2l - 2$ codewords of $I(G_n; x)$ and, since $|E_n^1| \leq t$, there is at least one element in $I(G; X) \setminus I(G; \Gamma)$ because $(2l - 2) + t < 2l + t - 1 \leq |I(G_n; x)|$. Thus the edges of P and $e \in E_n^1 \setminus P$ can add or remove together at most two elements (and at most one if $t = 1$) from $(I(G_n; \alpha) \cap S_2) \setminus I(G_n; X)$. Unless $l = 3$ and $t = 1$, the assumptions of the theorem give $(2l + t - 3) - 2 - |X \setminus \{x\}| \geq l + t - 4 \geq 1$ and we are done. If $l = 3$ and $t = 1$, then $(2l + t - 3) - 1 - |X \setminus \{x\}| \geq l + t - 3 \geq 1$. Hence always $I(G; X) \neq I(G; \Gamma)$. □

The following infinite families of optimal t-edge-robust $(1, \leq l)$-identifying codes are obtained by applying Theorem 1 to the previous theorem and Theorem 2.

THEOREM 4. *Let*

$$l \geq 3 \quad \textit{and} \quad t \geq 1$$

or

$$l = 2 \quad \textit{and} \quad t \geq 3.$$

The smallest possible cardinality of a t-edge-robust $(1, \leq l)$-identifying code equals

$$(2l + t - 1)\frac{2^n}{n + 1}$$

for the lengths $n = \mu_0 2^i - 1$, where the integers $i \geq 0$ and $\mu_0 > 0$ satisfy $\mu_0 \mid (2l + t - 1)$ and $2l + t - 1 \leq 2^i \mu_0$.

If $l = 2$ and $t = 2$, then all 5-fold 1-coverings are *not* 2-edge-robust $(1, \leq 2)$-identifying. This case is considered next.

THEOREM 5. *Suppose $C \subseteq F^n$ is a 5-fold 1-covering for which there do not exist four distinct codewords x', y', α', and β' such that*

$$\begin{aligned} I(G_n, C; x') &= \{x', c_1, c_2, c_3, c_4\}, \\ I(G_n, C; y') &= \{y', c_5, c_6, c_7, c_8\}, \\ I(G_n, C; \alpha') &= \{\alpha', c_1, c_2, c_5, c_6\}, \ \textit{and} \\ I(G_n, C; \beta') &= \{\beta', c_3, c_4, c_7, c_8\}. \end{aligned} \tag{2.1}$$

Then C is 2-edge-robust $(1, \leq 2)$-identifying.

Proof. Let C be a code satisfying the assumptions of the theorem. We need to verify that $I(G; X) \neq I(G; \Gamma)$ for all $X, \Gamma \subseteq F^n$ ($X \neq \Gamma$, $|X| \leq 2$, $|\Gamma| \leq 2$), where G is G_n or any G_n' where at most two edges have been deleted from or added to E_n or

one is added and another deleted. Assume to the contrary that $I(G;X) = I(G;\Gamma)$. Facts 1 to 3 of Theorem 3 are again valid (where $l = t = 2$). By Fact 2, we can assume $X = \{x, y\}$, $x \neq y$, and $\Gamma = \{\alpha, \beta\}$, $\alpha \neq \beta$. Let $x \in X \setminus \Gamma$ and $\alpha \in \Gamma \setminus X$. Fact 1 implies again $\Gamma \subseteq S_1 \cup S_2$ (recall the notations S_i and $S_{\geq k}$ from the proof of Theorem 3). Furthermore, we denote a set of edges incident with w and having the other endvertex in $A \subseteq F^n$ by $wA := \{wv \mid v \in A\}$.

If $y \in S_3 \cup S_4$, then without loss of generality we can assume that $I(G_n; y) \cap S_{\geq 4} = \{a\}$ for some $a \in C$ (a can be equal to y). Indeed, Γ covers nothing in $S_{\geq 4}$ using E_n, the edges in P are not incident with words in $S_{\geq 4}$, and $e \in E_n^1 \setminus P$ can remove (resp., add) at most one element from $I(G_n; y) \cap S_{\geq 4}$ (resp., to $I(G;\Gamma)$). In addition, by Lemma 1(iv), $|I(G_n; y) \cap S_{\geq 4}| \geq 1$. Consequently, a is an endvertex of $e \in E_n^1 \setminus P$ and

$$\begin{array}{ll} e = ay & \text{if } e \text{ is deleted from } E_n,\ a \neq y, \\ e \in a\Gamma & \text{if } e \text{ is added to } E_n. \end{array} \tag{2.2}$$

We examine separately the cases $\alpha \in S_2$ and $\alpha \in S_1$.

Case 1. Assume first that $\alpha \in S_2$. Since $I(G_n; \alpha) \cap S_3$ contains at least two codewords by Lemma 1(iv), y must cover, using E_n ,at least one of them, because e can either add at most one element to $I(G;X)$ or remove at most one from $I(G_n; \alpha) \cap S_3$ (and P none). Thus $y \in S_2 \cup S_3 \cup S_4$.

(i) Suppose that y does not cover, using E_n, all the codewords of $I(G_n; \alpha) \cap S_3$; say c is not covered. Since $I(G;X) = I(G;\Gamma)$, necessarily c is incident with e (notice that an edge in P cannot be incident with c).

If $y \in S_3 \cup S_4$, then $e = ac$ by (2.2). However, $c \neq y$ and $c \notin \Gamma$, a contradiction.

Let $y \in S_2$. If $\alpha \notin C$, then $|I(G_n; \alpha) \cap S_3| \geq 3$ but y can cover, using E_n, at most one by Lemma 1(iii), and e can either add or remove at most one (and P none), so we are done. Thus $\alpha \in C$. Neither y nor x can cover α using E_n. In addition, $e = \alpha c$ or $e \in cX$ so it does not help. Therefore, $f \in P$ (now $P = \{f\}$) is incident with α, and further, the only choice is $f = \alpha x$. If $y = \beta$, then by Lemma 1(iii) the words α and β cover, using E_n, a common word in S_1 (the other common word is in S_3) and two others in S_1 by Lemma 1(iv). But $|I(G_n; x)| \geq 5$, so $I(G; x)$ contains an element not in $I(G;\Gamma)$. Let $y \neq \beta$. Then α covers, using $E_n' = E_n \triangle E_n^1$, at most two words of $I(G_n; y)$, and f and e do not help β to cover the at least three codewords left in $I(G_n; y)$ which now lie also in $I(G;X)$; so β can use only E_n, and thus by Lemma 1(ii) we obtain $y = \beta$, which is a contradiction.

(ii) Assume next that y covers all of the codewords in $I(G_n; \alpha) \cap S_3$ using E_n. Because $y \neq \alpha$, we obtain by Lemma 1(ii) that $I(G_n; \alpha) \cap S_3$ consists of two elements, say c_5 and c_6. Furthermore, $y \in S_4$ by Lemma 1(iii). Thus (2.2) is valid and, moreover, $I(G_n; y) \cap S_3$ consists of four elements (notice that more than four is not possible according to Lemma 1(iv)). Hence we write $I(G_n; y) = \{a, c_5, c_6, c_7, c_8\}$ for some $c_7, c_8 \in C \cap S_3$. Since neither e nor $f \in P$ is incident with a word in S_3, in particular, not with c_7 and c_8, they must be covered by β using E_n. Hence, $\beta \in S_2$ and the set $I(G_n; \beta) \cap S_3$ consists of c_7 and c_8 (we are done if there are more than two codewords in the set). This implies that $\beta \in C$ and that there exist two codewords, say c_3 and c_4, in $I(G_n; \beta) \cap S_1$. We can also assume that $|I(G_n; \alpha) \cap S_3| = 2$ and, similarly, $\alpha \in C$ and there exist two codewords, say c_1 and c_2, in $I(G_n; \alpha) \cap S_1$. Therefore, $\{c_1, c_2, c_3, c_4\} \subseteq I(G_n; x)$. Because $\alpha, \beta \notin I(G_n; X)$ (but $\alpha, \beta \in I(G;\Gamma)$) and both e and f can be incident with at most one of α and β, we can assume, by symmetry, that α is an endvertex of f and that β is an endvertex of e. Consequently, $e = y\beta$ and

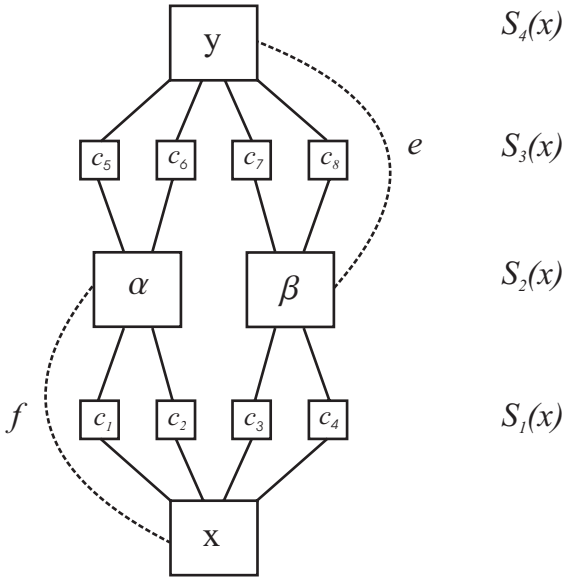


FIG. 1. *The solid lines denote edges in E_n and the dashed ones edges in E_n^1.*

$f = x\alpha$ or we are done. Thus $y = a$. Furthermore, it suffices to assume that $I(G_n; x) = \{x, c_1, c_2, c_3, c_4\}$. Now $I(G_n; \alpha) = \{\alpha, c_1, c_2, c_5, c_6\}$ and $I(G_n; \beta) = \{\beta, c_3, c_4, c_7, c_8\}$, so (see Figure 1) the assumption of the theorem gives the contradiction (setting $x = x'$, $y = y'$ and so on).

Case 2. Assume that $\alpha \in S_1$.

Suppose first that $\beta \notin I(G_n; \alpha) \cap S_2$. Then an edge of P is not incident with any word of $I(G_n; \alpha) \cap S_2$ and so only e can add or remove one of the at least three codewords in $I(G_n; \alpha) \cap S_2$. Thus y must cover, using E_n, two of them (more is not possible) giving $y \in S_3$ and one element in $(I(G_n; \alpha) \cap S_2) \setminus I(G_n; y)$, say c, must be incident with e. However, this is not possible since (2.2) applies and $c \neq y$ and $c \notin \Gamma$.

Assume finally that $\beta \in I(G_n; \alpha) \cap S_2$. Then $P = E_n^1$ because there exist at least two elements in $I(G_n; x) \setminus I(G_n, \Gamma)$. Thus at most one of the at least three elements of $I(G_n; \alpha) \cap S_2$ can be incident with an edge of P and we get again $y \in S_3$, but this would require an edge satisfying (2.2), which does not exist due to the fact that $P = E_n^1$. □

Example 1. Let H be the $[7, 4, 3]$-Hamming code [4, p. 8]. It is easy to check (with a computer) that the code

$$H \cup (1000000 + H) \cup (0100000 + H) \cup (0010000 + H) \cup (0001000 + H)$$

in F^7 satisfies the conditions of the previous theorem and so it is 2-edge-robust $(1, \leq 2)$-identifying. The cardinality 80 is optimal by Theorem 2.

Next we consider a construction (cf. [4, p. 67]), which gives us an infinite sequence of optimal 2-edge-robust $(1, \leq 2)$-identifying codes starting from the code of the example. Let $\pi(u)$ equal 0 (resp., 1) if the number of ones in $u \in F^n$ is even (resp., odd).

THEOREM 6. *Let $C \subseteq F^n$ be 2-edge-robust $(1, \leq 2)$-identifying in G_n. Then $D = \{(\pi(u), u, u + c) \mid u \in F^n, c \in C\} \subseteq F^{2n+1}$ is 2-edge-robust $(1, \leq 2)$-identifying in G_{2n+1}.*

Proof. We show that the code $D \subseteq F^{2n+1}$ is 2-edge-robust $(1, \leq 2)$-identifying by checking the conditions of the previous theorem. Because C is a 5-fold 1-covering, we know, by [4, Theorems 3.4.3 and 14.4.3], that the code D is as well. Assume that there exist four codewords $x, y, \alpha, \beta \in D$ such that (2.1) is satisfied, i.e., $I(G_{2n+1}, D; x) = \{x, c_1, c_2, c_3, c_4\}$, $I(G_{2n+1}, D; y) = \{y, c_5, c_6, c_7, c_8\}$, $I(G_{2n+1}, D; \alpha) = \{\alpha, c_1, c_2, c_5, c_6\}$, and $I(G_{2n+1}, D; \beta) = \{\beta, c_3, c_4, c_7, c_8\}$.

For a codeword $a \in D$ we use the notation $a = (\pi(u_a), u_a, u_a + c_a)$, where $c_a \in C$. Obviously, for any $a \in D$,

$$I(G_{2n+1}, D; a) = \{(\pi(u_a), u_a, u_a + c) \mid c \in C,\ d(c, c_a) \leq 1\},$$

where the distance is with respect to G_n. Note that the first $n+1$ coordinates are the same in all of the words of $I(G_{2n+1}, D; a)$.

Now $|I(G_{2n+1}, D; x) \cap I(G_{2n+1}, D; \alpha)| = 2$. Therefore, looking at the first $n+1$ coordinates in these I-sets of x and α, we obtain $u_x = u_\alpha$. Analogously, $u_x = u_\beta$ and $u_y = u_\beta$. Since $\alpha \neq x \neq \beta \neq y$, we must have $c_\alpha \neq c_x \neq c_\beta$ and $c_y \neq c_\beta$. But we shall see next that this leads to a contradiction. To this end, we observe first that if $I(G_{2n+1}, D; a) = \{(b_i, t_i, s_i) \mid i = 1, \ldots, k\}$, where $a \in D$, then $I(G_n, C; c_a) = \{t_i + s_i \mid i = 1, \ldots, k\}$. Applying this, we obtain

$$I(G, C; \{c_x, c_y\}) = I(G, C; \{c_\alpha, c_\beta\}),$$

where G is obtained from G_n adding to E_n two edges $c_x c_\alpha$ and $c_y c_\beta$. Because C is 2-edge-robust $(1, \leq 2)$-identifying, necessarily $\{c_x, c_y\} = \{c_\alpha, c_\beta\}$. But now $c_x = c_\alpha$ or $c_x = c_\beta$, a contradiction. This completes the proof. □

Combining the previous theorem and the example with Theorem 2, we obtain the following result.

COROLLARY 2.1. *The optimal cardinality of a 2-edge-robust $(1, \leq 2)$-identifying code of length $2^r - 1$, $r \geq 3$, equals $5 \cdot 2^{2^r - r - 1}$.*

3. On vertex-robust identification. Next we consider vertex-robust identification in G_n. Notice that now the underlying graph does not change.

THEOREM 7. *Let*

$$l \geq 2 \text{ and } t \geq 1$$

or

$$l = 1 \text{ and } t \geq 2.$$

Furthermore, let $2l + 2t \leq n + 2$. A code $C \subseteq F^n$ is t-vertex-robust $(1, \leq l)$-identifying (in G_n) if and only if it is a $(2l + 2t - 1)$-fold 1-covering.

Proof. ($\Rightarrow$) Let C be t-vertex-robust $(1, \leq l)$-identifying. By (1.1), the cases $l = 1$ and $t \geq 2$ are immediately clear.

Then let $l \geq 2$ and $t \geq 1$. Suppose that there is a word $x \in F^n$ such that $|I(x)| \leq 2l + 2t - 2$. Let p_i's $(i = 1, \ldots, 2l + 2t - 2)$ be distinct elements of $B_1(x)$ such that $I(x) \subseteq \{p_i \mid i = 1, \ldots, 2l + 2t - 2\}$ and $p_{2l+2t-2} = x$ if $x \in I(x)$, and $p_i \in S_1(x)$ for all $i = 1, \ldots, 2l + 2t - 2$ if $x \notin I(x)$. Denote $y_i = x + p_{2i-1} + p_{2i}$ for $i = 1, \ldots, l - 1$. Thus y_i $(\neq x)$ covers p_{2i-1} and p_{2i}. Let $A = \{p_i \mid i = 2l - 1, \ldots, 2l + t - 2\}$ and $B = \{p_i \mid i = 2l + t - 1, \ldots, 2l + 2t - 2\}$. Then

$$I(y_1, \ldots, y_{l-1}) \triangle (A \cap C) = I(y_1, \ldots, y_{l-1}, x) \triangle (B \cap C),$$

which is not allowed because $|A| = |B| = t$. Thus $|I(x)| \geq 2l + 2t - 1$ for all $x \in F^n$, and C is a $(2l + 2t - 1)$-fold 1-covering.

($\Leftarrow$) Let C be a $(2l + 2t - 1)$-fold 1-covering. Suppose that

$$I(X) \triangle A = I(\Gamma) \triangle B \tag{3.1}$$

for two distinct sets $X, \Gamma \subseteq F^n$ of cardinality at most l each, where $A, B \subseteq C$ and $|A|, |B| \leq t$. Without loss of generality, we can assume that $|X| \geq |\Gamma|$ and $x \in X \setminus \Gamma$. Denote again S_i and $S_{\geq k}$ as in the proof of Theorem 3.

By Lemma 1(i), the set $I(x)$ contains at least $2t - 1$ codewords which do not belong to $I(\Gamma)$. By virtue of (3.1), we immediately see that $A \cup B$ contains these codewords and, further, $|A| = t$ and $|B| \geq t - 1$, or symmetrically, $|B| = t$ and $|A| \geq t - 1$. Thus $|(A \cup B) \cap S_{\geq 2}| \leq 1$. Furthermore, $\Gamma \subseteq S_1 \cup S_2$ and $|\Gamma| = |X| = l$ or else $I(X)$ contains at least $2t + 1$ codewords not in $I(\Gamma)$, which is impossible because $|A|, |B| \leq t$. Let $\alpha \in \Gamma \setminus X$.

All other cases except $l = 2$ and $t = 1$: First we show that $X \cap S_{\geq 3} = \emptyset$. For $l = 1$ this is trivial, so let $l \geq 3$ and $t \geq 1$ or $l = 2$ and $t \geq 2$. If $y \in S_{\geq 3}$, then $|I(y) \cap S_{\geq 4}| \geq 2l + 2t - 5$ by Lemma 1(iv) and none of them belongs to $I(\Gamma)$. In addition, $A \cup B$ can contain at most one of these codewords. Hence $I(X) \triangle A$ contains a codeword not in $I(\Gamma) \triangle B$, because $2l + 2t \geq 7$. Thus $X \setminus \{x\} \subseteq S_1 \cup S_2$.

Consequently, any element of $X \setminus \{x\}$ covers at most one of $I(\alpha) \cap S_2$ (resp., $I(\alpha) \cap S_3$) if $\alpha \in S_1$ (resp., if $\alpha \in S_2$). Because $|(A \cup B) \cap S_{\geq 2}| \leq 1$ and $l + 2t \geq 5$, the set $I(\Gamma) \triangle B$ contains at least one codeword not in $I(X) \triangle A$. Thus (3.1) cannot hold.

The case $l = 2$ and $t = 1$: Let $X = \{x, y\}$. If $y \in S_{\geq 3}$, then it is enough to assume that $I(y) \cap S_{\geq 4} = \{c\} \subseteq A \cup B$.

Let $|I(\alpha) \cap S_2| \geq 3$ (resp., $|I(\alpha) \cap S_3| \geq 3$) for $\alpha \in S_1$ (resp., $\alpha \in S_2$). By Lemma 1(i), y can cover at most two of these words, so there exists c' among these at least three codewords such that $c' \notin I(X)$ and hence $c' \in A \cup B$. In addition, y must cover more than one of the at least three codewords and therefore $y \in S_{\geq 3}$. But now $c \neq c'$ gives the contradiction.

Since $|I(\alpha) \cap S_2| \geq 3$ always for $\alpha \in S_1$, it suffices to check the case $|I(\alpha) \cap S_3| = 2$ (less than two is not possible) for $\alpha \in S_2$. Consequently, $\alpha \in C$. Denote $\{c_1, c_2\} = I(\alpha) \cap S_3$. Since $|(A \cup B) \cap S_{\geq 2}| \leq 1$, the word y must cover at least one of c_1 and c_2. Suppose first that c_1 is not covered by y. Evidently, $c_1 \in A \cup B$. Now y must cover c_2 and α. Therefore, it suffices to assume that $y = c_2$. But then $y \in S_3$. Now $c \neq c_1$ leads to a contradiction. Assume next that y covers both c_1 and c_2, which requires that $y \in S_4$. Necessarily, $\alpha \in A \cup B$, but $c \neq \alpha$, so (3.1) is not possible.

Therefore, C is t-vertex-robust $(1, \leq l)$-identifying in G_n for the given parameters t and l. □

Combining the previous theorem with Theorem 1, we get the following.

THEOREM 8. *Let*

$$l \geq 2 \text{ and } t \geq 1$$

or

$$l = 1 \text{ and } t \geq 2.$$

The smallest possible cardinality of a t-vertex-robust $(1, \leq l)$-identifying code equals

$$(2l + 2t - 1)\frac{2^n}{n + 1}$$

for the lengths $n = \mu_0 2^i - 1$, *where the integers* $i \geq 0$, $\mu_0 > 0$ *are such that* $\mu_0 \mid (2l + 2t - 1)$ *and* $2l + 2t - 1 \leq \mu_0 2^i$.

The case $l = 1$ and $t = 1$ is different in its nature and will not be treated in this paper.

Acknowledgments. The author would like to thank Iiro Honkala for useful discussions, and a referee for the remarks.

REFERENCES

[1] U. Blass, I. Honkala, and S. Litsyn, *Bounds on identifying codes*, Discrete Math., 241 (2001), pp. 119–128.

[2] U. Blass, I. Honkala, and S. Litsyn, *On binary codes for identification*, J. Combin. Des., 8 (2000), pp. 151–156.

[3] I. Charon, I. Honkala, O. Hudry and A. Lobstein, *General bounds for identifying codes in some infinite regular graphs*, Electron. J. Combin., 8 (2001), R39.

[4] G. Cohen, I. Honkala, S. Litsyn, and A. Lobstein, *Covering Codes*, North-Holland, Amsterdam, 1997.

[5] G. D. Cohen, I. Honkala, A. Lobstein, and G. Zémor, *On identifying codes*, in Codes and Association Schemes, A. Barg and S. Litsyn, eds., DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 56, AMS, Providence, RI, 2001, pp. 97–109.

[6] I. Honkala, M. Karpovsky, and L. Levitin, *On robust identifying codes*, IEEE Trans. Inform. Theory, submitted.

[7] I. Honkala and T. Laihonen, *On identifying codes in the triangular and square grids*, SIAM J. Comput., 33 (2004), pp. 304–312.

[8] I. Honkala, T. Laihonen, and S. Ranto, *On codes identifying sets of vertices in Hamming spaces*, Des. Codes Cryptogr., 24 (2001), pp. 193–204.

[9] I. Honkala and A. Lobstein, *On identifying codes in binary Hamming spaces*, J. Combin. Theory, Ser. A, 99 (2002), pp. 232–243.

[10] I. Honkala and A. Lobstein, *On the complexity of the identification problem in Hamming spaces*, Acta Inform., 38 (2002), pp. 839–845.

[11] M. Karpovsky, K. Chakrabarty, and L. Levitin, *On a new class of codes for identifying vertices in graphs*, IEEE Trans. Inform. Theory, 44 (1998), pp. 599–611.

[12] T. Laihonen, *Sequences of optimal identifying codes*, IEEE Trans. Inform. Theory, 48 (2002), pp. 774–776.

[13] T. Laihonen, *On edge-robust* $(1, \leq l)$*-identifying codes in binary Hamming spaces*, submitted.

[14] S. Ranto, *Optimal linear identifying codes*, IEEE Trans. Inform. Theory, 49 (2003), pp. 1544–1547.

[15] S. Ranto, I. Honkala, and T. Laihonen, *Two families of optimal identifying codes in binary Hamming spaces*, IEEE Trans. Inform. Theory, 48 (2002), pp. 1200–1203.

[16] S. Ray, R. Ungrangsi, F. De Pellegrini, A. Trachtenberg, and D. Starobinski, *Robust location detection in emergency sensor networks*, in Proceedings of INFOCOM 2003, San Francisco, 2003.

SIAM J. DISCRETE MATH.
Vol. 18, No. 4, pp. 835–843

FOUR CHARACTERS SUFFICE TO CONVEXLY DEFINE A PHYLOGENETIC TREE*

KATHARINA T. HUBER†, VINCENT MOULTON†, AND MIKE STEEL‡

Abstract. It was recently shown that just five characters (functions on a finite set X) suffice to convexly define a trivalent tree with leaf set X. Here we show that four characters suffice which, since three characters are not enough in general, is the best possible.

Key words. phylogenetic tree, X-tree, convexly define, display, semidyadic closure, character compatibility

AMS subject classifications. 92B15, 92B10, 05C05

DOI. 10.1137/S0895480102416696

1. Introduction. The field of *phylogenetics* compares observable characteristics of (biological) species in order to reconstruct and analyze their evolutionary history. Generally this history is represented by a tree, with leaves labeled by the species. If each of the comparisons between the species involve just two possible character states (for example, "wings" vs. "no-wings") and each state has evolved only once, then there is a direct equivalence between such data and leaf-labeled trees. This equivalence was described by Peter Buneman in his classic paper [4] from 1971. More recently there has been considerable interest, from both computer scientists and mathematicians, in extending these results to data in which there may be many character states—so-called "multistate characters" [1], [7], [8], [10]. Recent whole genome data has given rise to extensive data sets of multistate characters, often with a large number of states (such as those obtained by comparing gene order between species).

This leads to the natural question of how many multistate characters are required to completely determine an underlying evolutionary tree, under the assumption that each state has evolved just once. In a surprising result, the authors of [10] recently showed that just *five* such characters suffice, regardless of the number of species (we describe this result more precisely in section 5). Their result applied a graph-theoretic argument involving chordal graphs to a specific edge-coloring of trees based on the cyclic group of order 5. However, the tantalizing question of whether this five character result could be improved to four characters was left as a posed problem [10, Problem 6.2], as the methods used in that paper did not seem to readily apply.

In this paper we employ a different approach, and show that four characters are indeed sufficient, a result that is optimal since three characters are not sufficient to completely define all trees [10]. We reproduce the tree topology in [10] that illustrates four as lower bound for Figure 1. In particular, we describe an edge-coloring of a tree

*Received by the editors October 29, 2002; accepted for publication (in revised form) August 31, 2004; published electronically May 20, 2005. This research was supported by The Swedish Foundation for International Cooperation in Research and Education (STINT).

http://www.siam.org/journals/sidma/18-4/41669.html

†School of Computing Sciences, University of East Anglia, Norwich, NR4 7TJ, UK (katharina.huber@cmp.uea.ac.uk, vincent.moulton@cmp.uea.ac.uk). The research of these authors presented in this paper was supported by The Linnaeus Centre for Bioinformatics, Uppsala University, Sweden, where the results were established, and The Swedish Research Council (VR).

‡Biomathematics Research Centre, University of Canterbury, Box 4800, Christchurch, New Zealand (m.steel@math.canterbury.ac.nz). The research of this author was supported by the New Zealand Marsden Fund.

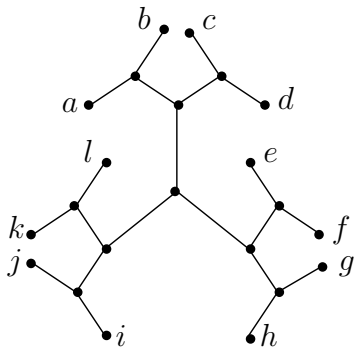


FIG. 1. *The figure depicts an evolutionary tree on the set* $X = \{a, b, \ldots, l\}$ *that cannot be convexly defined by three characters (see* [10] *for more details).*

using four colors, which induces characters in the same way as the edge coloration using five colors in [10]. To establish that the induced characters can be used to completely reconstruct the tree, we consider a set of small subtrees (each with four leaves) that are generated by the edge-coloring, and show that these subtrees determine the tree. This then allows us to establish that the characters induced by the edge-coloring determine the underlying tree.

The structure of this paper is as follows. In section 2, we introduce some terminology for trees and describe a closure operation on subtrees. Next, in section 3, we describe an edge-coloring of trees that produces subtrees on which this closure operation is applied. In section 4, we establish our main technical tool (Theorem 4.1), and in section 5, we use this result to show that four characters suffice to completely reconstruct a tree (Theorem 5.2).

2. Quartet trees and semidyadic closure. Throughout the paper, X denotes a nonempty finite set and $n = |X|$. A *phylogenetic tree (on X)* is a tree $\mathcal{T}$ that has X as its set of labeled leaves and interior vertices that are unlabeled and of degree at least three. If each interior vertex has degree exactly three, we say that $\mathcal{T}$ is *trivalent*. Two phylogenetic trees for X are *isomorphic* if the identity map on X induces a graph isomorphism on the underlying tree.

A (qualitative or discrete) *character on X* is a function χ from X into a set C of *character states*. Suppose that $\mathcal{T}$ is a phylogenetic tree on X, and let $\chi : X \to C$ be a character on X. For each state α in $\chi(X)$, let $\mathcal{T}_\alpha$ denote the minimal subtree of $\mathcal{T}$ containing the leaves that are assigned state α by χ. We say that χ is *convex on $\mathcal{T}$* if the subtrees in $\{\mathcal{T}_\alpha \mid \alpha \in \chi(X)\}$ are pairwise disjoint (see Figure 2). A collection of characters $\mathcal{C}$ on X is *compatible* if there is a phylogenetic tree $\mathcal{T}$ such that each character in $\mathcal{C}$ is convex on $\mathcal{T}$. If, in addition, $\mathcal{T}$ is the only phylogenetic tree on X with this property, we say that $\mathcal{C}$ *convexly defines* $\mathcal{T}$. The biological relevance of these concepts is explained further in [10] and [11].

We call a trivalent phylogenetic tree on a 4-set a *quartet tree*. If $\mathcal{T}$ is a quartet tree on the set $\{i, j, k, l\}$ and removal of the interior edge e of $\mathcal{T}$ results in the sets $\{i, j\}$ and $\{k, l\}$ labeling the different components of $\mathcal{T}\backslash\{e\}$, then we denote $\mathcal{T}$ by $ij|kl$. Now, given a phylogenetic tree $\mathcal{T}$ on X and a subset Y of X, let $\mathcal{T}|Y$ denote the minimal subtree of $\mathcal{T}$ that connects the leaves in Y, in which any resulting degree 2 vertices are suppressed. In particular, $\mathcal{T}|Y$ is a trivalent phylogenetic tree on Y and we say that $\mathcal{T}$ *displays* $\mathcal{T}|Y$. Given a collection $\mathcal{Q}$ of quartet trees, we say that a

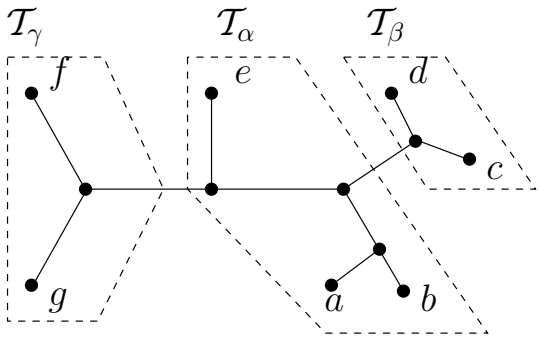


FIG. 2. *For $X = \{a, b, c, d, e, f, g\}$ and $C = \{\alpha, \beta, \gamma\}$, the character $\chi : X \to C$ with $\chi^{-1}(\alpha) = \{a, b, e\}$, $\chi^{-1}(\beta) = \{c, d\}$ and $\chi^{-1}(\gamma) = \{f, g\}$ is convex on the phylogenetic tree depicted in the figure.*

phylogenetic tree $\mathcal{T}$ *displays* $\mathcal{Q}$ precisely if $\mathcal{T}$ displays each quartet tree in $\mathcal{Q}$. For a trivalent phylogenetic tree $\mathcal{T}$ on X, let $\mathcal{Q}(\mathcal{T}) = \{\mathcal{T}|Y : Y \subseteq X, |Y| = 4\}$ be the set of all $\binom{n}{4}$ quartet trees displayed by $\mathcal{T}$.

For $\mathcal{Q}$ a set of quartet trees, let $\mathrm{scl}_2(\mathcal{Q})$ be the *semidyadic closure* of $\mathcal{Q}$, that is, the minimal set of quartet trees that contains $\mathcal{Q}$ and for which we have

$$ab|cd, ac|de \in \mathrm{scl}_2(\mathcal{Q}) \Rightarrow ab|ce, ab|de, bc|de \in \mathrm{scl}_2(\mathcal{Q}).$$

The following lemma summarizes some straightforward properties of the semidyadic closure that are part of the folklore (see [2], [5], [6], and [12]).

LEMMA 2.1. *For any set $\mathcal{Q}$ of quartet trees and any subsets $A, B \subseteq \mathcal{Q}$,*

(i) $A \subseteq \mathrm{scl}_2(A)$,
(ii) $A \subseteq B \Rightarrow \mathrm{scl}_2(A) \subseteq \mathrm{scl}_2(B)$,
(iii) $\mathrm{scl}_2(\mathrm{scl}_2(A)) = \mathrm{scl}_2(A)$,
(iv) $\mathrm{scl}_2(A \cup B) = \mathrm{scl}_2(\mathrm{scl}_2(A) \cup B)$.
(v) *If $\mathcal{Q} = \mathcal{Q}(\mathcal{T})$ for some trivalent phylogenetic tree $\mathcal{T}$, then* $\mathrm{scl}_2(\mathcal{Q}) = \mathcal{Q}$.

We recall one further useful property of the semidyadic closure that will be of use later. Suppose i, j is a *cherry* (a pair of leaves that are adjacent to a common vertex) of a trivalent phylogenetic $\mathcal{T}$ and select leaves u, v as shown in Figure 3(a). Let $\mathcal{T}' = \mathcal{T}|(X - \{j\})$ be the tree as shown in Figure 3(b). Then $\mathcal{T}$ is the only phylogenetic tree that displays both $\mathcal{T}'$ and $ij|uv$ and so, by [3, Lemma 3], we have the following result.

LEMMA 2.2. *For a trivalent phylogenetic tree $\mathcal{T}'$ and quartet tree $ij|uv$ as described,*

$$\mathrm{scl}_2(\mathcal{Q}(\mathcal{T}') \cup \{ij|uv\}) = \mathcal{Q}(\mathcal{T}).$$

For a set $\mathcal{Q}$ of quartet trees let $\mathrm{co}(\mathcal{Q})$ be the set of phylogenetic trees on X (up to isomorphism) that display $\mathcal{Q}$. We close this section with a lemma that summarizes an easily established property of $\mathrm{co}(\mathcal{Q})$.

LEMMA 2.3. *If $\mathcal{Q}$ is a set of quartet trees and $\mathrm{scl}_2(\mathcal{Q}) = \mathcal{Q}(\mathcal{T})$ for some trivalent phylogenetic tree $\mathcal{T}$, then $\mathrm{co}(\mathcal{Q}) = \{\mathcal{T}\}$.*

3. Quartet trees from handy edge-colorings. An *edge-coloring* of a graph is an assignment of colors to the edges of the graph so that two adjacent edges are assigned different colors. We begin this section by giving a method for edge-coloring

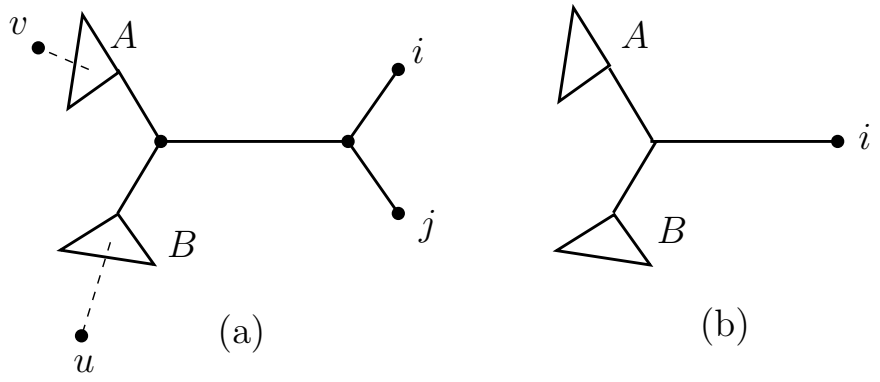


FIG. 3.

a trivalent phylogentic tree $\mathcal{T}$ on X with four colors R, R', L, L'. This edge-coloring is similar to the edge-coloring in [10] based on five colors.

Choose any leaf r of $\mathcal{T}$ and regard $\mathcal{T}$ as a rooted directed tree with r as its root and all edges directed away from r. Color the edge containing r by R. Given any vertex v of $\mathcal{T}$ with degree 3 that is at the end of an even (respectively, odd) length edge path starting at r and ending at v, arbitrarily color the two edges coming out of v by L and R (respectively, L' and R'). This gives an edge-coloring of $\mathcal{T}$ by the colors R, R', L, L', and we call any edge-coloring produced in this way a *handy edge-coloring* of $\mathcal{T}$.

Now, given a handy edge-coloring of $\mathcal{T}$, we describe how to associate a quartet tree with leaves in X to each interior edge of $\mathcal{T}$ (see Figure 4). Assume $e = (u, v)$ is an interior edge of $\mathcal{T}$ colored by R (we will consider the cases where e is colored by L, R', or L' below). The edge coming into u is colored by either (i) R' or (ii) L'. In case (i), we associate the quartet tree $ab|cd$ to edge e as follows: a is the last vertex in the directed path that starts at v and has first edge colored R' and all subsequent edges colored alternately by L and L'; b is the last vertex of the directed path that starts at v and has edges colored alternately by L' and L; c is the last vertex of the directed path that starts at u and has edges colored alternately by L and L'; d is the last vertex of the undirected path that starts at u and has first edge colored R' and all subsequent edges colored alternately by L' and L. In case (ii), a, b, c are all obtained in the same way and d is the last vertex of the undirected path that starts at u, has first two edges colored L' and R', respectively, and has all subsequent edges colored alternately by L and L'.

In case the edge $e = (u, v)$ is labeled by R', the quartet tree $ab|cd$ is obtained in a similar way by following the four distinct paths whose first vertices are either u or v and whose last edges are alternately colored using only the colors L and L'. In case the edge $e = (u, v)$ is labeled by either L or L', a similar procedure is followed in which colors L and R and L' and R' are interchanged so that, in particular, the quartet tree $ab|cd$ is obtained by following the four distinct paths whose first vertices are either u or v and whose last edges are alternately colored using only the colors R and R'.

We denote the collection of $n - 3$ quartet trees obtained in this way by $\mathcal{Q}_0(\mathcal{T})$. Note that in all cases the paths obtained are colored always by at most three colors. Whenever we picture a phylogenetic tree with a handy edge-coloring, we always regard

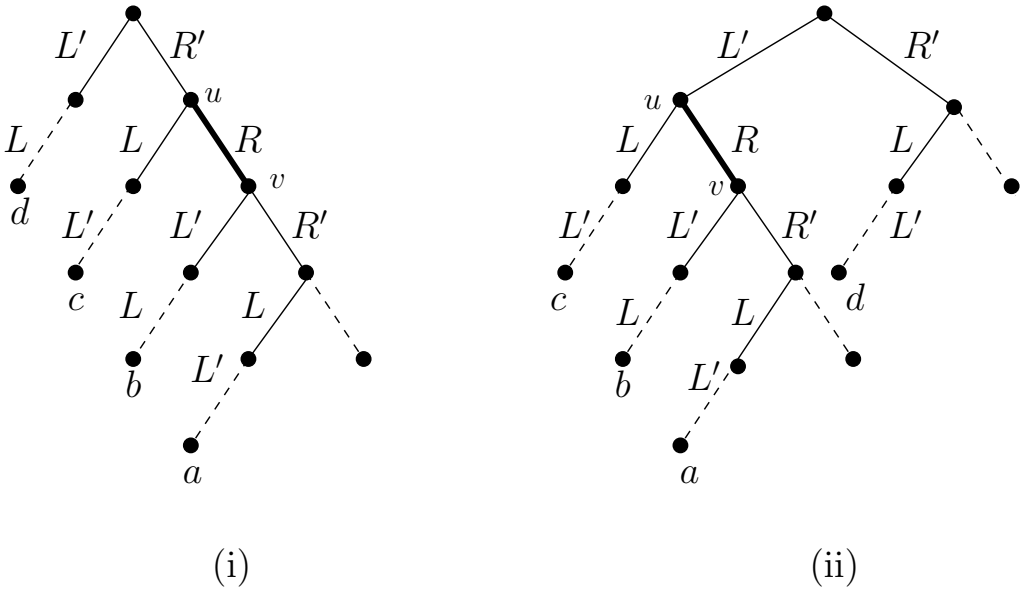


FIG. 4. *The figure depicts the two cases for associating a quartet tree $ab|cd$ to an interior edge e of $\mathcal{T}$, here in bold, that is labeled by R.*

edges below a particular vertex to be colored with R or R' when they are on the right or L or L' when they are on the left.

4. $\mathcal{Q}_0(\mathcal{T})$ determines $\mathcal{Q}(\mathcal{T})$ via semidyadic closure. Suppose that $\mathcal{T}$ is a trivalent phylogenetic tree on X with a handy edge-coloring. In the next section we describe (at most) four characters that convexly define $\mathcal{T}$ and come from the handy edge-coloring of $\mathcal{T}$. The proof that these four characters convexly define $\mathcal{T}$ is based on the following result.

THEOREM 4.1. *Suppose that $\mathcal{T}$ is a trivalent phylogenetic tree on X. Then*

$$\mathrm{scl}_2(\mathcal{Q}_0(\mathcal{T})) = \mathcal{Q}(\mathcal{T}).$$

Proof. We use induction on n. It is easily checked that the result holds when $n = 4$, since in this case $\mathcal{Q}_0(\mathcal{T}) = \mathcal{Q}(\mathcal{T}) = \{\mathcal{T}\}$.

Suppose the theorem holds for any trivalent phylogenetic tree on X with strictly less than $n \geq 5$ leaves. Suppose also that $\mathcal{T}$ is a trivalent phylogenetic tree on X with n leaves. Select a cherry i, j whose central vertex is at maximal edge distance from the reference leaf. If we now consider the handy edge-coloring of $\mathcal{T}$, then there are four cases (plus their mirror images) for the local tree structure around the cherry i, j, as depicted in Figure 5.

Note that in case (b) we could have instead selected the cherry k, l and this produces (the mirror image of) case (a) so we can "transform" case (b) into (a). It thus suffices to consider only cases (a), (c), and (d). For these cases, let $\mathcal{T}' = \mathcal{T}|(X - \{j\})$. Note that the edge-coloring of $\mathcal{T}$ induces a valid handy edge-coloring of $\mathcal{T}'$, where the color assigned to the edge containing i is the same as that assigned to the edge in $\mathcal{T}$ adjacent to the cherry i, j.

First consider cases (a) and (c). It is straightforward to check using the definition of a handy edge-coloring that the only interior edge of $\mathcal{T}$ yielding a quartet tree in

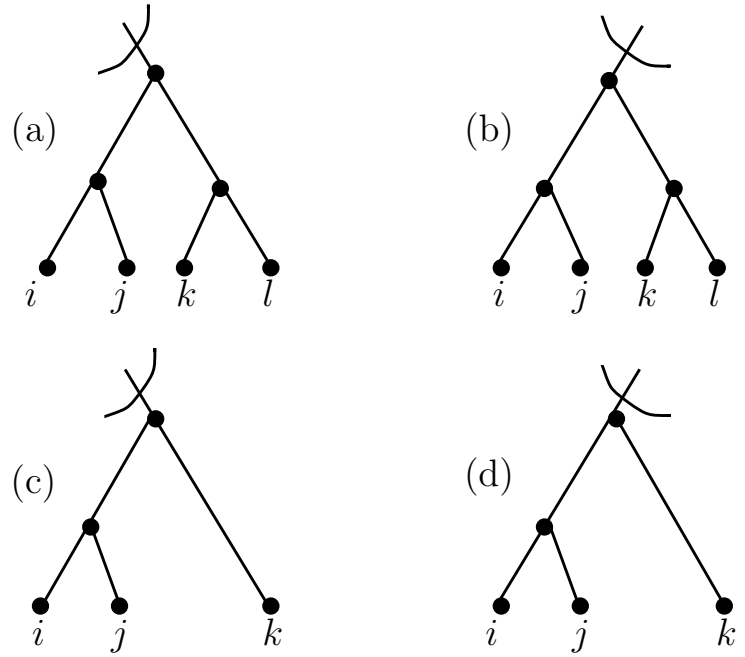


FIG. 5.

$\mathcal{Q}_0(\mathcal{T})$ that contains j is the interior edge that is adjacent to the cherry i, j. Moreover, every interior edge of $\mathcal{T}'$ corresponds to an interior edge of $\mathcal{T}$ and each of these edges gives rise to the same quartet tree in $\mathcal{Q}_0(\mathcal{T}')$ as it does in $\mathcal{Q}_0(\mathcal{T})$. From these observations it easily follows that

$$\mathcal{Q}_0(\mathcal{T}') = \mathcal{Q}_0(\mathcal{T}) - \{ij|kx\} \tag{1}$$

for some $x \in X$ (and with $x \neq l$ in case (a)).

Now, by the induction hypothesis applied to $\mathcal{T}'$,

$$\mathrm{scl}_2(\mathcal{Q}_0(\mathcal{T}')) = \mathcal{Q}(\mathcal{T}')$$

and by Lemma 2.1 (iv) and Lemma 2.2,

$$\mathrm{scl}_2(\mathcal{Q}_0(\mathcal{T}') \cup \{ij|kx\}) = \mathcal{Q}(\mathcal{T}).$$

Thus, by (1),

$$\mathrm{scl}_2(\mathcal{Q}_0(\mathcal{T})) = \mathcal{Q}(\mathcal{T}),$$

and so the induction step is established for cases (a) and (c).

Thus it suffices to consider now just case (d). The edge e coming into the cherry i, j induces the quartet tree $ij|ku \in \mathcal{Q}_0(\mathcal{T})$ and the edge e' incident to e but not containing k induces the quartet tree $jk|uv \in \mathcal{Q}_0(\mathcal{T})$, for some pair of leaves $u, v \in X$ (see Figure 6).

Thus,

$$\mathrm{scl}_2(\{ij|ku, jk|uv\}) \subseteq \mathrm{scl}_2(\mathcal{Q}_0(\mathcal{T})).$$

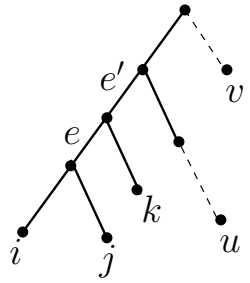


FIG. 6.

But $ik|uv \in \mathrm{scl}_2(\{ij|ku, jk|uv\})$ and so

$$ik|uv \in \mathrm{scl}_2(\mathcal{Q}_0(\mathcal{T})). \tag{2}$$

Now, it is straightforward to check using the definition of a handy edge-coloring that the only interior edges of $\mathcal{T}$ yielding quartet trees in $\mathcal{Q}_0(\mathcal{T})$ that contain j are the edges e and e'. Moreover, every interior edge of $\mathcal{T}'$ corresponds to an interior edge of $\mathcal{T}$ and each of these gives rise to the same quartet tree in $\mathcal{Q}_0(\mathcal{T}')$ as it does in $\mathcal{Q}_0(\mathcal{T})$ except e', which gives rise to $ik|uv$ in $\mathcal{Q}_0(\mathcal{T}')$. From these observations it easily follows that

$$\mathcal{Q}_0(\mathcal{T}') = (\mathcal{Q}_0(\mathcal{T}) - \{ij|ku, jk|uv\}) \cup \{ik|uv\}. \tag{3}$$

Combining (2), (3), and Lemma 2.1 (parts (i), (ii), and (iii)) we have

$$\mathrm{scl}_2(\mathcal{Q}_0(\mathcal{T}') \cup \{ik|uv\}) \subseteq \mathrm{scl}_2(\mathcal{Q}_0(\mathcal{T})). \tag{4}$$

On the other hand, if we apply Lemma 2.2, the induction hypothesis for $\mathcal{T}'$, and Lemma 2.1 (iv), we obtain (respectively) the following three equalities:

$$\begin{aligned}\mathcal{Q}(\mathcal{T}) &= \mathrm{scl}_2(\mathcal{Q}(\mathcal{T}') \cup \{ik|uv\}) \\ &= \mathrm{scl}_2(\mathrm{scl}_2(\mathcal{Q}_0(\mathcal{T}')) \cup \{ik|uv\}) \\ &= \mathrm{scl}_2(\mathcal{Q}_0(\mathcal{T}') \cup \{ik|uv\}).\end{aligned}$$

Combining these equalities with (4) gives

$$\mathcal{Q}(\mathcal{T}) \subseteq \mathrm{scl}_2(\mathcal{Q}_0(\mathcal{T})).$$

However, this implies $\mathcal{Q}(\mathcal{T}) = \mathrm{scl}_2(\mathcal{Q}_0(\mathcal{T}))$ in view of $\mathcal{Q}_0(\mathcal{T}) \subseteq \mathcal{Q}(\mathcal{T})$ and using Lemma 2.1 (parts (ii) and (v)). This establishes the induction step and thereby completes the proof of Theorem 4.1. □

5. Handy edge-colorings convexly define trees. We now relate characters and quartet trees. Given a character $\chi : X \to C$ on X, we denote by $\pi(\chi)$ the partition $\{\chi^{-1}(\alpha) : \alpha \in C\}$ of X. Suppose that $\mathcal{T}$ is a phylogenetic tree on X and that $\mathcal{C}$ is a set of characters on X. We say that $\mathcal{T}$ *displays* $\mathcal{C}$ if each character in $\mathcal{C}$ is convex on $\mathcal{T}$. Note that $\mathcal{T}$ displays $\mathcal{C}$ precisely if for each $\chi \in \mathcal{C}$ there exists some set $\mathcal{E}$ of edges of $\mathcal{T}$ such that, for all distinct $A, B \in \pi(\chi)$, A and B are subsets of different connected components of $\mathcal{T}\backslash\mathcal{E}$.

For any collection $\mathcal{C}$ of characters on X, let

$$\mathcal{Q}(\mathcal{C}) = \{ij|kl : \text{ there exists some } \chi \in \mathcal{C} \text{ and some } A, B \in \pi(\chi) \text{ such that } i, j \in A \text{ and } k, l \in B\}.$$

LEMMA 5.1. *Let $\mathcal{C}$ be a collection of characters on X, and suppose that $\mathcal{T}$ is a trivalent phylogenetic tree that displays $\mathcal{C}$. If there exists some $\mathcal{Q}_1 \subseteq \mathcal{Q}(\mathcal{C})$ with $\mathrm{scl}_2(\mathcal{Q}_1) = \mathcal{Q}(\mathcal{T})$, then $\mathcal{C}$ convexly defines $\mathcal{T}$.*

Proof. Note that Lemma 2.1 (ii) gives $\mathrm{scl}_2(\mathcal{Q}_1) \subseteq \mathrm{scl}_2(\mathcal{Q}(\mathcal{C}))$. Thus,

$$\mathcal{Q}(\mathcal{T}) \subseteq \mathrm{scl}_2(\mathcal{Q}(\mathcal{C})). \tag{5}$$

On the other hand, since each character in $\mathcal{C}$ is convex on $\mathcal{T}$, we have $\mathcal{Q}(\mathcal{C}) \subseteq \mathcal{Q}(\mathcal{T})$ and so

$$\mathrm{scl}_2(\mathcal{Q}(\mathcal{C})) \subseteq \mathcal{Q}(\mathcal{T}), \tag{6}$$

by Lemma 2.1 (parts (ii), (iii), and (v)). Combining (5) and (6) gives $\mathrm{scl}_2(\mathcal{Q}(\mathcal{C})) = \mathcal{Q}(\mathcal{T})$, and so, by Lemma 2.3 we have $\mathrm{co}(\mathcal{Q}(\mathcal{C})) = \{\mathcal{T}\}$. But from [12, Proposition 2(1)], if $\mathrm{co}(\mathcal{Q}(\mathcal{C})) = \{\mathcal{T}\}$, then $\mathcal{C}$ convexly defines $\mathcal{T}$. This completes the proof. ∎

We now specialize to a set of (at most four) characters that are induced by any handy edge-coloring of a trivalent phylogenetic tree $\mathcal{T}$ on X and show that these characters convexly define $\mathcal{T}$.

Suppose that we are given a handy edge-coloring of $\mathcal{T}$. To each color $F \in \{L, L', R, R'\}$ that is assigned to at least one edge of $\mathcal{T}$, we associate a character on X in the following way. Denote by $\sim_F$ the equivalence relation on X defined by $x \sim_F y$ $(x, y \in X)$ if the path in $\mathcal{T}$ from x to y does not contain an edge that is assigned color F. Let π_F denote the partition of X that arises from the equivalence classes of $\sim_F$ and let χ_F denote the character on X for which $\pi(\chi_F) = \pi_F$. We denote by $\mathcal{C}(\mathcal{T})$ the (at most) four characters induced by this edge-coloring.

The main result from [10] is that, for any trivalent phylogenetic tree $\mathcal{T}$ on X, there exists a set $\mathcal{C}$ of at most five characters on X, such that $\mathcal{T}$ is the only phylogenetic tree on X that displays $\mathcal{C}$. The following theorem shows that, by taking $\mathcal{C} = \mathcal{C}(\mathcal{T})$, we can improve the result by replacing "five" with "four."

THEOREM 5.2. *Suppose that $\mathcal{T}$ is a trivalent phylogenetic tree on X. Then the (at most) four characters in $\mathcal{C}(\mathcal{T})$ convexly define $\mathcal{T}$.*

Proof. First note that each character in $\mathcal{C}(\mathcal{T})$ is convex on $\mathcal{T}$. Note also that since $\mathcal{Q}_0(\mathcal{T})$ is the set of quartet trees corresponding to the handy edge-coloring of $\mathcal{T}$, we have

$$\mathcal{Q}_0(\mathcal{T}) \subseteq \mathcal{Q}(\mathcal{C}(\mathcal{T})).$$

Also, by Theorem 4.1, $\mathrm{scl}_2(\mathcal{Q}_0(\mathcal{T})) = \mathcal{Q}(\mathcal{T})$. Thus, since $\mathcal{T}$ displays $\mathcal{C}(\mathcal{T})$ we may apply Lemma 5.1 to deduce that $\mathcal{C}(\mathcal{T})$ convexly defines $\mathcal{T}$. ∎

Note that the proof of this result shows how to construct $\mathcal{T}$ from $\mathcal{C}(\mathcal{T})$ in polynomial time using the semidyadic closure operation. Alternatively, since $|\mathcal{Q}_0(\mathcal{T})| = |X| - 3$ the "split-closure" approach described by Semple and Steel [9] would also apply. It can also be shown that $\mathcal{C}(\mathcal{T})$ "strongly" defines $\mathcal{T}$ in the sense of [10].

Acknowledgment. The authors thank Charles Semple for some helpful comments on an earlier version of this manuscript.

REFERENCES

[1] R. Agarwala and D. Fernández-Baca, *A polynomial-time algorithm for the phylogeny problem when the number of character states is fixed*, SIAM J. Comput., 23 (1994), pp. 1216–1224.

[2] H.-J. Bandelt and A. W. M. Dress, *Reconstructing the shape of a tree from observed dissimilarity data*, Adv. Appl. Math., 7 (1986), pp. 309–343.

[3] S. Böcker, D. Bryant, A. W. M. Dress, and M. A. Steel, *Algorithmic aspects of tree amalgamation*, J. Algorithms, 37 (2000), pp. 522–537.

[4] P. Buneman, *The recovery of trees from measures of dissimilarity*, in Mathematics in the Archaeological and Historical Sciences, F. R. Hodson, D. G. Kendall, and P. Tautu, eds., Edinburgh University Press, Edinburgh, 1971, pp. 387–395.

[5] H. Colonius and H. H. Schulze, *Tree structures for proximity data*, British J. Math. Statist. Psych., 34 (1981), pp. 167–180.

[6] M. C. H. Dekker, *Reconstruction Methods for Derivation Trees*, unpublished Masters thesis, Vrije Universiteit, Amsterdam, Netherlands, 1986.

[7] S. Kannan and T. Warnow, *A fast algorithm for the computation and enumeration of perfect phylogenies*, SIAM J. Comput., 26 (1997), pp. 1749–1763.

[8] F. R. McMorris, T. J. Warnow, and T. Wimer, *Triangulating vertex-colored graphs*, SIAM J. Discrete Math., 7 (1994), pp. 296–306.

[9] C. Semple and M. Steel, *Tree reconstruction via a closure operation on partial splits*, in Proceedings of journées ouvertes: Biologie, informatique et mathématique, Lecture Notes in Comput. Sci. 2066, O. Gascuel and M.-F. Sagot, eds., Springer-Verlag, Berlin, 2001, pp. 126–134.

[10] C. Semple and M. Steel, *Tree reconstruction from multi-state characters*, Adv. Appl. Math., 28 (2002), pp. 169–184.

[11] C. Semple and M. Steel, *Phylogenetics*, Oxford University Press, Oxford, 2003.

[12] M. Steel, *The complexity of reconstructing trees from qualitative characters and subtrees*, J. Classification, 9 (1992), pp. 91–116.

SIAM J. DISCRETE MATH.
Vol. 18, No. 4, pp. 844–878

ON THE OPTIMALITY OF COLORING WITH A LATTICE*

YAEL BEN-HAIM† AND TUVI ETZION‡

Abstract. For $z_1, z_2, z_3 \in \mathbb{Z}^2$, the *tristance* $d_3(z_1, z_2, z_3)$ is a generalization of the L_1-distance on $\mathbb{Z}^2$ to a quality that reflects the relative dispersion of three points rather than two. In this paper we prove that at least $3k^2$ colors are required to color the points of $\mathbb{Z}^2$, such that the tristance between any three distinct points, colored with the same color, is at least $4k$. We prove that $3k^2 + 3k + 1$ colors are required if the tristance is at least $4k + 2$. For the first case we show an infinite family of colorings with $3k^2$ colors and conjecture that these are the only colorings with $3k^2$ colors.

Key words. coloring, lattice, Lee sphere, tristance

AMS subject classifications. 05C15, 11H31, 52C15

DOI. 10.1137/S0895480104439589

1. Introduction.

Consider the *grid graph* $\mathcal{G} = (V, E)$ whose vertex set is $V = \mathbb{Z}^2$ and $\{(x_1, y_1), (x_2, y_2)\} \in E$ if $|x_1 - x_2| + |y_1 - y_2| = 1$. A coloring $\mathcal{F}$ is an onto function $\mathcal{F} : \mathbb{Z}^2 \to \{1, 2, \ldots, \chi\}$, where χ is the number of colors. We ask the following question: Given a positive integer t, what is the smallest number of colors required to color $\mathbb{Z}^2$, such that for any three points colored with the same color, the size of the minimum spanning tree which connects them is at least t?

This problem has an application in two-dimensional cluster error-correcting codes [1], [3]. For each color φ we assign a two-error-correcting code to the points colored with φ. We obtain an array which corrects any number of errors, if there exists a cluster of size t, which contains all the errors.

The problem has also combinatorial interest, as the coloring structure obtained is a generalization of perfect codes of $\mathbb{Z}^2$ in the L_1-metric and tiling of $\mathbb{Z}^2$ with Lee spheres (see [4]).

Lee spheres and lattices have an important role in our discussion.

The L_1-*distance* between two elements of $\mathbb{Z}^2$, $z_1 = (x_1, y_1)$, $z_2 = (x_2, y_2)$, is defined by

$$d_2(z_1, z_2) = |x_1 - x_2| + |y_1 - y_2|.$$

Clearly, the length of the shortest path which connects z_1 and z_2 in $\mathcal{G}$ is $d_2(z_1, z_2)$.

For a given element $\varsigma \in \mathbb{Z}^2$, the *Lee sphere* of radius k, $\mathcal{S}_k(\varsigma)$ (or $\mathcal{S}(\varsigma)$ if k is known), is defined by [4]

$$\mathcal{S}_k(\varsigma) = \{z : d_2(\varsigma, z) \leq k\}.$$

Clearly, $|\mathcal{S}_k(\varsigma)| = 2k^2 + 2k + 1$.

*Received by the editors January 12, 2004; accepted for publication (in revised form) August 28, 2004; published electronically May 20, 2005. The results of this paper were presented in part at the IEEE International Symposium on Information Theory, Chicago, 2004.
http://www.siam.org/journals/sidma/18-4/43958.html

†Department of Electrical Engineering-Systems, Tel Aviv University, Tel Aviv 69978, Israel (yaelm@eng.tau.ac.il). This research was conducted for partial fulfillment of the author's requirements for the degree of Master of Science in Computer Science at the Technion–Israel Institute of Technology.

‡Department of Computer Science, Technion–Israel Institute of Technology, Haifa 32000, Israel (etzion@cs.technion.ac.il). This author's research was supported in part by the Technion V. P. R. Fund.

A *lattice* of $\mathbb{Z}^2$ is a linear subspace of $\mathbb{Z}^2$. A lattice Λ with dimension two is defined by $\Lambda = \{a_1 v_1 + a_2 v_2 : a_1, a_2 \in \mathbb{Z}\}$, where $v_1 = (v_{11}, v_{12})$, $v_2 = (v_{21}, v_{22})$ are two linearly independent vectors in $\mathbb{Z}^2$, called the *basis* of Λ. The matrix

$$G = \begin{pmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{pmatrix}$$

having these vectors as rows is said to be a *generator matrix* of Λ. It is well known that $|\det G|$ is the number of cosets of Λ in $\mathbb{Z}^2$, i.e., $|\mathbb{Z}^2 / \Lambda| = |\det G|$. A lattice Λ with dimension two defines a coloring as follows: The points of each distinct coset of Λ are colored with the same color. Thus, there are $|\det G|$ distinct colors.

The solution for the following simpler question is known [1]:

Given a positive integer t, what is the smallest number of colors required to color $\mathbb{Z}^2$, such that for any two points colored with the same color, the length of the shortest path which connects them is at least t?

If $t = 2k + 1$, then the number of colors is $2k^2 + 2k + 1$ [1]. A coloring is given by a lattice whose generator matrix is

$$\begin{pmatrix} 1 & 2k+1 \\ 0 & 2k^2 + 2k + 1 \end{pmatrix}.$$

This lattice defines also a tiling of $\mathbb{Z}^2$ with Lee spheres of radius k [1], [4]. If $t = 2k$, then the number of colors is $2k^2$ [1]. A coloring is given by a lattice whose generator matrix is

$$\begin{pmatrix} k & k \\ 0 & 2k \end{pmatrix}.$$

Let $z_1, z_2, z_3 \in \mathbb{Z}^2$. The *tristance* $d_3(z_1, z_2, z_3)$ is a generalization of the L_1-distance. $d_3(z_1, z_2, z_3)$ is defined as the number of edges in a minimum spanning tree of z_1, z_2, z_3 in the grid graph $\mathcal{G}$. It is known [3] that if $z_1 = (x_1, y_1)$, $z_2 = (x_2, y_2)$, $z_3 = (x_3, y_3)$, then

$$d_3(z_1, z_2, z_3) = \left(\max_{1 \le i \le 3} x_i - \min_{1 \le i \le 3} x_i \right) + \left(\max_{1 \le i \le 3} y_i - \min_{1 \le i \le 3} y_i \right).$$

For a coloring $\mathcal{F} : \mathbb{Z}^2 \to \{1, 2, \ldots, \chi\}$, $d_3(\mathcal{F})$ is defined by

$$d_3(\mathcal{F}) = \min_{\substack{\mathcal{F}(z_1) = \mathcal{F}(z_2) = \mathcal{F}(z_3) \\ |\{z_1, z_2, z_3\}| = 3}} d_3(z_1, z_2, z_3).$$

For a given t, a coloring $\mathcal{F}$ will be called a *t-coloring* if $d_3(\mathcal{F}) \ge t$.

Etzion and Vardy [3] proved that if $t = 4k$ ($t = 4k + 2$), then any t-coloring defined by a lattice has at least $3k^2$ ($3k^2 + 3k + 1$) colors. Schwartz and Etzion [6] proved that if $t = 4k + 1$ ($t = 4k + 3$), then any t-coloring defined by a lattice has at least $3k^2 + 2k$ ($3k^2 + 5k + 2$) colors. For each t an optimal lattice coloring was given in [3].

In this paper we prove that if $t = 4k$ ($t = 4k + 2$), then any t-coloring (and not just t-coloring defined by a lattice) has at least $3k^2$ ($3k^2 + 3k + 1$) colors. The result for $t = 4k$ is proved in section 2. In section 3 we show an infinite family of optimal colorings which we believe are the only optimal colorings. We conclude in sections 4 and 5 with extensions, some related questions, and problems for further research. There are also three appendices. In Appendix A we give a short description of the various types of geometric shapes used in our proofs. In Appendices B and C we give the detailed proofs of some of our results.

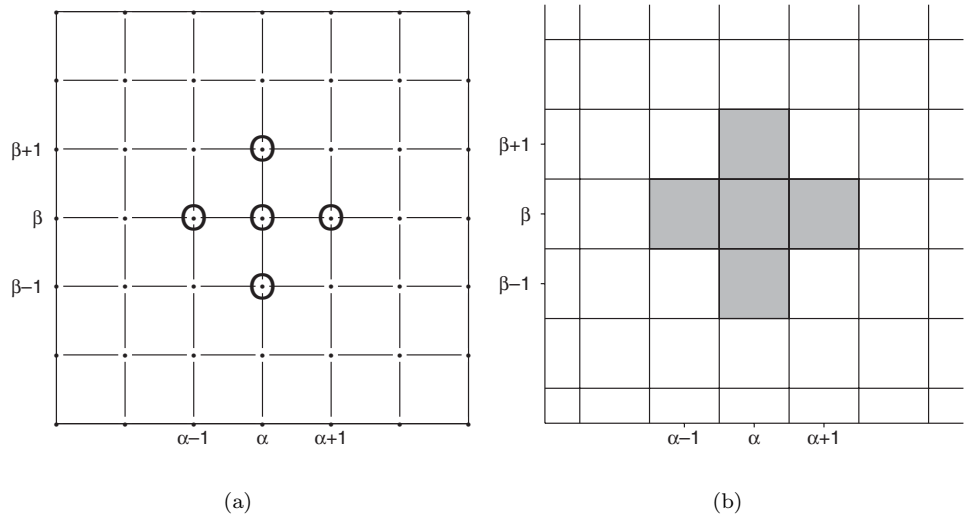


FIG. 1. $\mathcal{S}_1((\alpha,\beta))$ *in the grid graph and in the array.*

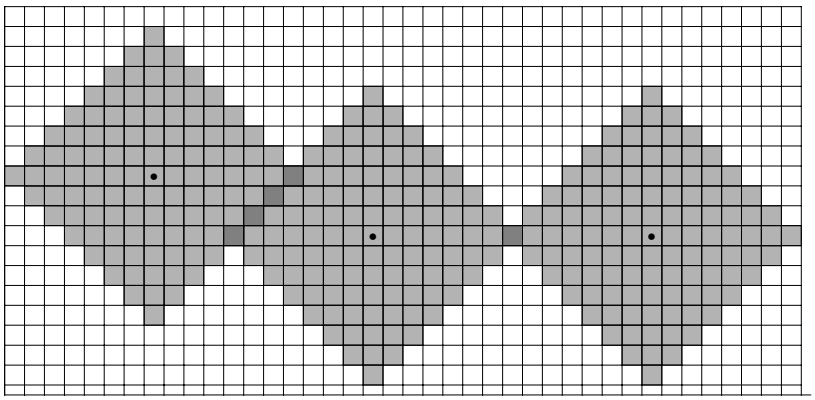
2. Optimality of the coloring. The main result of this paper is the following theorem.

THEOREM 1. *If $\mathcal{F}$ is a $4k$-coloring of $\mathbb{Z}^2$, then the number of colors in $\mathcal{F}$ is at least $3k^2$.*

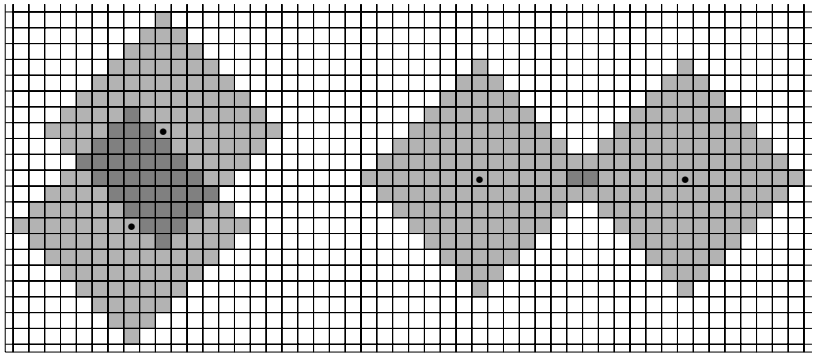
As the proof is very detailed, we first sketch the outline of the proof. For the simplicity of our presentation we will use an infinite array instead of the grid graph. Each element of $\mathbb{Z}^2$ is mapped into a corresponding cell of the array (see Figure 1). Let $\mathcal{C}$ be the set of cells in $\mathbb{Z}^2$ which are colored with the first color. These cells will be called *black cells*. For each black cell ς we define a neighborhood which contains ς. Each neighborhood contains either one black cell or two black cells, and two different neighborhoods are disjoint. We will prove that the size of a neighborhood with one black cell is at least $3k^2$ and the size of a neighborhood with two black cells is greater than $6k^2$. These properties will lead to an immediate proof of the theorem.

Hence the main part of the proof includes the definition of a neighborhood and the computation of its size. For this purpose, in what follows we consider all the Lee spheres with radius k whose centers are exactly all the black cells. Each one of these spheres, $\mathcal{S}(\varsigma)$, satisfies one of the following:

- If the sphere does not intersect another sphere, then the neighborhood of ς includes $\mathcal{S}(\varsigma)$ and additional cells above $\mathcal{S}(\varsigma)$.
- $\mathcal{S}(\varsigma)$ intersects other spheres, and any such intersection with another sphere contains cells which are on the same diagonal line, as depicted in Figure 2(a). This type of intersection will be called a *line-intersection*. The neighborhood of ς in this case includes $\mathcal{S}(\varsigma)$ (except maybe some of the intersection) and additional cells above $\mathcal{S}(\varsigma)$.
- $\mathcal{S}(\varsigma)$ intersects exactly one sphere, $\mathcal{S}(\varsigma_1)$, on more than one diagonal line, as depicted in Figure 2(b). This type of intersection will be called a *deep-intersection*. In this case we define a neighborhood which includes the union of $\mathcal{S}(\varsigma)$ and $\mathcal{S}(\varsigma_1)$ and additional cells above this union.



(a) Line-intersections



(b) Deep-intersections

FIG. 2. *Various types of intersections.* (a) *Line-intersections.* (b) *Deep-intersections.*

2.1. Intersections of spheres. For two black cells ς_1, ς_2, let $\mathcal{I}(\varsigma_1, \varsigma_2) = \mathcal{S}(\varsigma_1) \cap \mathcal{S}(\varsigma_2)$. By the definition of spheres with radius k we clearly have the following lemma.

LEMMA 1. *Given two black cells* ς_1, ς_2,

(a) $\mathcal{I}(\varsigma_1, \varsigma_2) = \varnothing$ *if and only if* $d_2(\varsigma_1, \varsigma_2) > 2k$;
(b) $\mathcal{I}(\varsigma_1, \varsigma_2)$ *is a line-intersection if and only if* $d_2(\varsigma_1, \varsigma_2) = 2k$;
(c) $\mathcal{I}(\varsigma_1, \varsigma_2)$ *is a deep-intersection if and only if* $d_2(\varsigma_1, \varsigma_2) < 2k$.

The next lemma is an immediate result from the definition of the tristance.

LEMMA 2. *Let* t *be a positive integer, and let* $z_0 = (0,0)$, $z_1 = (\alpha, \beta)$, $z_2 = (x, y)$ *be three cells in* $\mathbb{Z}^2$ *such that* $\alpha, \beta \geq 0$ *and* $\alpha + \beta < t$. *The tristance* $d_3(z_0, z_1, z_2) < t$

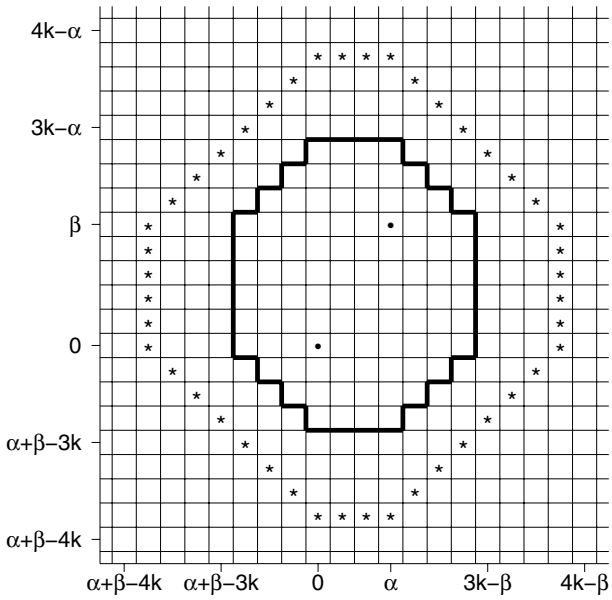


FIG. 3. *The polygon* $\mathcal{P}(\varsigma_0, \varsigma_1)$ *for* $(\alpha, \beta) = (3, 5)$ *and* $k = 4$.

iff

$$\begin{array}{rcccl} \alpha+\beta-t & < & x & < & t-\beta, \\ \alpha+\beta-t & < & y & < & t-\alpha, \\ \alpha+\beta-t & < & x+y & < & t, \\ \beta-t & < & y-x & < & t-\alpha. \end{array} \tag{1}$$

Note that if z_0 and z_1 are black cells, then there is no black cell inside the polygon defined by (1), i.e., inside the area defined by $*$ (inclusive) in Figure 3.

Let z_1, z_2 be two distinct black cells such that $d_2(z_1, z_2) < 3k$. We define $\mathcal{P}(z_1, z_2)$ to be the set of cells such that for each cell ς, if $\mathcal{P}(z_1, z_2) \cap \mathcal{S}(\varsigma) \neq \varnothing$, then $d_3(z_1, z_2, \varsigma) < 4k$.

LEMMA 3. *Let* $\varsigma_0 = (0, 0)$, $\varsigma_1 = (\alpha, \beta)$ *be two black cells such that* $|\alpha| \leq 2k$, $0 \leq \beta \leq 2k$, $d_2(\varsigma_0, \varsigma_1) = |\alpha| + \beta < 3k$.

If $\alpha \geq 0$, *then*

$$\mathcal{P}(\varsigma_0, \varsigma_1) = \left\{ (x, y) : \begin{array}{rcccl} \alpha+\beta-3k & < & x & < & 3k-\beta \\ \alpha+\beta-3k & < & y & < & 3k-\alpha \\ \alpha+\beta-3k & < & x+y & < & 3k \\ \beta-3k & < & y-x & < & 3k-\alpha \end{array} \right\}.$$

If $\alpha < 0$, *then*

$$\mathcal{P}(\varsigma_0, \varsigma_1) = \left\{ (x, y) : \begin{array}{rcccl} \beta-3k & < & x & < & 3k-|\alpha|-\beta \\ |\alpha|+\beta-3k & < & y & < & 3k-|\alpha| \\ \beta-3k & < & x+y & < & 3k-|\alpha| \\ |\alpha|+\beta-3k & < & y-x & < & 3k \end{array} \right\}.$$

The cells of $\mathcal{P}(\varsigma_0, \varsigma_1)$ are inside the bold lines in Figure 3. If $|\alpha| + \beta < 3k - 1$, $|\alpha|, \beta > 0$, then $\mathcal{P}(\varsigma_0, \varsigma_1)$ is an octagon, as depicted in Figure 3. If $|\alpha| + \beta < 3k - 1$ and either $\alpha = 0$ or $\beta = 0$, then $\mathcal{P}(\varsigma_0, \varsigma_1)$ is a hexagon. If $|\alpha| + \beta = 3k - 1$, then $\mathcal{P}(\varsigma_0, \varsigma_1)$ is a rectangle whose opposite vertices are ς_0 and ς_1. Note that $\mathcal{P}(\varsigma_0, \varsigma_1)$ always contains the rectangle whose opposite vertices are ς_0 and ς_1. $\mathcal{P}(\varsigma_0, \varsigma_1)$ will be called the *polygon* of ς_0 and ς_1.

COROLLARY 1. *If $\varsigma_1, \varsigma_2, \varsigma_3$ are three distinct black cells, then $\mathcal{P}(\varsigma_1, \varsigma_2) \cap \mathcal{S}(\varsigma_3) = \varnothing$.*

COROLLARY 2. *Let $\varsigma_1, \varsigma_2, \varsigma_3$ be three distinct black cells. If $\mathcal{I}(\varsigma_1, \varsigma_2)$ is a deep-intersection, then $\mathcal{P}(\varsigma_1, \varsigma_2) \supset \mathcal{S}(\varsigma_1) \cup \mathcal{S}(\varsigma_2)$.*

COROLLARY 3. *Let $\varsigma_1, \varsigma_2, \varsigma_3$ be three distinct black cells. If $\mathcal{I}(\varsigma_1, \varsigma_2)$ is a deep-intersection, then $(\mathcal{S}(\varsigma_1) \cup \mathcal{S}(\varsigma_2)) \cap \mathcal{S}(\varsigma_3) = \varnothing$.*

COROLLARY 4. *Let $\varsigma_1, \varsigma_2, \varsigma_3$ be three distinct black cells. If $\mathcal{I}(\varsigma_1, \varsigma_2)$ and $\mathcal{I}(\varsigma_2, \varsigma_3)$ are line-intersections, then one of the following holds:*

1. $|\mathcal{I}(\varsigma_1, \varsigma_2)| = |\mathcal{I}(\varsigma_2, \varsigma_3)| = 1$.
2. *$\mathcal{I}(\varsigma_1, \varsigma_2)$ is on the line $x + y = l_1$ and $\mathcal{I}(\varsigma_2, \varsigma_3)$ is on the line $x + y = l_2$ for some l_1, l_2 such that $|l_1 - l_2| = 2k$.*
3. *$\mathcal{I}(\varsigma_1, \varsigma_2)$ is on the line $y - x = l_1$ and $\mathcal{I}(\varsigma_2, \varsigma_3)$ is on the line $y - x = l_2$ for some l_1, l_2 such that $|l_1 - l_2| = 2k$.*

2.2. Definition of neighborhood. In this subsection we define the *neighborhood* $\mathcal{N}(\varsigma)$ for any given black cell ς. First, we give some definitions concerning the sphere of a black cell. The right (left) *tip* of the sphere is the rightmost (leftmost) cell in the sphere. The *top* (*bottom*) of the sphere is the highest (lowest) cell in the sphere. A cell is called *free* if it is not contained in any sphere. For each black cell $\varsigma = (\alpha, \beta)$ we define a set

$$\mathcal{U}(\varsigma) = \{(x, y) : |x - \alpha| \le k - 1,\ \beta + 2 \le y \le \beta + k,\ (x, y) \notin \mathcal{S}(\varsigma)\}.$$

An example of $\mathcal{U}(\varsigma)$, $\varsigma = (0, 0)$, is depicted in Figure 4. We partition $\mathcal{U}(\varsigma)$ into two subsets $\mathcal{U}_l(\varsigma)$ and $\mathcal{U}_r(\varsigma)$, where

$$\mathcal{U}_r(\varsigma) = \{(x, y) : 0 < x - \alpha \le k - 1,\ \beta + 2 \le y \le \beta + k,\ (x, y) \notin \mathcal{S}(\varsigma)\}.$$

If $\mathcal{S}(\varsigma)$ has a deep-intersection with another sphere $\mathcal{S}(\varsigma_1)$, then $\mathcal{N}(\varsigma) = \mathcal{N}(\varsigma_1)$; i.e., ς and ς_1 have a joint neighborhood. In any other case each black cell has its own neighborhood. Note that by Corollary 3, if $\mathcal{I}(\varsigma, \varsigma_1)$ is a deep-intersection, then no other sphere intersects $\mathcal{S}(\varsigma) \cup \mathcal{S}(\varsigma_1)$.

The definition of $\mathcal{N}(\varsigma)$ will be done by assigning each cell in $\mathbb{Z}^2$ to at most one neighborhood. This assignment of a cell $z_0 = (a, b_0)$ is done as follows:

1. If z_0 is not a free cell
   - If z_0 belongs to exactly one sphere $\mathcal{S}(\varsigma)$, then $z_0 \in \mathcal{N}(\varsigma)$.
   - If $z_0 \in \mathcal{I}(\varsigma_1, \varsigma_2)$ and $\mathcal{I}(\varsigma_1, \varsigma_2)$ is a deep-intersection, then $z_0 \in \mathcal{N}(\varsigma_1) = \mathcal{N}(\varsigma_2)$.
   - If $z_0 \in \mathcal{I}(\varsigma_1, \varsigma_2)$, $\varsigma_1 = (\alpha, \beta)$, $\varsigma_2 = (\gamma, \delta)$, $\beta < \delta$, and $\mathcal{I}(\varsigma_1, \varsigma_2)$ is a line-intersection, then $z_0 \in \mathcal{N}(\varsigma_2)$.
   - If $z_0 \in \mathcal{I}(\varsigma_1, \varsigma_2)$, $\varsigma_1 = (\alpha, \beta)$, $\varsigma_2 = (\gamma, \beta)$, $\alpha < \gamma$, and $\mathcal{I}(\varsigma_1, \varsigma_2)$ is a line-intersection, then $z_0 \in \mathcal{N}(\varsigma_1)$ (note that z_0 is the right tip of $\mathcal{S}(\varsigma_1)$ and the left tip of $\mathcal{S}(\varsigma_2)$).
2. If z_0 is a free cell, then let $z_1 = (a, b_1)$, $b_1 < b_0$, be a cell in a sphere such that all the cells in the set $\{(a, d) : b_1 < d < b_0\}$ are free. If such a cell z_1 does not exist, then z_0 does not belong to any neighborhood.

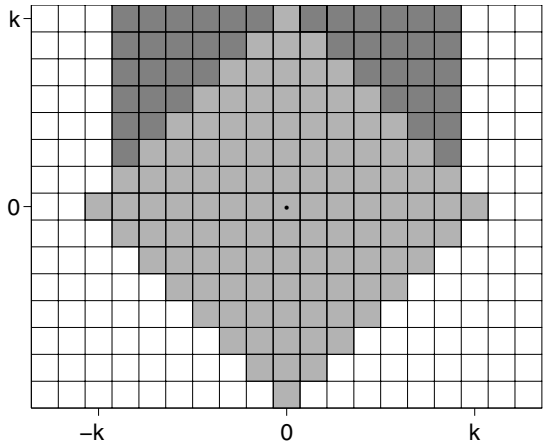


FIG. 4. $\mathcal{U}(\varsigma)$, *where* ς *is located at* $(0,0)$.

- If z_1 is not a tip of any sphere and $z_1 \in \mathcal{S}(\varsigma)$, then $z_0 \in \mathcal{N}(\varsigma)$.
- If z_1 is a tip of two spheres, $\mathcal{S}(\varsigma_1)$ and $\mathcal{S}(\varsigma_2)$, such that $\varsigma_1 = (\alpha, \beta)$ and $\varsigma_2 = (\alpha + 2k, \beta)$, then $z_0 \in \mathcal{N}(\varsigma_2)$.
- If z_1 is a tip of exactly one sphere, $\mathcal{S}(\varsigma)$, then let $z_2 = (a, b_2)$, $b_2 < b_1$, be a cell in a sphere such that all the cells in the set $\{(a, d) : b_2 < d < b_1\}$ are free. If such a cell z_2 does not exist, then $z_0 \in \mathcal{N}(\varsigma)$.
  - If z_2 is a cell of two distinct spheres, then $z_0 \in \mathcal{N}(\varsigma)$.
  - If z_2 is a cell of exactly one sphere $\mathcal{S}(\varsigma_1)$ and $\mathcal{U}(\varsigma_1) \cap \mathcal{S}(\varsigma) = \varnothing$, then $z_0 \in \mathcal{N}(\varsigma)$.
  - If z_2 is a cell of exactly one sphere $\mathcal{S}(\varsigma_1)$ and $\mathcal{U}(\varsigma_1) \cap \mathcal{S}(\varsigma) \neq \varnothing$, then $z_0 \in \mathcal{N}(\varsigma_1)$ if $z_0 \in \mathcal{P}(\varsigma_1, \varsigma)$ and $z_0 \in \mathcal{N}(\varsigma)$ if $z_0 \notin \mathcal{P}(\varsigma_1, \varsigma)$.

Note that in all cases, except one, z_0 is assigned to some neighborhood. The case analysis of the definition makes it clear that z_0 is assigned to at most one neighborhood. We would like to clarify that if z_0 is a free cell and z_1 is not a tip, then $z_1 \in \mathcal{I}(\varsigma_1, \varsigma_2)$, $\varsigma_1 \neq \varsigma_2$, only if $\mathcal{I}(\varsigma_1, \varsigma_2)$ is a deep-intersection and hence z_0 is assigned to exactly one neighborhood, $\mathcal{N}(\varsigma_1) = \mathcal{N}(\varsigma_2)$, in this case. Thus we have the following lemma.

LEMMA 4. *Each cell* z_0 *in* $\mathbb{Z}^2$ *belongs to at most one neighborhood.*

COROLLARY 5. *For two distinct black cells* ς_1, ς_2, $\mathcal{N}(\varsigma_1) \cap \mathcal{N}(\varsigma_2) = \mathcal{N}(\varsigma_1) = \mathcal{N}(\varsigma_2)$ *iff* $\mathcal{I}(\varsigma_1, \varsigma_2)$ *is a deep-intersection, and* $\mathcal{N}(\varsigma_1) \cap \mathcal{N}(\varsigma_2) = \varnothing$ *iff* $\mathcal{I}(\varsigma_1, \varsigma_2)$ *is not a deep-intersection.*

As a consequence of Corollary 5, we will denote by $\mathcal{N}(\varsigma_1, \varsigma_2)$ the common neighborhood of two black cells ς_1, ς_2 for which $\mathcal{I}(\varsigma_1, \varsigma_2)$ is a deep-intersection.

LEMMA 5. *If* $\mathcal{S}(\varsigma_1) \cap \mathcal{S}(\varsigma_2)$ *is a tip, where* $\varsigma_1 = (\alpha, \beta)$, $\varsigma_2 = (\alpha + 2k, \beta)$, *then* $(\alpha + k, \beta) \in \mathcal{N}(\varsigma_1)$ *and* $(\alpha + k, \beta + 1) \in \mathcal{N}(\varsigma_2)$.

Proof. By the definition of $\mathcal{N}(\varsigma_1)$ it is obvious that $(\alpha + k, \beta) \in \mathcal{N}(\varsigma_1)$. By Lemma 12 (see Appendix B), $(\alpha + k, \beta + 1) \in \mathcal{P}(\varsigma_1, \varsigma_2)$; therefore by Corollary 1 we have that $(\alpha + k, \beta + 1)$ is a free cell. Thus, by the definition of $\mathcal{N}(\varsigma_2)$ we have that $(\alpha + k, \beta + 1) \in \mathcal{N}(\varsigma_2)$. □

2.3. The size of a neighborhood. In this subsection we give a lower bound on the sizes of the neighborhoods defined in subsection 2.2. We first sketch the outline

of the proof. There are two cases:

- If ς is a black cell for which $\mathcal{S}(\varsigma)$ does not have a deep-intersection with another sphere, we want to show a lower bound of $3k^2$ on $|\mathcal{N}(\varsigma)|$. If all the cells of $\mathcal{U}(\varsigma)$ are free, then $|\mathcal{N}(\varsigma) \cap \mathcal{S}(\varsigma)| + |\mathcal{U}(\varsigma)|$ is sufficient to obtain the bound. If $\mathcal{U}(\varsigma) \cap \mathcal{S}(\varsigma_1) \neq \varnothing$ (w.l.o.g. $\mathcal{U}_r(\varsigma) \cap \mathcal{S}(\varsigma_1) \neq \varnothing$) for some black cell ς_1, then $|\mathcal{N}(\varsigma) \cap \mathcal{P}(\varsigma, \varsigma_1)| + |\mathcal{U}_l(\varsigma) \setminus \mathcal{P}(\varsigma, \varsigma_1)|$ is sufficient to obtain the bound. This will complete the proof in most cases. If $\mathcal{U}_l(\varsigma) \cap \mathcal{S}(\varsigma_2) \neq \varnothing$ for some black cell ς_2, then in some cases we consider $|\mathcal{N}(\varsigma) \cap (\mathcal{P}(\varsigma, \varsigma_2) \setminus \mathcal{P}(\varsigma, \varsigma_1))|$.
- If ς_1, ς_2 are black cells for which $\mathcal{I}(\varsigma_1, \varsigma_2)$ is a deep-intersection, then we have to show a lower bound of $6k^2$ on $|\mathcal{N}(\varsigma_1, \varsigma_2)|$. In this case we consider $|\mathcal{N}(\varsigma_1, \varsigma_2) \cap \mathcal{P}(\varsigma_1, \varsigma_2)| + |(\mathcal{U}(\varsigma_1) \cup \mathcal{U}(\varsigma_2)) \setminus \mathcal{P}(\varsigma_1, \varsigma_2)|$. This will complete the proof in most cases. If $\mathcal{U}(\varsigma_1) \cap \mathcal{S}(\varsigma_3) \neq \varnothing$ (or $\mathcal{U}(\varsigma_2) \cap \mathcal{S}(\varsigma_4) \neq \varnothing$) for some black cell ς_3 (ς_4), then we consider $|\mathcal{N}(\varsigma_1, \varsigma_2) \cap (\mathcal{P}(\varsigma_1, \varsigma_3) \setminus \mathcal{P}(\varsigma_1, \varsigma_2))|$ (or $|\mathcal{N}(\varsigma_1, \varsigma_2) \cap (\mathcal{P}(\varsigma_2, \varsigma_4) \setminus \mathcal{P}(\varsigma_1, \varsigma_2))|$).

Each case will be proved in a separate lemma. For the proof of the first lemma, we also need the following result, which can be easily verified.

LEMMA 6. *Let $\varsigma_1 = (\alpha, \beta)$, $\varsigma_2 = (\gamma, \delta)$ be two black cells such that $\mathcal{I}(\varsigma_1, \varsigma_2)$ is not a deep-intersection. If $\mathcal{U}(\varsigma_1) \cap \mathcal{S}(\varsigma_2) \neq \varnothing$, then the following three conditions hold:*

$$\begin{array}{rcccl} 2k & \leq & |\gamma - \alpha| + |\delta - \beta| & < & 3k, \\ 0 & < & |\gamma - \alpha| & < & 2k, \\ 2 & \leq & \delta - \beta & \leq & 2k. \end{array}$$

Let $F(\varsigma)$ denote the set of free cells in $\mathcal{N}(\varsigma)$.

LEMMA 7. *If $k \geq 2$, and if ς is a black cell for which $\mathcal{S}(\varsigma)$ does not have a deep-intersection with another sphere, then $|\mathcal{N}(\varsigma)| \geq 3k^2$.*

Proof. Let ς be a black cell for which $\mathcal{S}(\varsigma)$ does not have a deep-intersection with another sphere. W.l.o.g. we can assume that ς is located at $(0, 0)$. We distinguish between two cases.

Case 1. $\mathcal{S}(\varsigma)$ does not intersect any other sphere $\mathcal{S}(\varsigma_1)$.

We have to compute the number of free cells in the neighborhood of ς. For this computation we first consider the set of cells $\mathcal{U}(\varsigma)$, defined earlier and depicted in Figure 4. We distinguish between the following subcases.

Case 1.1. All the cells of $\mathcal{U}(\varsigma)$ are free.

Therefore all the cells of $\mathcal{U}(\varsigma)$ belong to the neighborhood of ς. Hence $\mathcal{N}(\varsigma) \supseteq \mathcal{S}(\varsigma) \cup \mathcal{U}(\varsigma)$ and

$$|\mathcal{N}(\varsigma)| \geq (2k^2 + 2k + 1) + k(k - 1) = 3k^2 + k + 1 > 3k^2$$

as required.

In the following two subcases there is a sphere $\mathcal{S}(\varsigma_1)$, $\varsigma_1 = (\alpha, \beta)$, which intersects the set $\mathcal{U}(\varsigma)$. W.l.o.g. we can assume that if $\mathcal{S}(\varsigma_2)$, $\varsigma_2 = (\gamma, \delta)$, also intersects $\mathcal{U}(\varsigma)$, then $|\gamma| \geq |\alpha|$. W.l.o.g. we can also assume that $\alpha > 0$.

Case 1.2. $\alpha \leq k$.

We claim that the number of cells in the union of the sphere of ς with the free cells of $\mathcal{N}(\varsigma)$ inside the polygon of ς and ς_1, $\mathcal{P}(\varsigma, \varsigma_1)$, is at least $3k^2$, i.e.,

$$|(F(\varsigma) \cap \mathcal{P}(\varsigma, \varsigma_1)) \cup \mathcal{S}(\varsigma)| \geq 3k^2.$$

The set $(F(\varsigma) \cap \mathcal{P}(\varsigma, \varsigma_1)) \cup \mathcal{S}(\varsigma)$ (an example is given in Figure 5), which is only a

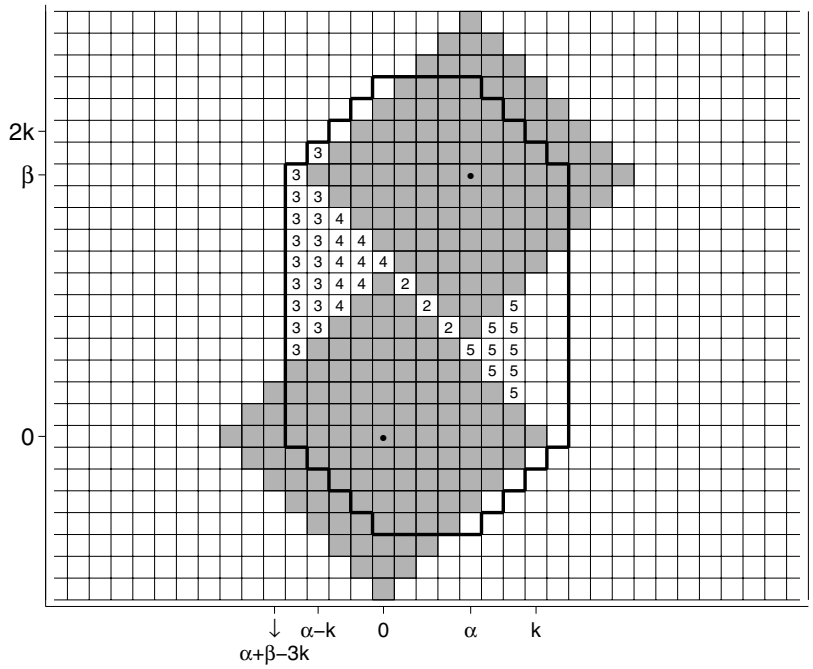


FIG. 5. *Case* 1.2 *with* $(\alpha, \beta) = (4, 12)$ *and* $k = 7$.

subset of $\mathcal{N}(\varsigma)$, contains the following disjoint subsets of cells:

(1) The sphere of ς, $\mathcal{S}(\varsigma)$, whose size is $2k^2 + 2k + 1$.
(2) The free cells of $\mathcal{N}(\varsigma)$ whose columns are between the top of $\mathcal{S}(\varsigma)$ and the bottom of $\mathcal{S}(\varsigma_1)$ (exclusive). This set of cells,

$$\{(x, y) : 0 < x < \alpha,\ k < x + y < \alpha + \beta - k\},$$

defines a parallelogram whose size is $(\alpha - 1)(\alpha + \beta - 2k - 1)$.
(3) The free cells of $\mathcal{N}(\varsigma)$ inside $\mathcal{P}(\varsigma, \varsigma_1)$ whose columns are between the left column of $\mathcal{P}(\varsigma, \varsigma_1)$ and the left tip of $\mathcal{S}(\varsigma_1)$ (inclusive). This set of cells,

$$\{(x, y) : \alpha + \beta - 3k + 1 \le x \le \alpha - k,\ k < y - x < 3k - \alpha\} \setminus \{(\alpha - k, \beta)\},$$

defines a parallelogram with a missing cell, $(\alpha - k, \beta)$, which is the left tip of $\mathcal{S}(\varsigma_1)$. The size of this set is $(2k - \beta)(2k - \alpha - 1) - 1$ if $\beta < 2k$ and 0 if $\beta = 2k$.
(4) The free cells of $\mathcal{N}(\varsigma)$ whose columns are between the left tip of $\mathcal{S}(\varsigma_1)$ (exclusive) and the top of $\mathcal{S}(\varsigma)$ (inclusive). This set is an arithmetic progression whose size is $(k - \alpha)(\beta - k - 2)$.
(5) The free cells of $\mathcal{N}(\varsigma)$ whose columns are between the bottom of $\mathcal{S}(\varsigma_1)$ (inclusive) and the right tip of $\mathcal{S}(\varsigma)$ (exclusive). This set is an arithmetic progression whose size is $(k - \alpha)(\beta - k - 2)$.

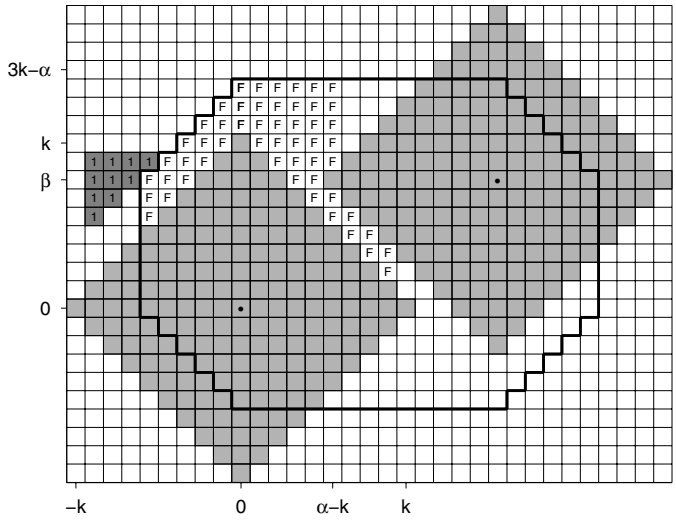


FIG. 6. *Case* 1.3 *with* $(\alpha, \beta) = (14, 7)$ *and* $k = 9$.

Note that all the sets in (2) through (5) are indeed contained in $\mathcal{P}(\varsigma, \varsigma_1)$. Therefore

$$\begin{aligned} &|(F(\varsigma) \cap \mathcal{P}(\varsigma, \varsigma_1)) \cup \mathcal{S}(\varsigma)| \\ &\geq 2k^2 + 2k + 1 \\ &+ (\alpha - 1)(\alpha + \beta - 2k - 1) \\ &+ (2k - \beta)(2k - \alpha - 1) - 1 \\ &+ 2(k - \alpha)(\beta - k - 2) \\ &= 4k^2 - 2k(\alpha + 1) + (\alpha + 1)^2. \end{aligned}$$

The minimum of $4k^2 - 2k(\alpha + 1) + (\alpha + 1)^2$ is when $\alpha + 1 = k$, and hence

$$|\mathcal{N}(\varsigma)| \geq |(F(\varsigma) \cap \mathcal{P}(\varsigma, \varsigma_1)) \cup \mathcal{S}(\varsigma)| \geq 3k^2$$

as claimed.

Case 1.3. $\alpha > k$.

In a similar way to Case 1.2 we consider the set $(F(\varsigma) \cap \mathcal{P}(\varsigma, \varsigma_1)) \cup \mathcal{S}(\varsigma)$ (an example is given in Figure 6) and compute its size. We obtain

$$|(F(\varsigma) \cap \mathcal{P}(\varsigma, \varsigma_1)) \cup \mathcal{S}(\varsigma)| \geq \frac{5k^2}{2} + \frac{k}{2}(2\alpha - 1) + \frac{1}{2}(-\alpha^2 + \alpha + 2).$$

Next, we consider $\mathcal{U}_l(\varsigma) \setminus \mathcal{P}(\varsigma, \varsigma_1)$, which contains the following isosceles right triangle (depicted in Figure 6):

$$TR_1 = \{(x, y) : -k + 1 \leq x,\ y \leq k - 1,\ y - x \geq 3k - \alpha\}.$$

The size of TR_1 is $\frac{1}{2}(\alpha - k)(\alpha - k - 1)$. If all the cells in $\mathcal{U}_l(\varsigma) \setminus \mathcal{P}(\varsigma, \varsigma_1)$ are free, then they are clearly free cells of $\mathcal{N}(\varsigma)$. Hence

$$|\mathcal{N}(\varsigma)| \geq |(F(\varsigma) \cap \mathcal{P}(\varsigma, \varsigma_1)) \cup \mathcal{S}(\varsigma)| + |TR_1| \geq 3k^2 + 1.$$

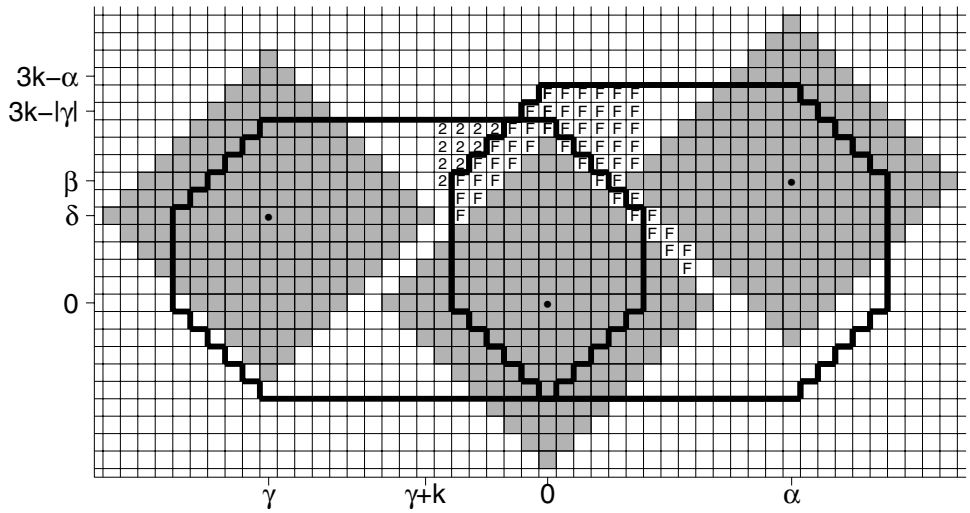


FIG. 7. *Case* 1.3 *with* $(\alpha,\beta)=(14,7)$, $(\gamma,\delta)=(-16,5)$, $k=9$. *The cells marked by "*2*" belong to* TR_2.

To complete the proof we have to consider the case of a black cell $\varsigma_2=(\gamma,\delta)$ such that $\mathcal{S}(\varsigma_2)\cap(\mathcal{U}_l(\varsigma)\setminus\mathcal{P}(\varsigma,\varsigma_1))$ is not empty. By the minimality of α we have $|\gamma|\geq\alpha$ and hence $|\gamma|>k$. We consider now another isosceles right triangle (depicted in Figure 7):

$$TR_2=\{(x,y):\gamma+k+1\leq x,\ y\leq 3k-|\gamma|-1,\ y-x\geq 3k-\alpha\}.$$

Note that $TR_2\subset\mathcal{P}(\varsigma,\varsigma_2)$ since $\alpha\leq|\gamma|$ and TR_2 does not intersect $\mathcal{P}(\varsigma,\varsigma_1)\cup\mathcal{S}(\varsigma_2)$. Hence, all the cells of TR_2 are free cells of $\mathcal{N}(\varsigma)$. The size of TR_2 is $\frac{1}{2}(\alpha-k)(\alpha-k-1)=|TR_1|$. Therefore

$$|\mathcal{N}(\varsigma)|\geq|(F(\varsigma)\cap\mathcal{P}(\varsigma,\varsigma_1))\cup\mathcal{S}(\varsigma)|+|TR_2|\geq 3k^2+1,$$

which completes the proof of this case.

Case 2. $\mathcal{S}(\varsigma)$ intersects other spheres and any such intersection is a line-intersection.

The proof is very similar to the proof of Case 1. If each cell (x,y), $|x|+y=k$ and $y>0$, belongs only to $\mathcal{S}(\varsigma)$ (and not to other spheres), then by the definition of $\mathcal{N}(\varsigma)$ and Lemma 5 we have $\mathcal{S}(\varsigma)\subset\mathcal{N}(\varsigma)$ or $(\mathcal{S}(\varsigma)\cup\{(-k,1)\})\setminus\{(-k,0)\}\subset\mathcal{N}(\varsigma)$, and hence the proof is identical to the one of Case 1.

Therefore, we assume that there exists $(x,y)\in\mathcal{I}(\varsigma,\varsigma_1)$, $\varsigma\neq\varsigma_1$, $\varsigma_1=(\alpha,\beta)$, $y>0$, and $|x|+y=k$. W.l.o.g. we assume that $\alpha\geq 0$. We distinguish between the following subcases.

Case 2.1. $\alpha=0$.

By Lemma 1(b) we have $\beta=2k$ and hence $\mathcal{U}(\varsigma)\subset\mathcal{P}(\varsigma,\varsigma_1)$ by Lemma 3. It follows by Corollary 4 and the definition of $\mathcal{N}(\varsigma)$ that $\mathcal{N}(\varsigma)\supset(\mathcal{S}(\varsigma)\cup\mathcal{U}(\varsigma))\setminus\{(0,k),(-k,0)\}$. Thus, as in Case 1.1 we have $|\mathcal{N}(\varsigma)|\geq 3k^2+k-1>3k^2$, as required.

Case 2.2. $0<\alpha\leq k$.

The reasonings are identical to the ones in Case 1.2 with the following exceptions:

- The value in (2) is negative since it represents part of $\mathcal{I}(\varsigma,\varsigma_1)$ rather than free cells, which should be substracted from $\mathcal{S}(\varsigma)$ and, as a consequence, from $\mathcal{N}(\varsigma)$.

- Also in (4) one cell belongs to $\mathcal{I}(\varsigma, \varsigma_1)$, which causes the arithmetic progression to start in -1. The same is true for (5).
- By Corollary 4, $\mathcal{S}(\varsigma)$ can intersect another sphere $\mathcal{S}(\varsigma_2)$. If $\mathcal{I}(\varsigma, \varsigma_2)$ is a tip of both spheres, then we can always assume for the sake of the proof that $\mathcal{I}(\varsigma, \varsigma_2) \subset \mathcal{N}(\varsigma)$ by Lemma 5. By Corollary 4 and the definition of $\mathcal{N}(\varsigma)$, $\mathcal{I}(\varsigma, \varsigma_2) \subset \mathcal{N}(\varsigma)$ also if $\mathcal{I}(\varsigma, \varsigma_2)$ is not a tip. Therefore, we compute the size of $\mathcal{S}(\varsigma)$ as part of the size of $\mathcal{N}(\varsigma)$ and substract $\mathcal{I}(\varsigma, \varsigma_1)$ from $\mathcal{N}(\varsigma)$ in (2), (4), and (5).

$*$ The only case where the computation is different is when $\alpha = k$. In this case the values in (4) and (5) should be -1 and not 0. In (1) the left tip of $\mathcal{S}(\varsigma_1)$, $(0, k)$, does not belong to the set of cells, and hence the size of the set should be $k(k-1)$ and not $k(k-1)-1$.

Therefore,

$$|\mathcal{N}(\varsigma)| \geq 4k^2 - 2k(\alpha+1) + (\alpha+1)^2 - 1 = 3k^2.$$

Case 2.3. $\alpha > k$.

In a similar way to Cases 1.3 and 2.2 we consider the set $(F(\varsigma) \cap \mathcal{P}(\varsigma, \varsigma_1)) \cup (\mathcal{S}(\varsigma) \setminus \mathcal{I}(\varsigma, \varsigma_1))$ and compute its size. We obtain

$$|(F(\varsigma) \cap \mathcal{P}(\varsigma, \varsigma_1)) \cup (\mathcal{S}(\varsigma) \setminus \mathcal{I}(\varsigma, \varsigma_1))| \geq \frac{5k^2}{2} + \frac{k}{2}(2\alpha - 1) + \frac{1}{2}(-\alpha^2 + \alpha).$$

By Lemmas 13, 14, and 15 (see Appendix B), we have that

$$|F(\varsigma) \setminus \mathcal{P}(\varsigma, \varsigma_1)| \geq \frac{1}{2}(\alpha - k)(\alpha - k - 1)$$

and hence

$$|\mathcal{N}(\varsigma)| \geq |(F(\varsigma) \cap \mathcal{P}(\varsigma, \varsigma_1)) \cup (\mathcal{S}(\varsigma) \setminus \mathcal{I}(\varsigma, \varsigma_1))| + |F(\varsigma) \setminus \mathcal{P}(\varsigma, \varsigma_1)| \geq 3k^2.$$

This completes the proof of the lemma. □

Note that Case 1.3 can be solved similarly to Case 2.3, but the proof given in Case 1.3 is much simpler.

The proof of the following lemma has some similarity to the proof of Lemma 7. It is also very detailed, and hence it will be given in Appendix B.

LEMMA 8. *If $k \geq 2$, and if ς_1 and ς_2 are two black cells such that $\mathcal{I}(\varsigma_1, \varsigma_2)$ is a deep-intersection, then $|\mathcal{N}(\varsigma_1, \varsigma_2)| > 6k^2$.*

Proof of Theorem 1. If $k = 1$, the proof is trivial, and we leave it to the reader. Therefore, we assume that $k \geq 2$. Let $A(n)$ be any $n \times n$ subarray of $\mathbb{Z}^2$. For a color i let μ_i be the density of the cells in $\mathbb{Z}^2$ colored by i, i.e.,

$$\mu_i = \lim_{n \to \infty} \sup \frac{|\mathcal{F}^{-1}(i) \cap A(n)|}{n^2}.$$

Let χ be the number of colors in $\mathcal{F}$. Clearly, $\sum_{i=1}^{\chi} \mu_i = 1$, and by Lemmas 7 and 8 $\mu_i \leq \frac{1}{3k^2}$ for each i, $1 \leq i \leq \chi$. Hence, $1 = \sum_{i=1}^{\chi} \mu_i \leq \frac{\chi}{3k^2}$, and thus $\chi \geq 3k^2$. □

3. An infinite family of optimal coloring. In this section we consider again only $4k$-colorings. We already know that any such coloring requires $3k^2$ colors. We would like to identify all the optimal $4k$-colorings, i.e., $4k$-colorings with $3k^2$ colors. First, note that the size of a neighborhood is at least $3k^2$.

CONJECTURE 1. *In an optimal $4k$-coloring each neighborhood has size $3k^2$.*

Remarks.

1. Note that a neighborhood can be defined for each cell of $\mathbb{Z}^2$ and not just for black cells.
2. The idea contained in Conjecture 1 is that in an optimal $4k$-coloring the "average" size of a neighborhood for each cell is $3k^2$. This idea is preserved if we rephrase the conjecture as "In an optimal $4k$-coloring each neighborhood with one black cell has size $3k^2$, and each neighborhood with two black cells has size $6k^2$." However, by Lemma 8, the size of a neighborhood with two black cells is greater than $6k^2$. Hence, we consider only the case where all the neighborhoods have size $3k^2$.

A coloring will be called *strongly optimal* if it satisfies the conjecture. One optimal coloring defined by a lattice Λ^R was given in [3]. The generator matrix of this lattice is

$$G^R = \begin{pmatrix} k & k \\ 0 & 3k \end{pmatrix}.$$

An isomorphic lattice Λ^L which also defines an optimal coloring has the generator matrix

$$G^L = \begin{pmatrix} -k & k \\ 0 & 3k \end{pmatrix}.$$

The cells of a given color in a strongly optimal coloring have a certain structure, as will be proved in what follows. Examples are depicted in Figure 8. We define the following sets:

$$\begin{aligned} \mathcal{D}_0^R &= \{(i,i) : i \in \mathbb{Z}\}, \\ \mathcal{D}_j^R &= (0,j) + \mathcal{D}_0^R, \quad j \in \mathbb{Z}, \\ \mathcal{D}_0^L &= \{(-i,i) : i \in \mathbb{Z}\}, \\ \mathcal{D}_j^L &= (0,j) + \mathcal{D}_0^L, \quad j \in \mathbb{Z}. \end{aligned}$$

A *shift vector* $\overrightarrow{s} = (\dots, \overrightarrow{s}(-1), \overrightarrow{s}(0), \overrightarrow{s}(1), \dots)$ is a function $\overrightarrow{s} : \mathbb{Z} \to \{0, 1, \dots, k-1\}$. For each shift vector and integer h, $0 \le h \le 3k-1$, we define two one-to-one functions, $T^R_{\overrightarrow{s},h} : \Lambda^R \to \mathbb{Z}^2$, $T^L_{\overrightarrow{s},h} : \Lambda^L \to \mathbb{Z}^2$, as follows:

$$\begin{aligned} T^R_{\overrightarrow{s},h}((ik, 3jk+ik)) &= (ik + \overrightarrow{s}(j), h + 3jk + ik + \overrightarrow{s}(j)), \\ T^L_{\overrightarrow{s},h}((-ik, 3jk+ik)) &= (-ik - \overrightarrow{s}(j), h + 3jk + ik + \overrightarrow{s}(j)). \end{aligned}$$

The images $T^R_{\overrightarrow{s},h}(\Lambda^R)$ and $T^L_{\overrightarrow{s},h}(\Lambda^L)$ will be called *templates*.

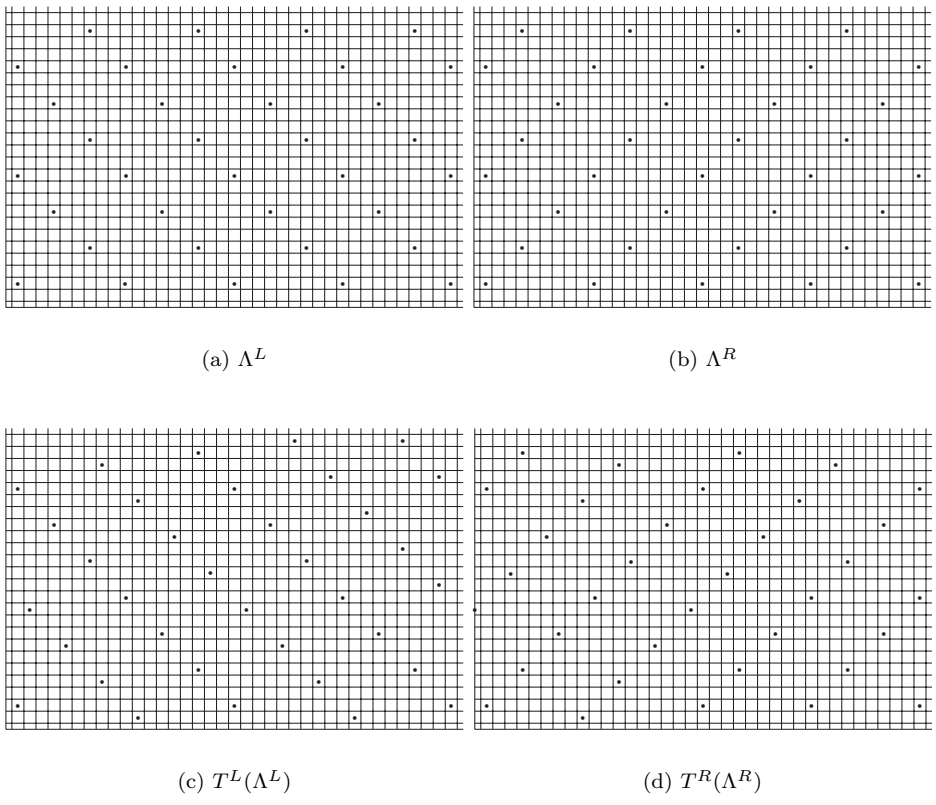
LEMMA 9. *If $\mathcal{F}$ is a strongly optimal coloring and φ is one of its colors, then the set of cells colored with φ, i.e., $\mathcal{F}^{-1}(\varphi)$, is a template.*

The proof of Lemma 9 is given in Appendix C. Examples of templates are given in Figure 8. Λ^R and Λ^L are templates with $h = 0$ and an allzero shift vector. Other templates are obtained from Λ^R in two steps:

- We lift the lattice by h; i.e., we obtain the set $(0,h) + \Lambda^R$.
- We shift the cells of $(0,h) + \Lambda^R$ in the diagonal $\mathcal{D}^R_{h+3jk}$ by $\overrightarrow{s}(j)$ to the right.

Similar templates are obtained from Λ^L.

Given a color φ, we say that φ is *R-oriented* (L-oriented) if the set of cells colored with φ is a template obtained from Λ^R (Λ^L).



(a) Λ^L (b) Λ^R

(c) $T^L(\Lambda^L)$ (d) $T^R(\Lambda^R)$

FIG. 8. *Lattices and templates.*

COROLLARY 6. *An R-oriented (L-oriented) color φ appears in the diagonal $\mathcal{D}_j^R$ ($\mathcal{D}_j^L$) iff it appears in the diagonal $\mathcal{D}_{j+3k}^R$ ($\mathcal{D}_{j+3k}^L$).*

LEMMA 10. *All the colors of the cells in the lattice $\Delta_{2k} = \{(ik, jk) : i + j \equiv 0 \pmod 2\}$ have the same orientation (R-oriented or L-oriented).*

Proof. Let $\varsigma_1 = (\alpha, \beta) \in \Delta_{2k}$; W.l.o.g. we assume that $\varphi_1 = \mathcal{F}(\varsigma_1)$ is R-oriented. By definition also $(\alpha + k, \beta + k)$ is colored with φ_1. Let $\varsigma_2 = (\alpha + k, \beta - k)$ and $\varphi_2 = \mathcal{F}(\varsigma_2)$. If φ_2 is L-oriented, then (α, β) is also colored with φ_2, a contradiction. Hence, φ_2 is R-oriented, and it can be easily verified that all the colors of the cells in Δ_{2k} have the same orientation. □

COROLLARY 7. *For $(\alpha, \beta) \in \mathbb{Z}^2$ all the colors of the cells in the set $(\alpha, \beta) + \Delta_{2k}$ have the same orientation.*

There are $2k^2$ disjoint cosets of Δ_{2k}. By Corollary 7, all the colors of the cells in a given coset have the same orientation. We say that the coset $(\alpha, \beta) + \Delta_{2k}$ is R-oriented (L-oriented) if the colors of the cells in the coset are R-oriented (L-oriented). We say that a cell $(\alpha, \beta) \in \mathbb{Z}^2$ is R-oriented (L-oriented) if it belongs to an R-oriented (L-oriented) coset. One can easily verify that a possible set of coset representatives is the set $\{(-j + i, j + i) : 0 \le i, j < k\} \cup ((0, 1) + \{(-j + i, j + i) : 0 \le i, j < k\})$. Note that the coset representative $(-j + i, j + i)$ lies in the intersection of the lines $y = x + 2j$ and $y = -x + 2i$ (an example is depicted in Figure 9). Clearly, we have the following lemma.

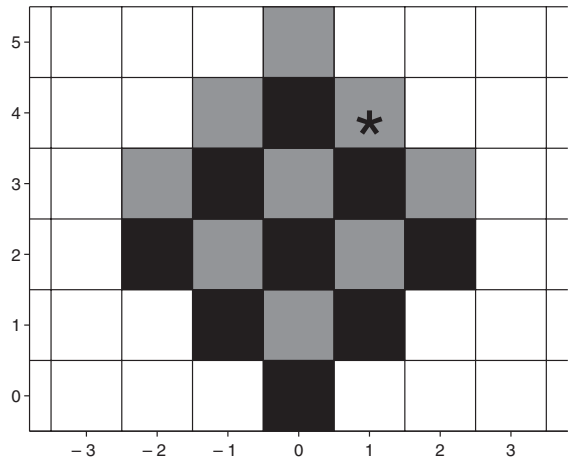


FIG. 9. *The set of coset representatives for $k = 3$. The coset representative marked by $*$ is $(1, 4) = (-1+2, 1+2+1)$; it lies in the intersection of the lines $y = x+2\cdot 1+1$ and $y = -x+2\cdot 2+1$.*

LEMMA 11. *The number of R-oriented (L-oriented) colors on the diagonal $\mathcal{D}_j^R$ ($\mathcal{D}_j^L$) is equal to the number of R-oriented (L-oriented) cosets of Δ_{2k} which intersect $\mathcal{D}_j^R$ ($\mathcal{D}_j^L$).*

Since $(0, 2k) \in \Delta_{2k}$, it follows from Lemma 11 that the number of R-oriented (L-oriented) colors on the diagonal $\mathcal{D}_j^R$ ($\mathcal{D}_j^L$) is equal to the number of R-oriented (L-oriented) cosets of Δ_{2k} which intersect $\mathcal{D}_{j-2k}^R$ ($\mathcal{D}_{j-2k}^L$). Hence, by Corollary 6, we have the following corollary.

COROLLARY 8. *The number of R-oriented (L-oriented) cosets of Δ_{2k} which intersect $\mathcal{D}_j^R$ ($\mathcal{D}_j^L$) is equal to the number of R-oriented (L-oriented) cosets of Δ_{2k} which intersect $\mathcal{D}_{j+k}^R$ ($\mathcal{D}_{j+k}^L$).*

Since the number of cosets of Δ_{2k} which intersect $\mathcal{D}_j^R$ ($\mathcal{D}_j^L$) is exactly k, we have the following corollary.

COROLLARY 9. *The number of R-oriented (L-oriented) cosets of Δ_{2k} which intersect $\mathcal{D}_j^L$ ($\mathcal{D}_j^R$) is equal to the number of R-oriented (L-oriented) cosets of Δ_{2k} which intersect $\mathcal{D}_{j+k}^L$ ($\mathcal{D}_{j+k}^R$).*

An example for possible assignment of orientations to the cosets as forced by Corollaries 8 and 9 is given in Figure 10. Once the orientation of each coset is determined, we can color $\mathbb{Z}^2$. We first color all the R-oriented cells. Consider the parallelogram $M^R = \{(x, y) : 0 \le x < k,\ 0 \le y - x < 3k\}$. By Lemma 9, each R-oriented color appears exactly once in M^R. Let ψ be the number of colors which are R-oriented. We assign the ψ R-oriented colors arbitrarily to R-oriented cells in M^R. By Lemma 9, $\mathcal{F}((\alpha, \beta)) = \mathcal{F}((\alpha + k, \beta + k))$, and hence we have an assignment of colors to all the R-oriented cells in the strip $\{(x, y) : 0 \le y - x < 3k\}$. By Corollary 6, $\mathcal{D}_j^R$ and $\mathcal{D}_{j+3k}^R$ have the same R-oriented colors, but the assignment of the R-oriented colors in one diagonal is independent in the assignment in the other diagonals; i.e., in each diagonal $\mathcal{D}_j^R$ the order of the R-oriented colors along the diagonal can be different. In the same way we assign the L-oriented colors to the L-oriented cells, by using a parallelogram $M^L = \{(x, y) : 0 \le x < k,\ 0 \le x + y < 3k\}$.

We will now give a procedure which will enable us to determine all the possible assignments of orientations to the $2k^2$ cosets of Δ_{2k}. Recall that each coset $(-j +$

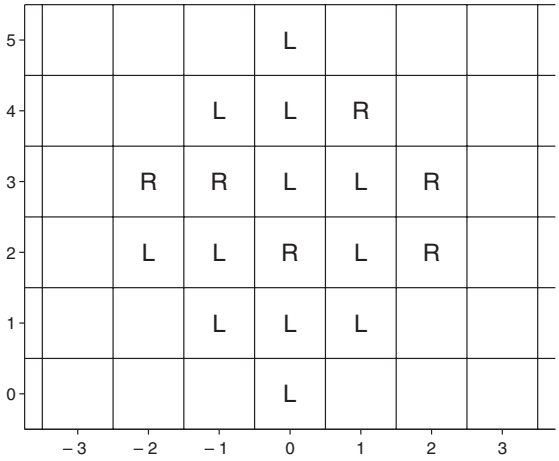


FIG. 10. *A possible assignment of orientations to the coset representatives for* $k = 3$*, which satisfies Corollaries* 8 *and* 9.

$i, j+i)$ (or $(-j+i, j+i+1)$) is associated with the lines $y = x+2j$ $(y = x+2j+1)$ and $y = -x+2i$ $(y = -x+2i+1)$.

We define a bipartite graph $G(V, E)$ (coset orientation graph) as follows: $V = V^L \cup V^R$, where

$$V^L = \{(2i+b, L) : i \in \mathbb{Z}_k,\ b \in \mathbb{Z}_2\},$$
$$V^R = \{(2j+b, R) : j \in \mathbb{Z}_k,\ b \in \mathbb{Z}_2\},$$
$$E = \{\{(2i+b, L),\ (2j+b, R)\} : i, j \in \mathbb{Z}_k,\ b \in \mathbb{Z}^2\}.$$

The edge $\{(2i+b, L),\ (2j+b, R)\}$ corresponds to the coset representative $(-j+i, j+i+b)$, which lies on the intersection of the lines $y = x+2j+b$ and $y = -x+2i+b$.

Each edge will get an assignment of either R or L, where the assignment will indicate the orientation of the corresponding coset. The only constraints on any assignment of orientations are Corollaries 8 and 9. For a vertex $v \in V$, let $\deg_L[v]$ ($\deg_R[v]$) denote the number of L-oriented edges incident to v. Note that $\deg_L[v] + \deg_R[v] = k$ for each $v \in V$. Corollary 8 implies that for $v = (\xi, L)$, $0 \le \xi \le k-1$, $\deg_L[(\xi, L)] = \deg_L[(\xi+k, L)]$, and Corollary 9 implies that for $v = (\xi, R)$, $0 \le \xi \le k-1$, $\deg_L[(\xi, R)] = \deg_L[(\xi+k, R)]$.

The procedure given below is not deterministic. It will produce all possible assignments. However, note that most of the assignments can be generated by different choices of the procedure.

THE ASSIGNMENT PROCEDURE.

- **Initialization:** $\deg_L[v] = 0$ for all $v \in V$.

(P.1) If $\deg_L[v] = k$ for all $v \in V$, then stop.
Either goto (P.2) or goto (P.5).

(P.2) Let $v_1 = (\xi, L)$ and $v_2 = (\xi+k, L)$ be two vertices such that $\deg_L[v_1] < k$.
Let $side := R$.

(P.3) Let $u_1 = (\xi_1, side)$ be a vertex such that $\deg_L[u_1] < k$.
Assign L to the edge $\{v_1, u_1\}$; increase $\deg_L[v_1]$ and $\deg_L[u_1]$.
Let $u_2 = (\xi_2, side)$ be a vertex such that $\deg_L[u_2] < k$.
Assign L to the edge $\{v_2, u_2\}$; increase $\deg_L[v_2]$ and $\deg_L[u_2]$.
If $u_1 \neq u_2$ and $\xi_1 \equiv \xi_2 \pmod{k}$, then goto (P.1).

(P.4) $v_1 := (\xi_1 + k \pmod{2k}, side)$.
$v_2 := (\xi_2 + k \pmod{2k}, side)$.
If $side = R$, then $side := L$, else $side := R$.
Goto (P.3).

(P.5) Assign R to all the edges which have not been assigned.

Note that the constraints of Corollaries 8 and 9 are satisfied since after (P.3) is performed either $\deg_L[(\xi, L)] = \deg_L[(\xi + k, L)]$ for all $\xi,\ 0 \le \xi \le k-1$, or $\deg_L[(\xi, R)] = \deg_L[(\xi + k, R)]$ for all $\xi,\ 0 \le \xi \le k-1$. Whenever (P.1) is reached both conditions hold.

From the discussion we have, it is clear that all the strongly optimal colorings are derived from "the assignment procedure." If Conjecture 1 is true, then these are all the optimal colorings. Hence we have the following conjecture.

CONJECTURE 2. *All the optimal $4k$-colorings are derived from "the assignment procedure."*

4. Some related results.

4.1. t-colorings with $t \neq 4k$.

THEOREM 2. *If $\mathcal{F}$ is a $(4k+2)$-coloring of $\mathbb{Z}^2$, then the number of colors in $\mathcal{F}$ is at least $3k^2 + 3k + 1$.*

Proof. Assume that for some k there exists a $(4k+2)$-coloring $\mathcal{F}$ of $\mathbb{Z}^2$ with $3k^2 + 3k$ colors. Let A be the set of $3k^2 + 3k$ colors of $\mathcal{F}$. We define the following coloring $\mathcal{F}' : \mathbb{Z}^2 \to A \times \mathbb{Z}_4$:

$$\mathcal{F}'((x,y)) = \begin{cases} \left(\mathcal{F}\left(\left\lfloor \frac{x}{2} \right\rfloor, \left\lfloor \frac{x}{2} \right\rfloor\right), 0\right), & x, y \text{ even}, \\ \left(\mathcal{F}\left(\left\lfloor \frac{x}{2} \right\rfloor, \left\lfloor \frac{x}{2} \right\rfloor\right), 1\right), & x \text{ even}, y \text{ odd}, \\ \left(\mathcal{F}\left(\left\lfloor \frac{x}{2} \right\rfloor, \left\lfloor \frac{x}{2} \right\rfloor\right), 2\right), & x \text{ odd}, y \text{ even}, \\ \left(\mathcal{F}\left(\left\lfloor \frac{x}{2} \right\rfloor, \left\lfloor \frac{x}{2} \right\rfloor\right), 3\right), & x, y \text{ odd}. \end{cases}$$

F' is an $(8k+4)$-coloring of $\mathbb{Z}^2$ with $12k^2 + 12k$ colors. However, by Theorem 1 an $(8k+4)$-coloring requires at least $3(2k+1)^2 = 12k^2 + 12k + 3$ colors, a contradiction. □

CONJECTURE 3.

- *If $\mathcal{F}$ is a $(4k+1)$-coloring of $\mathbb{Z}^2$, then the number of colors in $\mathcal{F}$ is at least $3k^2 + 2k$.*
- *If $\mathcal{F}$ is a $(4k+3)$-coloring of $\mathbb{Z}^2$, then the number of colors in $\mathcal{F}$ is at least $3k^2 + 5k + 2$.*

Colorings defined by lattices which attain the bounds of Theorem 3 and Conjecture 3 were given in [3]. A proof for Conjecture 3 will imply the optimality of colorings in a graph similar to the grid graph (see [6]).

4.2. Colorings in finite arrays. A t-coloring $\mathcal{F}$ can be defined similarly on finite arrays. Theorem 1 can be also proved on finite arrays.

THEOREM 3. *If $\mathcal{F}$ is a $4k$-coloring of a large enough $m \times n$ array, then the number of colors in $\mathcal{F}$ is at least $3k^2$.*

Proof. Let $L = 3k-1$, $R = 3k-1$, $D = 3k-1$, and $U = 4k-2$, and let m, n be integers such that

$$0 < m - U - D, \tag{2}$$

$$0 < n - R - L, \tag{3}$$

$$3k^2 - 1 < 3k^2 \frac{(m-U-D)(n-R-L)}{mn}. \tag{4}$$

Let A be an $m \times n$ array and $\mathcal{F}$ be a $4k$-coloring of A with χ colors. W.l.o.g. A is bounded by the cells $(-L,-D),(-L,-D+m),(-L+n,-D+m),(-L+n,-D)$. Let B be the $(m-U-D)\times(n-R-L)$ subarray of A, which is bounded by the cells $(0,0),(0,m-U-D),(n-R-L,m-U-D),(n-R-L,0)$. Note that by the detailed proofs of Lemmas 7 and 8, the neighborhood of any cell in B is contained in A.

For each color i, let a_i be the number of cells in B colored by i. By Lemmas 7 and 8, $3k^2 a_i \le mn$ for each color i. Obviously,

$$\sum_{i=1}^{\chi} a_i = (m-U-D)(n-R-L).$$

Hence

$$3k^2(m-U-D)(n-R-L) = 3k^2 \sum_{\varphi=1}^{\chi} a_\varphi \le \chi mn,$$

$$3k^2 - 1 < 3k^2 \frac{(m-U-D)(n-R-L)}{mn} \le \chi. \qquad \square$$

Clearly, Theorem 1 is an immediate consequence of Theorem 3, and hence we have an alternative proof of Theorem 1.

5. Conclusions. We have proved that the number of colors in a $4k$-coloring is at least $3k^2$. We have shown an infinite family of $4k$-colorings which attain this bound.

The original question can be asked more generally.

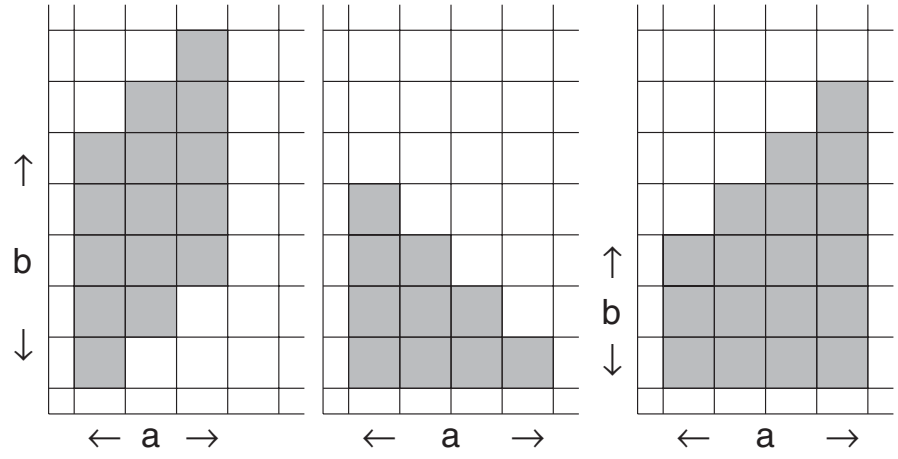
Given a positive integer r and a positive integer t, $t \ge r$, what is the smallest number of colors required to color $\mathbb{Z}^2$, such that for any r points colored with the same color, the size of the minimum spanning tree which connects them is at least t?

In this paper we discussed the case $r = 3$. Colorings with lattices for $r = 2$ are discussed in [1] and for $r \ge 3$ are discussed in [3]. For $r = 2$ all the optimal colorings can be identified with techniques which are similar to those used in section 3. Any generalization of the results in this paper for $r > 3$ would be very interesting. If a coloring defined by a lattice satisfies certain conditions, then the technique presented in section 3 can be used to obtain an infinite family of colorings with the same number of colors. Some of the colorings defined by lattices and conjectured to be optimal in [3] satisfy these conditions.

Appendix A. In the proofs of Lemmas 7 and 8 we use the areas of some geometric shapes. As these shapes are in fact polyominoes, their size is not necessarily equal to the size of the standard geometric shape.

- *Parallelogram.* The size of the parallelogram in Figure 11(a) is ab.
- *Isosceles right triangle.* The size of the isosceles right triangle in Figure 11(b), with legs of length a, is $\frac{a(a+1)}{2}$.
- *Right trapezoid.* In all the right trapezoids which are considered, the difference between any two consecutive columns is one. The size of the trapezoid in Figure 11(c) is $\frac{a(a+2b-1)}{2}$.

Note that the size of all these shapes can be computed as the sum of arithmetic progression. We also consider some arithmetic progressions with difference two between consecutive elements.



(a) A parallelogram. (b) An isosceles right triangle. (c) A right trapezoid.

FIG. 11. *Geometric shapes.*

Appendix B. In this appendix we will prove Lemma 8. First, we will give a few lemmas, some of which are also used in the proof of Lemma 7.

LEMMA 12. *Let* $\varsigma_1 = (\alpha, \beta)$, $\varsigma_2 = (\gamma, \delta)$ *be two black cells such that* $|\gamma - \alpha| \leq 2k$, $0 \leq \delta - \beta \leq 2k$, $d_2(\varsigma_1, \varsigma_2) = |\gamma - \alpha| + \delta - \beta < 3k$.

If $\gamma - \alpha \geq 0$, *then*

$$\mathcal{P}(\varsigma_1, \varsigma_2) = \left\{ (x, y) : \begin{array}{ccccc} \gamma + \delta - \beta - 3k & < & x & < & 3k + \alpha + \beta - \delta \\ \gamma + \delta - \alpha - 3k & < & y & < & 3k + \alpha + \beta - \gamma \\ \gamma + \delta - 3k & < & x + y & < & 3k + \alpha + \beta \\ \delta - \alpha - 3k & < & y - x & < & 3k + \beta - \gamma \end{array} \right\}.$$

If $\gamma - \alpha < 0$, *then*

$$\mathcal{P}(\varsigma_1, \varsigma_2) = \left\{ (x, y) : \begin{array}{ccccc} \alpha - \beta + \delta - 3k & < & x & < & 3k + \beta + \gamma - \delta \\ \alpha - \gamma + \delta - 3k & < & y & < & 3k - \alpha + \beta + \gamma \\ \alpha + \delta - 3k & < & x + y & < & 3k + \beta + \gamma \\ \delta - \gamma - 3k & < & y - x & < & 3k - \alpha + \beta \end{array} \right\}.$$

LEMMA 13. *Let* $\varsigma_0 = (0, 0)$ *and* $\varsigma_1 = (\alpha, \beta)$, $\alpha > k$, $\beta \geq 0$, *be two black cells. If* $\mathcal{I}(\varsigma_0, \varsigma_1) \neq \varnothing$, *then* $|\mathcal{U}_l(\varsigma_0) \setminus \mathcal{P}(\varsigma_0, \varsigma_1)| \geq \frac{1}{2}(\alpha - k)(\alpha - k + 1)$.

Proof.

$$\mathcal{U}_l(\varsigma_0) \setminus \mathcal{P}(\varsigma_0, \varsigma_1) \supseteq \{(x, y) : x \geq -k + 1,\ y \leq k,\ y - x \geq 3k - \alpha\},$$

which is an isosceles right triangle with legs of length $\alpha - k$. □

An example for Lemma 13 is depicted in Figure 12.

LEMMA 14. *Let* $\varsigma_0 = (0, 0)$ *and* $\varsigma_1 = (\alpha, \beta)$, $\alpha > k$, $\beta \geq 0$, *be two black cells, such that* $\mathcal{I}(\varsigma_0, \varsigma_1) \neq \varnothing$, *and let* $\varsigma_2 = (\gamma, \delta)$ *be a black cell such that* $\mathcal{U}_l(\varsigma_0) \cap \mathcal{S}(\varsigma_2) \neq \varnothing$. *If* $|\gamma| \geq k$, *then* $|\mathcal{N}(\varsigma_0) \cap (\mathcal{P}(\varsigma_0, \varsigma_2) \setminus \mathcal{P}(\varsigma_0, \varsigma_1))| \geq \frac{1}{2}(\alpha - k + 1)(\alpha - k) - 1$.

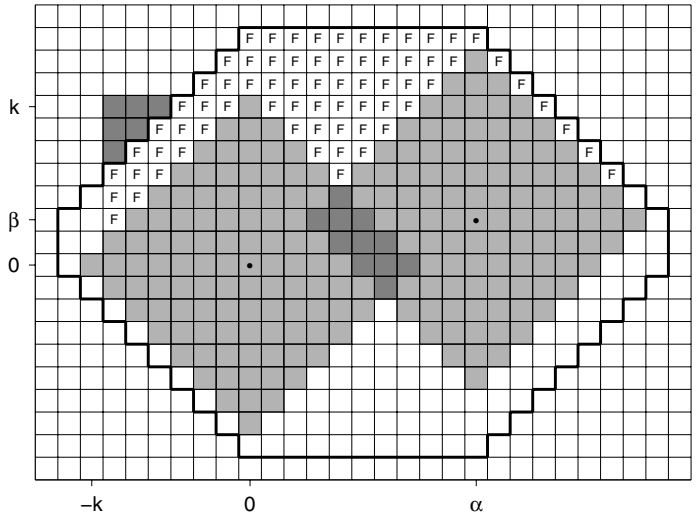


FIG. 12. *An example for Lemma* 13 *with* $(\alpha, \beta) = (10, 2)$ *and* $k = 7$.

Proof. We consider the set of free cells of $\mathcal{N}(\varsigma_0)$ inside $\mathcal{P}(\varsigma_0, \varsigma_2)$ and above $\mathcal{P}(\varsigma_0, \varsigma_1)$. This set contains the following two disjoint subsets V_1 and V_2:

$$V_1 = \{(x, y) : 1 \le x,\ y \ge 3k - \alpha,\ x + y \le 3k + \gamma - 1\},$$
$$V_2 = \{(x, y) : \gamma + k \le x \le 0,\ y \le 3k - |\gamma| - 1,\ y - x \ge 3k - \alpha\} \setminus \{(\gamma + k, \delta)\}.$$

If $\alpha > |\gamma| + 1$, then V_1 is an isosceles right triangle and V_2 is a right trapezoid with a missing cell, as depicted in Figure 13(a). If $\alpha \le |\gamma| + 1$, then V_1 is an empty set and V_2 is an isosceles right triangle with a missing cell, as depicted in Figure 13(b). In both cases

$$|\mathcal{N}(\varsigma_0) \cap (\mathcal{P}(\varsigma_0, \varsigma_2) \setminus \mathcal{P}(\varsigma_0, \varsigma_1))| \ge |V_1| + |V_2| = \frac{1}{2}(\alpha - k + 1)(\alpha - k) - 1. \qquad \square$$

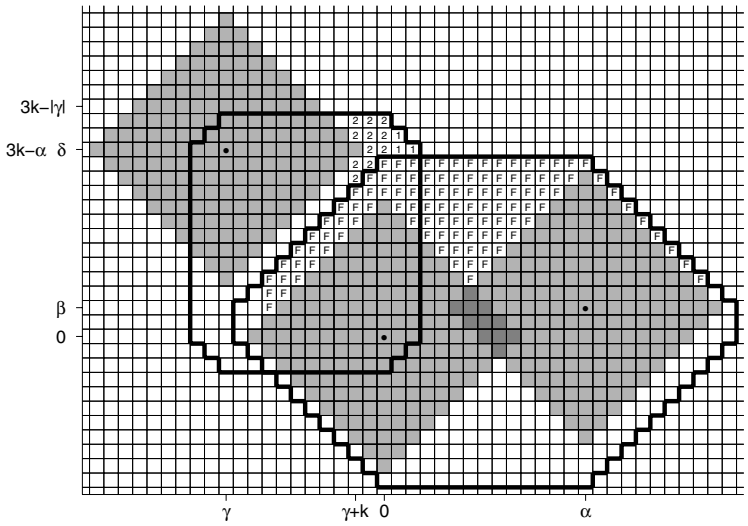
LEMMA 15. *Let* $\varsigma_0 = (0, 0)$ *and* $\varsigma_1 = (\alpha, \beta)$, $\alpha > k, \beta \ge 0$, *be two black cells, such that* $\mathcal{I}(\varsigma_0, \varsigma_1) \ne \varnothing$, *and let* $\varsigma_2 = (\gamma, \delta)$ *be a black cell such that* $\mathcal{U}_l(\varsigma_0) \cap \mathcal{S}(\varsigma_2) \ne \varnothing$. *If* $|\gamma| < k$, *then* $|\mathcal{N}(\varsigma_0) \cap (\mathcal{P}(\varsigma_0, \varsigma_2) \setminus \mathcal{P}(\varsigma_0, \varsigma_1))| \ge \frac{1}{2}(\alpha - k)(\alpha - k - 1)$.

Proof. Since $\mathcal{U}_l(\varsigma_0) \cap \mathcal{S}(\varsigma_2) \ne \varnothing$ it follows that $\delta \le 2k$. If $\delta < 2k$, we consider the set of free cells of $\mathcal{N}(\varsigma_0)$ inside $\mathcal{P}(\varsigma_0, \varsigma_2)$ and above $\mathcal{P}(\varsigma_0, \varsigma_1)$. This set contains the following two disjoint subsets V_1 and V_2:

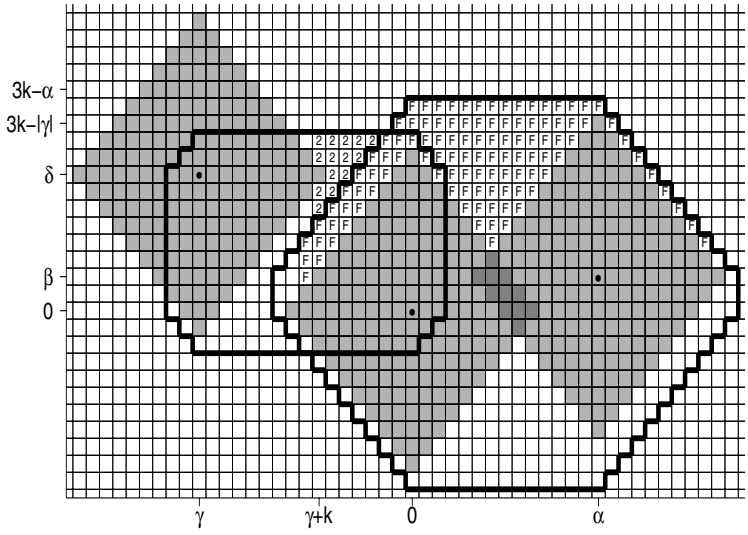
$$V_1 = \{(x, y) : x \le \gamma + k,\ y \ge 3k - \alpha,\ y - x \le \delta - \gamma - k - 1\},$$
$$V_2 = \{(x, y) : \gamma + k + 1 \le x \le 3k + \gamma - \delta - 1,\ y \ge 3k - \alpha,\ x + y \le 3k + \gamma - 1\}.$$

V_1 is an isosceles right triangle and V_2 is a right trapezoid, as depicted in Figure 14, and hence

$$|\mathcal{N}(\varsigma_0) \cap (\mathcal{P}(\varsigma_0, \varsigma_2) \setminus \mathcal{P}(\varsigma_0, \varsigma_1))| \ge |V_1| + |V_2| = \frac{1}{2}(\alpha - k)(\alpha - k - 1).$$



(a) $(\gamma, \delta) = (-11, 13)$



(b) $(\gamma, \delta) = (-16, 8)$

FIG. 13. *Lemma* 14 *applied on* $(\alpha, \beta) = (14, 2)$ *and* $k = 9$.

If $\delta = 2k$, then $\mathcal{P}(\varsigma_0, \varsigma_2)$ does not include cells of the form $(\gamma + k, y)$, and hence the set V_1 as defined above is not contained in $\mathcal{P}(\varsigma_0, \varsigma_2)$. Moreover, $|\mathcal{U}_l(\varsigma_0) \cap \mathcal{S}(\varsigma_2)| = 1$ and $\mathcal{N}(\varsigma_0) \cap \mathcal{P}(\varsigma_0, \varsigma_2) \supset \mathcal{U}_l(\varsigma_0) \setminus \mathcal{S}(\varsigma_2)$. Hence, it follows by Lemma 13 that

$$|\mathcal{N}(\varsigma_0) \cap (\mathcal{P}(\varsigma_0, \varsigma_2) \setminus \mathcal{P}(\varsigma_0, \varsigma_1))| \geq |\mathcal{U}_l(\varsigma_0) \setminus \mathcal{P}(\varsigma_0, \varsigma_1)| - 1 = \frac{1}{2}(\alpha - k + 1)(\alpha - k) - 1. \qquad \square$$

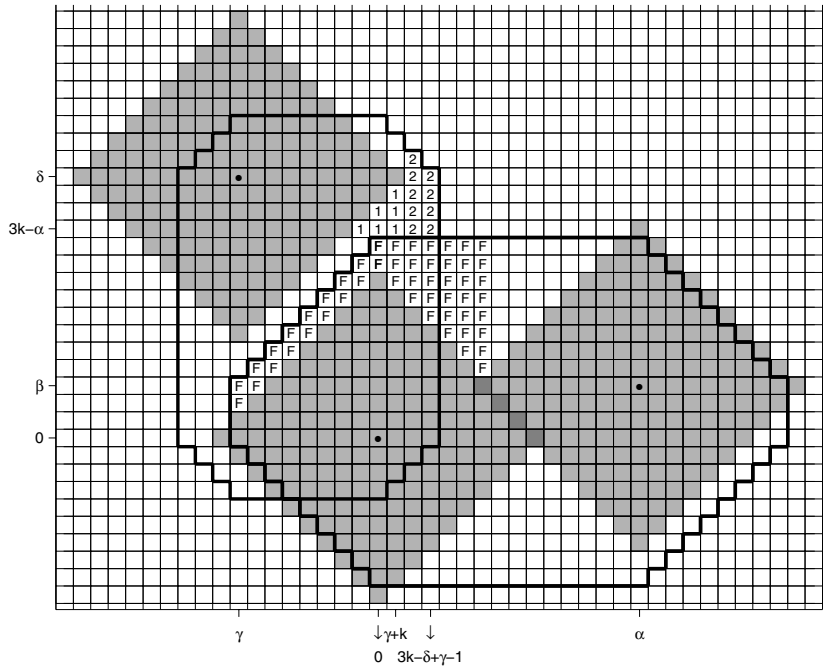


FIG. 14. *Case* 2.3 *with* $(\alpha,\beta)=(15,3)$, $(\gamma,\delta)=(-8,15)$, $k=9$. *The cells of* V_1 *and* V_2 *are marked "*1*" and "*2*."*

Note that in Lemmas 13, 14, and 15, if $\mathcal{I}(\varsigma_0,\varsigma_1)$ is a deep-intersection, then $\mathcal{N}(\varsigma_0)=\mathcal{N}(\varsigma_0,\varsigma_1)$.

We now proceed to prove more results which will be useful in the proof of Lemma 8. In what follows we assume that $\varsigma_0=(0,0)$ and $\varsigma_1=(\alpha,\beta)$, $\alpha,\beta\geq 0$, are black cells for which $\mathcal{I}(\varsigma_0,\varsigma_1)$ is a deep-intersection, i.e., $\alpha+\beta<2k$. First, we have to compute the size of $|\mathcal{I}(\varsigma_0,\varsigma_1)|$. We have found a few different methods to compute $|\mathcal{I}(\varsigma_0,\varsigma_1)|$; none of them is elegant. Therefore we leave the proof of the following lemma to the reader.

LEMMA 16.

$$|\mathcal{I}(\varsigma_0,\varsigma_1)|=\begin{cases}2(k-\beta+1)k-\beta+\left\lfloor\frac{\beta^2-\alpha^2}{2}\right\rfloor+1 & \text{if } \alpha<\beta,\\ 2(k-\alpha+1)k-\alpha+\left\lfloor\frac{\alpha^2-\beta^2}{2}\right\rfloor+1 & \text{if } \beta\leq\alpha.\end{cases}$$

For a black cell $\varsigma=(\gamma,\delta)$, let

$$T_l(\varsigma)=\{(\gamma-k,y):y>\delta,\ (\gamma-k,\eta)\text{ is a free cell for }\delta+1\leq\eta\leq y\},$$
$$T_r(\varsigma)=\{(\gamma+k,y):y>\delta,\ (\gamma+k,\eta)\text{ is a free cell for }\delta+1\leq\eta\leq y\}.$$

An example is depicted in Figure 15.

LEMMA 17. *If* $\alpha+\beta\leq k+1$, *then* $T_l(\varsigma_0)\subset\mathcal{N}(\varsigma_0,\varsigma_1)$ *and* $|\mathcal{P}(\varsigma_0,\varsigma_1)\cap T_l(\varsigma_0)|=2k-\alpha-1$.

Proof. A cell $z\in T_l(\varsigma_0)$ does not belong to $\mathcal{N}(\varsigma_0,\varsigma_1)$ if there exists a black cell ς_2 such that $\mathcal{U}(\varsigma_2)\cap\mathcal{S}(\varsigma_0)\neq\varnothing$ and $z\in\mathcal{P}(\varsigma_2,\varsigma_0)$. Assume that such a cell $\varsigma_2=(\gamma,\delta)$ exists. Clearly, $\gamma,\delta<0$. $d_3(\varsigma_0,\varsigma_1,\varsigma_2)=\alpha-\gamma+\beta-\delta\geq 4k$, and hence

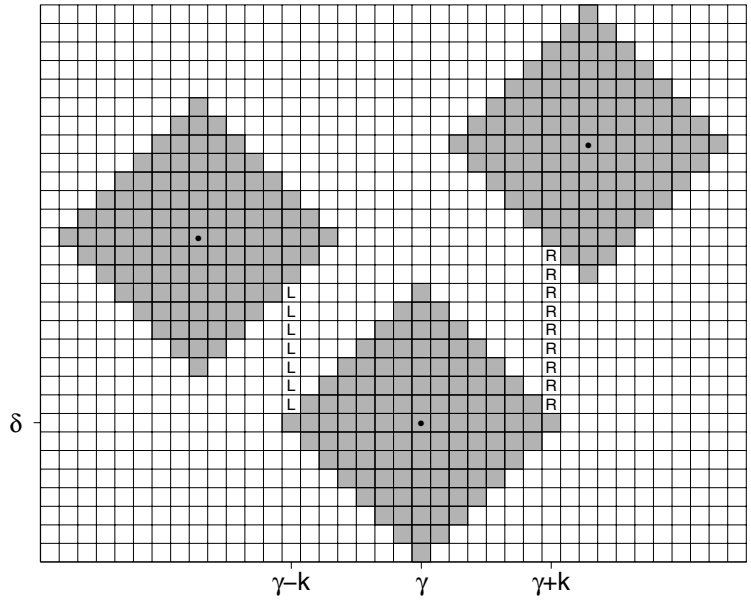


FIG. 15. $T_l(\varsigma)$ *and* $T_r(\varsigma)$.

$|\gamma| + |\delta| = -(\gamma + \delta) \geq 4k - (\alpha + \beta) \geq 3k - 1$. By Lemma 6, if $|\gamma| + |\delta| > 3k - 1$, then $\mathcal{U}(\varsigma_2) \cap \mathcal{S}(\varsigma_0) = \varnothing$. If $|\gamma| + |\delta| = 3k - 1$ and $\mathcal{U}(\varsigma_2) \cap \mathcal{S}(\varsigma_0) \neq \varnothing$, then $\mathcal{P}(\varsigma_2, \varsigma_0)$ is a rectangle and $z \notin \mathcal{P}(\varsigma_2, \varsigma_0)$, a contradiction.

Thus, such a black cell ς_2 does not exist, $T_l(\varsigma_0) \subset \mathcal{N}(\varsigma_0, \varsigma_1)$, and one can easily verify that $|\mathcal{P}(\varsigma_0, \varsigma_1) \cap T_l(\varsigma_0)| = 2k - \alpha - 1$. □

LEMMA 18. *If* $\alpha \leq k + 1$, $\alpha + \beta \geq k + 1$, *then* $|\mathcal{N}(\varsigma_0, \varsigma_1) \cap \mathcal{P}(\varsigma_0, \varsigma_1) \cap T_l(\varsigma_0)| \geq 3k - 2\alpha - \beta$.

Proof. A cell $z \in T_l(\varsigma_0)$ does not belong to $\mathcal{N}(\varsigma_0, \varsigma_1)$ if there exists a black cell ς_2 such that $\mathcal{U}(\varsigma_2) \cap \mathcal{S}(\varsigma_0) \neq \varnothing$ and $z \in \mathcal{P}(\varsigma_2, \varsigma_0)$. Assume that such a cell $\varsigma_2 = (\gamma, \delta)$ exists. Clearly, $\gamma, \delta < 0$.

By the definition of $\mathcal{P}(\varsigma_2, \varsigma_0)$, if $k < |\gamma| < 2k$, then

$$\mathcal{N}(\varsigma_2) \cap T_l(\varsigma_0) = \{(-k, y) : 1 \leq y \leq 3k - |\gamma| - |\delta| - 1\},$$

and if $|\gamma| \leq k$, then $\mathcal{N}(\varsigma_2) \cap T_l(\varsigma_0) = \{(-k, y) : 1 \leq y \leq 2k - |\delta| - 1\}$. Hence, $|\mathcal{N}(\varsigma_2) \cap T_l(\varsigma_0)| \leq 3k - |\gamma| - |\delta| - 1$. It is easy to verify that $|\mathcal{P}(\varsigma_0, \varsigma_1) \cap T_l(\varsigma_0)| = 2k - \alpha - 1$, and therefore

$$\begin{aligned} |\mathcal{N}(\varsigma_0, \varsigma_1) \cap \mathcal{P}(\varsigma_0, \varsigma_1) \cap T_l(\varsigma_0)| &\geq |\gamma| + |\delta| - \alpha - k \\ &= \alpha + \beta + |\gamma| + |\delta| - 2\alpha - \beta - k \\ &= d_3(\varsigma_0, \varsigma_1, \varsigma_2) - 2\alpha - \beta - k \\ &\geq 3k - 2\alpha - \beta. \end{aligned}$$

Note that $3k - 2\alpha - \beta \geq 0$ since $\alpha \leq k + 1$ and $\alpha + \beta < 2k$. □

LEMMA 19. *If* $\alpha \le k+1$, *then* $T_r(\varsigma_1) \subset \mathcal{N}(\varsigma_0,\varsigma_1)$ *and* $|\mathcal{P}(\varsigma_0,\varsigma_1) \cap T_r(\varsigma_1)| = 2k-\alpha-\beta-1$.

Proof. A cell $z \in T_r(\varsigma_1)$ does not belong to $\mathcal{N}(\varsigma_0,\varsigma_1)$ if there exists a black cell ς_2 such that $\mathcal{U}(\varsigma_2) \cap \mathcal{S}(\varsigma_1) \neq \varnothing$ and $z \in \mathcal{P}(\varsigma_2,\varsigma_1)$. Assume that such a cell $\varsigma_2 = (\gamma,\delta)$ exists. Clearly, $\alpha < \gamma$ and $\delta < \beta$.

If $\delta > 0$, then $\gamma - \alpha = d_3(\varsigma_0,\varsigma_1,\varsigma_2) - \alpha - \beta \ge 4k - (\alpha+\beta) > 2k$, and therefore by Lemma 6 we have $\mathcal{U}(\varsigma_2) \cap \mathcal{S}(\varsigma_1) = \varnothing$, a contradiction. If $\delta \le 0$, then $|\alpha-\gamma| + |\beta-\delta| = \gamma - \alpha + \beta - \delta = d_3(\varsigma_0,\varsigma_1,\varsigma_2) - \alpha \ge 4k - \alpha \ge 3k-1$. By Lemma 6, if $|\alpha-\gamma| + |\beta-\delta| > 3k-1$, then $\mathcal{U}(\varsigma_2) \cap \mathcal{S}(\varsigma_1) = \varnothing$. If $|\alpha-\gamma| + |\beta-\delta| = 3k-1$ and $\mathcal{U}(\varsigma_2) \cap \mathcal{S}(\varsigma_1) \neq \varnothing$, then $\mathcal{P}(\varsigma_2,\varsigma_1)$ is a rectangle and $z \notin \mathcal{P}(\varsigma_2,\varsigma_1)$, a contradiction.

Thus, such a black cell ς_2 does not exist, $T_r(\varsigma_1) \subset \mathcal{N}(\varsigma_0,\varsigma_1)$, and one can easily verify that $|\mathcal{P}(\varsigma_0,\varsigma_1) \cap T_r(\varsigma_1)| = 2k-\alpha-\beta-1$. □

LEMMA 20. *If* $\alpha < \beta \le k$, *then* $T_r(\varsigma_0) \subset \mathcal{N}(\varsigma_0,\varsigma_1)$ *and* $|\mathcal{P}(\varsigma_0,\varsigma_1) \cap T_r(\varsigma_0)| = \beta-\alpha-1$.

Proof. A cell $z \in T_r(\varsigma_0)$ does not belong to $\mathcal{N}(\varsigma_0,\varsigma_1)$ if there exists a black cell ς_2 such that $\mathcal{U}(\varsigma_2) \cap \mathcal{S}(\varsigma_0) \neq \varnothing$ and $z \in \mathcal{P}(\varsigma_2,\varsigma_0)$. Assume that such a cell $\varsigma_2 = (\gamma,\delta)$ exists. Clearly, $\gamma > 0$ and $\delta < 0$.

If $\gamma \le \alpha$, then $|\delta| = d_3(\varsigma_0,\varsigma_1,\varsigma_2) - \alpha - \beta \ge 4k - (\alpha+\beta) > 2k$, and by Lemma 6 we have $\mathcal{U}(\varsigma_2) \cap \mathcal{S}(\varsigma_0) = \varnothing$, a contradiction. If $\alpha < \gamma$, then $\gamma + |\delta| = d_3(\varsigma_0,\varsigma_1,\varsigma_2) - \beta \ge 4k - \beta \ge 3k$, and by Lemma 6 we have $\mathcal{U}(\varsigma_2) \cap \mathcal{S}(\varsigma_0) = \varnothing$, a contradiction.

Thus, such a black cell ς_2 does not exist, $T_r(\varsigma_0) \subset \mathcal{N}(\varsigma_0,\varsigma_1)$, and it is easy to verify that $|\mathcal{P}(\varsigma_0,\varsigma_1) \cap T_r(\varsigma_0)| = \beta-\alpha-1$. □

LEMMA 21. *If* $\alpha < \beta$ *and* $k < \beta$, *then* $|\mathcal{N}(\varsigma_0,\varsigma_1) \cap \mathcal{P}(\varsigma_0,\varsigma_1) \cap T_r(\varsigma_0)| \ge k-\alpha$.

Proof. A cell $z \in T_r(\varsigma_0)$ does not belong to $\mathcal{N}(\varsigma_0,\varsigma_1)$ if there exists a black cell ς_2 such that $\mathcal{U}(\varsigma_2) \cap \mathcal{S}(\varsigma_0) \neq \varnothing$ and $z \in \mathcal{P}(\varsigma_2,\varsigma_0)$. Assume that such a cell $\varsigma_2 = (\gamma,\delta)$ exists. Clearly, $\gamma > 0$ and $\delta < 0$.

By symmetry and similar arguments to those of Lemma 18 we have that $|\mathcal{N}(\varsigma_2) \cap \mathcal{P}(\varsigma_0,\varsigma_1) \cap T_r(\varsigma_0)| \le 2k - |\delta| - 1$ if $\gamma \le k$ and $|\mathcal{N}(\varsigma_2) \cap \mathcal{P}(\varsigma_0,\varsigma_1) \cap T_r(\varsigma_0)| \le 3k - \gamma - |\delta| - 1$ otherwise. $T_r(\varsigma_0) = \{(k,y) : 1 \le y \le \beta-\alpha-1\}$, and hence $|T_r(\varsigma_0)| = \beta-\alpha-1$. Therefore, we have the following:

- If $\gamma > \alpha$, then

$$\begin{aligned} |\mathcal{N}(\varsigma_0,\varsigma_1) \cap T_r(\varsigma_0)| &\ge (\beta-\alpha-1) - (3k-\gamma-|\delta|-1) \\ &= d_3(\varsigma_0,\varsigma_1,\varsigma_2) - \alpha - 3k \\ &\ge k-\alpha. \end{aligned}$$

- If $\gamma \le \alpha$, then $\gamma < k$;

$$\begin{aligned} |\mathcal{N}(\varsigma_0,\varsigma_1) \cap \mathcal{P}(\varsigma_0,\varsigma_1) \cap T_r(\varsigma_0)| &\ge (\beta-\alpha-1) - (2k-|\delta|-1) \\ &= d_3(\varsigma_0,\varsigma_1,\varsigma_2) - 2\alpha - 2k \\ &\ge 2k - 2\alpha \\ &> k-\alpha. \end{aligned}$$

Note that $k-\alpha > 0$ since $\alpha < \beta$ and $\alpha+\beta < 2k$. □

LEMMA 22. *If* $0 < \alpha < \beta$, *then*

$$|\mathcal{N}(\varsigma_0,\varsigma_1) \cap \mathcal{P}(\varsigma_0,\varsigma_1) \setminus (T_l(\varsigma_0) \cup T_r(\varsigma_0) \cup T_r(\varsigma_1))| = 6k^2 + (\alpha-2)k - \frac{\alpha^2}{2} + \frac{3\alpha}{2} - \alpha\beta + 3. \tag{5}$$

Proof. Let $V = \mathcal{N}(\varsigma_0,\varsigma_1) \cap \mathcal{P}(\varsigma_0,\varsigma_1) \setminus (T_l(\varsigma_0) \cup T_r(\varsigma_0) \cup T_r(\varsigma_1))$. V is partitioned into seven subsets $\Pi_1, \Pi_2, \Pi_3, \Pi_4, \Pi_5, \Pi_6, \Pi_7$ as follows (see Figure 16):

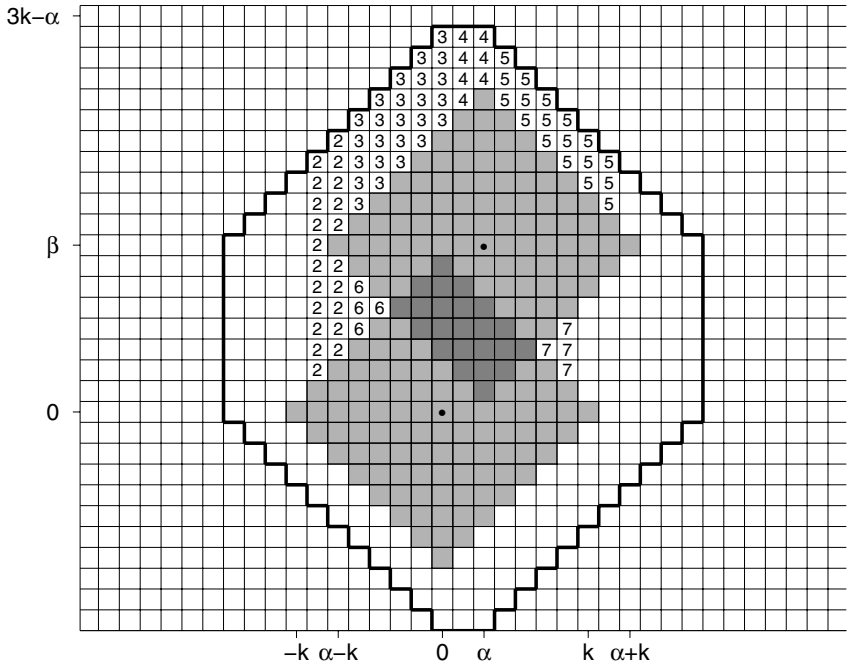


FIG. 16. *Lemma* 22 *applied on* $(\alpha,\beta)=(2,8)$ *and* $k=7$.

1. $\Pi_1=\mathcal{S}(\varsigma_0)\cup\mathcal{S}(\varsigma_1)$, and by Lemma 16
$$|\Pi_1|=|\mathcal{S}(\varsigma_0)|+|\mathcal{S}(\varsigma_1)|-|\mathcal{I}(\varsigma_0,\varsigma_1)|=2k^2+2(\beta+1)k+\beta-\left\lfloor\frac{\beta^2-\alpha^2}{2}\right\rfloor+1.$$
2. $\Pi_2=\{(x,y):-k+1\le x\le\alpha-k,\ k+1\le y-x\le 3k-\alpha-1\}\setminus\{(\alpha-k,\beta)\}$. Π_2 is a parallelogram with a missing cell and $|\Pi_2|=(2k-\alpha-1)\alpha-1$.
3. $\Pi_3=\{(x,y):\alpha-k+1\le x\le 0,\ \beta-\alpha+k+1\le y-x\le 3k-\alpha-1\}$. Π_3 is also a parallelogram and $|\Pi_3|=(2k-\beta-1)(k-\alpha)$.
4. $\{(x,y):1\le x\le\alpha,\ \beta-\alpha+k+1\le y-x,\ y\le 3k-\alpha-1\}$. Π_4 is a right trapezoid and $|\Pi_4|=(2k-\beta-\frac{\alpha+3}{2})\alpha$.
5. $\Pi_5=\{(x,y):\alpha+1\le x\le\alpha+k-1,\ \alpha+\beta+k+1\le x+y\le 3k-1\}$. Π_5 is a parallelogram and $|\Pi_5|=(2k-\alpha-\beta-1)(k-1)$.
6. $\Pi_6=\{(x,y):\alpha-k+1\le x,\ k+1\le y-x,\ x+y\le\alpha+\beta-k-1\}$. Π_6 is an arithmetic progression and $|\Pi_6|=\lfloor\frac{(\beta-\alpha-2)^2}{4}\rfloor$.
7. $\Pi_7=\{(x,y):x\le k-1,\ k+1\le x+y,\ y-x\le\beta-\alpha-k-1\}$ and clearly $|\Pi_7|=|\Pi_6|$.

Therefore,
$$|V|=\sum_{i=1}^{7}|\Pi_i|=6k^2+(\alpha-2)k-\frac{\alpha^2}{2}+\frac{3\alpha}{2}-\alpha\beta+3.\qquad\square$$

LEMMA 23. *If* $0<\alpha<\beta$ *and* $\alpha+\beta\le k+1$, *then* $|\mathcal{N}(\varsigma_0,\varsigma_1)|>6k^2$.

Proof. By Lemmas 17 and 19 we have that $T_l(\varsigma_0)\cup T_r(\varsigma_1)\subset\mathcal{N}(\varsigma_0,\varsigma_1)$, $|\mathcal{P}(\varsigma_0,\varsigma_1)\cap T_l(\varsigma_0)|=2k-\alpha-1$, $|\mathcal{P}(\varsigma_0,\varsigma_1)\cap T_r(\varsigma_1)|=2k-\alpha-\beta-1$, and hence by Lemma 22

we have

$$|\mathcal{N}(\varsigma_0,\varsigma_1)| \geq |(\mathcal{N}(\varsigma_0,\varsigma_1)\cap\mathcal{P}(\varsigma_0,\varsigma_1))\backslash T_r(\varsigma_0)| = 6k^2+(\alpha+2)k-\frac{\alpha^2}{2}-\frac{\alpha}{2}-\beta(\alpha+1)+1. \tag{6}$$

From (6) and since $\alpha+\beta \leq k+1$ it follows that

$$\begin{aligned}|\mathcal{N}(\varsigma_0,\varsigma_1)| &\geq 6k^2+(\alpha+2)k-\frac{\alpha^2}{2}-\frac{\alpha}{2}-(k+1-\alpha)(\alpha+1)+1\\ &= 6k^2+k+\frac{\alpha(\alpha-1)}{2} > 6k^2. \qquad \square\end{aligned}$$

LEMMA 24. *If $0 \leq \beta \leq \alpha$, then*

$$\begin{aligned}&|(\mathcal{N}(\varsigma_0,\varsigma_1)\cap\mathcal{P}(\varsigma_0,\varsigma_1))\setminus(T_l(\varsigma_0)\cup T_r(\varsigma_1))|\\ &\geq 6k^2+(2\alpha-\beta-2)k+\alpha+\tfrac{2\beta-5\alpha^2-2\alpha\beta+\beta^2}{4}+2.\end{aligned} \tag{7}$$

Proof. Let $V=(\mathcal{N}(\varsigma_0,\varsigma_1)\cap\mathcal{P}(\varsigma_0,\varsigma_1))\setminus(T_l(\varsigma_0)\cup T_r(\varsigma_1))$. V is partitioned into five subsets $\Pi_1,\Pi_2,\Pi_3,\Pi_4,\Pi_5$ as follows (see Figure 17):

1. $\Pi_1=\mathcal{S}(\varsigma_0)\cup\mathcal{S}(\varsigma_1)$, and by Lemma 16, $|\Pi_1|=2k^2+2(\alpha+1)k+\alpha-\lfloor\frac{\alpha^2-\beta^2}{2}\rfloor+1$.
2. $\Pi_2=\{(x,y): -k+1\leq x\leq 0\ , k+1\leq y-x\leq 3k-\alpha-1\}$. Π_2 is a parallelogram and $|\Pi_2|=(2k-\alpha-1)k$.
3. $\Pi_3=\{(x,y): 1\leq x\leq\lfloor\frac{\alpha-\beta}{2}\rfloor,\ k+1\leq x+y,\ y\leq 3k-\alpha-1\}$. Π_3 is a right trapezoid and $|\Pi_3|=\frac{1}{2}(4k-\lfloor\frac{3\alpha+\beta+3}{2}\rfloor)\lfloor\frac{\alpha-\beta}{2}\rfloor$.
4. $\Pi_4=\{(x,y): \lfloor\frac{\alpha-\beta}{2}\rfloor+1\leq x\leq\alpha-1,\ \beta-\alpha+k+1\leq y-x,\ y\leq 3k-\alpha-1\}$. Π_4 is a right trapezoid and $|\Pi_4|=\frac{1}{2}(4k-\lfloor\frac{3\alpha+3\beta+4}{2}\rfloor)\lfloor\frac{\alpha+\beta-1}{2}\rfloor$.
5. $\Pi_5=\{(x,y): \alpha\leq x\leq\alpha+k-1,\ \alpha+\beta+k+1\leq x+y\leq 3k-1\}$. Π_5 is a parallelogram and $|\Pi_5|=(2k-\alpha-\beta-1)k$.

Therefore,

$$\begin{aligned}|V|=\sum_{i=1}^{5}|\Pi_i| &= 6k^2+(2\alpha-\beta-2)k+\alpha+\left\lfloor\frac{2\beta-3\alpha^2-2\alpha\beta-\beta^2}{4}\right\rfloor-\left\lfloor\frac{\alpha^2-\beta^2}{2}\right\rfloor+2\\ &\geq 6k^2+(2\alpha-\beta-2)k+\alpha+\frac{2\beta-5\alpha^2-2\alpha\beta+\beta^2}{4}+2. \qquad \square\end{aligned}$$

LEMMA 25. *If $0\leq\beta\leq\alpha$ and $\alpha+\beta\leq k+1$, then $|\mathcal{N}(\varsigma_0,\varsigma_1)|>6k^2$.*

Proof. By Lemmas 17 and 19 we have that $T_l(\varsigma_0)\cup T_r(\varsigma_1)\subset\mathcal{N}(\varsigma_0,\varsigma_1)$, $|\mathcal{P}(\varsigma_0,\varsigma_1)\cap T_l(\varsigma_0)|=2k-\alpha-1$, $|\mathcal{P}(\varsigma_0,\varsigma_1)\cap T_r(\varsigma_1)|=2k-\alpha-\beta-1$, and hence by Lemma 24 we have

$$\begin{aligned}|\mathcal{N}(\varsigma_0,\varsigma_1)| &\geq |\mathcal{N}(\varsigma_0,\varsigma_1)\cap\mathcal{P}(\varsigma_0,\varsigma_1)|\\ &\geq 6k^2+(2\alpha-\beta+2)k-\alpha-\beta+\frac{2\beta-5\alpha^2-2\alpha\beta+\beta^2}{4}.\end{aligned} \tag{8}$$

A lower bound on $|\mathcal{N}(\varsigma_0,\varsigma_1)|$ for a fixed α and in the given range of β is obtained for the largest possible value of β. We distinguish between two cases:

- If $\alpha\geq\frac{k+1}{2}$, then the largest value of β is $k-\alpha+1$. Substituting $\beta=k-\alpha+1$ in (8) implies that

$$|\mathcal{N}(\varsigma_0,\varsigma_1)| \geq 5\frac{1}{4}k^2+(2\alpha+1)k-\frac{\alpha^2}{2}-1\frac{1}{2}\alpha-\frac{1}{4}. \tag{9}$$

  A lower bound on $|\mathcal{N}(\varsigma_0,\varsigma_1)|$ in the given range of α is obtained for $\alpha=\frac{k+1}{2}$. Substituting $\alpha=\frac{k+1}{2}$ in (9) we obtain $|\mathcal{N}(\varsigma_0,\varsigma_1)|>6k^2$.

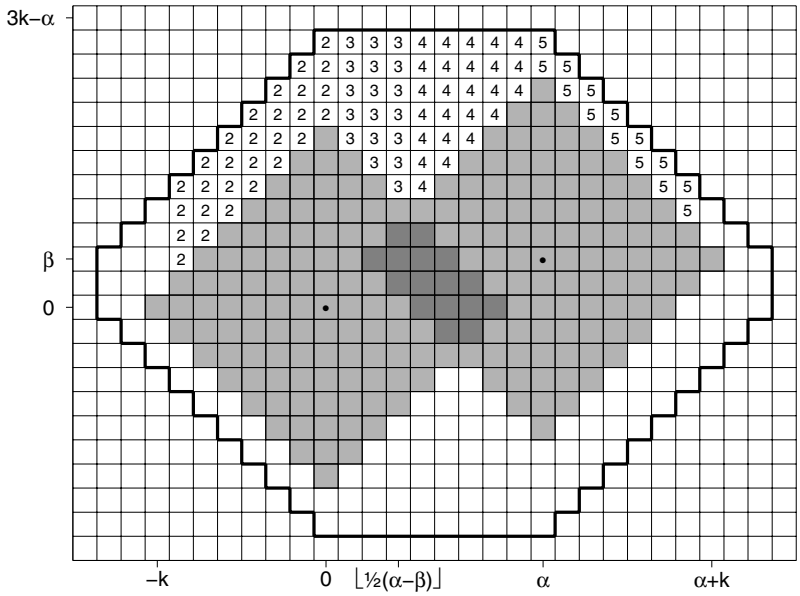


FIG. 17. *Lemma* 24 *applied on* $(\alpha, \beta) = (9, 2)$ *and* $k = 7$.

- If $1 \leq \alpha \leq \frac{k}{2}$, then the largest value of β is α, and

$$|\mathcal{N}(\varsigma_0, \varsigma_1)| \geq 6k^2 + (\alpha + 2)k - \frac{3\alpha(\alpha+1)}{2}. \tag{10}$$

A lower bound on $|\mathcal{N}(\varsigma_0, \varsigma_1)|$ in the given range of α is obtained for $\alpha = 1$ if $k > 10$ and for $\alpha = \left\lfloor \frac{k}{2} \right\rfloor$ if $2 \leq k \leq 10$. Substituting $\alpha = 1$ or $\alpha = \frac{k}{2}$, respectively, in (10) we obtain $|\mathcal{N}(\varsigma_0, \varsigma_1)| > 6k^2$. □

LEMMA 26. *If* $\alpha + \beta > k$, *then* $|\mathcal{U}_r(\varsigma_1) \setminus \mathcal{P}(\varsigma_0, \varsigma_1)| = \frac{1}{2}(\alpha + \beta - k)(\alpha + \beta - k + 1)$.

Proof.

$$\mathcal{U}_r(\varsigma_1) \setminus \mathcal{P}(\varsigma_0, \varsigma_1) = \{(x, y) : x \leq \alpha + k - 1,\ y \leq \beta + k,\ x + y \geq 3k\}$$

is an isosceles right triangle with legs of length $\alpha + \beta - k$. □

An example of $\mathcal{U}_r(\varsigma_1) \setminus \mathcal{P}(\varsigma_0, \varsigma_1)$ is depicted in Figure 18.

LEMMA 27. *If* $\varsigma_2 = (\gamma, \delta)$ *is a black cell such that* $\mathcal{U}_r(\varsigma_1) \cap \mathcal{S}(\varsigma_2) \neq \varnothing$, *then* $\alpha + \beta > k$.

Proof. By Lemma 6, $\alpha + \beta > \gamma + \delta - 3k = d_3(\varsigma_0, \varsigma_1, \varsigma_2) - 3k \geq k$. □

LEMMA 28. *Let* $\varsigma_2 = (\gamma, \delta)$ *be a black cell such that* $\mathcal{U}_r(\varsigma_1) \cap \mathcal{S}(\varsigma_2) \neq \varnothing$. *If* $\gamma \geq \alpha + k$, *then* $|\mathcal{N}(\varsigma_0, \varsigma_1) \cap (\mathcal{P}(\varsigma_1, \varsigma_2) \setminus \mathcal{P}(\varsigma_0, \varsigma_1))| \geq \frac{1}{2}(\alpha + \beta - k)(\alpha + \beta - k + 1) - 1$.

Proof. We consider the set of free cells of $\mathcal{N}(\varsigma_0, \varsigma_1)$ inside $\mathcal{P}(\varsigma_1, \varsigma_2)$ and above $\mathcal{P}(\varsigma_0, \varsigma_1)$. This set contains the following two disjoint subsets V_1 and V_2 (see Figure 19):

$$V_1 = \{(x, y) : x \leq \alpha - 1,\ y \geq 3k - \alpha,\ y - x \leq 3k - \gamma - 1 + \beta\},$$
$$V_2 = \{(x, y) : \alpha \leq x \leq \gamma - k,\ y \leq 3k - \gamma - 1 + \alpha + \beta,\ x + y \geq 3k\} \setminus \{(\gamma - k, \delta)\}.$$

If $\gamma < 2\alpha + \beta - 1$, then V_1 is an isosceles right triangle and V_2 is a right trapezoid with a missing cell, as depicted in Figure 19(a). If $\gamma \geq 2\alpha + \beta - 1$, then V_1 is an empty set

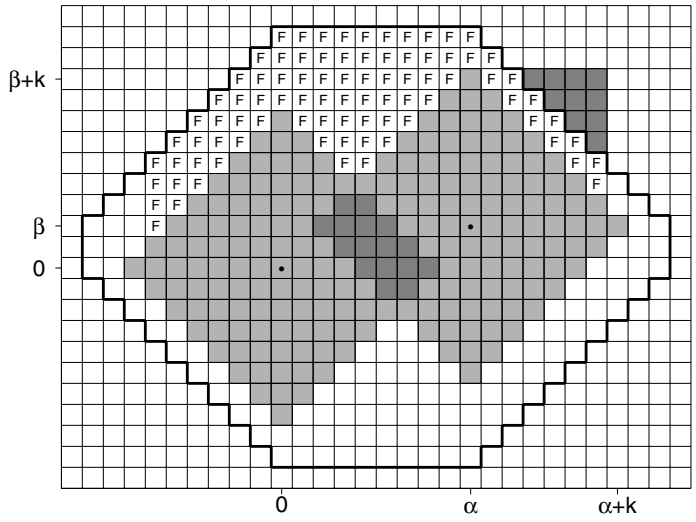


FIG. 18. *An example of* $\mathcal{U}_r(\varsigma_1) \setminus \mathcal{P}(\varsigma_0, \varsigma_1)$ *with* $(\alpha, \beta) = (9, 2)$ *and* $k = 7$.

and V_2 is an isosceles right triangle with a missing cell, as depicted in Figure 19(b). In both cases

$$|\mathcal{N}(\varsigma_0, \varsigma_1) \cap (\mathcal{P}(\varsigma_1, \varsigma_2) \setminus \mathcal{P}(\varsigma_0, \varsigma_2))| \geq |V_1| + |V_2| = \frac{1}{2}(\alpha+\beta-k)(\alpha+\beta-k+1) - 1. \qquad \square$$

LEMMA 29. *Let* $\varsigma_2 = (\gamma, \delta)$ *be a black cell such that* $\mathcal{U}_r(\varsigma_1) \cap \mathcal{S}(\varsigma_2) \neq \varnothing$. *If* $\gamma < \alpha + k$, *then* $|\mathcal{N}(\varsigma_0, \varsigma_1) \cap (\mathcal{P}(\varsigma_1, \varsigma_2) \setminus \mathcal{P}(\varsigma_0, \varsigma_1))| \geq \frac{1}{2}(\alpha+\beta-k)(\alpha+\beta-k+1) - 2$.

Proof. Since $\mathcal{U}_r(\varsigma_1) \cap \mathcal{S}(\varsigma_2) \neq \varnothing$, it follows that $\delta \leq \beta + 2k$. If $\delta < \beta + 2k$, we consider the set of free cells inside $\mathcal{P}(\varsigma_1, \varsigma_2)$ and above $\mathcal{P}(\varsigma_0, \varsigma_1)$. This set contains the following four disjoint subsets V_1, V_2, V_3, V_4 (see Figure 20):

$$V_1 = \{(x, y) : x \geq \gamma - k,\ y \geq 3k - \alpha,\ x + y \leq \gamma + \delta - k - 1\},$$
$$V_2 = \{(x, y) : \gamma + \delta - \beta - 3k + 1 \leq x \leq \gamma - k - 1,\ y \geq 3k - \alpha,\ y - x \leq 3k - \gamma - 1 + \beta\},$$
$$V_3 = T_l(\varsigma_2) \cap \mathcal{P}(\varsigma_1, \varsigma_2) = \{(\gamma - k, y) : \delta + 1 \leq y \leq 2k + \beta - 1\},$$
$$V_4 = \{(\alpha + k - 1, y) : 2k - \alpha + 1 \leq y \leq \alpha + \delta - \gamma - 2\}.$$

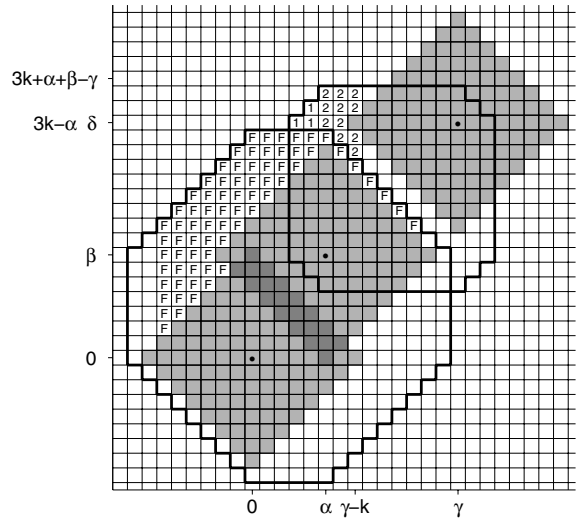
V_1 is a isosceles right triangle, V_2 is a right trapezoid, and

$$|V_1| + |V_2| = \frac{1}{2}(\alpha+\beta-k)(\alpha+\beta-k-1),$$
$$|V_3| + |V_4| = 2\alpha + \beta - \gamma - 3 = \alpha + \beta - (\gamma - \alpha) - 3 \geq \alpha + \beta - k - 2.$$

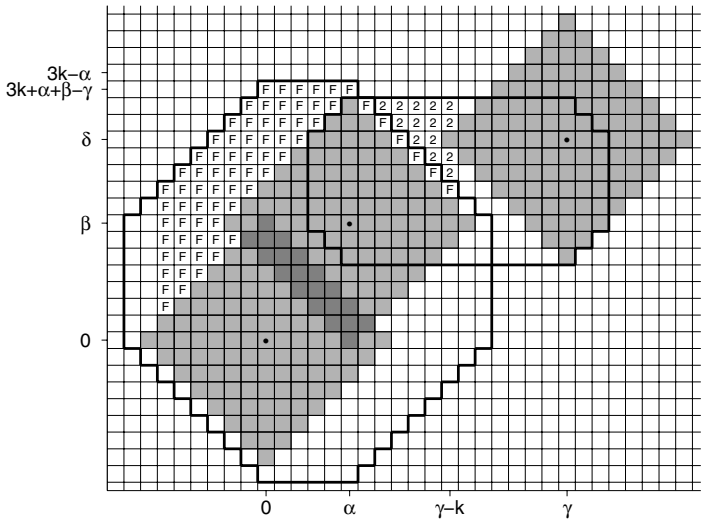
Hence

$$|\mathcal{N}(\varsigma_0, \varsigma_1) \cap (\mathcal{P}(\varsigma_1, \varsigma_2) \setminus \mathcal{P}(\varsigma_0, \varsigma_1))| \geq |V_1| + |V_2| + |V_3| + |V_4| \geq \frac{1}{2}(\alpha+\beta-k)(\alpha+\beta-k+1) - 2.$$

If $\delta = \beta + 2k$, then $\mathcal{P}(\varsigma_1, \varsigma_2)$ does not include cells of the form $(\gamma - k, y)$, and hence the set V_1 as defined above is not contained in $\mathcal{P}(\varsigma_1, \varsigma_2)$. Moreover, $|\mathcal{U}_r(\varsigma_1) \cap \mathcal{S}(\varsigma_2)| = 1$



(a) $(\gamma, \delta) = (14, 16)$



(b) $(\gamma, \delta) = (18, 12)$

FIG. 19. *Lemma* 28 *applied on* $(\alpha, \beta) = (5, 7)$ *and* $k = 7$. *The cells of* V_1 *and* V_2 *are marked "*1*" and "*2*."*

and $\mathcal{N}(\varsigma_0, \varsigma_1) \supset \mathcal{U}_r(\varsigma_1) \setminus \mathcal{S}(\varsigma_2)$. Hence, it follows by Lemmas 26 and 27 that

$$\begin{aligned} |\mathcal{N}(\varsigma_0, \varsigma_1) \cap (\mathcal{P}(\varsigma_1, \varsigma_2) \setminus \mathcal{P}(\varsigma_0, \varsigma_1))| &\geq |\mathcal{U}_r(\varsigma_1) \setminus \mathcal{P}(\varsigma_0, \varsigma_1)| - 1 \\ &= \frac{1}{2}(\alpha + \beta - k)(\alpha + \beta - k + 1) - 1. \qquad \square \end{aligned}$$

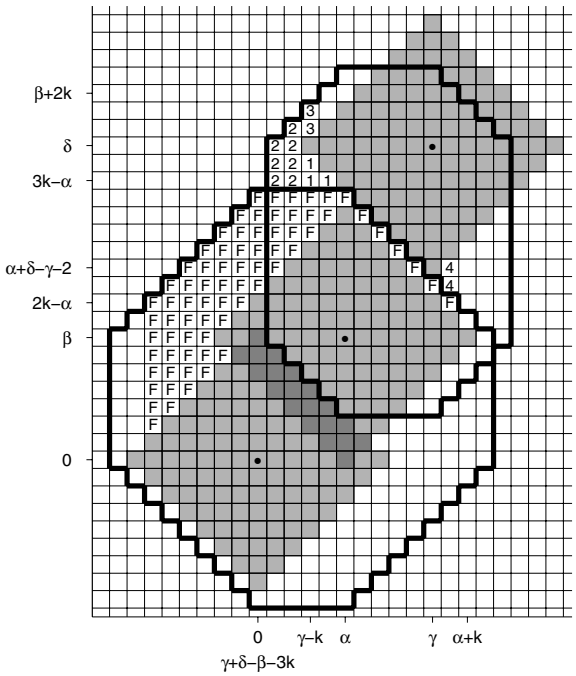


FIG. 20. *Lemma* 29 *applied on* $(\alpha,\beta)=(5,7),\ (\gamma,\delta)=(10,18),\ k=7$.

LEMMA 30. *If* $0<\alpha<\beta\le k$ *and* $\alpha+\beta>k+1$, *then* $|\mathcal{N}(\varsigma_0,\varsigma_1)|>6k^2$.

Proof. By Lemmas 18, 19, and 20 we have that

$$|\mathcal{N}(\varsigma_0,\varsigma_1)\cap\mathcal{P}(\varsigma_0,\varsigma_1)\cap(T_l(\varsigma_0)\cup T_r(\varsigma_0)\cup T_r(\varsigma_1))|\ge 5k-4\alpha-\beta-2. \tag{11}$$

By Lemmas 26, 28, and 29 we have that

$$|\mathcal{N}(\varsigma_0,\varsigma_1)\setminus\mathcal{P}(\varsigma_0,\varsigma_1)|\ge\frac{1}{2}(\alpha+\beta-k)(\alpha+\beta-k+1)-2. \tag{12}$$

Combining (5) with (11) and (12) we obtain

$$|\mathcal{N}(\varsigma_0,\varsigma_1)|\ge 6\frac{1}{2}k^2+\left(2\frac{1}{2}-\beta\right)k-2\alpha+\frac{\beta^2}{2}-\frac{\beta}{2}-1.$$

Since $\alpha<\beta$ it follows that

$$|\mathcal{N}(\varsigma_0,\varsigma_1)|\ge 6\frac{1}{2}k^2+\left(2\frac{1}{2}-\beta\right)k+\frac{\beta^2}{2}-\frac{5\beta}{2}+1. \tag{13}$$

A lower bound on $|\mathcal{N}(\varsigma_0,\varsigma_1)|$ in the given range of β is obtained for $\beta=k$. Substituting $\beta=k$ in (13) we obtain $|\mathcal{N}(\varsigma_0,\varsigma_1)|\ge 6k^2+1$. □

LEMMA 31. *If* $0<\alpha<\beta$ *and* $k<\beta$, *then* $|\mathcal{N}(\varsigma_0,\varsigma_1)|>6k^2$.

Proof. By Lemmas 18, 19, and 21 we have that

$$|\mathcal{N}(\varsigma_0,\varsigma_1)\cap\mathcal{P}(\varsigma_0,\varsigma_1)\cap(T_l(\varsigma_0)\cup T_r(\varsigma_0)\cup T_r(\varsigma_1))|\ge 6k-4\alpha-2\beta-1. \tag{14}$$

By Lemmas 26, 28, and 29 we have that

$$|\mathcal{N}(\varsigma_0,\varsigma_1)\setminus\mathcal{P}(\varsigma_0,\varsigma_1)|\geq\frac{1}{2}(\alpha+\beta-k+1)(\alpha+\beta-k)-2. \tag{15}$$

Combining (5) with (14) and (15) we obtain

$$|\mathcal{N}(\varsigma_0,\varsigma_1)|\geq 6\frac{1}{2}k^2+\left(3\frac{1}{2}-\beta\right)k-2\alpha+\frac{\beta^2}{2}-\frac{3\beta}{2}.$$

Since $\alpha<2k-\beta$ it follows that

$$|\mathcal{N}(\varsigma_0,\varsigma_1)|\geq 6\frac{1}{2}k^2-\left(\frac{1}{2}+\beta\right)k+\frac{\beta^2}{2}+\frac{\beta}{2}+2. \tag{16}$$

A lower bound on $|\mathcal{N}(\varsigma_0,\varsigma_1)|$, in the given range of β, is obtained for $\beta=k+1$. Substituting $\beta=k+1$ in (16) we obtain $|\mathcal{N}(\varsigma_0,\varsigma_1)|\geq 6k^2+3$. □

LEMMA 32. *If $\alpha=0$, then $|\mathcal{N}(\varsigma_0,\varsigma_1)|>6k^2$.*

Proof. This case is the only one in which some of the cells in $\mathcal{P}(\varsigma_0,\varsigma_1)$ above the left tip of ς_0 belong to $T_l(\varsigma_1)$. Thus we have the following variant of Lemma 22:

$$\begin{aligned}&|\mathcal{N}(\varsigma_0,\varsigma_1)\cap\mathcal{P}(\varsigma_0,\varsigma_1)\setminus(T_l(\varsigma_0)\cup T_l(\varsigma_1)\cup T_r(\varsigma_0)\cup T_r(\varsigma_1))|\\&=6k^2+(\alpha-2)k-\frac{\alpha^2}{2}+\frac{3\alpha}{2}-\alpha\beta+4\\&=6k^2-2k+4\end{aligned}$$

since in the proof $|\Pi_2|=0$. On the other hand, Lemma 17 takes the following form:

If $\beta\leq k+1$, then $T_l(\varsigma_0)\cup T_l(\varsigma_1)\subset\mathcal{N}(\varsigma_0,\varsigma_1)$ and $|T_l(\varsigma_0)\cup T_l(\varsigma_1)|=2k-2$.

Lemma 18 takes the following form:

If $\beta>k+1$, then $|\mathcal{N}(\varsigma_0,\varsigma_1)\cap\mathcal{P}(\varsigma_0,\varsigma_1)\cap(T_l(\varsigma_0)\cup T_l(\varsigma_1))|\geq 3k-\beta-1$.

Therefore, $|\mathcal{N}(\varsigma_0,\varsigma_1)|>6k^2$ as computed in Lemmas 23 and 31. □

LEMMA 33. *If $\beta\leq\alpha\leq k+1$ and $\alpha+\beta>k+1$, then $|\mathcal{N}(\varsigma_0,\varsigma_1)|>6k^2$.*

Proof. By Lemmas 18 and 19 we have that

$$|\mathcal{N}(\varsigma_0,\varsigma_1)\cap\mathcal{P}(\varsigma_0,\varsigma_1)\cap(T_l(\varsigma_0)\cup T_r(\varsigma_1))|\geq 5k-3\alpha-2\beta-1. \tag{17}$$

By Lemma 26, 28, and 29 we have that

$$|\mathcal{N}(\varsigma_0,\varsigma_1)\setminus\mathcal{P}(\varsigma_0,\varsigma_1)|\geq\frac{1}{2}(\alpha+\beta-k+1)(\alpha+\beta-k)-2. \tag{18}$$

Combining (7) with (17) and (18) we obtain

$$|\mathcal{N}(\varsigma_0,\varsigma_1)|\geq 6\frac{1}{2}k^2+\left(\alpha-2\beta+2\frac{1}{2}\right)k+\frac{3\beta^2-3\alpha^2+2\alpha\beta-4\beta-6\alpha}{4}-1. \tag{19}$$

A lower bound on $|\mathcal{N}(\varsigma_0,\varsigma_1)|$ for a fixed α and in the given range of β is obtained for

the largest possible value of β. We distinguish between three cases:

- If $\beta \le \alpha - 3$, then the largest value of β is $\alpha - 3$. Substituting $\beta = \alpha - 3$ in (19) implies that

$$|\mathcal{N}(\varsigma_0, \varsigma_1)| \ge 6\frac{1}{2}k^2 + \left(8\frac{1}{2} - \alpha\right)k + \frac{\alpha^2 - 17\alpha}{2} + 8\frac{3}{4}. \tag{20}$$

  A lower bound on $|\mathcal{N}(\varsigma_0, \varsigma_1)|$ in the given range of α is obtained for $\alpha = k+1$. Substituting $\alpha = k+1$ in (20) we obtain $|\mathcal{N}(\varsigma_0, \varsigma_1)| \ge 6k^2 + \frac{3}{4}$.
- If $\alpha - 2 \le \beta \le \alpha - 1$, then the largest value of β is $\alpha - 1$. Substituting $\beta = \alpha - 1$ in (19) implies that

$$|\mathcal{N}(\varsigma_0, \varsigma_1)| \ge 6\frac{1}{2}k^2 + \left(4\frac{1}{2} - \alpha\right)k + \frac{\alpha^2 - 9\alpha}{2} + \frac{3}{4}. \tag{21}$$

  A lower bound on $|\mathcal{N}(\varsigma_0, \varsigma_1)|$ in the given range of α is obtained for $\alpha = k$ (note that $\alpha + \beta < 2k$). Substituting $\alpha = k$ in (21) we obtain $|\mathcal{N}(\varsigma_0, \varsigma_1)| \ge 6k^2 + \frac{3}{4}$.
- If $\alpha = \beta$, then

$$|\mathcal{N}(\varsigma_0, \varsigma_1)| \ge 6\frac{1}{2}k^2 + \left(2\frac{1}{2} - \alpha\right)k + \frac{\alpha^2 - 5\alpha}{2} - 1. \tag{22}$$

  A lower bound on $|\mathcal{N}(\varsigma_0, \varsigma_1)|$ in the given range of α is obtained for $\alpha = k-1$ (note that $\alpha + \beta < 2k$). Substituting $\alpha = k-1$ in (22) we obtain $|\mathcal{N}(\varsigma_0, \varsigma_1)| \ge 6k^2 + 2$.

Thus, $|\mathcal{N}(\varsigma_0, \varsigma_1)| > 6k^2$. □

LEMMA 34. *If $\varsigma_2 = (\gamma_1, \delta_1)$ and $\varsigma_3 = (\gamma_2, \delta_2)$ are two black cells such that $\mathcal{U}_l(\varsigma_0) \cap \mathcal{S}(\varsigma_2) \ne \varnothing$ and $\mathcal{U}_r(\varsigma_1) \cap \mathcal{S}(\varsigma_3) \ne \varnothing$, then $\mathcal{P}(\varsigma_0, \varsigma_2) \cap \mathcal{P}(\varsigma_1, \varsigma_3) = \varnothing$.*

Proof. $\mathcal{P}(\varsigma_0, \varsigma_1) \cap \mathcal{S}(\varsigma_2) = \varnothing$ by Corollary 1, and hence $\delta_1 - \gamma_1 - k \ge 3k - \alpha$, i.e., $3k - |\gamma_1| - \delta_1 - 1 \le \alpha - k - 1$.

$\mathcal{P}(\varsigma_0, \varsigma_1) \cap \mathcal{S}(\varsigma_3) = \varnothing$ by Corollary 1, and hence $\gamma_2 + \delta_2 - k \ge 3k$, i.e., $\gamma_2 + \delta_2 - \beta - 3k + 1 \ge k - \beta + 1$.

Since $\alpha + \beta < 2k$, it follows that $\alpha - k - 1 < k - \beta + 1$.

Therefore, $3k - |\gamma_1| - \delta_1 - 1 < \gamma_2 + \delta_2 - \beta - 3k + 1$, and it can be readily verified that $\mathcal{P}(\varsigma_0, \varsigma_2) \cap \mathcal{P}(\varsigma_1, \varsigma_3) = \varnothing$ by Lemmas 3 and 12. □

LEMMA 35. *If $\beta \le \alpha$ and $\alpha \ge k + 2$, then $|\mathcal{N}(\varsigma_0, \varsigma_1)| > 6k^2$.*

Proof. By Lemmas 13, 14, 15, 26, 28, 29, and 34 we have that

$$|\mathcal{N}(\varsigma_0, \varsigma_1)| \setminus \mathcal{P}(\varsigma_0, \varsigma_1)| \ge \frac{1}{2}(\alpha - k)(\alpha - k - 1) + \frac{1}{2}(\alpha + \beta - k + 1)(\alpha + \beta - k) - 2.$$

Combining with (7) we obtain

$$|\mathcal{N}(\varsigma_0, \varsigma_1)| \ge 7k^2 - 2(\beta + 1)k + \alpha + \beta + \frac{3\beta^2 - \alpha^2 + 2\alpha\beta}{4}. \tag{23}$$

Since $\alpha + \beta < 2k$ and $\alpha \ge k + 2$ it follows that $\beta \le k - 3$. A lower bound on $|\mathcal{N}(\varsigma_0, \varsigma_1)|$ for a fixed β and in the given range of α is obtained for the largest possible value of α, i.e., $2k - 1 - \beta$. Substituting $\alpha = 2k - 1 - \beta$ in (23) implies that

$$|\mathcal{N}(\varsigma_0, \varsigma_1)| \ge 6k^2 + k - \beta - 1\frac{1}{4} \ge 6k^2 + 1\frac{3}{4}.$$ □

Proof of Lemma 8. W.l.o.g. we assume that $\varsigma_1 = (0,0)$ and $\varsigma_2 = (\alpha,\beta),\ \alpha,\beta \geq 0$.

- If $\alpha+\beta \leq k+1$, then by Lemmas 23, 25, and 32, $|\mathcal{N}(\varsigma_1,\varsigma_2)| > 6k^2$.
- If $\alpha+\beta > k+1$ and $\alpha < \beta$, then by Lemmas 30, 31, and 32, $|\mathcal{N}(\varsigma_1,\varsigma_2)| > 6k^2$.
- If $\alpha+\beta > k+1$ and $\alpha \geq \beta$, then by Lemmas 33 and 35 $|\mathcal{N}(\varsigma_1,\varsigma_2)| > 6k^2$.

Thus, $|\mathcal{N}(\varsigma_1,\varsigma_2)| > 6k^2$. □

Appendix C. In this appendix we will prove Lemma 9. By considering the various cases in the proof of Lemma 7, one can readily verify that in most cases the size of the neighborhood is greater than $3k^2$. In some cases, where the formulas indicate that the size of a neighborhood can be $3k^2$, we can find some cells in the neighborhood which were not counted, e.g., above the tips. There are two cases in which the size of the neighborhood of a given black cell $\varsigma = (\alpha,\beta)$ can be $3k^2$, depending on the position of the black cells surrounding it:

- Either $(\alpha+k,\beta+k)$ is a black cell or $(\alpha-k,\beta+k)$ is a black cell (Case 2.2 item marked with $*$ in Lemma 7).
- Either $(\alpha+k+1,\beta+k-1)$ is a black cell or $(\alpha-k-1,\beta+k-1)$ is a black cell (a specific value in Case 2.3 of Lemma 7).

In the latter case one can verify that the coloring cannot be strongly optimal. Therefore, we have the following lemma.

LEMMA 36. *If $\mathcal{F}$ is a coloring for which all the neighborhoods have size $3k^2$, then for each black cell (α,β), either $(\alpha+k,\beta+k)$ is a black cell or $(\alpha-k,\beta+k)$ is a black cell.*

W.l.o.g. we assume that (α,β) and $(\alpha+k,\beta+k)$ are black cells for some $(\alpha,\beta) \in \mathbb{Z}^2$.

LEMMA 37. *If (α,β) and $(\alpha+k,\beta+k)$ are black cells, then $(\alpha+ik,\beta+ik)$ is a black cell for each $i \geq 0$.*

Proof. By Lemma 36, if $(\alpha+k,\beta+k)$ is a black cell, then either $(\alpha+2k,\beta+2k)$ is a black cell or $(\alpha,\beta+2k)$ is a black cell. But $(\alpha,\beta+2k)$ cannot be a black cell since $d_3((\alpha,\beta),(\alpha+k,\beta+k),(\alpha,\beta+2k)) = 3k < 4k$. □

For each $i \geq 0$ let $\varsigma_i = (\alpha+ik,\beta+ik)$; by Lemma 12 we have the following lemma.

LEMMA 38. *All the cells on the line $y-x = 2k+\beta-\alpha-1$, where $x > \alpha-k$, are inside $\bigcup_{i=0}^{\infty}\mathcal{P}(\varsigma_i,\varsigma_{i+1})$. None of the cells on the line $y-x = 2k+\beta-\alpha$ belongs to $\bigcup_{i=0}^{\infty}\mathcal{P}(\varsigma_i,\varsigma_{i+1})$.*

LEMMA 39. *There exists $j,\ -2k+1 \leq j \leq -k+1$, such that $(\alpha+j+ik,\beta+3k+j+ik)$ are black cells for all $i \geq 0$.*

Proof. Recall that all the black cells satisfy Case 2.2 in the proof of Lemma 7. Note that in this case if $|\mathcal{N}(\varsigma_i)| = 3k^2$, then $\mathcal{N}(\varsigma_i) \subset \mathcal{P}(\varsigma_i,\varsigma_{i+1}) \cup \mathcal{S}(\varsigma_i)$, and hence all the cells on the line $y-x = 2k+\beta-\alpha$ for $x > \alpha-k$ belong to some spheres whose black cells are on the line $y-x = 3k+\beta-\alpha$. In particular, the cell $(\alpha-k+1,\beta+k+1)$ belongs to some sphere. Therefore, there is a black cell $(\alpha+j,\beta+3k+j)$ for some $j,\ -2k+1 \leq j \leq -k+1$. The next black cell on the line $y-x = 3k+\beta-\alpha$ must be either $(\alpha+j+k,\beta+3k+j+k)$ or $(\alpha+j+k+1,\beta+3k+j+k+1)$. Since the size of the neighborhood of $(\alpha+j,\beta+3k+j)$ is $3k^2$, it follows that $(\alpha+j+k,\beta+3k+j+k)$ is a black cell, and thus by Lemma 37, for each $i \geq 0$, $(\alpha+j+ik,\beta+3k+j+ik)$ is a black cell. □

COROLLARY 10. *For each $l \geq 0$ there exists $j_l, -2k+1 \leq j_l \leq -k+1$, such that $(\alpha+\Sigma_{m=1}^{l}j_m,\beta+3kl+\Sigma_{m=1}^{l}j_m)$ is a black cell.*

Let $\mathcal{R}_{\alpha,\beta} = \{(x,y) : y \geq \frac{2k+1}{k-1}(x-\alpha)+\beta,\ y \geq \frac{2k+1}{-k+1}(x-\alpha)+\beta\}$.

COROLLARY 11. *There exist a shift vector $\overrightarrow{s}$ and an integer $h,\ 0 \leq h \leq 3k-1$, such that $T^R_{\overrightarrow{s},h}(\Lambda^R) \cap \mathcal{R}_{\alpha,\beta}$ consists of all the black cells inside $\mathcal{R}_{\alpha,\beta}$.*

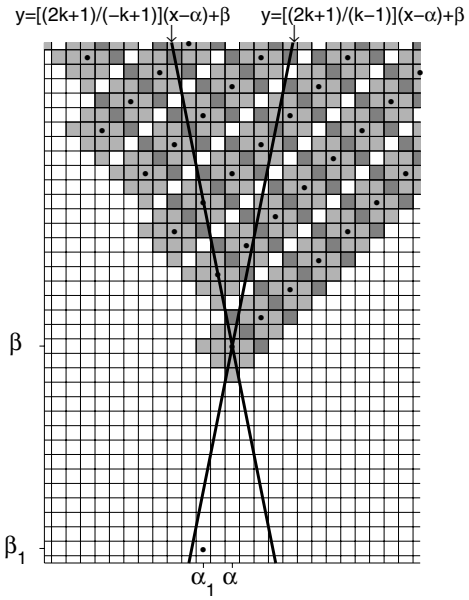


FIG. 21. *Corollary* 11 *with* $k = 2$.

By Theorem 3, there exists a black cell (α_1, β_1) in the region

$$\left\{(x,y) : y < \frac{2k+1}{k-1}(x-\alpha)+\beta,\ y < \frac{2k+1}{-k+1}(x-\alpha)+\beta\right\}.$$

We can start our discussion in this appendix with (α_1, β_1) instead of (α, β). The scenario is depicted in Figure 21.

COROLLARY 12. *There exist a shift vector* $\overrightarrow{s}$ *and an integer* h, $0 \le h \le 3k-1$, *such that either* $T^R_{\overrightarrow{s},h}(\Lambda^R) \cap \mathcal{R}_{\alpha_1,\beta_1}$ *or* $T^L_{\overrightarrow{s},h}(\Lambda^L) \cap \mathcal{R}_{\alpha_1,\beta_1}$ *consists of all the black cells inside* $\mathcal{R}_{\alpha_1,\beta_1}$.

Note that $\mathcal{R}_{\alpha_1,\beta_1} \supset \mathcal{R}_{\alpha,\beta}$; we can define an infinite sequence of cells (α_i, β_i), $i \ge 0$, such that $\mathcal{R}_{\alpha_{i+1},\beta_{i+1}} \supset \mathcal{R}_{\alpha_i,\beta_i}$ and $\mathcal{R}_{\alpha_i,\beta_i} \to_{i\to\infty} \mathbb{Z}^2$, and hence Lemma 9 is proved by Corollary 12.

Acknowledgment. We would like to thank an anonymous referee for his valuable comments.

REFERENCES

[1] M. BLAUM, J. BRUCK, AND A. VARDY, *Interleaving schemes for multidimensional cluster errors*, IEEE Trans. Inform. Theory, 44 (1998), pp. 730–743

[2] T. ETZION, M. SCHWARTZ, AND A. VARDY, *Optimal tristance anticodes in certain graphs*, J. Combin. Theory Ser. A, to appear.

[3] T. ETZION AND A. VARDY, *Two-dimensional interleaving schemes with repetitions: Constructions and bounds*, IEEE Trans. Inform. Theory, 48 (2002), pp. 428–457.

[4] S. W. GOLOMB AND L. R. WELCH, *Perfect codes in the Lee metric and the packing of polyominoes*, SIAM J. Appl. Math., 18 (1970), pp. 302–317.

[5] A. JIANG, M. COOK, AND J. BRUCK, *Optimal t-interleaving on tori*, in Proceedings of the IEEE International Symposium on Information Theory, Chicago, IL, 2004, p. 22.

[6] M. Schwartz and T. Etzion, *Optimal* 2*-dimensional* 3*-dispersion lattices*, in Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, Lecture Notes in Comput. Sci. 2643, Springer, Berlin, 2003, pp. 216–225.
[7] A. Slivkins and J. Bruck, *Interleaving schemes on circulant graphs*, IEEE Trans. Inform. Theory, to appear.