SIAM J. DISC. MATH.
Vol. 6, No. 1, pp. 1-23, February 1993

001

ON ERROR-CORRECTING CODES AND INVARIANT LINEAR FORMS*

A. R. CALDERBANK† AND P. DELSARTE‡

Abstract. Given a code C, invariant linear forms are used to study the designs afforded by codewords of a fixed weight. The most important theorem relating codes and designs is due to Assmus and Mattson [*J. Combin. Theory*, 6 (1969), pp. 122–151], and this theorem is extended in different ways. For extremal self dual codes over the fields $\mathbb{F}_2$ and $\mathbb{F}_3$, it is proved that the t-designs afforded by the codewords of any fixed weight exhibit extra regularity with respect to $(t+2)$-sets. The same is true for the design afforded by the codewords of minimum weight in an extremal self-dual code over $\mathbb{F}_4$. The invariant linear forms are also used to construct Boolean designs with several block sizes, extending previous work by Safavi-Naini and Blake [*Utilitas Math.*, 14 (1978), pp. 49–63], [*Ars Combin.*, 7 (1979), pp. 135–151], [*Inform. and Control*, 42 (1986), pp. 261–282].

Key words. algebraic coding theory, t-designs, invariant linear forms

AMS(MOS) subject classification. 05B20

1. Introduction.

We analyze the relationship between linear codes and the designs afforded by codewords of a fixed weight using invariant linear forms. Given a t-subset x of the coordinate set $[1, n] = \{1, 2, \cdots, n\}$, let $M_j^w(x)$ be the number of codewords in C of weight w with exactly j nonzero entries in the set x. Let L_t be the space of linear forms in the variables M_j^w and let I_t be the space of invariant linear forms; those linear forms $\sum_{w,j} a_{w,j}M_j^w$ for which $\sum_{w,j} a_{w,j}M_j^w(x)$ is independent of x. The codimension $\Delta_t(C)$ of I_t in L_t, and the space $I_t^\perp$ itself are fundamental to this analysis. The reason is that, for *every* weight w in C, the index $\Delta_t(C)$ determines the regularity with which the design afforded by the codewords of weight w meets an arbitrary subset of t points. For example, if $\Delta_t(C) = 0$, then the codewords of any fixed weight form a t-design.

A companion paper [3] considers families of w-subsets of an n-set within the Johnson scheme $J(n, w)$ and describes the structure of the space of invariant forms in the variables $M_j^w, j = 0, \cdots, l$, under the assumption that the codimension of this space is known.

The usefulness of invariant linear forms as an analytical tool stems from the fact that the algebraic theory of error-correcting codes provides a great many invariant linear forms (see Delsarte [7], [8] or Goethals [12] and MacWilliams and Sloane [16, Chap. 21] for an introduction to this theory). Section 2 contains all the results that we need from this theory.

The most important theorem relating codes and designs is the Assmus–Mattson theorem [2]. If C is a linear code satisfying the Assmus–Mattson hypotheses, then the codewords of any fixed weight form a t-design ($\Delta_t(C) = 0$). The main result of § 3 is Theorem 3.5, which extends the Assmus–Mattson theorem; the conclusion of Theorem 3.5 is an upper bound on $\Delta_{t+2}(C)$, given the extra assumption that the weights w_j in C satisfy $w_j - w_{j-1} \geqq 2$ for all j. If C is an extremal self-dual binary code with all weights divisible by 4, then Theorem 3.5 implies that $\Delta_{t+2}(C) = 1$. This means that the t-designs afforded by the codewords of any fixed weight have the extra property of regularity with respect to $(t+2)$-sets. For octads in the Golay code, the extra regularity on 7-sets is expressed in the invariance of the 7-form $6M_7^8 + M_6^8$. If $\Delta_{t+2}(C) = 1$, then every vector in $I_{t+2}^\perp$ is a multiple of a single nonzero vector $a = (a_j^{w_m})$ with integer entries. This vector allows us to read off the invariant $(t+2)$-forms in the variables $M_j^w, j = 0, \cdots, t+2$.

* Received by the editors November 3, 1989; accepted for publication (in revised form) January 21, 1992.

† Mathematical Sciences Research Center, AT&T Bell Laboratories, Murray Hill, New Jersey 07974.

‡ Department of Computer Science, Université Catholique de Louvain, B1348 Louvain la Neuve, Belgium.

We calculate the one-dimensional space $I_7^{\perp}$ for the [24, 12, 8] Golay code and for the [48, 24, 12] extended quadratic residue code (or any extremal self-dual [48, 24, 12] code with all weights divisible by 4).

A second reason for studying the spaces $I_l^{\perp}$ is to construct "Boolean designs with several block sizes" [14], [17], [21], which are analogous to "Euclidean designs with several radii" [10]. For the [24, 12, 8] Golay code, the 7-form $6M_7^8 + M_7^{12}$ is invariant. This means that the collection of Golay codewords with weights 8 and 12, where the octads are taken with multiplicity 6, is a 7-design. We note that, if C is a linear code and if d' is the minimum distance in $C^{\perp}$, then, for any $l < d'$, the collection of codewords of C (all taken with multiplicity 1) is an l-design (in the projection of C onto an arbitrary l-set, every l-tuple appears the same number of times).

Safavi-Naini and Blake [18]–[20] have described methods of constructing t-designs with different block sizes from the supports of codewords in a binary linear code. Their theory uses the four fundamental parameters of a code described by Delsarte [8]. We extend this theory by exploiting the full distance spectrum of a code, rather than just the minimum weight and the number of nonzero weights. We also make use of a result proved in [4], which characterizes collections of subsets that satisfy certain 2-term invariant forms.

In § 5 we use this characterization to derive a strengthening of the Assmus–Mattson theorem (different from that given in [4]) under the assumption that C is a binary linear code for which the first gap in the weight distribution is nontrivial.

In § 6 we extend the analysis given in §§ 2–5 to nonbinary linear codes. We prove that, if C is an extremal self-dual ternary code, then the t-designs afforded by the codewords of any fixed weight also have the extra property of regularity with respect to $(t + 2)$-sets. For extremal self-dual codes over $\mathbb{F}_4$, we prove that the t-design afforded by codewords of *minimum* weight exhibits extra regularity on $(t + 2)$-sets.

In § 7 we extend the analysis given for linear codes to nonlinear codes that are distance invariant. This we do by means of an example: the Nordstrom–Robinson code.

2. Algebraic theory of error-correcting codes. Given an alphabet R_q of q letters, and given a code C in R_q^n, let A_i be the average number of codewords in C at distance i from a given codeword. If C is a linear code over the field $\mathbb{F}_q$, then A_i is just the number of codewords in C with weight i. We can apply the MacWilliams transform to the distance distribution $A = (A_0, \cdots, A_n)$ to obtain the dual distribution $B = (B_0, \cdots, B_n)$, where

$$B_i = \frac{1}{|C|} \sum_{j=0}^{n} A_j P_i(j) \tag{2.1}$$

and where

$$P_i(\zeta) = \sum_{j=0}^{i} (-1)^j (q-1)^{i-j} \binom{\zeta}{j} \binom{n-\zeta}{i-j} \tag{2.2}$$

is the ith Krawtchouk polynomial. If C is a linear code over the field $\mathbb{F}_q$, then B_i is just the number of codewords of weight i in the dual code $C^{\perp}$, and (2.1) expresses the MacWilliams identities.

Let $w_1 = d, w_2, \cdots, w_s$ be those nonzero indices i for which $A_i \neq 0$, and let $w'_1, \cdots, w'_{s'}$ be those nonzero indices j for which $B_j \neq 0$. The *annihilator polynomial* $\alpha(\zeta)$ of C is defined to be

$$\alpha(\zeta) = \frac{q^n}{|C|} \prod_{i=1}^{s'} \left(1 - \frac{\zeta}{w'_i}\right). \tag{2.3}$$

The parameter s' is called the *external distance* of C (it is greater than or equal to the covering radius of C), and the parameter w'_1 (sometimes denoted d') is called the *dual distance* of C.

Below, we expand the shifted annihilator polynomial $\zeta^m\alpha(\zeta)$ as a linear combination of Krawtchouk polynomials:

$$\zeta^m\alpha(\zeta) = \sum_{i=0}^{s'+m} \alpha_i^m P_i(\zeta). \tag{2.4}$$

Given a vector $z \in R_q^n$, let $b_i(z)$ be the number of codewords in C at distance i from z. Delsarte [7] proved that

$$\sum_{i=0}^{s'+m} \alpha_i^m b_i(z) = \begin{cases} 1, & \text{if } m = 0, \\ 0, & \text{otherwise,} \end{cases} \tag{2.5}$$

and we use this result continually, after changing variables so as to better study the design properties of codewords of C.

A code C is said to be *distance invariant* if the number of codewords at distance i from a fixed codeword only depends on i, and not on the particular codeword chosen. All linear codes are distance invariant. A sufficient condition for distance invariance is that the number s of distances is at most equal to the dual distance d' (see Delsarte [7, Thm. 5.4]). For simplicity, we only consider nonlinear codes, over a field alphabet $\mathbb{F}_q$, that are distance invariant and that contain the zero vector. For these codes, the weight spectrum coincides with the distance spectrum.

Given a t-subset x of the coordinate set $[1, n] = \{1, 2, \cdots, n\}$, let $M_j^w(x)$ be the number of codewords in C of weight w with exactly j nonzero entries in the set x. Let L_t be the space of linear forms in the variables $M_j^{w_i}$, where $1 \leqq i \leqq s$ and $0 \leqq j \leqq t$. We are interested in the subspace I_t of invariant linear forms; formally,

$$I_t = \left\{ \sum_{i=1}^{s} \sum_{j=0}^{t} a_{ij} M_j^{w_i} \;\middle|\; \sum_{i=1}^{s} \sum_{j=0}^{t} a_{ij} M_j^{w_i}(x) \text{ is a constant independent of the } t\text{-set } x \right\}. \tag{2.6}$$

The *support* supp (c) of a codeword $c \in C$, with weight w, is the set of w coordinates where the entries of c are nonzero. The design formed by the codewords in C of weight w is the set of supports of codewords of weight w. If R_q is a field, and if C is closed under multiplication by nonzero scalars in $\mathbb{F}_q$, then we count supports with multiplicity $1/(q-1)$, since the $q-1$ scalar multiples of a given codeword have the same support. If the code C contains two codewords of weight w with the same support that are not scalar multiples of each other, then the design will have multiple blocks (blocks with integer multiplicity greater than 1).

Let $v_0 \leqq n$ be the largest integer satisfying

$$v_0 - \left[\frac{v_0 + q - 2}{q - 1}\right] < d, \tag{2.7}$$

where if $q = 2$ we take $v_0 = n$. If $w \leqq v_0$, then two codewords of weight w with the same support must be scalar multiples of one another. The designs formed by codewords of C with weight $w \leqq v_0$ are simple (that is, all blocks have multiplicity 1).

The design properties of the codewords of C are determined by the codimension of I_t in L_t and by the dual space $I_t^\perp$ itself. We begin with a straightforward but important observation.

PROPOSITION 2.1. *The codewords of any fixed weight in C form a t-design if and only if $I_t = L_t$. (We emphasize again that when we speak of t-designs, we mean to allow t-designs with multiple blocks.)*

When $q = 2$, then, for any m, t, we obtain invariant linear forms

$$\sum_{i=0}^{s'+m} \alpha_i^m \sum_{\substack{l,j \\ w_l+t-2j=i}} M_j^{w_l} \in I_t \tag{2.8}$$

directly from (2.5). When $q \neq 2$, we obtain invariant linear forms in the variables $M_j^{w_i}$ by summing (2.5) over all vectors $z \in \mathbb{F}_q^n$ with support x: We obtain

$$\sum_{\substack{z \\ \operatorname{supp}(z)=x}} b_i(z) = \sum_{w_r,l} \sum_{\substack{b \\ w_r+t-2l+b=i}} M_l^{w_r}(x) \binom{l}{b} (q-2)^b \tag{2.9}$$

by counting in two ways the pairs (z, c), where $z \in \mathbb{F}_q^n$ is a vector with $\operatorname{supp}(z) = x$, and $c \in C$ is a codeword with $d(z, c) = i$ (the weight $wt(c) = w_r$, $|\operatorname{supp}(c) \cap x| = l$, and $l - b$ nonzero entries of z and c agree).

The most important theorem relating codes and designs is the Assmus–Mattson theorem, given below. Our statement of this theorem differs from statements given elsewhere (for example in [2], [5], and [16]). In the other versions of the Assmus–Mattson theorem, the conclusion applies only to codewords $c \in C$ with weight $wt(c) \leqq v_0$, where v_0 is defined in (2.7). The extra condition appears when the authors of [2], [5], and [16], when referring to t-designs, wish to *exclude* t-designs with multiple blocks. Since we mean to allow t-designs with repeated blocks, we may omit the extra condition.

THEOREM 2.2 (Assmus–Mattson). *Let C be a linear $[n, k, d]$ code over $\mathbb{F}_q$, where the weights of the nonzero codewords are $w_1 = d, w_2, \cdots, w_s$. Let $w'_1, w'_2, \cdots, w'_{s'}$ be the nonzero weights in $C^\perp$. Let t be the greatest integer in the range $0 < t < d$ such that there are at most $d - t$ weights w'_i with $0 < w'_i \leqq n - t$. Then the codewords of any weight w_i in C form a t-design.*

In later sections, we need the coefficients of ζ^j, ζ^{j-1} in the Krawtchouk polynomial $P_j(\zeta)$. It can be shown that

$$\operatorname{coeff}_j(P_j(\zeta)) = \frac{(-q)^j}{j!}, \tag{2.10}$$

$$\operatorname{coeff}_{j-1}(P_j(\zeta)) = \frac{(-q)^{j-1}}{(j-1)!}\left\{(q-1)n - \frac{(q-2)(j-1)}{2}\right\}, \tag{2.11}$$

where $\operatorname{coeff}_l(p(\zeta))$ denotes the coefficient of ζ^l in the polynomial $p(\zeta)$. For $q = 2$, we also need the identity

$$\operatorname{coeff}_{j-2}(P_j(\zeta)) = \frac{(-2)^{j-2}}{24(j-2)!}\{12n(n-1) + 8(j-2)\}. \tag{2.12}$$

3. Binary codes and invariant linear forms. In this section, we consider binary linear codes, and we begin by using invariant linear forms to prove the Assmus–Mattson theorem (in the case where $q = 2$). Actually, we prove slightly more.

THEOREM 3.1 (Assmus–Mattson). *Let C be a binary linear $[n, k, d]$ code, where the weights of the nonzero codewords are $w_1 = d, w_2, \cdots, w_s$. Let $w'_1, w'_2, \cdots, w'_{s'}$ be the nonzero weights in $C^\perp$. Let t be the greatest integer in the range $0 < t < d$ such that* *there are at most $d - t$ weights w'_i with $0 < w'_i \leqq n - t$. Then $s' = d - t$ or $s' = d - t + 1$, and the codewords in C of any fixed weight w_i form a t-design.*

Proof. We prove that $s' = d - t$ or $s' = d - t + 1$ by proving that there are no codewords in $C^\perp$ with weight w'_i satisfying $n - t < w'_i < n$.

Suppose that $w'_{s'} \neq n$. Let $s' = d - t + l$ and suppose that $l \geqq 1$. Then $w'_{s'} = n - t + g$, where $g \geqq l$. We apply (2.5) to an arbitrary $(t - l)$-set x and obtain

$$(3.1) \qquad \alpha^0_{d-t+l} M^d_{t-l}(x) = \begin{cases} 1 - \alpha^0_{t-l}, & \text{if } 2(t-l) \leqq d, \\ 1, & \text{otherwise.} \end{cases}$$

The two possibilities in (3.1) correspond to $d(x, 0) \leqq d - t + l$ and $d(x, 0) > d - t + l$. Since $\alpha^0_{d-t+l} \neq 0$ (the degree of the annihilator polynomial is $d - t + l$, and the degree of the Krawtchouk polynomial $P_i(\zeta)$ is i), it follows that the codewords in C of weight d form a $(t - l)$-design. Since $g \geqq l$, the codewords of weight d also form a $(t - g)$-design. Let c be a codeword in $C^\perp$ with weight $w'_{s'} = n - t + g$ and let x be the complementary $(t - g)$-set. Since $M^d_i(x) \neq 0$ for all $i = 0, \cdots, t - g$, there exists a codeword $y \in C$, for which the inner product (y, c) satisfies $(y, c) \equiv 1 \pmod 2$. This is impossible; hence $l = 0$ and $s' = d - t$.

Suppose that $w'_{s'} = n$. Let $s' = d - t + l$ and suppose that $l \geqq 2$. Then $w'_{s'-1} = n - t + g$, where $g \geqq l - 1$. We apply (2.5) to an arbitrary $(t - l + 1)$-set x and obtain

$$(3.2) \qquad \alpha^0_{d-t+l-1} M^d_{t-l+1}(x) = \begin{cases} 1 - \alpha^0_{t-l+1}, & \text{if } 2(t-l) < d, \\ 1, & \text{otherwise} \end{cases}$$

(since $w'_{s'} = n$, it follows that $w_2 \geqq d + 2$). We must prove that (3.2) is not identically zero. It follows directly from the definition of the annihilator polynomial that

$$\frac{\alpha^0_{d-t+l-1}}{\alpha^0_{d-t+l}} = \sum_{i=1}^{d-t+l} \frac{(2w'_i - n)}{d - t + l}$$

(as above, $\alpha^0_{d-t+l} \neq 0$). Since $w'_{d-t+l-i} = n - w'_i$, we have that

$$(3.3) \qquad \alpha^0_{d-t+l-1} = n\alpha^0_{d-t+l}/(d - t + l),$$

and, in particular, $\alpha^0_{d-t+l-1} \neq 0$. Now it follows from (3.2) that the codewords in C of weight d form a $(t - g)$-design. Given a codeword $c \in C$ with weight $w'_{s'-1} = n - t + g$, the above argument produces a codeword $y \in C$ for which $(y, c) \equiv 1 \pmod 2$. This is impossible; hence $l = 1$ and $s' = d - t + 1$.

Given Proposition 2.1, we must prove that $I_t = L_t$, or, equivalently, that $I_t^\perp = \{0\}$. Let $\delta = 0$ or 1, according to whether C is even ($w'_{s'} = n$) or not even ($w'_{s'} \neq n$). We apply (2.5) to an arbitrary t-set x and obtain

$$\alpha^0_{d-t} M^d_t(x) = \begin{cases} 1 - \alpha_t, & \text{if } 2t \leqq d + \delta, \\ 1, & \text{otherwise} \end{cases}$$

(where $\alpha^0_{d-t} \neq 0$), so that the codewords in C of weight d form a t-design. It follows that the space I_t of invariant linear forms contains the triangular system

$$\sum_{f=j}^{t} \binom{f}{j} M^d_f, \qquad j = 0, 1, \cdots, t.$$

Hence the restriction $I_t[M^d_j; j = 0, 1, \cdots, t]$ of the space I_t to the variables $M^d_j, j = 0, 1, \cdots, t$ is full, and $I_t^\perp[M^d_j; j = 0, 1, \cdots, t] = \{0\}$.

It is important to observe that, if $I_t^{\perp}[M_t^{w_l}] = \{0\}$, then $M_t^{w_l} \in I_t$, and so the codewords in C of weight w_l form a t-design. We then have that

$$(3.4) \qquad \sum_{f=j}^{t} \binom{f}{j} M_f^{w_l} \in I_t \quad \text{for } j = 0, 1, \cdots, t,$$

from which it follows that $I_t^{\perp}[M_j^{w_l}; j = 0, 1, \cdots, t] = \{0\}$.

Now suppose that $I_t \neq L_t$. Then there exists $e \geqq 2$ such that $I_t^{\perp}[M_t^{w_e}] = \mathbb{R}$, and $I_t^{\perp}[M_j^{w_l}; j = 0, 1, \cdots, t \text{ and } l < e] = \{0\}$. We apply (2.8) with $m = w_e - d - \delta$ to an arbitrary t-set x and obtain the invariant linear form

$$(3.5) \qquad \left\{ \sum_{i=0}^{w_e - t - 1} \alpha_i^m \sum_{\substack{0 \leqq j \leqq t \\ l < e \\ w_l + t - 2j = i}} M_j^{w_l} \right\} + \alpha_{w_e - t}^m M_t^{w_e},$$

where $\alpha_{w_e - t}^m \neq 0$. However, this contradicts the assumption that $I_t^{\perp}[M_t^{w_e}] \neq \{0\}$. Hence $I_t = L_t$, and the proof is complete.

This is certainly not the shortest available proof of the Assmus–Mattson theorem. However, it serves to introduce the viewpoint that the best way to study the design properties of codewords is through the invariant forms (2.8) and (2.9), provided by the algebraic theory of error-correcting codes.

DEFINITION 3.2. Given a code C, we define the *index* Δ_m $(=\Delta_m(C))$ by $\Delta_m = \dim(L_m) - \dim(I_m)$.

The indices Δ_m and the spaces $I_m^{\perp}$ are fundamental to this study of the relationship between codes and designs. If C satisfies the hypotheses of the Assmus–Mattson theorem, then the index $\Delta_t(C)$ equals zero, and the codewords of any fixed weight form a t-design. Sometimes these t-designs have the extra property that certain linear forms are constant on l-sets, for some $l > t$. This extra regularity is determined by the index $\Delta_l(C)$ and by the space $I_l^{\perp}$ itself. Linear forms defined over l-sets will be called l-forms.

A companion paper [3] considers w-subsets of an n-set (the Johnson scheme $J(n, w)$) and describes the structure of the space of invariant l-forms under the assumption that the codimension of this space is known. What is remarkable about codes C satisfying the hypotheses of the Assmus–Mattson theorem is the simultaneous existence of s interesting spaces of l forms (one space for each nonzero weight in C), each with codimension no more than $\Delta_l(C)$, and all linked together by the Krawtchouk coefficients α_i^m. This interaction between the Hamming and Johnson schemes is still mysterious.

Of particular interest is the case where the index $\Delta_l(C)$ equals 1. In this case, vectors in $I_l^{\perp}$ are multiples of single vector a with integer entries $a_j^{w_m}$. Given a weight w_m in C, the linear form

$$(3.6) \qquad a_{l-1}^{w_m} M_l^{w_m}(x) - a_l^{w_m} M_{l-1}^{w_m}(x)$$

is independent of the l-set x. Furthermore, if $a_l^{w_j} > 0$ and $a_l^{w_m} < 0$, then the linear form

$$(3.7) \qquad a_{l_j}^{w} M_l^{w_m}(x) - a_l^{w_m} M_{l_j}^{w}(x)$$

is independent of the l-set x. In this case, consider the collection Ω of codewords in C with weight w_j or w_m, where codewords of weight w_m are taken with multiplicity $a_{l_j}^{w}$, and codewords of weight w_j are taken with multiplicity $-a_l^{w_m}$. Then Ω is an l-design with two different block sizes. If $a_{l_j}^{w}$, $a_l^{w_m}$ have the same sign, then we can still construct an l-design Ω if we are willing to assign a negative multiplicity to codewords of a particular weight.

LEMMA 3.3. *Let C be a binary linear $[n, k, d]$ code, where the weights of the nonzero codewords are $w_1 = d, w_2, \cdots, w_s$. Let $w'_1, w'_2, \cdots, w'_{s'}$ be the nonzero weights* *in $C^\perp$. Let t be the greatest integer in the range $0 < t < d$ such that there are at most $d - t$ weights w'_i with $0 < w'_i \leqq n - t$. Then*

(1) $\quad \Delta_{t+2}(C) = \dim (I^\perp_{t+2}[M^{w_m}_{t+2}, M^{w_m}_{t+1}; m = 1, \cdots, s]);$

if $w_s = n$, then

(2) $\quad \Delta_{t+2}(C) = \dim (I^\perp_{t+2}[M^{w_m}_{t+2}, M^{w_m}_{t+1}; w_m \leqq \lfloor n/2 \rfloor]).$

Proof. Let $a = (a^{w_m}_j) \in I^\perp_{t+2}$ and suppose that $a^{w_m}_{t+2} = a^{w_m}_{t+1} = 0$, for $m = 1, \cdots, s$. The Assmus–Mattson theorem provides invariant linear forms

$$\sum_{f=j}^{t+2} \binom{f}{j} M^{w_m}_f, \qquad m = 1, \cdots, s, \quad j = 0, \cdots, t.$$

Hence $a = 0$, and this proves part (1). To prove part (2), we observe that, if $w_s = n$, then we have the invariant forms

$$M^{w_m}_0 - M^{n-w_m}_{t+2}, \quad M^{w_m}_1 - M^{n-w_m}_{t+1}, \qquad m = 1, \cdots, s.$$

In the example that follows, we calculate the space $I^\perp_7$ for the [24, 12, 8] Golay code. Since we use (2.8) to produce invariant forms in I_7, we must be able to calculate the Krawtchouk coefficients α^m_j. The Krawtchouk recurrence (for $q = 2$)

$$(3.8) \qquad (i + 1)P_{i+t}(\zeta) = (n - 2\zeta)P_i(\zeta) - (n - i + 1)P_{i-1}(\zeta)$$

provides the recursion

$$(3.9) \qquad 2\alpha^m_i = -(n - 1)\alpha^{m-1}_{i+1} + n\alpha^{m-1}_i - i\alpha^{m-1}_{i-1} \quad \text{for } m \geqq 1.$$

Example 3.4. Here C is the [24, 12, 8] Golay code. The annihilator polynomial $\alpha(\zeta)$ is given by

$$(3.10) \quad \alpha(\zeta) = 2^{12}\left(1 - \frac{\zeta}{8}\right)\left(1 - \frac{\zeta}{12}\right)\left(1 - \frac{\zeta}{16}\right)\left(1 - \frac{\zeta}{24}\right) = \left(\sum_{i=0}^{3} P_i(\zeta)\right) + \frac{1}{6} P_4(\zeta).$$

Since C satisfies the hypotheses of the Assmus–Mattson theorem with $t = 5$, the codewords in C of any fixed weight w_l form a 5-design, and we have the invariant linear forms

$$(3.11) \qquad \sum_{f=j}^{7} \binom{f}{j} M^{w_l}_f, \qquad j = 0, \cdots, 5.$$

The matrix A of coefficients is given by

$$(3.12) \qquad \begin{array}{c|cccccccc}
j \backslash f & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
\hline
5 & 21 & 6 & 1 & & & & & \\
4 & 35 & 15 & 5 & 1 & & & & \\
3 & 35 & 20 & 10 & 4 & 1 & & & \\
2 & 21 & 15 & 10 & 6 & 3 & 1 & & \\
1 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}$$

and is independent of the weight w_l. The solutions $x = [x_7, \cdots, x_0]$ to the equation $Ax^T = 0$ form a two-dimensional space V, which we may parametrize as

$$(3.13) \quad V = \{x_7(1, 0, -21, 70, -105, 84, -35, 6) + x_6(0, 1 -6, 15, -20, 15, -6, 1)\}$$

or as

$$(3.14) \quad V = \{x_0(6, -35, 84, -105, 70, -21, 0, 1) + x_1(1, -6, 15, -20, 15, -6, 1, 0)\}.$$

We prove that every vector in $I_7^\perp$ is a multiple of a single nonzero vector with integer entries. Let $a = (a_j^{w_l}) \in I_7^\perp$ and let $a^{w_l} = (a_7^{w_l}, \cdots, a_0^{w_l})$. Since $a_7^8 \neq 0$ (the octads in the Golay code do not form a 7-design), we may multiply through to obtain $a_7^8 = 1$.

We apply (2.8) with $m = 0$ and obtain the invariant 7-form

$$\alpha_1^0 M_7^8 + \alpha_3^0 M_6^8 = M_7^8 + M_6^8. \tag{3.15}$$

Since $a^8 \in V$ is orthogonal to $(1, 1, 0, \cdots, 0)$, we have that

$$a^8 = (1, -1, -15, 55, -85, 69, -29, 5). \tag{3.16}$$

Since the codewords in C of weight 16 are the complements of the codewords of weight 8, we have the invariant 7-form $M_0^{16} + M_1^{16}$. Since $a^{16} \in V$ is orthogonal to $(0, \cdots, 0, 1, 1)$, it follows from (3.14) that

$$a^{16} = y_{16}(5, -29, 69, -85, 55, -15, -1, 1) \tag{3.17}$$

for some constant y_{16}. Since the 7-forms $M_j^8 - M_{7-j}^{16}$, $j = 0, \cdots, 7$ are invariant, we have $y_{16} = 1$.

Since codewords in C of weight 12 come in complementary pairs, the 7-form $M_7^{12} - M_0^{12}$ is invariant. Thus $a^{12} \in V$ is orthogonal to $(1, 0, \cdots, 0, -1)$, and so

$$a^{12} = y_{12}(1, -5, 9, -5, -5, 9, -5, 1) \tag{3.18}$$

for some constant y_{12}. Next, we calculate the Krawtchouk coefficients α_i^1 using the recurrence (3.9) and obtain

$$\alpha_0^1 = \alpha_1^1 = \alpha_2^1 = 0, \quad \alpha_3^1 = \frac{35}{4}, \quad \alpha_4^1 = 0, \quad \alpha_5^1 = \frac{-5}{12}.$$

We apply (2.8) with $m = 1$ and obtain the invariant 7-form $21M_6^8 - M_5^8 - M_7^{12}$. Since this form is orthogonal to $a = (a_j^{w_l})$, we have that

$$21a_6^8 - a_5^8 - a_7^{12} = -6 - y_{12} = 0,$$

so that $y_{12} = -6$. We have now shown that every vector in $I_7^\perp$ is a multiple of the vector $a = (a_j^{w_l})$, given below:

(3.19)

$w_l \backslash j$	7	6	5	4	3	2	1	0
8	1	−1	−15	55	−85	69	−29	5 .
12	−6	30	−54	30	30	−54	30	−6
16	5	−29	69	−85	55	−15	−1	1
24	0							

Note that every column of (3.19) sums to zero. The 7-forms $M_j^8 + M_j^{12} + M_j^{16}$, $j = 0, \cdots, 7$ are invariant because, in the projection of the Golay code C onto an arbitrary 7-set, every 7-tuple appears 32 times.

Consider the collection Ω of codewords in C with weight 8 or 12, where codewords of weight 8 are taken with multiplicity 6, and codewords of weight 12 with multiplicity 1. Since $6M_7^8 + M_7^{12} \in I_7$, the collection Ω is a 7-design (with $\lambda_7 = 6$).

Let C be a binary linear code satisfying the hypotheses of the Assmus–Mattson theorem. Suppose that $w_j - w_{j-1} \geqq 2$ for all j. Let

$$\gamma(C) = |\{j \mid w_j = w_{j-1} + 2\}|, \tag{3.20}$$

$$\begin{aligned}\phi(C) = |\{i \mid i \leqq s' &\text{ and there exists a weight } w_j \text{ in } C \text{ such that}\\ &i = w_j - t - 2 \text{ or } i = w_j - t\}|,\end{aligned} \tag{3.21}$$

$$\varepsilon(C) = \begin{cases} 0, & \text{if } \alpha_i^0 = 0 \text{ for all } i \leqq s' \text{ for which there exists}\\ & \text{a weight } w_j \text{ in } C \text{ such that } i = w_j - t - 2 \text{ or } i = w_j - t,\\ 1, & \text{otherwise.}\end{cases} \tag{3.22}$$

THEOREM 3.5. *Let C be a binary linear $[n, k, d]$ code, where the weights of the nonzero codewords are $w_1 = d, w_2, \cdots, w_s$. Let $w'_1, w'_2, \cdots, w'_{s'}$ be the nonzero weights in $C^{\perp}$. Let t be the greatest integer in the range $0 < t < d$ such that there are at most $d - t$ weights w'_i with $0 < w'_i \leqq n - t$. Suppose that $w_j - w_{j-1} \geqq 2$ for all j. Then*

$$\Delta_{t+2}(C) \leqq \phi(C) + \gamma(C) - \varepsilon(C).$$

Proof. If $\Delta_{t+2}(C) > \phi(C) + \gamma(C) - \varepsilon(C)$, then there exists $a = (a_j^{w_m}) \neq 0$ in $I_{t+2}^{\perp}$ such that $a_j^{w_m} = 0$ if either (i) $w_{m-j} \leqq s'$, where $j = t + 1$ or $t + 2$, or (ii) $j = t + 1$ and $w_{m+1} = w_m + 2$. The Assmus–Mattson theorem provides the invariant $(t + 2)$-forms

$$\sum_{f=j}^{t+2} \binom{f}{j} M_f^{w_m}, \qquad m = 1, \cdots, s, \quad j = 0, \cdots, t. \tag{3.23}$$

It follows that, if $w_m - t - 2 \leqq s'$, then $a_j^{w_m} = 0$ for all $j = 0, \cdots, t + 2$. Let e be the greatest integer, with $e \geqq 2$, such that

$$a_j^{w_l} = 0 \qquad \text{for } l < e \quad \text{and} \quad j = 0, \cdots, t + 2.$$

We apply (2.8) with $m = w_e - s' - t - 2$ to obtain an invariant $(t + 2)$-form

$$\left\{\sum_{i=0}^{w_e-t-3} \alpha_i^m \sum_{\substack{0 \leqq j \leqq t+2\\ l<e\\ w_l+t+2-2j=i}} M_j^{w_l}\right\} + \alpha_{w_e-t-2}^m M_{t+2}^{w_e}, \tag{3.24}$$

where $\alpha_{w_e-t-2}^m \neq 0$. Since (3.24) is orthogonal to a, we have that $a_{t+2}^{w_e} = 0$. If $w_{e+1} = w_e + 2$, then, by definition, $a_{t+1}^{w_e} = 0$, and it follows from (3.23) that $a_j^{w_e} = 0$ for all $j = 0, \cdots, t + 2$. Since this contradicts the definition of e, we have that $w_{e+1} \neq w_e + 2$. Next, we apply (2.8) with $m = w_e - s' - t$ to obtain an invariant $(t + 2)$-form

$$\left\{\sum_{i=0}^{w_e-t} \alpha_i^m \sum_{\substack{0 \leqq j \leqq t+2\\ l<e\\ w_l+t+2-2j=i}} M_j^{w_l}\right\} + \alpha_{w_e-t-2}^m M_{t+2}^{w_e} + \alpha_{w_e-t}^m M_{t+1}^{w_e}. \tag{3.25}$$

Since (3.25) is orthogonal to a, we have that $a_{t+1}^{w_e} = 0$, and again it follows from (3.23) that $a_j^{w_e} = 0$ for all j, contradicting the definition of e. Hence $\Delta_{t+2}(C) \leqq \phi(C) + \gamma(C) - \varepsilon(C)$, as required.

Given a binary code C with $\Delta_l(C) = 1$, then, for any weight w_m in C, there is an invariant l-form

$$a_{l-1}^{w_m} M_l^{w_m} - a_l^{w_m} M_{l-1}^{w_m}. \tag{3.26}$$

The invariant forms that appear in (3.26) are far from arbitrary. To be more specific, we need a little representation theory of the symmetric group (see Calderbank, Delsarte, and Sloane [4] for more details).

Let Ω be the set of w-subsets of the n-set $[1, n] = \{1, 2, \cdots, n\}$ with $w \leqq \lfloor n/2 \rfloor$ (if $w > \lfloor n/2 \rfloor$ then take complements). We sometimes identify Ω with the set of all points $\phi = (\phi_1, \cdots, \phi_n)$ that satisfy $\phi_p \in \{0, 1\}$, for all p, and $\sum_{p=1}^{n} \phi_p = w$. The space $\mathbb{R}^{\Omega}$, consisting of all mappings from Ω to $\mathbb{R}$, is invariant under the natural action of the symmetric group S_n. The irreducible S_n-invariant subspaces of $\mathbb{R}^{\Omega}$ are the *harmonic subspaces* harm (i), $i = 0, 1, \cdots, w$, and we begin with a brief description of these subspaces.

First, we define the *homogeneous space* hom (i) to be the set of functions $f\colon \Omega \rightarrow \mathbb{R}$ represented by homogeneous polynomials $f(z) = f(z_1, \cdots, z_n)$ of total degree i and degree $\leqq 1$ in each variable z_p. The monomials $z_{p_1} \cdots z_{p_i}$ are linearly independent, and so hom (i) is an $\binom{n}{i}$-dimensional vector space over $\mathbb{R}$. The *harmonic space* harm (i) is the subspace of hom (i) containing all functions f that satisfy the "Laplace equation" $\Delta f(z) = 0$, where the differential operator Δ is given by

$$\Delta f(z) = \sum_{p=1}^{n} \frac{\partial f(z)}{\partial z_p}.$$

The harmonic space harm (i) is the eigenspace of degree i of the Johnson scheme $J(n, w)$ (see Dunkl [11] and Delsarte [9]). We have the orthogonal decomposition

$$\text{hom}\,(i) = \text{harm}\,(i) \oplus \text{hom}\,(i-1),$$

with respect to the inner product $\langle f, g \rangle = \sum_{\phi \in \Omega} f(\phi) g(\phi)$, from which it follows that the dimension of harm (i) is $\binom{n}{i} - \binom{n}{i-1}$.

If ψ is an S_n-invariant subspace of $\mathbb{R}^{\Omega}$, then we can write

$$\psi = \mathop{\textcircled{\scriptsize$\Sigma$}}_{i \in T} \text{harm}\,(i), \tag{3.27}$$

where T is a well-defined subset of $\{0, 1, \cdots, w\}$, and $\textcircled{\scriptsize$\Sigma$}$ denotes the orthogonal sum.

Next, consider a nonempty subset $\mathfrak{B}$ of Ω. A subspace ψ of $\mathbb{R}^{\Omega}$ is said to be $\mathfrak{B}$-*regular* if it satisfies

$$\langle \pi(\mathfrak{B}), f \rangle = \frac{|\mathfrak{B}|}{|\Omega|} \langle \pi(\Omega), f \rangle \quad \text{for all } f \in \zeta, \tag{3.28}$$

where $\pi(\quad)$ is the characteristic function of a subset of Ω. Since $\pi(\Omega)$ is the all-one function (which spans harm (0)), the inner product $\langle \pi(\Omega), f \rangle$ vanishes for all $f \in$ harm (j) with $j \geqq 1$. If ψ is S_n-invariant and $\mathfrak{B}$-regular, then it follows from (3.27) and (3.28) that we have that

$$\langle \pi(\mathfrak{B}), f \rangle = 0 \quad \text{for all } f \in \text{harm}\,(j) \text{ with } j \in T, j \neq 0. \tag{3.29}$$

In other words, $\mathfrak{B}$ is a T-design, as defined by Delsarte in [9]. When $0 \in T$, a T-design is defined to be a T'-design with $T' = T \setminus \{0\}$. Delsarte [9] proved that a classical t-design is a T-design, with $T = \{1, \cdots, t\}$. Note that, when $0 \notin T$, then (3.29) means that

$$\pi(\mathfrak{B}) \in \mathop{\textcircled{\scriptsize$\Sigma$}}_{i \notin T} \text{harm}\,(i).$$

The above argument (used in both directions) shows that the concept of a T-design $\mathfrak{B}$ is equivalent to that of an S_n-invariant $\mathfrak{B}$-regular subspace of $\mathbb{R}^{\Omega}$.

Given a subset $\mathfrak{B}$ of Ω and an integer l with $1 \leqq l \leqq w$, let us suppose that the linear form

$$bM_l^w(x) + b'M_{l-1}^w(x) \tag{3.30}$$

is independent of the l-set x. With the l-set $x = \{p_1, \cdots, p_l\}$, we associate the function

$$f_x(\phi) = b\phi_{p_1}\phi_{p_2}\cdots\phi_{p_l} + b'[(1-\phi_{p_1})\phi_{p_2}\cdots\phi_{p_l} + \cdots + (1-\phi_{p_l})\phi_{p_1}\cdots\phi_{p_{l-1}}],$$

which represents (3.30). Calderbank, Delsarte, and Sloane [4] proved the following theorem by analyzing the harmonic decomposition of the linear space ψ spanned by the functions f_x (for all l-sets x).

THEOREM 3.6. *Let $\mathfrak{B}$ be a nonempty collection of w-subsets of the n-set $\{1, 2, \cdots, n\}$. Given an l-set x, suppose that there exist real numbers b, b′, c, not all zero, such that*

$$bM_l^w(x) + b'M_{l-1}^w(x) = c$$

for all l-subsets x of $\{1, 2, \cdots, n\}$. Then

$$\mathfrak{B} \text{ is an } \{l\}\text{-design} \qquad \text{if } b \neq lb',$$

$$\mathfrak{B} \text{ is an } \{l-1\}\text{-design} \quad \text{if } b = lb'.$$

In particular, if $\mathfrak{B}$ is not an $\{l-1\}$-design, then $\mathfrak{B}$ is an $\{l\}$-design.

Note that, if $\mathfrak{B}$ is an $(l-2)$-design that satisfies the hypotheses of Theorem 3.6, then either $\mathfrak{B}$ is an $(l-1)$-design, in which case

$$M_l^w + lM_{l-1}^w \in I_l, \tag{3.31}$$

or $\mathfrak{B}$ is a $\{1, 2, \cdots, l-2, l\}$-design. Calderbank, Delsarte, and Sloane [4] proved that $\mathfrak{B}$ is a $\{1, \cdots, l-2, l\}$-design if and only if

$$[l(w-l-1)-(n-2l+2)]M_l^w + (w-l-1)M_{l-1}^w \in I_l. \tag{3.32}$$

We now specialize these results to binary linear codes C that satisfy the hypotheses of the Assmus–Mattson theorem.

COROLLARY 3.7. *Let C be a binary linear $[n, k, d]$ code, where the weights of the nonzero codewords are $w_1 = d, w_2, \cdots, w_s$. Let $w'_1, w'_2, \cdots, w'_{s'}$ be the nonzero weights in $C^\perp$. Let t be the greatest integer in the range $0 < t < d$ such that there are at most $d - t$ weights w'_i with $0 < w'_i \leqq n - t$. If $\Delta_{t+2}(C) = 1$, then, for any weight w_m in C with $w_m \leqq \lfloor n/2 \rfloor$, either*

(1) *The codewords of weight w_m in C form a $(t+1)$-design, and*

$$(t+2)M_{t+2}^{w_m} + M_{t+1}^{w_m} \in I_{t+2}; \qquad \textit{or} \tag{3.33}$$

(2) *The codewords of weight w_m in C form a $\{1, \cdots, t, t+2\}$-design, and*

$$[(t+2)(w_m-t-1)-(n-2t-2)]M_{t+2}^{w_m} + (w_m-t-1)M_{t+1}^{w_m} \in I_{t+2}. \tag{3.34}$$

Remarks. (1) If w_m is a weight in C with $w_m > \lfloor n/2 \rfloor$, then either the codewords of weight w_m in C form a $(t+1)$-design and (3.33) holds, or

(3.35)

$$[(t+2)(n-w_m-t-1)-(n-2t-2)]M_0^{w_m} + (n-w_m-t-1)M_1^{w_m} \in I_{t+2}.$$

(2) If $\Delta_{t+2}(C) = 1$, then vectors in $I_{t+2}^\perp$ are multiples of a single nonzero vector $a = (a_j^{w_m})$ with integer entries. These entries are determined by the Krawtchouk expansion of the annihilator polynomial, which is, in turn, determined by the weights w'_i in the

dual code $C^{\perp}$. Corollary 3.7 implies that there are just two possibilities for the space $I^{\perp}_{t+2}[M_j^{w_m}; j = 0, 1, \cdots, t+2]$. We conjecture that it is possible to use this restriction to classify the parameters of codes satisfying the hypotheses of the Assmus–Mattson theorem, for which $\Delta_{t+2}(C) = 1$.

Example 3.8. Here $t = 3$, and C is the $[16, 11, 4]$ extended Hamming code. The weight distribution of C is

$$A_0 = A_{16} = 1, \quad A_4 = A_{12} = 140, \quad A_6 = A_{10} = 448, \quad A_8 = 870,$$

and the Krawtchouk expansion of the annihilator polynomial is given by

$$\alpha(\zeta) = 2^5\left(1 - \frac{\zeta}{8}\right)\left(1 - \frac{\zeta}{16}\right) = P_0(\zeta) + P_1(\zeta) + \frac{1}{8}P_2(\zeta).$$

By Lemma 3.3, we can calculate the index $\Delta_5(C)$ from the restriction of the spaces L_5, I_5 to the variables M_5^w, M_4^w. Since the 5-forms $M_j^w - M_{5-j}^{16-w}$ are invariant, we can, in fact, calculate $\Delta_5(C)$ by restricting to the variables M_4^4, M_5^6, M_4^6, M_5^8, M_4^8. First, we calculate the Krawtchouk coefficients α_j^1 using the recurrence (3.9), shown below:

$$(3.36) \qquad \alpha_0^1 = \alpha_2^1 = 0, \quad \alpha_1^1 = \frac{105}{8}, \quad \alpha_3^1 = \frac{-3}{8}.$$

Since C satisfies the hypotheses of the Assmus–Mattson theorem with $t = 3$, the codewords in C of any fixed weight w_l form a 3-design. The matrix A of coefficients appearing in the invariant 5-forms (3.23) is given by

(3.37)

$j\backslash f$	5	4	3	2	1	0
3	10	4	1			
2	10	6	3	1		
1	5	4	3	2	1	
0	1	1	1	1	1	1

and is independent of the weight w_l. The solutions $x = [x_5, \cdots, x_0]$ to the equation $Ax^T = 0$ form a two-dimensional space V, which we may parametrize as

$$(3.38) \qquad V = \{x_5(1, 0, -10, 20, -15, 4) + x_4(0, 1, -4, 6, -4, 1)\}$$

or as

$$(3.39) \qquad V = \{x_0(4, -15, 20, -10, 0, 1) + x_1(1, -4, 6, -4, 1, 0)\}.$$

Let $a = (a_j^{w_l})$ be a nonzero vector in $I_5^{\perp}$ and let $a^{w_l} = (a_5^{w_l}, \cdots, a_0^{w_l})$. Since the codewords in C of weight 4 do not form a 4-design, we may assume that

$$(3.40) \qquad a^4 = y^4(0, 1, -4, 6, -4, 1)$$

for some constant y_4. Since codewords of weight 8 come in complementary pairs, the 5-form $M_5^8 - M_0^8$ is invariant, and

$$(3.41) \qquad a^8 = y_8(1, -3, 2, 2, -3, 1)$$

for some constant y_8. Finally, we write

$$(3.42) \qquad a^6 = y_6(1, 0, -10, 20, -15, 4) + y_6'(0, 1, -4, 6, -4, 1)$$

for some constants y_6, y_6'. Note that $a_j^w = a_{5-j}^{16-w}$, since the 5-form $M_5^w - M_0^{16-w}$ is invariant.

Now we apply (2.8) and (3.36) to obtain the following system of invariant linear forms:

$$(3.43) \qquad M_4^4 + M_5^6, \qquad 35M_4^4 + 35M_5^6 - M_4^6 - M_5^8 - M_3^4.$$

These invariant 5-forms are orthogonal to a, and so we obtain

$$(3.44) \qquad y_6 = -y_4, \qquad -y_6' - y_8 + 4y_4 = 0.$$

It follows that $\Delta_5 \leqq 2$ and that any vector in $I_5^\perp$ is a linear combination of the two vectors given below:

$w_l \backslash j$	5	4	3	2	1	0
4	0	0	0	0	0	0
6	0	1	−4	6	−4	1
8	−1	3	−2	−2	3	−1
10	1	−4	6	−4	1	0
12	0	0	0	0	0	0
16	0					

and

$w_l \backslash j$	5	4	3	2	1	0
4	0	1	−4	6	−4	1
6	−1	0	10	−20	15	−4
8	4	−12	8	8	−12	4
10	−4	15	−20	10	0	−1
12	1	−4	6	−4	1	0
16	0					

.

The collection of codewords in C with weights 8 and 10 is a 5-design with $\lambda_5 = 39$, and the collection of codewords in C with weights 6 and 12 is a 5-design with $\lambda_5 = 26$.

4. Extremal binary self-dual codes. Let C be an extremal binary self-dual $[n, n/2, 4\lfloor n/24 \rfloor + 4]$ code and let $t = 1, 3$, or 5, according to whether $n \equiv 16, 8, 0 \pmod{24}$. Then the Assmus–Mattson theorem implies that the codewords in C of any fixed weight form a t-design. For a list of the known extremal codes, see Conway and Sloane [6, p. 194].

COROLLARY 4.1. *Let C be an extremal binary self-dual $[n, n/2, 4\lfloor n/24 \rfloor + 4]$ code and let $t = 1, 3$, or 5, according to whether $n \equiv 16, 8, 0 \pmod{24}$. Then $\Delta_{t+2}(C) \leqq 1$.*

Proof. We apply Theorem 3.5; we have that $\gamma(C) = 0$, $\phi(C) = 2$, and $\varepsilon(C) = 1$, since $\alpha_{d-t}^0 = n\alpha_{d-t+1}^0/(d - t + 1) \neq 0$ (see (3.3)). Hence $\Delta_{t+2}(C) \leqq 1$.

Now we can apply Corollary 3.7 to the extremal self-dual codes of lengths 24 and 48 (with $t = 5$), to extremal self-dual codes of lengths 32, 56, 80, and 104 (with $t = 3$), and to extremal self-dual codes of lengths 16, 40, 54, 88, and 136 (with $t = 1$).

We conclude this section by computing the space $I_7^\perp$ for any self-dual [48, 24, 12] doubly even code.

Example 4.2. Here $t = 7$ and C is the [48, 24, 12] extended quadratic residue code (or any extremal self-dual [48, 24, 12] code). Again, $\Delta_7(C) = 1$, and vectors in $I_7^\perp$ are multiples of a single nonzero vector $a = (a_j^{w_m})$. The weight distribution of C is

$$A_{12} = A_{36} = 17296, \quad A_{16} = A_{32} = 535095, \quad A_{20} = A_{28} = 3995376, \quad A_{24} = 7681680$$

(see Mallows and Sloane [15]), and the integrality conditions exclude the possibility that the codewords in C of some fixed weight w_m form a 6-design. Hence $\Delta_7(C) = 1$, and Corollary 3.7, part (2) applies. Every vector in $I_7^\perp$ is a multiple of a single nonzero vector with rational entries. Let $a = (a_j^{w_l}) \in I_7^\perp$, and let $a^{w_l} = (a_7^{w_l}, \cdots, a_0^{w_l})$. Since $a_7^{12} \neq 0$ (the vectors of weight 12 in C do not form a 7-design), we may multiply through to obtain $a_7^{12} = 1$. (Since we choose the normalization $a_7^{12} = 1$, we cannot assume that a has integer entries.)

For $w_l = 12$, the vector a^{12} is in the two-dimensional space V given in (3.13) and (3.14). Here (3.34) provides the invariant 7-form $M_7^{12} + M_6^{12}$ so that a^{12} is orthogonal to $(1, 1, 0, \cdots, 0)$. Hence

$$\begin{aligned} a^{12} &= (1, -1, -15, 55, -85, 69, -29, 5), \\ a^{36} &= (5, -29, 69, -85, 55, -15, -1, 1). \end{aligned} \tag{4.1}$$

For $w_l = 16$, (3.34) provides the invariant 7-form $17M_7^{16} + 5M_6^{16}$, and so

$$\begin{aligned} a^{16} &= y_{16}(5, -17, -3, 95, -185, 165, -73, 13), \\ a^{32} &= y_{16}(13, -73, 165, -185, 95, -3, -17, 5) \end{aligned} \tag{4.2}$$

for some constant y_{16}.

For $w_l = 20$, (3.34) provides the invariant 7-form $31M_7^{20} + 7M_6^{20}$, and so

$$\begin{aligned} a^{20} &= y_{20}(7, -31, 39, 25, -115, 123, -59, 11), \\ a^{28} &= y_{20}(11, -59, 123, -115, 25, 39, -31, 7) \end{aligned} \tag{4.3}$$

for some constant y_{20}.

Finally, for $w_l = 24$, symmetry $(a_j^{24} = a_{7-j}^{24})$ gives

$$a^{24} = y_{24}(1, -5, 9, -5, -5, 9, -5, 1) \tag{4.4}$$

for some constant y_{24}.

Since the minimum weight in $C^\perp$ $(=C)$ is greater than 7, every 7-tuple occurs a constant number of times in the projection of C onto an arbitrary 7-set. This means that the linear forms

$$\sum_{l=1}^{7} M_j^{w_l}, \qquad j = 0, 1, \cdots, 7, \tag{4.5}$$

are invariant, and hence

$$y_{24} = -6 - 18y_{16} - 18y_{20}. \tag{4.6}$$

It remains to calculate y_{16} and y_{20}.

The Krawtchouk coefficients α_i^0 are as follows:

$$\alpha_0^0 = \alpha_1^0 = \alpha_2^0 = \alpha_3^0 = 1, \quad \alpha_4^0 = \frac{5}{27}, \quad \alpha_5^0 = \alpha_6^0 = \alpha_7^0 = \frac{1}{9}, \quad \alpha_8^0 = \frac{1}{54}.$$

For simplicity, we work with the coefficients β_i^m that are obtained from the α_i^m by clearing denominators and removing common factors (this does not change the space I_7, which is closed under scalar multiplication). Thus

$$\beta_0^0 = \beta_1^0 = \beta_2^0 = \beta_3^0 = 54, \quad \beta_4^0 = 10, \quad \beta_5^0 = \beta_6^0 = \beta_7^0 = 6, \quad \beta_8^0 = 1.$$

Below, we calculate the coefficients β_i^1 using the recurrence (3.9):

$$\beta_0^1 = \beta_1^1 = \beta_2^1 = 0, \quad \beta_3^1 = 1980, \quad \beta_4^1 = 0, \quad \beta_5^1 = -20,$$

$$\beta_6^1 = 0, \quad \beta_7^1 = 205, \quad \beta_8^1 = 0, \quad \text{and} \quad \beta_9^1 = -9.$$

We apply (2.8) with $m = 1$ and obtain the invariant linear form

$$-20M_7^{12} + 205M_6^{12} - 9M_5^{12} - 9M_7^{16} \in I_7.$$

Since this form is orthogonal to $a = (a_j^{w_l})$, we have that

$$-20a_7^{12} + 205a_6^{12} - 9a_5^{12} - 9a_7^{16} = -90 - 45y_{16} = 0,$$

so that $y_{16} = -2$. Below, we calculate the coefficients β_i^5 by iterating the recurrence (3.9):

$$\begin{array}{lll}
\beta_0^5 = -4931435520, & \beta_5^5 = 729050435, & \beta_{10}^5 = 3957120, \\
\beta_1^5 = 7632595080, & \beta_6^5 = -639907200, & \beta_{11}^5 = -1707255, \\
\beta_2^5 = -5893240320, & \beta_7^5 = 341853110, & \beta_{12}^5 = 285120, \\
\beta_3^5 = 2459153565, & \beta_8^5 = -99955200, & \beta_{13}^5 = -19305. \\
\beta_4^5 = -868612800, & \beta_9^5 = 9444042, &
\end{array}$$

We apply (2.8) with $m = 5$ and obtain the invariant linear form

$$\begin{aligned}
&\beta_5^5 M_7^{12} + \beta_7^5 M_6^{12} + \beta_9^5 M_5^{12} + \beta_{11}^5 M_4^{12} + \beta_{13}^5 M_3^{12} \\
&+ \beta_9^5 M_7^{16} + \beta_{11}^5 M_6^{16} + \beta_{13}^5 M_5^{16} \\
&+ \beta_{13}^5 M_7^{20}.
\end{aligned} \tag{4.7}$$

Since (4.7) is orthogonal to a, we have that $y_{20} = 5$. By (4.6), we have that $y_{24} = -60$, and the vector $a = (a_j^{w_l})$ is completely determined as follows:

$w_m \backslash j$	7	6	5	4	3	2	1	0
12	1	−1	−15	55	−85	69	−29	5
16	−10	34	6	−190	370	−330	146	−26
20	35	−155	195	125	−575	615	−295	55
24	−60	300	−540	300	300	−540	300	−60
28	55	−295	615	−575	125	195	−155	35
32	−26	146	−330	370	−190	6	34	−10
36	5	−29	69	−85	55	−15	−1	1
48	0							

.

Let $\Omega[p_{12}, \cdots, p_{36}]$ be the collection of codewords in C with weights 12, $\cdots$, 36 and where codewords of weight w_l are taken with multiplicity p_l. If

$$p_{12} - 10p_{16} + 35p_{20} - 60p_{24} + 55p_{28} - 26p_{32} + 5p_{36} = 0,$$

then $\Omega[p_{12}, \cdots, p_{36}]$ is a 7-design. In particular, $\Omega[10, 1, 0, \cdots, 0]$ is a block design with two block sizes and $\lambda_7 = 85$.

5. Strengthening the Assmus–Mattson theorem. If C is a code that satisfies the hypotheses of the Assmus–Mattson theorem, and if there is a nontrivial gap between the first two weights in C, then we can strengthen the conclusions of that theorem.

THEOREM 5.1. *Let C be a binary linear $[n, k, d]$ code, where the weights of the nonzero codewords are $w_1 = d, w_2, \cdots, w_s$. Let $w'_1, \cdots, w'_{s'}$ be the nonzero weights in $C^{\perp}$. Let t be the greatest integer in the range $0 < t < d$ such that there are at most $d - t$ weights w'_i with $0 < w'_i \leqq n - t$. Let $\delta = 0$ or 1, according to whether C is even or not even.*

Suppose that $w_2 > d + 2 - \delta$. Then either

$$\begin{aligned}
&(1)\ \sum_{i=1}^{d-t} w_i'^2 = \frac{(d-t)}{12}(3n^2 + (d-t+1)(3n - 2(d+2t+2))), \quad \textit{or} \\
&(2)\ \textit{the codewords in } C \textit{ of any fixed weight } w_i \textit{ form a } (t+1)\textit{-design.}
\end{aligned} \tag{5.1}$$

Proof. Suppose that the codewords of weight d form a $(t+1)$-design. Then $I^{\perp}_{t+1}[M^d_j;\, j = 0, \cdots, t+1] = \{0\}$. For $i = 2, 3, \cdots$, we apply (2.5) with $m = w_i - s' - t - 1$ to an arbitrary $(t+1)$-set and obtain $I^{\perp}_{t+1}[M^{w_i}_{t+1}] = \{0\}$. Hence $I^{\perp}_{t+1} = \{0\}$, and the codewords in C of any fixed weight w_i form a $(t+1)$-design.

Suppose now that the codewords of weight d do not form a $(t+1)$-design. If $\delta = 1$, then $s' = d - t$, and applying (2.5) with $m = 0$ to an arbitrary $(t+1)$-set x gives

$$\alpha^0_{d-t-1} M^d_{t+1}(x) = \begin{cases} 1, & \text{if } d < 2t+1, \\ 1 - \alpha^0_{t+1}, & \text{if } d \geqq 2t+1 \end{cases} \tag{5.2}$$

(if $c \in C$ satisfies $d(c, x) \leqq d - 1$, then $wt(c) = d$ or $c = 0$). It follows from (5.2) that $d \geqq 2t + 1$, $\alpha^0_{t+1} = 1$ and $\alpha^0_{d-t-1} = 0$. We now apply (2.5) with $m = 0$ to an arbitrary $(t+2)$-set x and obtain the invariant $(t+2)$-form

$$\alpha^0_{d-t-2} M^d_{t+2} + \alpha^0_{d-t} M^d_{t+1}, \tag{5.3}$$

which is nonzero, since $\alpha^0_{d-t} \neq 0$. By Corollary 3.7, we must have that

$$\frac{\alpha^0_{d-t-2}}{\alpha^0_{d-t}} = \frac{(t+2)(d-t-1) - (n-2t-2)}{d-t-1}; \tag{5.4}$$

otherwise, the codewords of weight d form a $(t+1)$-design.

If $\delta = 0$, then $s' = d - t + 1$, and applying (2.5) with $m = 0$ to an arbitrary $(t+1)$-set x gives the invariant $(t+1)$-form

$$\alpha^0_{d-t-1} M^d_{t+1} + \alpha^0_{d-t+1} M^d_t, \tag{5.5}$$

which is nonzero, since $\alpha^0_{d-t+1} \neq 0$. The Assmus–Mattson theorem implies that the codewords of weight d form a t-design, and so the $(t+1)$-form

$$(t+1) M^d_{t+1} + M^d_t \tag{5.6}$$

is invariant. Since the codewords of weight d do not form a $(t+1)$-design, the $(t+1)$-forms (5.5) and (5.6) are proportional, and so

$$\frac{\alpha^0_{d-t-1}}{\alpha^0_{d-t+1}} = (t+1). \tag{5.7}$$

It remains to verify that (5.1) is equivalent to (5.7) when $\delta = 0$ and that (5.1) is equivalent to (5.4) when $\delta = 1$. We do this by equating coefficients of $\zeta^{d-t-\delta-1}$ in the Krawtchouk expansion of the annihilator polynomial $\alpha(\zeta)$ as follows:

$$\alpha(\zeta) = 2^{n-k} \prod_{i=1}^{d-t+1-\delta} \left(1 - \frac{\zeta}{w_i}\right) = \sum_{i=0}^{d-t+1-\delta} \alpha^0_i P_i(\zeta). \tag{5.8}$$

It follows from (2.10) that

$$\alpha^0_{d-t+1-\delta} = \frac{(d-t+1-\delta)!\,2^{n-k}}{2^{d-t+1-\delta} \prod_{i=1}^{d-t+1-\delta} w'_i}, \tag{5.9}$$

and it follows from (2.10), (2.11) that

$$\frac{\alpha^0_{d-t-\delta}}{\alpha^0_{d-t+1-\delta}} = \frac{\sum_{i=1}^{d-t+1-\delta} 2w'_i - n}{d-t+1-\delta} = \begin{cases} \dfrac{n}{d-t+1}, & \text{if } \delta = 0, \\ 0, & \text{if } \delta = 1. \end{cases} \tag{5.10}$$

Given (2.10)–(2.12), we equate coefficients of $\zeta^{d-t-\delta-1}$ in (5.8) and obtain

$$\begin{aligned}
&\alpha^0_{d-t-1-\delta}\frac{(-2)^{d-t-1-\delta}}{(d-t-1-\delta)!}+\alpha^0_{d-t-\delta}\frac{(-2)^{d-t-1-\delta}}{(d-t-1-\delta)!}n\\
(5.11)\qquad &+\alpha^0_{d-t+1-\delta}\frac{(-2)^{d-t-1-\delta}}{24(d-t-1-\delta)!}\{12n(n-1)+8(d-t-1-\delta)\}\\
&=\frac{2^{n-k}(\sum_{i<j}w'_iw'_j)(-1)^{d-t-1-\delta}}{\prod_{i=1}^{d-t+1-\delta}w'_i}.
\end{aligned}$$

We use (5.9) to rewrite (5.11) as

$$\begin{aligned}
(5.12)\qquad &\alpha^0_{d-t-1-\delta}+n\alpha^0_{d-t-\delta}+\alpha^0_{d-t+1-\delta}\{12n(n-1)\\
&\quad+8(d-t-1-\delta)\}/24=\frac{4(\sum_{i<j}w'_iw'_j)\alpha^0_{d-t+1-\delta}}{(d-t+1-\delta)(d-t-\delta)}.
\end{aligned}$$

Now

$$(5.13)\qquad \frac{\alpha^0_{d-t-1-\delta}}{\alpha^0_{d-t+1-\delta}}=\begin{cases}[(t+2)(d-t-1)-(n-2t-2)]/(d-t-1), & \text{if } \delta=1,\\ t+1, & \text{if } \delta=0,\end{cases}$$

and by (5.10) we have that

$$\begin{aligned}
2\sum_{i<j}w'_iw'_j&=\left(\sum_{i=1}^{d-t+1-\delta}w'_i\right)^2-\sum_{i=1}^{d-t+1-\delta}w'^2_i\\
(5.14)\qquad &=\begin{cases}\dfrac{n^2(d-t+2)^2}{4}-\displaystyle\sum_{i=1}^{d-t}w'^2_i-n^2, & \text{if } \delta=0,\\[2ex] \dfrac{n^2(d-t)^2}{4}-\displaystyle\sum_{i=1}^{d-t}w'^2_i, & \text{if } \delta=1.\end{cases}
\end{aligned}$$

We obtain (5.1) by substituting (5.10), (5.13), and (5.14) into (5.12).

Remarks. (1) If we take $n=4m+2$ and assume C to be the code spanned by the blocks $\{1,\cdots,2m+1\}$, $\{2m+2,\cdots,4m+2\}$, then C satisfies the hypotheses of Theorem 5.1 with $t=1$. In this case, (5.1) reduces to the identity

$$\sum_{i=1}^{2m}w'^2_i=4\sum_{i=1}^{2m}i^2=\frac{4\cdot 2m(2m+1)(4m+1)}{6}.$$

In this case, the codewords of minimum weight $d=2m+1$ do not form a 2-design, so (5.2) must be satisfied.

(2) Let C be a binary self-dual $[24u, 12u, 4u+4]$ doubly even code. Suppose that all multiples of 4 in the range $[4u+4, 20u-4]$ occur as weights in C; that is, $w'_1=4u+4,\cdots,w'_{4u-1}=20u-4$, and $w'_{4u}=24u$. Then C satisfies the hypotheses of Theorem 5.1 with $t=5$. In this case, (5.1) becomes the identity

$$\sum_{i=1}^{4u-1}w'^2_i=\sum_{i=u+1}^{5u-1}(4i)^2=16\left[\frac{(5u-1)5u(10u-1)}{6}-\frac{u(u+1)(2u+1)}{6}\right].$$

6. Nonbinary codes. In this section, we generalize the results obtained in § 3 to nonbinary codes, starting with Theorem 3.5. The parameters $\gamma(C)$, $\phi(C)$, and $\varepsilon(C)$ are defined in (3.20)–(3.22).

THEOREM 6.1. *Let C be a linear $[n, k, d]$ code over $\mathbb{F}_q$, where the weights of the nonzero codewords are $w_1 = d, w_2, \cdots, w_s$. Let $w'_1, \cdots, w'_{s'}$ be the nonzero weights in $C^\perp$. Let t be the greatest integer in the range $0 < t < d$ such that there are at most $d - t$ weights w'_i with $0 < w'_i \leqq n - t$. Suppose that $w_j - w_{j-1} \geqq 2$ for all j, and that either*

(1) $w_2 - t - 2 \geqq s'$, or

(2) $w_2 - t - 2 = s' - 1$ and $\alpha^0_{s'-1} \neq -(t+2)(q-2)\alpha^0_{s'}$.

Then $\Delta_{t+2}(C) \leqq \phi(C) + \gamma(C) - \varepsilon(C)$.

Proof. If $\Delta_{t+2}(C) > \phi(C) + \gamma(C) - \varepsilon(C)$, then there exists $a = (a_j^{w_m}) \neq 0$ in $I_{t+2}^\perp$ such that $a_j^{w_l} = 0$ if either (i) $w_l - j \leqq s'$, where $j = t+1$ or $t+2$, or (ii) $j = t+1$ and $w_{l+1} = w_l + 2$. Following the proof of Theorem 3.5, let e be the greatest integer, with $e \geqq 2$, such that $a_j^{w_l} = 0$ for $l < e$ and $j = 0, \cdots, t+2$.

Let x be any $(t+2)$-set, and let $z \in \mathbb{F}_q^n$ be a vector with support x. If $w_e - t - 2 \geqq s'$, then we apply (2.5) with $m = w_e - t - 2 - s'$ and obtain

$$\left(\sum_{i < w_e - t - 2} \alpha_i^m b_i(z)\right) + \alpha^m_{w_e - t - 2} b_{w_e - t - 2}(z) = \begin{cases} 1, & \text{if } m = 0, \\ 0, & \text{if } m \neq 0, \end{cases} \tag{6.1}$$

where $\alpha^m_{w_e - t - 2 - s'} \neq 0$. Now

$$\sum_{\operatorname{supp}(z) = x} b_{w_e - t - 2}(z) = M^{w_e}_{t+2}(x), \tag{6.2}$$

since (6.2) just counts pairs (z, c), where $c \in C$ is a codeword that agrees with z on $x = \operatorname{supp}(z)$. Therefore (6.1) yields the invariant $(t+2)$-form

$$\left(\sum_{\substack{l<e \\ j \leqq t+2}} \varepsilon_{w_l, j} M_j^{w_l}\right) + \alpha^m_{w_e - t - 2} M^{w_e}_{t+2}. \tag{6.3}$$

Since (6.3) is orthogonal to a, we have that $a^{w_e}_{t+2} = 0$.

If $w_e - t - 2 = s' - 1$, then we apply (2.5) with $m = 0$ and obtain

$$\left(\sum_{i < w_e - t - 2} \alpha_i^0 b_i(z)\right) + \alpha^0_{s'-1} b_{w_e - t - 2}(z) + \alpha^0_{s'} b_{w_e - t - 1}(z) = 1. \tag{6.4}$$

Now

$$\begin{aligned} &\sum_{\operatorname{supp}(z) = x} (\alpha^0_{s'-1} b_{w_e - t - 2}(z) + \alpha^0_{s'} b_{w_{e-t-1}}(z)) \\ &\quad = (\alpha^0_{s'-1} + (t+2)(q-2)\alpha^0_{s'}) M^{w_e}_{t+2}, \end{aligned} \tag{6.5}$$

since (6.5) counts pairs (z, c), where $c \in C$ is a codeword that agrees with z in at least $(t+1)$ positions on $x = \operatorname{supp}(z)$. If $\alpha^0_{s'-1} \neq -(t+2)(q-2)\alpha^0_{s'}$, then we deduce that $a^{w_e}_{t+2} = 0$.

To prove $a^{w_e}_{t+1} = 0$ (we may assume that $w_{e+1} \neq w_e + 2$), we apply (2.5) with $m = w_e - t - s'$ and sum over all vectors z with support x to obtain the invariant $(t+2)$-form

$$\left(\sum_{\substack{l<e \\ j \leqq t+2}} \varepsilon_{w_l, j} M_j^{w_l}\right) + \varepsilon_{w_e, t+2} M^{w_e}_{t+2} + \alpha^m_{w_e - t}\left(1 + \binom{t+2}{2}(q-2)^2\right) M^{w_e}_{t+1}, \tag{6.6}$$

where the coefficient of $M^{w_e}_{t+1}$ is nonzero. Since (6.6) is orthogonal to a, we have that $a^{w_e}_{t+1} = 0$, and the proof is completed, as in Theorem 3.5.

Remarks. We may rewrite the condition $\alpha^0_{s'-1} \neq -(t+2)(q-2)\alpha^0_{s'}$ in terms of the weights w'_i. Equating coefficients of $\zeta^{s'}$ in

$$q^{n-k}\prod_{i=1}^{s'}\left(1-\frac{\zeta}{w'_i}\right)=\sum_{i=0}^{s'}\alpha^0_i P_i(\zeta), \tag{6.7}$$

we obtain

$$\frac{(-1)^{s'}q^{n-k}}{\prod_{i=1}^{s'} w'_i}=\alpha^0_{s'}\frac{(-q)^{s'}}{s'!}. \tag{6.8}$$

Equating coefficients of $\zeta^{s'-1}$ in (6.7) gives

$$\frac{(-1)^{s'}q^{n-k}(-\sum_{i=1}^{s'} w'_i)}{\prod_{i=1}^{s'} w'_i}$$

$$=\alpha^0_{s'}\frac{(-q)^{s'-1}}{(s'-1)!}\left\{(q-1)n-\frac{(q-2)(s'-1)}{2}\right\}+\alpha^0_{s'-1}\frac{(-q)^{s'-1}}{(s'-1)!}.$$

Since $\alpha^0_{s'} \neq 0$, we may divide through by $\alpha^0_{s'}$ and use (6.8) to obtain

$$\frac{\alpha^0_{s'-1}}{\alpha^0_{s'}}=-\frac{1}{s'}\sum_{i=1}^{s'}(n(q-1)-qw'_i)+\frac{(q-2)(s'-1)}{2}.$$

We may rewrite $\alpha^0_{s'-1} \neq -(t+2)(q-2)\alpha^0_{s'}$ as

$$\sum_{i=1}^{s'} n(q-1)-qw'_i \neq s'(q-2)\left(t+2+\frac{s'-1}{2}\right). \tag{6.9}$$

COROLLARY 6.2. *Let C be a linear $[n, k, d]$ code over $\mathbb{F}_q$, where the weights of the nonzero codewords are $w_1 = d, w_2, \cdots, w_s$. Let $w'_1, \cdots, w'_{s'}$ be the nonzero weights in $C^\perp$. Let t be the greatest integer in the range $0 < t < d$ such that there are at most $d - t$ weights w'_i with $0 < w'_i \leqq n - t$. Suppose that $w_j - w_{j-1} \geqq 3$ for all j, and that either*

(1) $w_2 - t - 2 \geqq s'$, *or*

(2) $w_2 - t - 2 = s' - 1$ *and*

$$\sum_{i=1}^{s'} n(q-1)-qw'_i \neq s'(q-2)\left(t+2+\frac{s'-1}{2}\right). \tag{6.10}$$

Then

$$\Delta_{t+2}(C)=\dim I^\perp_{t+2}[M^d_{t+2}, M^d_{t+1}].$$

Proof. The proof follows directly from Theorem 6.1 and (6.9).

COROLLARY 6.3. *Let C be an extremal ternary self-dual $[n, n/2, 3\lfloor n/12\rfloor + 3]$ code and let $t = 1$, 3, or 5, according to whether $n \equiv 8$, 4, or 0 (mod 12). Then the codewords in C of any fixed weight form a t-design. Furthermore, $\Delta_{t+2}(C) \leqq 1$, and, for every weight w in C, either*

(1) *The codewords of weight w in C form a $(t+1)$-design and*

$$(t+1)M^w_{t+2}+M^w_{t+1}\in I_{t+2}, \qquad or$$

(2) *The codewords of weight w in C form a $\{1, 2, \cdots, t, t+2\}$ design and*

(i) $[(t+2)(w-t-1)-(n-2t+2)]M^w_{t+2}$
$\qquad + (w-t-1)M^w_{t+1} \in I_{t+2}$, *if* $w \leqq \lfloor n/2\rfloor$,

(ii) $[(t+2)(n-w-t-1)-(n-2t-2)]M^w_0$
$\qquad + (n-w-t-1)M^w_1 \in I_{t+2}$, *if* $w > \lfloor n/2\rfloor$.

Proof. The codewords of weight w form a t-design because C satisfies the hypotheses of the Assmus–Mattson theorem.

Let $x = \{p_1, \cdots, p_{t+2}\}$ be any $(t+2)$-set. We puncture the code C by deleting coordinates $p_1, \cdots, p_{t+2}$, to obtain a code $\hat{C}$. Table 1, below, relates the parameters of $\hat{C}$ and C.

The code $\hat{C}^{\perp}$ is obtained by taking codewords $c = (c_1, \cdots, c_n)$ in $C^{\perp}$ with $c_{p_1} = \cdots = c_{p_{t+2}} = 0$, and deleting these $t + 2$ coordinates. In each case, the number of weights in $\hat{C}^{\perp}$ is one greater than the minimum distance in $\hat{C}$. The MacWilliams relations allow us to solve for the weight distribution of $\hat{C}$ in terms of one free parameter β, say. Thus, if $\hat{A}_i$ is the number of codewords of weight i in $\hat{C}$, then

$$\hat{A}_i = \varepsilon_{1,i}\beta + \varepsilon_{2,i} \tag{6.11}$$

for some constants $\varepsilon_{1,i}$, $\varepsilon_{2,i}$. Now $\hat{A}_{3u+1-t} = M_{t+2}^{3u+3}$, $\hat{A}_{3u+2-t} = M_{t+1}^{3u+3}$, and so we can produce a nonzero invariant $(t+2)$-form involving the variables M_{t+2}^{3u+3} and M_{t+1}^{3u+3}. Hence

$$\dim\left(I_{t+2}^{\perp}[M_{t+2}^{d}, M_{t+1}^{d}]\right) \leqq 1. \tag{6.12}$$

To prove that $\Delta_{t+2}(C) \leqq 1$, we apply Corollary 6.2. For $t = 1, 3$, we have that $w_2 - t - 2 \geqq s'$, and, for $t = 5$, we have that $w_2 - t - 2 = s' - 1$,

$$\sum_{i=u+1}^{4u} 24u - 9i = \frac{9u(u-3)}{2} \neq 3u\left(7 + \frac{3u-1}{2}\right).$$

Thus Corollary 6.2 applies, and $\Delta_{t+2}(C) \leqq 1$. The remainder of the proof follows Corollary 3.7, and we omit the details.

Let C be a linear $[n, k, d]$ code over $\mathbb{F}_q$ that satisfies the hypotheses of the Assmus–Mattson theorem. It is sometimes possible to prove that the t-design formed by the codewords of minimum weight d has the extra property that certain linear forms are constant on l-sets, for some $l > t$, without being able to prove that the t-designs formed by codewords of other weights have this extra regularity.

COROLLARY 6.4. *Let C be a linear $[n, k, d]$ code over $\mathbb{F}_q$, where the weights of the nonzero codewords are $w_1 = d, w_2, \cdots, w_s$. Let $w'_1, w'_2, \cdots, w'_{s'}$ be the nonzero weights*

TABLE 1
The parameters of the codes C and $\hat{C}$.

	C	$\hat{C}$
$t = 1$	$n = 12u + 8$ $k = 6u + 4$ $d = 3u + 3$ $s' = 3u + 2$	$\hat{n} = 12u + 5$ $\hat{k} = 6u + 4$ $\hat{d} = 3u$ $\hat{s}' = 3u + 1$
$t = 3$	$n = 12u + 4$ $k = 6u + 2$ $d = 3u + 3$ $s' = 3u + 1$	$\hat{n} = 12u - 1$ $\hat{k} = 6u + 2$ $\hat{d} = 3u - 2$ $\hat{s}' = 3u - 1$
$t = 5$	$n = 12u$ $k = 6u$ $d = 3u + 3$ $s' = 3u$	$\hat{n} = 12u - 7$ $\hat{k} = 6u$ $\hat{d} = 3u - 4$ $\hat{s}' = 3u - 3$

in $C^{\perp}$. Let l be an integer in the range $0 < l < d$ such that there are at most $d - l + 1$ weights w_i' with $0 < w_i' \leqq n - l$. Let $\mathfrak{B}$ be the collection of codewords in C with weight d. If $w_2 \geqq d + 2$, then either $\mathfrak{B}$ is an $\{l\}$-design or $\mathfrak{B}$ is an $\{l - 1\}$-design.

Proof. Puncture C with respect to any l-set, and deduce that there exists a nontrivial invariant l-form involving the variables M_l^d and M_{l-1}^d (the argument was used in Corollary 6.3 to derive (6.11)). Then apply Theorem 3.6.

Example 6.5. Let C be an extremal quaternary self-dual $[n, n/2, 2\lfloor n/6 \rfloor + 2]$ code and let $t = 1, 3, 5$, according to whether $n \equiv 4, 2$, or $0 \pmod 6$. Then C satisfies the hypotheses of the Assmus–Mattson theorem, and the codewords in C of any given weight form a t-design. Furthermore, C satisfies the hypotheses of Corollary 6.4 with $l = t + 2$, so that the codewords of weight d in C form a $(t + 1)$-design or a $\{1, \cdots, t, t + 2\}$ design.

7. Nonlinear codes. The algebraic theory of error-correcting codes outlined in § 2 applied to linear and nonlinear codes. The following theorem is due to Delsarte [8, Thm. 5.7], and it is an analogue of the Assmus–Mattson theorem.

THEOREM 7.1 (Delsarte). *Let C be a binary code of minimum distance d and external distance s', with $d \geqq s'$. Then C is distance invariant. Moreover, the codewords of a given weight form a t-design with $t = d - s'$.*

Goethals and van Tilborg [13] have extended Theorem 7.1 to *q-ary t-designs* (a q-ary t-(n, w, λ) design is a collection S of vectors of weight w with the property that every vector $x \in R_q^n$ of weight t is covered by exactly λ vectors y in S). Assmus, Goethals, and Mattson [1] have also given a more explicit proof of this extension.

Example 7.2. Here C is the Nordstrom–Robinson code with distance distribution

$$A_0 = A_{16} = 1, \quad A_6 = A_{10} = 112, \quad \text{and} \quad A_8 = 30. \tag{7.1}$$

The Krawtchouk expansion of the annihilator polynomial is given by

$$\begin{aligned} \alpha(\zeta) &= 2^8\left(1 - \frac{\zeta}{6}\right)\left(1 - \frac{\zeta}{8}\right)\left(1 - \frac{\zeta}{10}\right)\left(1 - \frac{\zeta}{16}\right) \\ &= P_0(\zeta) + P_1(\zeta) + \frac{3}{10}P_2(\zeta) + \frac{1}{5}P_3(\zeta) + \frac{1}{20}P_4(\zeta). \end{aligned} \tag{7.2}$$

Let x be an arbitrary 3-set. It follows directly from (2.5) that $\frac{1}{5}(M_3^6(x) + 1) = 1$, and so the codewords of weight 6 in C form a 3-design with $\lambda_3 = 4$. The standard argument proves that $\Delta_3(C) = 0$ and that the codewords of any given weight form a 3-design.

Next, we investigate regularity with respect to 5-sets. Let x be an arbitrary 5-set, and let $a = (a_j^{w_l})$ be an arbitrary nonzero vector in $I_5^{\perp}$. If $a^{w_l} = (a_j^{w_l})$, then the vectors a^{w_l} lie in the space V described in (3.38) and (3.39). We know that

$$a^6 = y_6(1, 0, -10, 20, -15, 4) + y_6'(0, 1, -4, 6, -4, 1),$$

and, since the Nordstrom–Robinson code is closed under taking complements, we have that

$$a^{10} = y_6(4, -15, 20, -10, 0, 1) + y_6'(1, -4, 6, -4, 1, 0),$$

for suitable constants y_6, y_6'. Symmetry gives $a^8 = y_8(1, -3, 2, 2, -3, 1)$ for some constant y_8.

We apply (2.5) with $m = 0$ and obtain the invariant 5-form

$$M_5^6 + \tfrac{1}{5}M_4^6 + \tfrac{1}{5}M_5^8. \tag{7.3}$$

Since (7.3) is orthogonal to the vector $a = (a_j^{w_i})$, we have that

$$(7.4) \qquad 5y_6 + y_6' + y_8 = 0.$$

It follows that every vector in $I_5^{\perp}$ is a linear combination of the two vectors given below:

$w_i \backslash j$	5	4	3	2	1	0
6	1	0	−10	20	−15	4
8	−5	15	−10	−10	15	−5
10	4	−15	20	−10	0	1
16	0					

and

$w_i \backslash j$	5	4	3	2	1	0
6	0	1	−4	6	−4	1
8	−1	3	−2	−2	3	−1
10	1	−4	6	−4	1	0
16	0					

.

In this case, we can prove that $\Delta_5(C) = 2$. If $\Delta_5(C) = 1$, there is a nontrivial invariant 5-form relating M_5^6 and M_4^6. Since the integrality conditions imply that the codewords of weight 6 do not form a 4-design, it follows from Theorem 3.6 that the 5-form $M_5^6 + M_4^6$ is invariant. Clearly, $M_5^6(x) + M_4^6(x) = 1$, since, if $M_5^6(x) = 1$, then $M_4^6(x) = 0$. However, counting pairs (c, x) with $c \in C$ and $|\operatorname{supp}(c) \cap \operatorname{supp}(x)| \geqq 4$, we see that

$$112\left(\binom{6}{5} + \binom{6}{4} \times 10\right) \Big/ \binom{16}{5} = 4\ (\neq 1).$$

Hence $\Delta_5(C) = 2$.

Acknowledgment. The authors thank Prof. E. F. Assmus, Jr. for bringing to their attention the references by Safavi-Naini and Blake.

Note added in proof. H. Koch [*Discrete Mathematics*, 83 (1990), pp. 291–300] has also analyzed the regularity of designs afforded by codewords of a fixed weight in self-dual doubly-even extremal codes. He employs a theorem of Venkov that requires weighted theta functions and the theory of modular forms.

REFERENCES

[1] E. F. Assmus, Jr., J.-M. Goethals, and H. F. Mattson, Jr., *Generalized t-designs and majority logic decoding of linear codes*, Inform. Control, 32 (1976), pp. 43–60.

[2] E. F. Assmus, Jr. and H. F. Mattson, Jr., *New 5-designs*, J. Combin. Theory, 6 (1969), pp. 122–151.

[3] A. R. Calderbank and P. Delsarte, *Extending the t-design concept*, Trans. Amer. Math. Soc., to appear.

[4] A. R. Calderbank, P. Delsarte, and N. J. A. Sloane, *A strengthening of the Assmus-Mattson theorem*, IEEE Trans. Inform. Theory, IT-37 (1991), pp. 1261–1268.

[5] P. J. Cameron and J. H. van Lint, *Graph Theory, Coding Theory, and Block Designs*, London Math. Soc. Lecture Note Series, No. 19, Cambridge University Press, London 1975.

[6] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, Springer-Verlag, New York, 1988.

[7] P. Delsarte, *An algebraic approach to the association schemes of coding theory*, Philips J. Res. Suppl., 10 (1973).

[8] ———, *Four fundamental parameters of a code and their combinatorial significance*, Inform. Control, 23 (1973), pp. 407–438.

[9] ———, *Hahn polynomials, discrete harmonics, and t-designs*, SIAM J. Appl. Math., 34 (1978), pp. 157–166.

[10] P. Delsarte and J. J. Seidel, *Fisher type inequalities for Euclidean t-designs*, Linear Algebra Appl., 114/115 (1989), pp. 213–230.

[11] C. F. DUNKL, *An addition theorem for Hahn polynomials: The spherical functions*, SIAM J. Math. Anal., 9 (1978), pp. 627–637.

[12] J.-M. GOETHALS, *Association schemes*, in Algebraic Coding Theory and Applications, G. Longo, ed., CISM Courses and Lectures 258, Springer-Verlag, Vienna, 1979, pp. 243–283.

[13] J.-M. GOETHALS AND H. C. A. VAN TILBORG, *Uniformly packed codes*, Philips J. Res., 30 (1975), pp. 9–26.

[14] E. S. KRAMER, *Some results on t-wise balanced designs*, Ars Combin., 15 (1983), pp. 179–192.

[15] C. L. MALLOWS AND N. J. A. SLOANE, *An upper bound for self-dual codes*, Inform. Control, 22 (1973), pp. 188–200.

[16] F. J. MACWILLIAMS AND N. J. A. SLOANE, *The Theory of Error-Correcting-Codes*, North–Holland, Amsterdam, 1979.

[17] H. J. RYSER, *New types of combinatorial designs*, Actes Congrès Internat. Math., 3, 1970, pp. 235–239.

[18] R. SAFAVI-NAINI AND I. F. BLAKE, *On designs from codes*, Utilitas Math., 14 (1978), pp. 49–63.

[19] ———, *Generalized t-designs and orthogonal arrays*, Ars Combin., 7 (1979), pp. 135–151.

[20] ———, *Generalized t-designs and weighted majority decoding*, Inform. Control, 42 (1986), pp. 261–282.

[21] D. R. WOODALL, *The λ-μ problem*, J. London Math. Soc. (2), 1 (1969), pp. 509–519.

SIAM J. DISC. MATH.
Vol. 6, No. 1, pp. 24–29, February 1993

ALGORITHMIC ASPECTS OF NEIGHBORHOOD NUMBERS*

GERARD J. CHANG†, MARTIN FARBER‡, AND ZSOLT TUZA§

Abstract. In a graph $G = (V, E)$, $E[v]$ denotes the set of edges in the subgraph induced by $N[v] \equiv \{v\} \cup \{u \in V: uv \in E\}$. The neighborhood-covering problem is to find the minimum cardinality of a set C of vertices such that $E = \cup\{E[v]: v \in C\}$. The neighborhood-independence problem is to find the maximum cardinality of a set of edges in which there are no two distinct edges belonging to the same $E[v]$ for any $v \in V$. Two other related problems are the clique-transversal problem and the clique-independence problem. It is shown that these four problems are NP-complete in split graphs with degree constraints and linear time algorithms for them are given in a strongly chordal graph when a strong elimination order is given.

Key words. neighborhood-covering, neighborhood-independence, clique-transversal, clique-independence, chordal graph, strongly chordal graph, split graph, NP-complete

AMS(MOS) subject classifications. 05C70, 68R10

1. Introduction. The concept of neighborhood number was first introduced by Sampathkumar and Neeralagi [SN]. Suppose that $G = (V, E)$ is a finite undirected graph with vertex set V and edge set E. The *(open) neighborhood* $N(v)$ of a vertex v is the set of vertices adjacent to v, and the *closed neighborhood* $N[v]$ is $\{v\} \cup N(v)$. A *neighborhood-covering set* C is a set of vertices such that $E = \cup\{E[v]: v \in C\}$, where $E[v]$ is the set of edges in the subgraph induced by $N[v]$. (This definition is slightly different from the original one in [SN]; we follow the terminology in [LT].) The *neighborhood-covering number* $\rho_N(G)$ of G is the minimum cardinality of a neighborhood-covering set in G. A *neighborhood-independent set* of G is a set of edges in which there are no two distinct edges belonging to the same $E[v]$ for any $v \in V$. The *neighborhood-independence number* $\alpha_N(G)$ of G is the maximum size of a neighborhood-independent set in G. These two parameters are related by a min-max duality inequality: $\alpha_N(G) \leqq \rho_N(G)$ for any graph G. A graph is called *neighborhood-perfect* if $\alpha_N(H) = \rho_N(H)$ for every induced subgraph H of G.

Two other related problems are defined as follows. In a graph $G = (V, E)$, a *clique* is a set of pairwise adjacent vertices. A *maximal clique* is a clique of size $\geqq 2$ that is maximal under inclusion. A *clique-transversal set* of G is a set of vertices that meets all maximal cliques of G. As defined in [T], the *clique-transversal number* $\tau_C(G)$ of G is the minimum cardinality of a clique-transversal set in G. We now introduce the concept of a *clique-independent set*, which means a collection of pairwise disjoint maximal cliques. The *clique-independence number* $\alpha_C(G)$ of G is the maximum size of a clique-independent set in G. There is also a min-max duality inequality: $\alpha_C(G) \leqq \tau_C(G)$ for any graph G. Note that the clique-independence number of a triangle-free graph is equal to its matching number and hence can be computed in polynomial time.

Various properties of $\rho_N(G)$, $\alpha_N(G)$, $\tau_C(G)$, and $\alpha_C(G)$ have been studied in [SN], [LT], [T], [AST], and [EGT]. The aim of this paper is to investigate some problems concerning the algorithmic complexity of determining these four parameters of a given

* Received by the editors May 6, 1991; accepted for publication (in revised form) January 21, 1992.

† Institute of Applied Mathematics, National Chiao Tung University, Hsinchu 30050, Taiwan, Republic of China. e-mail: gjchang@cc.nctu.edu.tw. This research was supported in part by National Science Council of the Republic of China grant NSC79-0208-M009-31.

‡ Operations Research Department, AT&T Bell Laboratories, Holmdel, New Jersey 07733.

§ Computer and Automation Institute, Hungarian Academy of Sciences, Kende U. 13–17, H-1111 Budapest, Hungary.

graph. Erdös, Gallai, and Tuza [EGT] proved that the problem of finding the clique-transversal number is NP-complete over the class of triangle-free graphs, and more generally over the class of graphs with girth at least g for any fixed $g \geqq 4$. Lehel and Tuza [LT] gave an $O(|V| + |E|)$ algorithm for finding $\rho_N(G)$ and $\alpha_N(G)$ of an interval graph G. Wu [W] gave an $O(|V|^3)$ algorithm for determining $\rho_N(G)$ and $\alpha_N(G)$ of a strongly chordal graph G.

In § 3 we prove that the problems of finding $\rho_N(G)$, $\alpha_N(G)$, $\tau_C(G)$, and $\alpha_C(G)$ are NP-complete over the class of split graphs with degree constraints. Section 4 gives linear time algorithms for determining $\rho_N(G)$, $\alpha_N(G)$, $\tau_C(G)$, and $\alpha_C(G)$ of a strongly chordal graph G if a strong elimination order is available.

2. Terminology. The concept of chordal graph was introduced by Hajnal and Surânyi [HS] in connection with the theory of perfect graphs; see [Go]. A graph is *chordal* (or *triangulated*) if every cycle of length greater than three has a chord (i.e., every induced cycle is a triangle). One of the most important properties of a chordal graph G is that its vertices have a *perfect* elimination order $v_1, v_2, \ldots, v_n$; i.e., for each i ($1 \leqq i \leqq n$), $N_i[v_i]$ is a clique, where $N_i[x]$ is the closed neighborhood of x in the subgraph G_i of G induced by $\{v_i, v_{i+1}, \ldots, v_n\}$. Note that any maximal clique of a chordal graph G is equal to some $N_i[v_i]$, but $N_i[v_i]$ is not necessarily an maximal clique.

Two interesting subclasses of chordal graphs discussed in this paper are strongly chordal graphs and split graphs. An *s-sun* (or *incomplete s-trampoline*) is a chordal graph with a Hamiltonian cycle $x_1, y_1, x_2, y_2, \ldots, x_s, y_s, x_1$ such that each y_i is of degree two. A *strongly chordal graph* (or *sun-free chordal graph*) is a chordal graph without any s-sun as an induced subgraph for all $s \geqq 3$. It was proved in [F1] that a graph is strongly chordal if and only if its vertices have a *strong elimination order* $v_1, v_2, \ldots, v_n$; i.e., for each i ($1 \leqq i \leqq n$), $N_i[v_j] \subseteq N_i[v_k]$ when $v_j, v_k \in N_i[v_i]$ and $j < k$. Note that a strong elimination order is always a perfect elimination order. Anstee and Farber [AF] gave $O(|V|^3)$ algorithms; Hoffman, Kolen, and Sakarovitch [HKS] gave an $O(|V|^3)$ algorithm; Lubiw [Lu] gave an $O(|E| \log^2 |E|)$ algorithm; Paige and Tarjan [PT] gave an $O(|E| \log |E|)$ algorithm; and Spinrad [S] gave an $O(|V|^2)$ algorithm for recognizing if a graph $G = (V, E)$ is strongly chordal and for finding a strong elimination order when G is strongly chordal.

A graph $G = (V, E)$ is *split* if its vertex set V can be partitioned into a clique V_1 and an independent set V_2. Every split graph is chordal, and a natural perfect elimination order is given by listing the vertices in V_2 first and then the vertices in V_1. Note that an s-sun in which $\{x_1, x_2, \ldots, x_s\}$ is a clique is a split graph.

3. Split graphs and NP-completeness. Let us recall the following two problems; see [CN1], [CN2], and [F2]. A *dominating set* D of a graph $G = (V, E)$ is a set of vertices such that every vertex not in D is adjacent to some vertex in D; i.e., $V = \cup\{N[v]: v \in D\}$. The *domination number* $\delta(G)$ of G is the minimum cardinality of a dominating set in G. A *2-stable set* of G is a set of vertices in which any two distinct vertices are of distance greater than 2. The *2-stability number* $\alpha_2(G)$ of G is the maximum cardinality of a 2-stable set in G. Note that $\alpha_2(G) \leqq \delta(G)$ for any graph G.

THEOREM 1. *It is* NP-*complete to determine the neighborhood-covering number, the clique-transversal number, and the domination number of a split graph with only degree-*2 *vertices in the independent set.*

Proof. Suppose that $G = (V, E)$ is a split graph without isolated vertices such that V is the disjoint union of a clique V_1 and an independent set V_2. Without loss of generality, we may assume that $N[x]$ is a proper subset of V_1 for any $x \in V_2$ (otherwise, we move x from V_2 to V_1). So the only maximal cliques of G are V_1 and $N[x]$ for all $x \in V_2$.

By the fact that $N[x] \subseteq N[y]$ for any $x \in V_2$ and $y \in N(x)$, we can always find a minimum neighborhood-covering set $C \subseteq V_1$. The same is true for clique-transversal sets and dominating sets. In fact, these three terms are then identical, and so $\rho_N(G) = \tau_C(G) = \delta(G)$.

Note that split graphs are in one-to-one correspondence to hypergraphs in which multiple edges are allowed. Vertices in the clique V_1 of a split graph G correspond to vertices of the hypergraph, and a nonisolated vertex y in the independent set V_2 corresponds to an edge, which is $N_G(y)$, of the hypergraph. It is then clear that $\delta(G)$ is equal to the transversal number of the corresponding hypergraph H_G, which is the minimum number of vertices meeting all edges. Hence the theorem follows from the fact that determining the transversal number of a 2-uniform hypergraph (i.e., a graph) is NP-complete; this problem is called the "vertex cover" problem and also the "hitting set" problem on pp. 190 and 222, respectively, of [GJ]. □

THEOREM 2. *It is* NP-*complete to determine the neighborhood-independence number, the clique-independence number, and the* 2-*stability number of a split graph with only degree*-3 *vertices in the independent set.*

Proof. A neighborhood-independent set of a split graph G must be of the form $\{x'x \in E: x \in S\}$ for some 2-stable set $S \subseteq V_2$. Moreover, a clique-independent set of G is of the form $\{N[x]: x \in S\}$ for some 2-stable set $S \subseteq V_2$. These, together with the fact that any 2-stable set of G is a subset of V_2, imply that $\alpha_N(G) = \alpha_C(G) = \alpha_2(G)$.

Also, $\alpha_2(G)$ is equal to the matching number, which is the maximum number of pairwise disjoint edges, of the corresponding hypergraph H_G as described in the proof of Theorem 1. Hence the theorem follows from the fact that determining the matching number of a 3-uniform hypergraph is NP-complete; a special case of this problem is called "three-dimensional matching" (see [GJ, p. 221]). □

Note that Chang and Nemhauser [CN1] proved that it is NP-complete to determine the domination number and the 2-stability number of a split graph without degree constraints. Moreover, the NP-completeness of the neighborhood-covering/independence problem was first observed by Lehel [L] by a different reduction. Let us note further that Theorems 1 and 2 remain valid under the assumption that the degrees of all vertices in the independent set are equal to k for some $k \geqq 3$.

For any graph $G = (V, E)$, we define the *neighborhood-split graph* $S(G)$ of G in the following way. The vertex set of $S(G)$ is $V \cup E$. In $S(G)$, any two vertices of V are adjacent, E is an independent vertex set, and an $e \in E$ is adjacent to a $v \in V$ if and only if $e \in E[v]$. Note that $S(G)$ has no isolated vertex if G has at least two vertices. The following statement is immediately seen from the definitions.

PROPOSITION 3. *For any graph G with at least one edge*, $\rho_N(G) = \delta(S(G))$ *and* $\alpha_N(G) = \alpha_2(S(G))$.

A structural relation between G and $S(G)$ is given by the following result.

THEOREM 4. *If G is strongly chordal, then so is* $S(G)$.

Proof. Since G is strongly chordal, its vertices have a strong elimination order $v_1, v_2, \ldots, v_n$. We order the vertices of $S(G)$ as $e_1, e_2, \ldots, e_m, v_1, v_2, \ldots, v_n$ in such a way that, for any $e_i = (v_{i_1}, v_{i_2})$, $e_j = (v_{j_1}, v_{j_2})$, $i < j$, $i_1 < i_2, j_1 < j_2$, we have that $i_1 < j_1$ or ($i_1 = j_1$ and $i_2 < j_2$). It is easy to check that this order is a strong elimination order of $S(G)$. Thus $S(G)$ is strongly chordal. □

Note that the strong elimination order of $S(G)$ in the proof of Theorem 4 can be obtained in linear time from a strong elimination order of G. By Proposition 3 and Theorem 4, we can use the linear algorithms [F2], [HKS] for the domination number and the 2-stability number to find the neighborhood-covering number and the neighborhood-independence number of a strongly chordal graph. However, $S(G)$ has $|V| +$

$|E|$ vertices and $O(|V||E|)$ edges. So this method gives an $O(|V||E|)$ algorithm. Actually, the algorithm in [W] is just this method without describing $S(G)$.

4. Efficient algorithms in strongly chordal graphs.

In this section, we derive efficient algorithms for finding $\rho_N(G)$, $\alpha_N(G)$, $\tau_C(G)$, $\alpha_C(G)$, and the corresponding optimum solution sets of a strongly chordal graph G. Suppose that a strong elimination order $v_1, v_2, \ldots, v_n$ of G is given. Note that this is also a perfect elimination order. For technical reasons, we add an isolated vertex v_0 to G.

Recall that $N_i[x]$ (respectively, $N_i(x)$) is the closed (respectively, open) neighborhood of vertex x in the subgraph G_i of G induced by $\{v_i, v_{i+1}, \ldots, v_n\}$. For simplicity, we call $v_i < v_j$ if $i < j$. For each $v_i \in V$, denote by $v_{m(i)}$ the maximum element in $N[v_i]$; i.e., $m(i) = \max\{j: v_j \in N[v_i]\}$.

LEMMA 5. *A clique-transversal set is a neighborhood-covering set for any graph.*

Proof. The lemma follows from the fact that each edge is contained in a maximal clique. □

LEMMA 6. *In a graph, replacing each edge of a neighborhood-independent set by a maximal clique containing it yields a clique-independent set.*

Lemmas 5 and 6, together with the min-max duality inequalities in § 1, give that, for any graph G,

$$\alpha_N(G) \leqq \rho_N(G) \leqq \tau_C(G) \quad \text{and} \quad \alpha_N(G) \leqq \alpha_C(G) \leqq \tau_C(G). \tag{4.1}$$

The idea of our algorithms is to find a clique-transversal set C, which is also a neighborhood-covering set by Lemma 5, a clique-independent set I_C, and a neighborhood-independent set I_N such that $|C| = |I_C| = |I_N|$. If such sets are found, then they are optimum solutions for the four problems, and all inequalities in (4.1) are equalities. This provides an algorithmic proof for a special case of the following result.

THEOREM 7 (see [LT]). $\alpha_C(G) = \alpha_N(G) = \rho_N(G) = \tau_C(G)$ *for any odd-sun-free chordal graph G.*

Algorithm NHD (NHD means NeighborHooD)

```
 1.  $C \leftarrow \emptyset$;
 2.  $I_C \leftarrow \emptyset$;
 3.  $I_N \leftarrow \emptyset$;
 4.  identify all $i$ such that $N_i[v_i]$ is a maximal clique;
 5.  for $i = 1$ to $n$ do
 6.    if $N_i[v_i]$ is a maximal clique and $N_i[v_i] \cap C = \emptyset$ then do
 7.      $v_p \leftarrow \max\{v_0\} \cup (N[v_i] \cap C)$; { Note that $v_p < v_i$ now. }
 8.      $v_j \leftarrow \min(N_i(v_i) - N_p[v_p])$;
 9.      $I_N \leftarrow I_N \cup \{v_i v_j\}$;
10.      $I_C \leftarrow I_C \cup \{N_i[v_i]\}$;
11.      $v_{m(i)} \leftarrow \max N[v_i]$;
12.      $C \leftarrow C \cup \{v_{m(i)}\}$;
13.      end if;
14.    end for.
```

THEOREM 8. *Algorithm* NHD *gives a minimum clique-transversal set C, a maximum clique-independent set I_C, and a maximum neighborhood-independent set I_N for a strongly chordal graph G in linear time when a strong elimination order is given.*

Proof. By steps 6, 11, and 12 of Algorithm NHD, the final C is a clique-transversal set of G.

In step 8, v_j must exist; otherwise, $N_i[v_i] \subseteq N_p[v_p]$ would imply that $N_i[v_i]$ is not a maximal clique. Suppose that v_iv_j and $v_{i'}v_{j'}$ (with $i' < i$) are two distinct edges of I_N that are both in some $E[v_q]$. Consider the set C at the beginning of iteration i, i.e., when step 8 is just done. For the case of $q \leqq i'$, since $q \leqq i' < i < j$, $v_{m(i')} \in N_{i'}[v_{i'}] \subseteq N_q[v_{i'}] \subseteq N_q[v_i] \subseteq N_q[v_j]$; i.e., v_iv_j and $v_{i'}v_{j'}$ both are in $E[v_{m(i')}]$. For the case of $i' < q$, since $i' < q \leqq m(i')$, v_i, $v_j \in N_{i'}[v_q] \subseteq N_{i'}[v_{m(i')}]$; i.e., v_iv_j and $v_{i'}v_{j'}$ both are in $E[v_{m(i')}]$. Note that $v_{m(i')} \in C$, since, in iteration i', we put $v_{i'}v_{j'}$ into I_N and $v_{m(i')}$ into C. By the choice of v_p and v_j (in steps 7 and 8), $v_pv_i \in E$ and $v_pv_j \notin E$, and $v_p \equiv v_{m(i'')}$ for some $v_{i''}v_{j''} \in I_N$ with $m(i') < m(i'') < i$. So $v_p = v_{m(i'')} \in N_{m(i')}[v_i] \subseteq N_{m(i')}[v_j]$, which contradicts $v_pv_j \notin E$. Therefore I_N is a neighborhood-independent set of G.

By Lemma 6, I_C is a clique-independent set of G. Since $|C| = |I_C| = |I_N|$, these three sets are optimum solutions of these four problems.

Next, we show that Algorithm NHD has running time linear in $|V| + |E|$. First, step 4 can be performed by Gavril's linear algorithm; see [G]. In iteration i, step 6 needs $|N_i[v_i]|$ operations to check if $N_i[v_i] \cap C = \varnothing$. This can be done if C is represented by a Boolean function f as follows:

$$f(i) = \begin{cases} 1, & \text{if } i \in C, \\ 0, & \text{if } i \notin C; \end{cases}$$

then we check if $f(q) = 0$ for all $v_q \in N_i[v_i]$. Step 7 can also be done in the same way.

For step 8, we keep an array $g(1{:}n)$ whose values are all initially zero. At the beginning of iteration i, $g(1{:}n)$ contains values $< i$. To find v_j of step 8, we first set $g(q) \leftarrow i$ for all $v_q \in N_p[v_p]$ and then check if $g(q) < i$ for each $v_q \in N_i[v_i]$ to obtain v_j. Note that $v_p \in N[v_i]$ and $v_p < v_i$ imply that $N_p[v_p] \subseteq N_p[v_i] \subseteq N[v_i]$. So step 8 needs $|N_i(v_i)| + |N_p[v_p]| \leqq 2|N[v_i]|$ operations.

Finally, steps 9, 10, and 12 need constant time, and step 11 needs $|N[v_i]|$ time. So the total running time is $O(\sum_i \deg(v_i) + 1) = O(|V| + |E|)$. □

We can modify Algorithm NHD slightly to get a simpler one as follows. First, we delete step 4 from the algorithm. Then we replace step 6 by step 6′ as follows:

6′. **if** $N_i[v_i] \cap C = \varnothing$ **then do**.

Also, insert step 8.5 between steps 8 and 9, shown below:

8.5. **if** v_j does not exist **then** go to 13.

All results are the same, except that we need not identify all maximal cliques.

THEOREM 9. *The modified algorithm gives a minimum clique-transversal set C, a maximum clique-independent set I_C, and a maximum neighborhood-independent set I_N for a strongly chordal graph G in linear time when a strong elimination order is given.*

Proof. The argument is the same as in the proof of Theorem 8, except that we must prove that, in iteration i, $N_i[v_i]$ is a maximal clique if and only if v_j exists.

Note that if v_j does not exist, then either $N_i(v_i) = \varnothing$, and so $N_i[v_i] = \{v_i\}$ is not a maximal clique; else $N_i[v_i] \subseteq N_p[v_p]$, and so $N_i[v_i]$ is not a maximal clique.

On the other hand, suppose that v_j exists. Then $N_i(v_i) \neq \varnothing$, and so $|N_i[v_i]| \geqq 2$. Suppose that $N_i[v_i]$ is not a maximal clique; i.e., $N_i[v_i]$ is a subset of some maximal clique $N_q[v_q]$, where $v_q < v_i$. Note that $N_q[v_q] \cap C \neq \varnothing$ by the algorithm now, say $v_{m(i')} \in N_q[v_q] \cap C$. Then $v_{i'}v_{j'}$, $v_iv_j \in E[v_{m(i')}]$. By a similar argument as in the proof of Theorem 8, to prove that I_N is neighborhood-independent, we obtain a contradiction. So $N_i[v_i]$ is a maximal clique.

5. Concluding remarks. According to Theorems 1 and 2, we cannot expect a good characterization for the class of graphs G satisfying $\rho_N(G) \leqq k$ (or $\alpha_N(G) \leqq k$) if k is large. We must note here that many graphs G contain some induced subgraph G' in which $\rho_N(G')$ is much larger than $\rho_N(G)$ (and the same holds even for $\alpha_N(G)$). The following problems, however, seem to be easier.

1. Let k by a given natural number. Characterize the graphs G in which $\rho_N(G')$ (and/or $\alpha_N(G')$) is at most k for all induced subgraphs G'. (For $k = 1$, the question is easy; cf. [LT].)

2. Prove that every neighborhood-perfect graph is perfect [LT].

3. Characterize neighborhood-perfect graphs.

4. Determine the algorithmic complexity of finding $\rho_N(G)$ and $\alpha_N(G)$ for planar graphs.

5. Find similar estimates and characterizations for covering and independence, when $E_k[v]$ is defined as the set of edges in the subgraph induced by the vertices of distance at most k from v. (With this notation, $E_1[v] = E[v]$.)

Acknowledgment. The authors thank J. Lehel for discussions on the subject.

REFERENCES

[AF] R. P. ANSTEE AND M. FARBER, *Characterizations of totally balanced matrices*, J. Algorithms, 5 (1984), pp. 215–230.

[AST] T. ANDREAE, M. SCHUGHART, AND ZS. TUZA, *Clique-transversal sets of line graphs and complements of line graphs*, Discrete Math., 88 (1991), pp. 11–20.

[CN1] G. J. CHANG AND G. L. NEMHAUSER, *k-domination and k-stability problems in sun-free chordal graphs*, SIAM J. Algebraic Discrete Meth., 5 (1984), pp. 332–345.

[CN2] ———, *Covering, packing and generalized perfection*, SIAM J. Algebraic Discrete Meth., 6 (1985), pp. 109–132.

[EGT] P. ERDÖS, T. GALLAI, AND ZS. TUZA, *Covering the cliques of a graph with vertices*, Discrete Math., to appear.

[F1] M. FARBER, *Characterization of strongly chordal graphs*, Discrete Math., 43 (1983), pp. 173–189.

[F2] ———, *Domination, independent domination and duality in strongly chordal graphs*, Discrete Appl. Math., 7 (1984), pp. 115–130.

[G] F. GAVRIL, *Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph*, SIAM J. Comput., 1 (1972), pp. 180–187.

[GJ] M. R. GAREY AND D. S. JOHNSON, *Computer and Intractability: A Guide to the Theory of* NP-*Completeness*, W. H. Freeman, San Francisco, 1979.

[Go] M. C. GOLUMBIC, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[HS] A. HAJNAL AND J. SURÁNYI, *Über die Auflösung von Graphen in Vollständige Teilgraphen*, Ann. Univ. Sci. Budapest Eötvös Sect. Math., 1 (1958), pp. 113–121.

[HKS] A. J. HOFFMAN, A. W. J. KOLEN, AND M. SAKAROVITCH, *Totally-balanced and greedy matrices*, SIAM J. Algebraic Discrete Meth., 6 (1985), pp. 721–730.

[L] J. LEHEL, private communication, 1987.

[LT] J. LEHEL AND ZS. TUZA, *Neighborhood perfect graphs*, Discrete Math., 61 (1986), pp. 93–101.

[Lu] A. LUBIW, *Doubly lexical ordering of matrices*, SIAM J. Comput., 16 (1987), pp. 854–879.

[PT] R. PAIGE AND R. E. TARJAN, *Tree partition refinement algorithms*, SIAM J. Comput., 16 (1987), pp. 973–989.

[SN] E. SAMPATHKUMAR AND P. S. NEERALAGI, *The neighborhood number of a graph*, Indian J. Pure Appl. Math., 16 (1985), pp. 126–132.

[S] J. P. SPINRAD, *Doubly lexical ordering of dense* 0-1 *matrics*, preprint.

[T] ZS. TUZA, *Covering all cliques of a graph*, Discrete Math., 86 (1990), pp. 117–126.

[W] J. WU, *Neighborhood covering and neighborhood independence in strongly chordal graphs*, preprint.

SIAM J. DISC. MATH.
Vol. 6, No. 1, pp. 30–43, February 1993

003

ON THE STRUCTURE OF THE STRONG ORIENTATIONS OF A GRAPH*

JOHN DONALD† AND JOHN ELWIN†

Abstract. Given a graph G, denote by Strong(G) the digraph whose vertices are the strong orientations of G, with a directed edge from orientation O to orientation O' if O' can be obtained from O by a "simple transformation," that is, by reversing the orientation on all edges in a directed cycle or on certain directed paths of G with orientation O. The main result is that Strong(G) is strongly connected. The computational complexity of determining a sequence of simple transformations between two strong orientations is shown to be polynomial in the size of G. These results depend on the notion of a minimal difference set, or MDS. A complete characterization of MDSs is given.

Key words. strongly connected, strong orientation, difference graph, handle basis

AMS(MOS) subject classifications. 05C20, 05C30, 05C50

Introduction. An orientation O of an undirected graph G is an assignment of a direction to each of the edges of G. This amounts to replacing each unordered pair representing an edge in the graph by an ordered pair. We denote the resulting digraph by (G, O). Given an undirected graph, we may ask whether it admits an orientation with given properties. For example, every undirected graph without self loops admits an acyclic orientation. A connected undirected graph admits a *strong orientation*, i.e., one for which the resulting digraph is strongly connected, if and only if the graph has no bridging edges [8].

This classical and elementary result invites certain refinements. Boesch and Tindell [2] and Chung, Gary, and Tarjan [4] ask when a partial orientation extends to a strong one. They provide a linear time algorithm for determining the answer. They also address the question of optimizing the orientation with respect to certain measures, such as the diameter of the resulting digraph, a problem shown to be NP-hard in Chvatal and Thomasen [3]. Roberts and Xu [9]–[12] consider optimizing the orientations of grid graphs with respect to various measures.

Given such interest, it is natural to view the set of strong orientations of a graph G as a search space Strong(G). Here we investigate the structure of this space. We endow Strong(G) with an operation we call *a simple transformation*, which connects pairs of its members. Our connectivity theorem, Theorem 3.1, states that the resulting graph or digraph is itself strongly connected; that is, we may connect any two strong orientations of a graph by a sequence of simple transformations.

If we reverse the orientations along any simple cycle in a strongly connected digraph, we get another strongly connected digraph. We should certainly call such an operation "simple." Moreover, cycles are easy to discover and recognize. However, the cycle reversal operation does not connect Strong(G).

We need the extra power provided by the handles of a *handle basis*. A handle basis $B = (h_i, i = 0, \cdots, d - 1)$ of a digraph G expresses G as the edge disjoint union of a sequence of subgraphs, where h_0 is a simple cycle, and h_i, $i > 0$, is a simple path that meets the union of $h_j, j < i$, exactly in its endpoints. The endpoints of h_i may be identical, in which case h_i is a cycle. The number d of handles is always the cyclomatic number

* Received by the editors August 17, 1988; accepted for publication (in revised form) March 12, 1992. This research was supported in part by National Science Foundation grant DCR-8602566.

† Department of Mathematical Sciences, San Diego State University, San Diego, California 92182.

(or topological dimension) of G. A digraph admits a handle basis if and only if it is strongly connected. The subgraph given by the union of any initial segment of handles in any handle basis for G is a strongly connected subgraph of G. Furthermore, given any strongly connected subgraph H of G, we may extend any handle basis for H to one for G. If the strongly connected graph G has n vertices and m edges, then it is easy to see that its dimension, and therefore the size of its handle bases, is $m - n + 1$. See Donald, Elwin, Hager, and Salamon [5] for applications of the handle basis concept.

Now let (G, O) be a strong orientation, let B be a handle basis of (G, O), and let h be any handle of B. This reversal of all the orientations of edges of h results in a new strong orientation. Accordingly, we define a *simple transformation* between strong orientations of a graph G as the reversal of all edges of some handle in some handle basis for some strongly connected graph (G, O). These are the operations that we show connect any two strong orientations of a given graph.

Given two strong orientations O and O' of G, there is a set of directed edges D of (G, O) that we must eventually reverse to get to (G, O'). We call D a *difference set* or a *difference graph*, according as we are referring only to the edge set in D or to the induced subgraph of G consisting of the edges of D together with their endpoints. We would like to know the structure of possible difference graphs D. D need not be a handle in some handle basis. For example, D need not be connected. Similarly, we would like to know what the *remainder graphs* $G \backslash D$ can look like, where $G \backslash D$ is the digraph remaining after deleting the edges (but not the vertices) of D.

A simple example illustrates the possible structure of both $G \backslash D$ and D (cf. Fig. 1). Let $G \backslash D$ consist of the vertex disjoint union of two strongly connected graphs G_1 and G_2. Let D consist of two edges, one connecting G_1 to G_2 and one connecting G_2 to G_1. Then G is strongly connected. D may be a cycle (Fig. 1(c)), a path of length two (Fig. 1(b)), or two disjoint edges (Fig. 1(a)), depending on where the endpoints of its edges are.

We have not succeeded in proving the connectivity theorem without first coming to understand something about the structure of possible difference graphs and remainder graphs. The critical concept is that of *minimal* difference set, that is, a nonempty difference set not properly containing a smaller difference set. The example Ds in the previous paragraph are minimal. Difference sets properly containing the edges of some handle in a handle basis are not. The difference set arising from a simple transformation, that is, the edges of just one handle, may or may not be minimal.

We first hoped to connect strong orientations by successively decreasing the size of the difference set. We would then have an orderly metric overseeing the process. Examples like the simple one above show, however, that we cannot do it; in certain situations, all handle reversals will enlarge the difference set.

What emerges instead (as Theorem 2.15) is a complete characterization of minimal difference graphs and their associated remainder graphs. Strangely, the only possible structures are generalizations of the above example. A minimal difference graph is either a cycle or a vertex disjoint union of simple paths. The associated remainder graph is a vertex disjoint union of strongly connected graphs, which the minimal difference graph connects in some cyclic order. Figure 1 illustrates the possibilities.

Given the characterizations embodied in Theorem 2.15, we may proceed directly to prove Theorem 3.1, basing the proof entirely on the structural characterization of Theorem 2.15.

The proof of Theorem 3.1 involves a bunch of structures that must exist, but the proof does not concern itself with how to find them. In § 4 we show that we can construct and/or identify these things in polynomial time. For example, we may identify handles

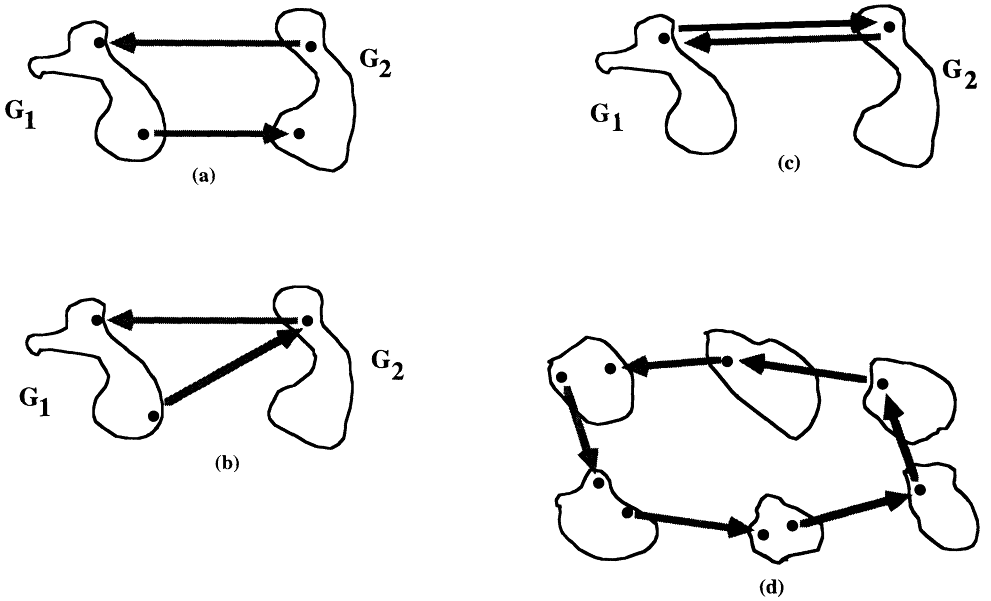


FIG. 1. *Each figure shows a minimal difference graph D (shown as heavy arrows) connecting the strong components (shown as blobs) of* $G\backslash D$. (a) *D is* 2 *disjoint edges*; (b) *D is a* 2*-path*; (c) *D is a* 2*-cycle*; (d) *A larger example, in which D is a vertex disjoint union of simple paths.*

and minimal difference graphs in polynomial time, and we may find a minimal difference set inside a difference set in polynomial time. Most importantly, in Theorem 4.5 we show that we may construct the sequence required by Theorem 3.1 in polynomial time.

We do not address here the issue of finding a shortest path in Strong(G) using simple transformations or of finding canonical paths. In fact, the inspiration for this work was [6], in which Irving and Leather use certain transformations to connect solutions to the stable marriage problem. Not only do they achieve a useful description of the solution space, but they also show, as a consequence, that the size of the solution space is #P-hard to compute. We have a preliminary description of the space Strong(G), which may prove useful, but so far we have failed to apply it to significant computational problems about that space.

1. Notation and conventions. Our notation and usages are, we hope, relatively standard and self explanatory. Graphs are normally undirected, unless otherwise indicated by context. Digraphs are directed graphs. Graphs and digraphs may have selfloops and multiple edges because all our results are invariant under subdivision of edges. Here, by a *subdivision* of the edge xy in a graph or digraph G, we mean the addition of some new point z to G and the replacement of edge xy with the new pair of edges xz and zy.

Let G be a graph or digraph. Let X be a set of edges on the vertices of G not belonging to the edges of G. Then $G[X]$ denotes the graph or digraph on the vertices of G whose edge set is the union of X and the edges of G.

Conversely, let X be a set of edges or (by abuse of notation?) a subgraph of the graph or digraph G. Then $G\backslash X$ denotes the graph obtained from G by deleting the edges in X (but *not* the vertices).

If X is a set of oriented edges or a digraph, the *reversal* X^R of X denotes the set of edges or the digraph obtained from X by reversing the orientations of all the edges.

A *source vertex* of a digraph is a vertex with indegree 0. Similarly a *sink vertex* is one with outdegree 0. Generalizing, a *source subgraph* of a digraph is a subgraph with no incoming edges. Similarly, a *sink subgraph* has no outgoing edges. In particular, if we construct the strong components of a digraph, *source components* are those with no incoming edges, and *sink components* are those with no outgoing ones.

Because we work with orientations of an undirected graph, we often speak of a graph G and an orientation O. In this case, (G, O) refers to the resulting digraph. When O is understood and the context is one of directed graphs, we may drop the explicit reference to O and refer simply to the digraph G. To shorten some of the terminology, when we have structures that are always directed, we may in some settings refer to them as *graphs*. For example, we say difference graphs rather than difference digraphs.

2. Minimal difference sets.

Let G be a graph and let O be a strong orientation. Then a *difference set* D for (G, O) is a set of O-oriented edges of G such that $(G\backslash D)[D^R]$ is strongly connected. Once we fix O, then the difference sets are in one-to-one correspondence with the orientations of G. Thus we may study difference sets instead of strong orientations. We call the subgraph of (G, O) induced by D a *difference graph.*

Certain obvious sets are always difference sets, regardless of the structure of the surrounding digraph. The point of the next proposition is that reversing part or all of a subdigraph so as to make it strongly connected will not destroy the strong connectivity of a containing digraph.

PROPOSITION 2.1. *Let (G, O) be a strongly connected digraph. If D is a strongly connected subdigraph, then D is a difference graph. More generally, suppose there exists a subdigraph H of G containing D such that $(H\backslash D)[D^R]$ is strongly connected. Then D is a difference graph for G.*

In terms of the structure of G, handle bases provide the critical source of difference sets. The next, equally obvious, proposition overlaps Proposition 2.1.

PROPOSITION 2.2. *Let D be the union of some of the handles of some handle basis of (G, O). Then D is a difference graph.*

There exist difference graphs that are neither strongly connected nor a union of some of the handles in a handle basis, such as the examples in Fig. 1. The situation with the more general statement in Proposition 2.1, however, is more subtle. The statement allows $H = G$, in which case all difference graphs trivially are of the given type. The import of the proposition, however, is that to find difference sets, we can seek proper subgraphs H that would become strongly connected after reversal of some subset D of its edges. Surprisingly, with a trivial exception, all difference graphs are of this type, i.e., their complements $G\backslash D$ are not "minimal" with respect to the ability of D to connect them (cf. Corollary 2.16), but proving this fact apparently requires Theorem 2.15, below.

We pursue some ideas relating to the connecting property of difference sets. We define a *connecting set* for the digraph H to be any set E of non-H edges on the vertices of H such that $H[E]$ is strongly connected. A connecting set is *minimal* if no proper subset is a connecting set. Obviously, any subset E of the edges of a strongly connected graph G is a connecting set for $G\backslash E$.

A connecting set E is *reversible* if E^R is also a connecting set. Thus a difference set D is just a reversible connecting set for the associated remainder graph $G\backslash D$. Such a connecting set need not exist if we disallow multiple edges. For example, let $G\backslash D$ be the complete acyclic graph consisting of points z_i, $i = 1, \cdots, n$, with edges z_iz_j, $i < j$. If we do not disallow multiple edges (as we do not), then, given a connecting set, we may always throw in its reverse to get a reversible connecting set. Reversible connecting sets enjoy one crucial property with respect to the digraph G they connect.

LEMMA 2.3. *Let D be a reversible connecting set for the nonstrongly connected digraph G. Then, for each source and sink point of G, D contains at least one inbound edge, and at least one outbound edge. More generally, for each source and sink component of G, D contains at least one inbound edge and at least one outbound edge.*

A second property of reversible connecting sets is a kind of duality.

LEMMA 2.4. *Let D be a reversible connecting set for G, which spans G; that is, such that every vertex of G is incident at some edge of D. Then G is a reversible connecting set for D.*

Certain connecting sets D automatically qualify as reversible.

LEMMA 2.5. *Let D be a connecting set for G for which the graph induced by the pairs in D is strongly connected. Then D is reversible.*

Such reversible connecting sets lack constraint—there are too many of them. For example, every strongly connected digraph is a reversible connecting set for its own vertices. We may even blow up those vertices into larger, strongly connected digraphs of arbitrary structure.

THEOREM 2.6. *Let D be a strongly connected digraph. Then D is a reversible connecting graph for its own vertices, and, in fact, D provides a reversible connecting graph for any vertex disjoint union of strongly connected graphs in one-to-one correspondence with the vertices of D.*

Proof. Let C_i be the strongly connected components whose vertex disjoint union forms G. Distinguish any vertex v_i of C_i and fix a correspondence between the vertices of D and the v_i. There results a correspondence between the edges of D and a set D' of non-D edges joining certain v_i. Then $G[D']$ is strongly connected, and the subgraph of $G[D']$ induced by D' is isomorphic to D.

To constrain our sets further, we define reversible *minimal* connecting sets (RMCs), i.e., reversible connecting sets that are also minimal connecting sets. The reversible connecting sets constructed in Theorem 2.6 are RMCs if and only if D is itself *minimally connected*; that is, if and only if for no edge e of D is $D\setminus e$ strongly connected. Difference sets need not, of course, be minimal connecting sets, but the difference sets of Fig. 1 *are* minimal connecting sets. Theorem 2.6 has an obvious generalization.

THEOREM 2.7. *Let D be a strongly connected digraph. Then some topologically equivalent digraph D is an* RMC *for its own vertices.*

Proof. By subdividing every edge of D, there results a minimally connected digraph.

We now ask what digraphs G with nonstrongly connected components admit RMCs and what their RMCs can look like. At first glance, RMCs may seem hard to construct in this case. Thus we offer the following conjecture.

CONJECTURE 2.8. *Let G be a topologically connected digraph that is not strongly connected. Then G has no* RMC.

EXAMPLE 2.9. It is not hard, however, to construct RMCs for disconnected digraphs whose components are not strongly connected. As a simple example, let G be two vertex disjoint edges vw and xy. Let D consist of the edges $\{wv, vy, yx, xw\}$. Then D is an RMC. Symmetry is lost, however. G is not an RMC for D. Furthermore, D^R is not an RMC for G.

CONJECTURE 2.10. *Let G have at least one topological component that is not strongly connected. Then G admits no* RMC *whose reversal is also an* RMC.

The point of the previous development is to focus on connectivity properties that are relevant to the issue of difference sets. In fact, although we take no position about the truth or falsity of Conjecture 2.10, we can prove its analogue for difference sets, and that is precisely the result we need.

We define a *minimal* difference set or MDS D for G to be a *nonempty* difference set no proper subset of which is a difference set. The notion of MDS for G differs slightly from that of RMC for $G \backslash D$ because RMCs may be empty. If $G \backslash D$ is strongly connected, then D is an MDS if and only if D is a singleton set. D cannot be an RMC for $G \backslash D$ in this case; the only RMC is empty. If $G \backslash D$ is not strongly connected, however, and D is an MDS for G, it is an RMC for $G \backslash D$. There is less anomaly here than we might think, because if $G \backslash D$ is not strongly connected, then D cannot be a singleton. This follows from our structure theorem to follow, but it is obvious if we note that in this case the unique edge of D must connect some sink component of $G \backslash D$ to some source component. D^R could then not connect $G \backslash D$.

MDSs enjoy a number of very constraining properties not shared by RMCs. We begin with the following lemma, which shows that only the simple cycle from among strongly connected digraphs can be an MDS.

LEMMA 2.11. *Let D be an* MDS. *If D is not a simple cycle, then D can contain no topological cycles.*

Proof. Suppose that D contains the topological cycle Z. If Z is coherently oriented in D, then, by Proposition 2.1, Z is a difference set. Otherwise, the reversal of some proper subset of the edges of Z will yield a coherently oriented cycle. Thus, again by Proposition 2.1, that subset of edges is a difference set.

MDSs enjoy a critical duality lacking for RMCs.

LEMMA 2.12. *Let D be an* MDS *for G. Then D^R is also an* MDS *for $G \backslash D[D^R]$.*

Proof. D^R is certainly a difference set. It is minimal because if a subset D_0 of D^R is reversible, then the subset $(D - D_0^R)$ of D is reversible.

The following property of MDSs is central for their characterization.

LEMMA 2.13. *Let D be an* MDS *for G. Then there can be no subdigraph H of $G \backslash D$ and proper subset D_0 of D such that D_0 is a connecting set for H. Equivalently, no proper strongly connected subdigraph K of G contains only a proper subset D_0 of D.*

Proof. If H and D_0 exist, then by Proposition 2.1, D_0^R is a difference set for $G \backslash D[D^R]$. Thus D^R is not minimal. Then, by Lemma 2.12, D is also not minimal. The equivalence follows by considering $H = K \backslash D$.

We now want to prove our characterization of MDSs. Let D be a difference set for the digraph G. The proof strategy depends on focussing not on the MDS per se, but on the remainder digraph $G \backslash D$. Specifically, we ask what structural restrictions, if any, obtain for $G \backslash D$. We aim to prove a theorem for difference sets that we can only conjecture for RMCs. The following lemma is the analogue for MDSs of Conjecture 2.10.

LEMMA 2.14. *Let D be an* MDS *for G. Then every topological component of $G \backslash D$ is strongly connected.*

Proof. Suppose the contrary and let C be such a topological component. Let X be a source component of C and let Y be some sink component (therefore, distinct from X) accessible in C from X via a simple path P_1 in C, whose internal vertices necessarily avoid X. Note that no edge of D is in P_1. By the strong connectivity of G, there is some simple path P_2 in G from Y to X whose internal vertices also avoid X. Now X, Y, P_1, and P_2 together form a strongly connected subdigraph K of G. K contains at least one edge of D because X is not accessible from Y in $G \backslash D$. K cannot contain D because, by Lemma 2.3 (with $G \backslash D$ playing the role of G in the statement of that lemma), the reversibility of D implies that D contains at least one outgoing edge e from X. The only such edge in K is the first edge of P_1, which we have observed contains no edges of D; i.e., e is not in K. Thus, K contains only a proper subset of D. By Lemma 2.13, D is not an MDS.

It is now easy to prove the main result of this section, a structural characterization of MDSs.

THEOREM 2.15. *Let D be an* MDS *for G. Then $G \backslash D$ is a vertex disjoint union of some number $r \geqq 1$ of strongly connected subdigraphs. If $r = 1$, then D is a singleton. Otherwise, indexing* mod r, *there is an ordering $C_1, \cdots, C_r$ of the components of $G \backslash D$ and there are pairs $\{x_i, y_i\}$ of points of C_i, $i = 1, \cdots, r$, such that the edges of D are exactly $y_i x_{i+1}$, $i = 1, \cdots, r$. D is a simple cycle or a vertex disjoint union of simple paths. Conversely, given a structure for $G \backslash D$ and D as just described, D is an* MDS.

Proof. That $G \backslash D$ is a vertex disjoint union of some number $r \geqq 1$ of strongly connected subdigraphs is the content of Lemma 2.14. If $r = 1$, then D must be singleton. We assume now that $r > 1$.

Consider now the digraph D' resulting from G by condensing the strong components C_i of $G \backslash D$ to points v_i. This is the quotient of G in which we identify all points belonging to the same strong component of $G \backslash D$. D' is also strongly connected. Suppose D' contains a proper simple cycle Z through some subset V_0 of the points v_i. Then, by lifting Z in some way to G and augmenting the result by paths in each C_j meeting the lift, we can construct a simple cycle in G, which meets D in a proper subset. By Lemma 2.13, no such simple cycle Z exists. Thus D', being strongly connected without proper cycles, must be a simple cycle. The order in which D' traverses its points provides the indexing by i, $i = 1, \cdots, r$, of the components C_i of $G \backslash D$.

Returning to D, we see that it must consist of edges $y_i x_{i+1}$, $i = 1, \cdots, r$, with x_i and y_i in C_i. The structure of D now depends on which i satisfy $x_i = y_i$. If all do, then D is a simple cycle. Otherwise, it consists of disjoint paths.

The converse is obvious.

As a result of Theorem 2.15, we see that there are no unusual species of MDS. A difference set may, of course, be more complicated, but if it is, then it must contain MDSs. It does not follow immediately from the theorem that we can efficiently recognize MDSs inside a given difference set. In fact, we can. We address this issue in § 4.

As a corollary to Theorem 2.15, we state a partial converse to Proposition 2.1 alluded to in the introduction to this section. In the context of a potential duality between difference sets D and remainder digraphs $G \backslash D$, it says that if D is minimal, then $G \backslash D$ is not, unless it is trivial.

COROLLARY 2.16. *Let D be an* MDS *for the digraph G. Then either $G = D$ or else there exists a proper subdigraph H of G containing D such that $(H \backslash D)[D^R]$ is strongly connected. If $G \neq D$, H can consist of D and simple paths from x_i to y_i for each i (cf. Thm.* 2.15).

Proof. This follows directly from the structure of G and $G \backslash D$.

The following easy fact also follows directly from the theorem, although it has a direct and obvious proof.

COROLLARY 2.17. *Let the* MDS *D consist of a single edge. Then $G \backslash D$ is strongly connected.*

Proof. By the theorem, D has the same number of edges as there are strong components of $G \backslash D$.

The following somewhat surprising corollary generalizes Corollary 2.17.

COROLLARY 2.18. *Let D be a nonempty difference set. Then D contains an edge e whose removal graph $G \backslash e$ consists of a linear chain of some number r of strongly connected subgraphs connected in order by $r - 1$ edges.*

Proof. D contains some MDS. Any edge in an MDS has this property by the theorem.

We conclude this section with an example.

We remind the reader that a subset D_0 of D is a difference set if and only if $(D^R - D_0^R)$ is a difference set in $(G\backslash D)[D^R]$ (cf. Lemma 2.12).

EXAMPLE 2.19. In Fig. 2(a), we show our strongly connected graph G together with an associated difference set D consisting of the heavy arrows. This difference set is an RMC, but it is not minimal as a difference set. Figure 2(a) emphasizes $G\backslash D$, whose structure is a vertex disjoint union of two edges and two noncyclic simple paths of length three. By our structure theorem, we know D is not an MDS. Figure 2(b) emphasizes D, whose structure is a vertex disjoint union of two noncyclic simple paths of length five. Figure 2(c) shows the target graph, where it is clear that the four diagonal members of D^R are not necessary for connectivity. Thus, as we may have suspected from Conjecture 2.10, D^R is not an RMC.

Returning now to the MDSs inside D, we note first that being minimal as a connecting set, D contains no singleton MDSs. Secondly, the left and right six-vertex rectangles in Fig. 2(a) are strongly connected, and $\{e_1, e_2\}$ connects them into a two cycle. Thus

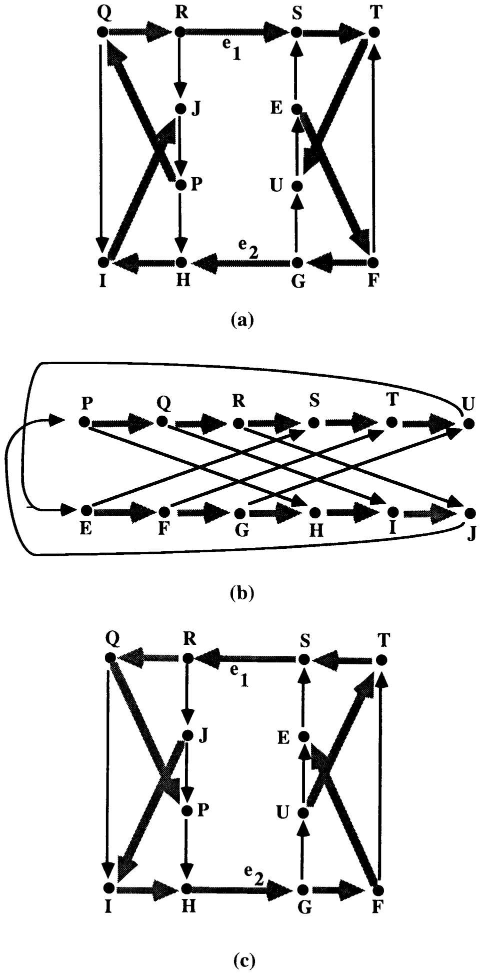


FIG. 2. (a) *A digraph G with the difference set D shown as heavy arrows. $G\backslash D$ has no strongly connected topological components. D is a minimal connecting set for $G\backslash D$.* (b) *A different view of G, showing D as a union of 5-paths.* (c) *The digraph after reversing D. D^R is not a minimal connecting set.*

$\{e_1, e_2\}$ is an MDS. Clearly if we reverse one of e_1 and e_2, we then must reverse the other to preserve connectivity. If we do not change e_1 or e_2, then we see by inspection that any other MDSs must lie inside one or the other six-vertex rectangle, preserving its strong connectivity. Further inspection reveals that there are no other possibilities.

As for MDSs inside D^R, the situation is more complicated. Each diagonal member of D^R is a singleton MDS, so these edges can belong to no other MDS. (MDSs need not be disjoint.) Every difference set of G contains some MDS, and thus contains $\{e_1, e_2\}$. Thus, by the remark preceding this example, every difference set inside D^R must avoid $\{e_1^R, e_2^R\}$. Reversing MDSs involving the other edges would create sources or sinks. Thus, there are only the four singleton MDSs inside D^R.

3. The space of strong orientations. Let Strong(G) denote the set of strong orientations of the graph G. We now consider a natural structure for this set.

Let O be an orientation in Strong(G). By Proposition 2.2, any handle in any handle basis of (G, O) is a difference set. We call a difference set corresponding to some handle a *simple* difference set. We say there is a *simple transformation* between the strong orientations O and O' if O' results from O by the reversal of some simple difference set for the strongly connected digraph (G, O).

We now endow Strong(G) with a digraph structure by saying that there is an edge connecting O to O' if there is a simple transformation between them.

THEOREM 3.1. *The digraph on* Strong(G) *with edges provided by simple transformations is strongly connected.*

Proof. Given two orientations O and O' in Strong(G), we must show there is a sequence of simple transformations connecting them. Equivalently, we consider the difference set D in (G, O) corresponding to O'. We must show there exists a sequence of simple difference sets whose reversals will accomplish the reversal of D.

It is sufficient to show that we can achieve the reversal of an MDS by a sequence of simple transformations. Accordingly, we may assume D is an MDS. By Theorem 2.15, we know that $G\backslash D$ is a union of strongly connected components C_i, $i = 1, \cdots, r$, which D connects into a cycle via r edges $e_i = y_i x_{i+1}$, with $x_i$and y_i points of C_i (here, we identify $r + 1$ with 1).

Consider first the case $r = 1$. Then $G\backslash D$ is strongly connected. Thus, we may extend a handle basis for $G\backslash D$ to one for G, in which, evidently, the unique edge of D is the last handle. Thus, reversing that edge is a simple transformation.

Assume now $r > 1$. Let P_i be a simple path in C_i connecting x_i to y_i. P_i will be trivial if $x_i = y_i$. Now we define a simple cycle Z by the sequence $P_1 e_1 P_2 e_2 \cdots P_r e_r$. We may let Z be the "zeroth" handle in some handle basis whose remaining elements we do not care about. Z defines a simple transformation. Applying that transformation leads to a new digraph consisting of the topologically connected subdigraphs C'_i resulting from the reversal in C_i of the edges of P_i, with the C'_i connected in reverse circular order of the indices by the edges e_i^R of D^R.

The effect of applying the simple transformation just described is to achieve the reversal of all of D, together with the reversal of each P_i. To complete the achievement of reversing D, we must reverse each P_i^R, thus returning the components C'_i to their original structure C_i. Note that each P_i^R is in fact a difference set by Proposition 2.1 applied to P_i^R inside C'_i. The C'_i need not, of course, be strongly connected. To show that we can effect the reversal of P_i^R, it is then sufficient to prove the following lemma.

LEMMA 3.2. *Let the difference graph D in the digraph G be a simple path of length r. Then we may effect the reversal of D by a sequence of at most r simple transformations.*

Proof of Lemma 3.2. We use induction on the length r of D. If $r = 1$, then D is an MDS, and, as proved above, reversal of D constitutes a simple transformation.

Assume $r > 1$. It suffices to show that there exists a simple difference set consisting of some initial portion of the path D. Then the associated simple transformation will yield a shorter difference graph that is also a simple path, and we can apply induction.

Let the path D consist of the edges $e_i = x_i x_{i+1}$, $i = 0, \cdots, r - 1$. By the strong connectivity of G, there exists a simple path P from x_r to x_0. Similarly, there exists a shortest simple path Q from x_0 to x_r in $(G\backslash D)[D^R]$. P and Q lie in distinct digraphs, but with regard only to their vertex sets, they need not be interior vertex disjoint, nor need they be disjoint from the interior of D. Let x_q be the first point of D after x_0 along Q. Let Q_0 be the segment of Q bounded by x_0 and x_q. Then Q_0 is edge disjoint from D^R, whence it may be viewed as a path in G, as well as in $(G\backslash D)[D^R]$.

The subdigraph H of G consisting of the not necessarily disjoint subdigraphs P, Q_0, and the tail $x_q \cdots x_r$ of D is strongly connected. Thus, we may begin a handle basis of G with any handle basis for H.

Now let p be the smallest index > 1 such that x_p is on P. The subgraph of D induced on $\{x_j \mid j \leqq \min(p, q)\}$ is edge disjoint from H and has both its endpoints in H so we may take this segment for the next handle h. The difference set given by h is an initial segment of D of length $q > 0$. Effecting the corresponding simple transformation shortens D, at which point we may apply induction.

To complete the proof of the theorem, we simply apply Lemma 3.2 to each P_i^R as difference set in G.

We remark that our use of the cycle Z in the preceding proof simplifies the notation and ideas, but we can replace Z by a handle properly contained in Z if $Z \neq G$. In fact, suppose some P_i is nontrivial. Then the corresponding C_i is not a point. Begin a handle basis for G with one for C_i. Now we can use the component of $Z\backslash P_i$ containing D as the next handle; that is, we use all of $Z\backslash P_i$ except for any isolated points inside P_i.

The diameter of a digraph is the maximum distance between two of its vertices, where the distance between vertices is the number of edges in a shortest path connecting them.

COROLLARY 3.3. *The diameter of* Strong(G) *is* $\leqq mn$, *where m is the number of edges in G, and n is the number of vertices in G.*

Proof. We have to bound the number of simple transformations required to connect two orientations O and O'. The process of applying simple transformations breaks down into stages, one for each MDS requiring treatment as in Lemma 3.2. Each stage decreases the size of the remaining difference set; therefore, there are at most m stages.

We now try to bound the number of simple transformations required in each stage. If the associated MDS has size one, then we require only one simple transformation, because, as seen previously, a singleton MDS is a handle in some handle basis. Otherwise the MDS is disconnected, and we have to apply one transformation based on a cycle containing all of the MDS edges, and then we have to reduce, in sequence, difference sets that are simple paths. The total length of these paths is at most $n - 2$. From Lemma 3.2, we can reverse a difference set that is a path of length r with at most r simple transformations. Thus, the total number of simple transformations required in one stage is at most $1 + (n - 2) = n - 1$.

The bound in Corollary 3.3 is not particularly sharp. If $n = 1$, then G consists of one vertex supporting a bouquet of m selfloops. In this case and when D equals all the selfloop edges of G, we must reverse the selfloops one at a time, and the distance between G and G^R is in fact $mn = m$.

If $n > 1$, then the dimension d of $G = m - n + 1 < m$. In particular, because we may move from G to G^R by reversing d handles, we may do this with fewer than m simple transformations. Difference sets in G that do not correspond to G^R have fewer than m edges. Moreover, each stage in the reversal process either involves a singleton D whose reversal takes just one simple transformation, or it shortens D by at least 2. Combining these observations, we see that we may sharpen our estimate of the diameter: if $n > 1$, then the diameter of Strong$(G) \leqq (m - 1)n/2$. Even this reasoning avoids the subtleties of the situation. For example, if D is large, then the paths constructed as the second part of an early phase will intersect D. Thus we will not have to return entire paths to their original condition. Also, it appears difficult to arrange for the independence of several phases in the transformation process without focussing each phase on a different part of the graph. The result is that the paths resulting from reversing the cycles at the beginning of a phase will also lie in different parts of the graph. We have no example proving that the diameter of G is on the order of mn.

We conclude this section with an example illustrating the main theorem.

EXAMPLE 3.4. The digraph G and difference set D are those of Example 2.19, depicted in Fig. 2(a). The difference set is not a simple path, so we have to find an MDS. In Example 2.19, we showed that the only MDS is $D_0 = \{e_1, e_2\}$. Following the proof of Theorem 3.1, we begin by reversing some simple cycle containing those edges. This cycle is unique, in this case, containing all of D, together with the edges JP and UE.

The first simple transformation yields a digraph like that in Fig. 2(c), but containing $(JP)^R$ and $(UE)^R$ instead of JP and UE. A moment's reflection shows that $(JP)^R$ and $(UE)^R$ are indeed difference sets, and thus, as in the argument in the proof of the theorem, their reversals are simple transformations.

4. Efficient determination of paths in Strong(G).

In Corollary 3.3, we showed that we may convert one strong orientation of G into another in at most mn simple transformations, where m is the number of edges in G, and n is the number of vertices. We have not shown, however, that there exists a computationally feasible way to find the simple transformations that will effect the conversion. Here, by computationally feasible, we mean computable in polynomial time. In this section, we show that all the objects in this paper may be identified or located in polynomial time. Here, by polynomial time (linear time, and so forth), we mean time polynomial in the number and size of all the objects in question. In the case of graphs, the size would typically be the sum $n + m$ of the number of vertices and edges. We denote the size of object X by size(X). If X_1 is a subobject of X, then because size$(X_1) \leqq$ size(X) in typical cases, a function polynomial of some degree in size(X_1) + size(X) is also polynomial of the same degree in size(X). It is standard (e.g., [1]) that we may find the strong components of a digraph G in time linear in size(G).

Throughout the discussion, m denotes the number of edges in G, and n denotes the number of vertices. If G is strongly connected and $n > 1$, then $m \geqq n$, so that a function polynomial of degree k in size(G) is polynomial of degree k in m.

We will use the standard "big-oh" notation: function $f(x)$ is $O(g(x))$ if for all sufficiently large x, there exists a constant C such that $f(x) \leqq Cg(x)$. Thus we may find the strong components of a digraph G in $O(n + m)$ time.

First, we consider the identification of handles. Simple cycles can always function as the first handle in some handle basis. We may thus consider only simple paths. Let P be a simple path in the strongly connected digraph G. Let G_P denote the induced subdigraph of G obtained by deleting interior points of P, that is, all points except the endpoints.

THEOREM 4.1. *Let P be a simple noncyclic path in the strongly connected digraph G. Then P is a handle if and only if both endpoints of P lie in the same strong component of G_P. In particular, we can tell in $O($size$(G))$ time whether P is a handle in some handle basis.*

Proof. Let P be a noncyclic handle in some handle basis. Consider the digraph $G_P = G - \text{interior}(P)$ obtained by deleting the interior vertices of P. Construct the strong components of G_P. Then P has both endpoints in the strong component containing all those handles in the handle basis that precede P.

Conversely, suppose both endpoints of P lie in the strong component C of G_P. Construct a handle basis for G by beginning with one for C. We may then take P for the next handle.

Because, given P, constructing G_P and determining its strong components can be done in time linear in $\text{size}(G) + \text{size}(P)$ (therefore linear in $\text{size}(G)$), we can tell in linear time whether P is a handle in some handle basis.

THEOREM 4.2. *Let G be a strongly connected digraph. Then we can construct a handle basis for G in linear time, i.e., in time that is $O(m)$.*

Proof. We claim, rather casually, that this may be done by an easy modification of standard depth first search (see Baase [1]). Every back edge encountered in the search produces another handle in the handle basis.

Because a set D of edges is a difference set in the strongly connected digraph G if and only if $(G\backslash D)[D^R]$ is strongly connected, we have a second obvious identification.

THEOREM 4.3. *Let D be a set of edges in the strongly connected digraph G. Then we can tell in linear time whether D is a difference set.*

From Theorem 2.15, we have the identification of MDSs.

THEOREM 4.4. *Let D be a difference set in the digraph G. Then we can tell in linear time whether D is an* MDS.

Proof. We only have to check the structure specified by Theorem 2.15. To do this, construct the strong components of $G\backslash D$ in time linear in $\text{size}(G) + \text{size}(D) = \text{size}(G)$ and then check (in linear time) whether D joins them in some circular order.

We now turn to the construction of paths in $\text{Strong}(G)$. Our goal is to show that we may construct them in polynomial time.

THEOREM 4.5. *Let O and O' be strong orientations of the graph G. Then we may construct a path from (G, O) to (G, O') in* $\text{Strong}(G)$ *in $O(m^3)$ time, i.e., in time cubic in size(G).*

Proof. We use a series of lemmas.

LEMMA 4.6. *Let D be a difference set that is a simple noncyclic path in the digraph (G, O). Then we may reverse D with $\leqq$ size(D) simple transformations. In particular, we may reverse D in $O(\text{size}(D)^* \text{ size}(G))$ time, and, in particular, in $O(mn)$ time.*

Proof of Lemma 4.6. Following the proof of Lemma 3.2, we see that we must construct paths P and Q. It is standard (e.g., Baase [1]) that such paths can be constructed in linear time. Now we need not construct a handle basis for the strongly connected digraph H as in that proof. We simply effect the reversal of the initial segment of that path or the segment from the initial vertex of D to the first point of the return path Q common to D and Q, as in that proof.

Because we need at most size(D) of these simple transformations (following, for example, the proof of Corollary 3.3), the total time is $O(\text{size}(D)^* \text{ size}(G))$.

LEMMA 4.7. *Let D be an* MDS *in the digraph G. Then we may reverse D using simple transformations in $O(mn)$ time.*

Proof of Lemma 4.7. Referring to the proof of Theorem 3.1, we may find the strong components C_i of $G\backslash D$ and then the cycle Z in linear time. We may reverse Z in linear time. Now the sum of the sizes of the paths P_i is at most n. Each reversed path P_i^R may, by Lemma 4.6, be returned to its unreversed state in time that is $O(\text{size}(P_i)^* \text{ size}(G))$. Adding these up, we see that all the paths P_i^R may be returned to their unreversed states in time, which is $O(n^* \text{ size}(G))$, i.e., in time $O(mn)$.

LEMMA 4.8. *Given a difference set D in the digraph G, we may find an* MDS *inside D in time linear in* size(D)* size(G), *and, in particular, in $O(m^2)$ time.*

Proof of Lemma 4.8. We apply Corollary 2.18 and analyze the structure of $G\backslash e$ for each edge e of D. Each analysis is $O(\text{size}(G))$. One of these analyses must yield the structure described in that corollary.

We complete the proof of Theorem 4.5 by using the above lemmas applied to the analysis in the proof of Corollary 3.3. In that proof, we see that the connection of O to O' occurs in at most size$(D) = O(m)$ stages. We must find the computational cost of each stage.

The work of one stage consists of two actions: (1) finding an MDS inside D and (2) carrying out the reversal of the MDS. The computational costs of these actions are $O(m^2)$ and $O(mn)$ respectively, by Lemmas 4.7 and 4.8. Thus, the cost of one stage is $O(m^2)$.

Because there are $O(m)$ stages, the total cost is $O(m^3)$.

There are several places where we could attempt to improve this bound, but for our purposes it serves to establish the computational feasibility of connecting orientations.

5. Concluding remarks. The space Strong(G) is quite large, containing at least 2^d points, where d is the topological dimension or cyclomatic number of G. This is because given a strong orientation, any subset of the handles in any handle basis specifies a difference set. Thus, because the basis contains d handles, it can specify 2^d difference sets. Because the topological dimension $d = m - n + 1$, where m and n are the numbers of edges and vertices of G, we see that if m is large in comparison with n, then d and m are comparable. In this case, the size of the space Strong(G) is at least on the order of 2^m.

Finding paths in such a large space could reasonably take time exponential in m, but by Theorem 4.5 it does not. Of course, by Corollary 3.3, we already knew that such paths could be short, that is, logarithmic in the size of the point set involved. However, the path finding construction also avoids all branching. The path in Strong(G) it finds need not, however, be optimal, because it does not check whether a difference set is already a path in G before looking for an MDS inside it.

We do not know a method for computing the size of Strong(G) in polynomial time. Such a method would provide even more information about the apparent simplicity of this large set.

As mentioned in the Introduction, one inspiration for the present work was the paper of Irving and Leather [6], describing the space of solutions to the stable marriage problem and showing, via that description, that counting that space is #P-hard. Other work related to ours in purpose deals with the set of acyclic orientations, which in Stanley [13] is shown to be counted by the chromatic polynomial evaluated at -1, and which in Lineal [7] is shown, by arguments based on Valiant [14], to be #P-hard to count. These arguments do not, however, depend on structural analysis of the space to be counted. Instead, they use algebraic ideas and manipulation of the generating polynomial for graph colorings. We do not know which style holds more promise for analysis of Strong(G).

REFERENCES

[1] S. BAASE, *Computer Algorithms, Introduction to Design and Analysis*, Addison-Wesley, Reading, MA, 1978.

[2] F. BOESCH AND R. TINDELL, *Robbins's theorem for mixed multigraphs*, Amer. Math. Monthly, 87 (1980), pp. 716–719.

[3] V. CHVATAL AND G. THOMASSEN, *Distances in orientations of graphs*, J. Combin. Theory Ser. B, 24 (1978), pp. 61–75.

[4] F. R. K. CHUNG, M. R. GAREY, AND R. E. TARJAN, *Strongly connected orientations of mixed multigraphs*, Networks, 15 (1985), pp. 477–484.

[5] J. DONALD, J. ELWIN, R. HAGER, AND P. SALAMON, *Handle bases and bounds on the number of subgraphs*, J. Combin. Theory Ser. B, 42 (1987), pp. 1–13.

[6] R. W. IRVING AND P. LEATHER, *The complexity of counting stable marriages*, SIAM J. Comput., 15 (1986), pp. 655–667.

[7] N. LINEAL, *Hard enumeration problems in geometry and combinatorics*, SIAM J. Algorithm Discrete Meth., 7 (1986), pp. 331–335.

[8] H. E. ROBBINS, *A theorem on graphs, with an application to a problem of traffic control*, Amer. Math. Monthly, 46 (1939), pp. 281–283.

[9] F. S. ROBERTS AND Y. XU, *On the optimal strongly connected orientations of city street graphs* I: *Large grids*, SIAM J. Discrete Math., 1 (1988), pp. 199–222.

[10] ———, *On the optimal strongly connected orientations of city street graphs* II: *Two east-west avenues or north-south streets*, Networks, 19 (1989), pp. 221–233.

[11] ———, *On the optimal strongly connected orientations of city street graphs* III: *Three east-west avenues or north-south streets*, Networks, 22 (1992), pp. 109–143.

[12] ———, *On the optimal strongly connected orientations of city street graphs* IV: *Four east-west avenues or north-south streets*, Discrete Appl. Math., to appear.

[13] R. P. STANLEY, *Acyclic orientations of graphs*, Discrete Math., 5 (1973), pp. 171–178.

[14] L. G. VALIANT, *The complexity of enumeration and reliability problems*, SIAM J. Comput., 8 (1979), pp. 410–421.

SIAM J. DISC. MATH.
Vol. 6, No. 1, pp. 44–56, February 1993

GARDEN OF EDEN CONFIGURATIONS FOR CELLULAR AUTOMATA ON CAYLEY GRAPHS OF GROUPS*

ANTONIO MACHÌ† AND FILIPPO MIGNOSI‡

Abstract. The tessellation of the plane given by square cells of equal size can be considered the Cayley graph of the free abelian group of rank 2. This group has polynomial growth. The theorems of Moore [*Symposium on Applied Mathematics*, Vol. XIV, American Mathematical Society, Providence, Rhode Island, 1962, pp. 17–33] and Myhill [*Proceedings of the American Mathematical Society*, 14 (1963), pp. 685–686] on the existence of Garden of Eden configurations for an automaton defined on such a graph are extended to Cayley graphs of groups whose growth function is not exponential. Examples are given of Cayley graphs of groups of exponential growth for which these theorems do not hold.

Key words. automata, groups, Cayley graphs, growth functions

AMS(MOS) subject classifications. 68Q80, 05C25

1. Introduction.

In the classical theory of cellular automata [10], we consider automata on the lattice of integer points of Euclidean n-space. This "universe" is discrete and homogeneous. Time is discrete, and the transition function is deterministic and local (the state at time $t + 1$ at any point only depends on the states of its neighbours at time t). For this structure, Moore [5] gave a sufficient condition for the existence of configurations, called Garden of Eden configurations, which can only appear at time $t = 0$; that is, there is no configuration at time $t - 1$ that will give rise to the given configuration at time t. Moore's condition (the existence of "mutually erasable patterns") was also proved to be necessary by Myhill [7].

The lattice of Euclidean n-space is the Cayley graph of the free abelian group of rank n. This group has polynomial growth (a concept due to Milnor [4]). The purpose of this paper is to show that the theorems of Moore and Myhill hold in the more general case of automata for which the universe is the Cayley graph of a finitely generated group whose growth is not exponential. In particular, this answers a question of Schupp, who asked whether these theorems hold for groups of polynomial growth [9].

In § 1 we give a proof of Moore's theorem. We modify the final part of the proof to reveal the point that allows both theorems to be generalized. The theorems are then proved in the more general setting. In the last section, examples are given to show that when the growth is exponential the theorems need not hold.

2. The tessellation structure.

Consider the Euclidean plane subdivided into square cells of equal size; call this the *Euclidean tessellation*. Let A, $|A| > 1$ be a finite set (the set of *states*), and let c be a map assigning to each cell of the tessellation one of the states of A; call c a *configuration*, and let $\mathscr{C}$ be the set of all such maps c. For any cell v, a set of nearby cells, the *neighbourhood* $N(v)$ of v, is specified. Figure 1 shows the neighbourhoods of a cell v according to von Neumann [10] and Moore [5]. (Note that v is included in $N(v)$.) Assume now that the state of each cell changes with time in such a way that the state of v at time t depends only on the states of the cells of $N(v)$ at time $t - 1$, and that this functional dependence, f, say, is the same for all cells of the plane.

* Received by the editors April 12, 1990; accepted for publication (in revised form) February 13, 1992.

† Dipartimento di Matematica, Università "La Sapienza," Piazzale Aldo Moro 2, I-00185 Roma, Italy, and Institut Blaise Pascal–Laboratoire d'Informatique Theorique et Programmation (IBP-LITP), Université Paris VII, 2, Place Jussieu, 75251 Paris, France.

‡ Dipartimento di Matematica, Università di Palermo, via Archirafi 34, I-90123 Palermo, Italy and IBP-LITP, Université Paris VII, 2, Place Jussieu, 75251 Paris, France.

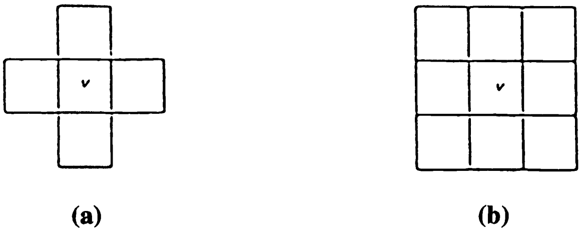


FIG. 1. *The von Neumann and Moore neighbourhoods of a cell v.*

An *array* F is a block of cells (perhaps the whole tessellation). A restriction $c|_F$ of a configuration c to an array F is called *Garden of Eden* (GOE) if, for any configuration c^* such that $c^*|_F = c|_F$, there is no c' at time $t-1$ that will give rise to c^* at time t. The restriction of a configuration to a finite array will be called a *pattern*. Two patterns p_1 and p_2 on an array F are said to be *mutually erasable* if they are distinct, and all pairs of configurations c_1 and c_2 that at time t agree outside F and agree with p_1 and p_2, respectively, on F, give rise at time $t+1$ to the same configuration. A pattern p' *contains m copies* of a pattern p if there exist m disjoint subsets of the array of p' and each of these subsets contains a copy of p.

THEOREM 1. *If there are mutually erasable patterns for a nearest neighbour cellular automaton on the Euclidean tessellation, then there are* GOE *patterns.*

Proof. Without loss of generality, we can assume that all patterns are square. Let n be an integer such that there is an array L of size $n \times n$, which is the support of two mutually erasable patterns. Consider, for an integer k to be chosen later, a square array B of size $kn \times kn$ consisting of k^2 copies of L. Let R be the equivalence relation defined on the set of patterns on L as follows: Two patterns are equivalent if they are equal or mutually erasable. By assumption, there are at least two mutually erasable patterns on L, so that the number of equivalence classes is at most $a^l - 1$, where $a = |A|$ and l is the number of cells of L. This relation R induces an equivalence relation R^* on the set of patterns with support B: p_1 and p_2 are equivalent if each of the k^2 patterns induced on L by p_1 has the relation R to the pattern induced by p_2 in the corresponding location. Then the number of classes of R^* is at most $(a^{n^2} - 1)^{k^2}$. Let B^- be the array obtained by removing a border of cells from B of width 1. Now any pattern of the kind described in the blocks L of B can be changed to any other such pattern by a sequence of exchanges using equivalence R one block at a time, and by mutual erasability the output configurations after each exchange coincide. Thus two patterns on B that are equivalent under R^* lead to the same pattern on B^- at time $t+1$. The number of all possible patterns on B^- is $a^{(kn-2)^2}$. Thus, at time $t+1$, there will be a pattern b on B^- that is not a successor of a pattern on B, and therefore a GOE pattern, provided that there exists a sufficiently large k such that

$$a^{(kn-2)^2} > (a^{n^2} - 1)^{k^2}. \tag{1}$$

By taking logarithms with base a, (1) is equivalent to the following inequalities:

$$(kn-2)^2 > k^2 \log(a^{n^2} - 1),$$

$$\frac{(kn)^2}{(kn-2)^2} \frac{\log(a^{n^2} - 1)}{n^2} < 1.$$

Now $\log(a^{n^2} - 1) < \log a^{n^2} = n^2$, and, a being greater than 1, $\log(a^{n^2} - 1)$ is positive.

Thus $\log(a^{n^2}-1)/n^2$ is positive and less than 1. However, $\lim_{k\to\infty}(kn)^2/(kn-2)^2 = 1$, so that such a k certainly exists where (1) holds. □

The converse of this theorem was proved by Myhill [7]: *If there are* GOE *patterns, then there are mutually erasable patterns*. A proof of this result is given in the next section (Theorem 2, part (ii)).

Remark. The proof of Moore's theorem given above used the following properties of the Euclidean tessellation.

(a) Given a square L, it is always possible to find a sequence of squares B of increasing size that are filled up completely by copies of L;

(b) For any sequence of squares (and, in particular, for a sequence of squares as in (a)), the ratio between the number of cells of a square B of the sequence and that of B^- tends to 1 as the size of the squares increases.

These two properties can be weakened; this will allow us to show that both Moore's theorem and Myhill's theorem hold for universes more general than Euclidean tessellations.

3. Cellular automata on Cayley graphs. Now consider as cells the vertices of a graph $\mathscr{G}$, and as neighbours of a vertex v the vertices connected to v by an edge. von Neumann's and Moore's models are represented by Figs. 2(a) and 2(b). These are examples of Cayley graphs of groups. Figure 2(a) is the Cayley graph of the group $Z \times Z$, with generators a and b and relation $ab = ba$. Figure 2(b) is the Cayley graph of the same group, with generators a, b, ab, and $a^{-1}b$, and the same relation.

DEFINITION 1. Let G be a group and X a set of generators for G. The *Cayley graph* $\mathscr{G}$ of G relative to X is the graph whose vertices are the elements of G, two vertices g and g' being joined by an arc if there exists $x \in X \cup X^{-1}$ such that $gx = g'$. The *ball* $B(v, n)$ with center v and radius n is the subgraph of $\mathscr{G}$ whose vertices are the vertices of $\mathscr{G}$ having (graph) distance at most n from v. For a subgraph $\mathscr{G}'$ of $\mathscr{G}$, the number of vertices of $\mathscr{G}'$ will be denoted by $|\mathscr{G}'|$.

Remark. If $\mathscr{G}$ is a Cayley graph of G and g is a fixed element of G, then multiplication on the left by g is an isomorphism of $\mathscr{G}$. Thus Cayley graphs are translation-invariant.

The following definitions are motivated by the discussion of the previous section.

DEFINITION 2. A (deterministic) *cellular automaton* on a graph $\mathscr{G}$ is a 4-tuple $\mathscr{A} = (A, \mathscr{G}, N, f)$, where

(i) A is a finite set, $|A| = a > 1$, called the *alphabet* or the set of *states*;

(ii) $\mathscr{G}$ is the Cayley graph of a finitely generated infinite group G;

(iii) For a vertex v of $\mathscr{G}$, $N(v)$ is the subgraph whose vertices are v and the vertices of $\mathscr{G}$ connected to v by an edge (the *neighbouring vertices* of v). Note that $N(v) = vN(1)$, where 1 is the identity element of G;

(iv) f is a function $A^{N(1)} \to A$. It extends to all $N(v)$, $v \in \mathscr{G}$, by translating v to 1. Such an f is called a *local map*.

DEFINITION 3. A *configuration* c is a map $c : \mathscr{G} \to A$ that associates a state with each vertex of $\mathscr{G}$. The set of all configurations will be denoted by $\mathscr{C}$. If F is a nonempty subgraph of $\mathscr{G}$, then $c|_F$ denotes the restriction of the configuration c to F; F is the *support* of $c|_F$. If $F = N(v)$ for some v, then a restriction $c|_{N(v)}$ is an element of $A^{N(v)}$. If F is finite, $c|_F$ is called a *pattern*; the number of vertices of F will be denoted by $|F|$.

DEFINITION 4. Given a cellular automaton $\mathscr{A}$, a *transition map* is a function on configurations that applies one timestep of the cellular automaton; i.e., it is a function $\tau : \mathscr{C} \to \mathscr{C}$ such that

$$\tau(c)(v) = f(c|_{N(v)});$$

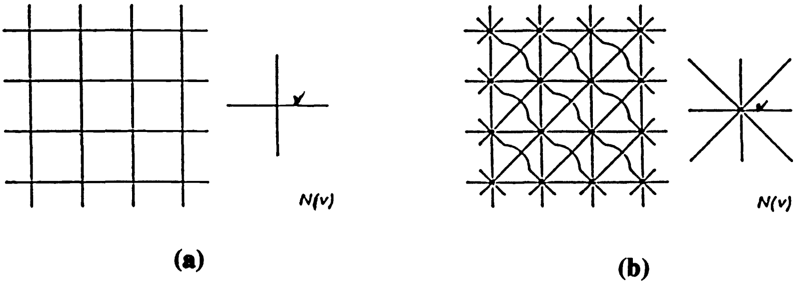


FIG. 2

in other words, the state at v in the configuration $\tau(c)$ only depends on the states of the neighbours of v in the configuration c. This dependence, expressed by f, is the same for all vertices of $\mathcal{G}$.

DEFINITION 5. Let F be a finite subgraph of $\mathcal{G}$, c a configuration, and τ a transition map. The restriction $c|_F$ is called a *Garden of Eden* (GOE) *pattern* if, for all configurations c^* such that $c^*|_F = c|_F$, there is no configuration c' such that $\tau(c') = c^*$. A configuration c is GOE if there is no configuration c' such that $\tau(c') = c$.

Remark. If τ is considered as time, and if we consider the sequence of configurations c, $\tau(c)$, $\tau(\tau(c))$, $\cdots$ as the sequence of the states of the universe as time passes, then a GOE pattern is a restriction that can only appear in the initial configuration c.

DEFINITION 6. Two patterns p_1 and p_2 are said to be *mutually erasable* if, for all pairs of configurations c and c^* such that $c|_F = p_1$, $c^*|_F = p_2$ and $c|_{\mathcal{G}\backslash F} = c^*|_{\mathcal{G}\backslash F}$, we have $\tau(c) = \tau(c^*)$. Note that, if $F \subseteq E$ and if $c|_F$ and $c^*|_F$ are two erasable patterns, then so are $c|_E$ and $c^*|_E$.

DEFINITION 7. For F a subgraph of $\mathcal{G}$, the subgraphs F^+ and F^- are defined as follows:

$$F^+ = \bigcup_{v \in F} N(v);$$

in other words, F^+ is obtained by adding to F the subgraph whose vertices are neighbours of some vertex of F. F^-, below, is the union of the $N(v)$, $v \in F$, that are contained in F:

$$F^- = \bigcup_{v \in F, N(v) \subseteq F} N(v).$$

The difference set $F \backslash F^-$ is the *boundary* of F. Clearly, $F \subseteq (F^+)^-$. However, equality does not always hold, nor do we always have $F \subseteq (F^-)^+$. Also, $\mathcal{G} \backslash F \subseteq (\mathcal{G} \backslash F^-)^-$.

DEFINITION 8. Let F_1 and F_2 be two subgraphs of $\mathcal{G}$. A mapping $\sigma : F_1 \to F_2$ *embeds* F_1 in F_2 if

(i) It is one-to-one;

(ii) It commutes with N: $\sigma(N(v)) = N\sigma(v)$.

An embedding map is also called an (injective) *isomorphism*. If $\sigma_1, \sigma_2, \ldots, \sigma_m$ embed F_1 in F_2, then F_2 *contains m copies* of F_1 if $\sigma_i(F_1) \cap \sigma_j(F_1) = \varnothing$, for $i \neq j$.

The latter definition extends to restrictions of configurations as follows.

DEFINITION 9. For σ, F_1, and F_2 as in Definition 8, and c a configuration, define

$$\sigma(c|_{F_1})(v) = c(\sigma^{-1}(v)), \qquad v \in F_2.$$

A restriction $c^*|_{F_2}$ is said to *contain m copies* of $c|_{F_1}$ if F_2 contains m copies of F_1 under m isomorphisms $\sigma_1, \sigma_2, \dots, \sigma_m$, and $\sigma_i(c|_{F_1}) = c^*|_{\sigma_i(F_1)}$, all i's.

It is clear that copies of GOE or of mutually erasable patterns still have these properties.

LEMMA 1. *Let c be a configuration and let $F \subseteq \mathscr{G}$. Then the restriction $\tau(c)|_{F^-}$ only depends on $c|_F$ and not on c.*

Proof. It is clear that, if c^* is such that $c^*|_F = c|_F$, then $\tau(c^*)|_{F^-} = \tau(c)|_{F^-}$. □

LEMMA 2. *Let F be a finite subgraph of $\mathscr{G}$, and assume that F^+ is not the support of two mutually erasable patterns (that is, for all c and c^* such that $c|_{F^+} \neq c^*|_{F^+}$ and $c|_{\mathscr{G}\setminus F^+} = c^*|_{\mathscr{G}\setminus F^+}$, we have $\tau(c) \neq \tau(c^*)$). If two configurations c_1 and c_2 agree on $F^+\setminus F^-$ and disagree on F^-, then their iterates $\tau(c_i)$ disagree on F.*

Proof. Suppose that there existed c_1 and c_2 that agree on $F^+\setminus F^-$ but disagree on F^-, having $\tau(c_1)$ and $\tau(c_2)$ agreeing on F. From a new configuration c_3 that agrees with c_2 on F^+ and with c_1 everywhere else. Then we claim that $\tau(c_1) = \tau(c_3)$. Equality holds outside F^+ because c_1 and c_3 agree on $\mathscr{G}\setminus\mathscr{F}$ since c_1 and c_2 agree on $F^+\setminus F^-$. It holds on F by assumption. Thus $c_1|_{F^+}$ and $c_2|_{F^+}$ are mutually erasable, since they differ only on F^-, a contradiction proving the lemma. □

Consider now two finite subgraphs L and B of $\mathscr{G}$, and assume that B contains m copies of L under $\sigma_1, \sigma_2, \dots, \sigma_m$. Let $l = |L|$, $b = |B|$, $a = |A|$, and $u = |B\setminus B^-|$. The following theorem includes Moore's theorem and Myhill's theorem as special cases (Theorem 3).

THEOREM 2. *Let L, B, and a be as above, and suppose that*

$$a^{b-u} > (a^l - 1)^m a^{b-ml}. \tag{2}$$

Then

(i) *If L is the support of two mutually erasable patterns, then B^- is the support of a* GOE *pattern;*

(ii) *If L is the support of a* GOE *pattern, then B^+ is the support of two mutually erasable patterns.*

Proof. (i) The proof closely follows that of Moore's theorem. Let R be the equivalence relation defined on the set of patterns having support L by declaring equivalent two patterns if they are equal or mutually erasable. The number of R-equivalence classes is at most $a^l - 1$. R induces an equivalence relation R^* on the set of patterns having support B as follows: $p_1 R^* p_2$ if $p_1(v) = p_2(v)$ for v not in one of the copies of L, and, if $\sigma_i^{-1}(p_1|_{\sigma_i(L)})$ has the relation R with $\sigma_i^{-1}(p_2|_{\sigma_i(L)})$, all i's. The number of equivalence classes of R^* is therefore at most $(a^l - 1)^m a^{b-ml}$. It is easy to see that two patterns with support B that are R^*-equivalent are either equal or mutually erasable. Therefore, if $c_1|_B R^* c_2|_B$, then Lemma 1 implies that $\tau(c_1)|_{B^-} = \tau(c_2)|_{B^-}$. The number of patterns having support B^- that are of the form $\tau(c)|_{B^-}$ for some c is at most equal to the number of R^*-equivalence classes. The total number of patterns on B^- being a^{b-u}, (2) implies that there exists a pattern on B^- that is not of the form $\tau(c)|_{B^-}$ for some configuration c, that is, a GOE pattern.

(ii) Assume the contrary. Lemma 2 applied to B^+ implies that the number of distinct patterns with support B of the form $\tau(c)|_B$ (and therefore not GOE) is at least equal to the number of distinct patterns on B^-; these are a^{b-u} in number. On the other hand, there can be no more than $(a^l - 1)^m a^{b-ml}$ patterns on B not containing at least one copy of the GOE pattern under one of the σ_i's. Since a pattern containing a GOE pattern is also GOE, the number h of non-GOE patterns with support B satisfies the inequality

$$a^{b-u} \leqq h \leqq (a^l - 1)^m a^{b-ml},$$

which contradicts (2). □

Taking logarithms, inequality (2) is equivalent to

$$\frac{b}{b-u}\left[1+\frac{ml}{b}\left(\frac{\log(a^l-1)}{l}\right)-1\right]<1. \tag{3}$$

Now $ml/b \leqq 1$ and $\log(a^l-1)/l<1$, so that the quantity in square brackets is positive and less than 1. Suppose that $ml/b \geqq q$, for some real number $q \in (0, 1]$. Then

$$1+q\left(\frac{\log(a^l-1)}{l}-1\right)=t<1 \tag{4}$$

is an upper bound for the above quantity. This leads us to the following definition.

DEFINITION 10. A graph $\mathscr{G}$ is *tight* if, for each finite subgraphs L of $\mathscr{G}$, there exist a real number $q \in (0, 1]$ and an integer n_0 both depending on L such that for all vertices $v \in \mathscr{G}$ and all $n \geqq n_0$ the ball $B(v, n)$ contains at least $m_n \geqq (b_n/l)q$ copies of L, where $b_n = |B(v, n)|$ and $l = |L|$.

The next theorem generalizes the theorems of Moore and Myhill.

THEOREM 3. *Let $\mathscr{G}$ be a tight graph, L a finite subgraph of $\mathscr{G}$, and let $\{B_i\}$ be a sequence of balls as in the above definition. Assume further that* $\liminf_{i\to\infty} b_i/(b_i-u_i)=1$, *where* $u_i = |B_i \backslash B_i^-|$. *Then*

(i) *If L is the support of two mutually erasable patterns, then there exists an i such that B_i^- is the support of a* GOE *pattern*;

(ii) *If L is the support of a* GOE *pattern, then there exists an i such that B_i^+ is the support of two mutually erasable patterns.*

Proof. By Theorem 2, it is sufficient to show that (3) holds for some i. By tightness, the upper bound (4) holds for all i, and the hypothesis on lim inf ensures the existence of an i such that $b_i/(b_i-u_i)<s$, where $s=1/t>1$. □

The remainder of this section is devoted to showing that Cayley graphs are tight. The following concept was introduced by Milnor [4].

DEFINITION 11. Let G be a group generated by a finite set X. The *growth function* of G for X is the function

$$\gamma(n)=\{g\in G \mid |g| \leqq n\};$$

that is, $\gamma(n)$ is the number of elements of G that can be expressed as words of length at most n in the elements of X and their inverses.

The notion of the growth of a group G becomes that of the growth of a graph when we consider a Cayley graph $\mathscr{G}$ of G. In this case, $\gamma(n)=|B(1,n)|$. (Due to the translation invariant character of $\mathscr{G}$, the ball can be centered at any vertex of $\mathscr{G}$.) For instance, the graph of Fig. 2(a) has growth $\gamma(n)=2n^2+2n+1$; that of Fig. 2(b) has growth $\gamma(n)=(2n+1)^2$.

It is easily seen that the function $\gamma(n)$ satisfies the following inequality:

$$\gamma(n+m)\leqq\gamma(n)\gamma(m). \tag{5}$$

LEMMA 3. *Cayley graphs are tight.*

Proof. Without loss of generality, we can assume that the subgraph L of Definition 10 is a ball of radius m. Consider $B_1 = B(g_1, m)$, and let $g_2, g_3, \cdots$ be the vertices at distance $m+1$ from B_1. The ball centered at g_2 and of radius m has no vertices in common with B_1; we write $B_1 \cap B_2 = \varnothing$. Let g_3 (after renumbering) be the first vertex such that B_3 has an empty intersection with $B_1 \cup B_2, \ldots, g_i$ the first vertex such that B_i has an empty intersection with $B_1 \cup B_2 \cdots \cup B_{i-1}$. When no vertices with this property are left, we have a set of balls $B_1, B_2, \ldots, B_t$ that are pairwise disjoint. Consider now the set of vertices at a distance $m+1$ from the union of the above balls. Proceeding as before, we construct a sequence of balls that are pairwise disjoint and disjoint from the balls of the previous step; let $B_1, B_2, \cdots$ be this sequence, and let $g_1, g_2, \cdots$ be the corresponding centers. Any point $g \in \mathscr{G}$ has a distance $< m$ from one of the B_i's, so that, for some i, a point g belongs to the ball $B(g_i, 2m)$.

Now let $B(h, n)$ be a ball of radius n, and let r be the number of copies of the ball of radius m contained in it. This number r is at least equal to the number of g_i's contained in $B(h, n - m)$. Now, if $g \in B(h, n - 3m)$, then, as seen above, $g \in B(g_i, 2m)$, some i, and this g_i is in $B(h, n - m)$. Since $|B(g_i, 2m)| = \gamma(2m)$, we have

$$\gamma(n - 3m) = |B(h, n - 3m)| \leqq |\{g_i \in B(h, n - m)\}|\gamma(2m).$$

As noted above, $|\{g_i \in B(h, n - m)\}| \leqq r$, so that $r \geqq \gamma(n - 3m)/\gamma(2m)$. Now from (5) it follows that $\gamma(n) = \gamma((n - 3m) + 3m) \leqq \gamma(n - 3m)\gamma(3m)$ and that $\gamma(n - 3m) \geqq \gamma(n)/\gamma(3m)$. Therefore

$$r \geqq \frac{\gamma(n)}{\gamma(2m)\gamma(3m)} = \frac{\gamma(m)}{\gamma(2m)\gamma(3m)} \frac{\gamma(n)}{\gamma(m)}.$$

With $q = \gamma(m)/\gamma(2m)\gamma(3m)$ (see Definition 10), this proves that a Cayley graph is tight. □

Remark. In the remark at the end of § 2, two properties of a Euclidean tessellation were noted. For the Cayley graph of a group, Property 1 becomes that of being tight. In the theorem of Moore, the sequence of squares B is chosen in such a way as to have $q = 1$. Property 2 is weakened in the sense that the existence of the limit is not required. It is sufficient that lim inf = 1.

4. The main theorem. It follows from inequality (3) that $\gamma(n) \leqq \gamma(1)^n$; that is, $\gamma(n)^{1/n} \leqq \gamma(1)$. The sequence $\gamma(n)^{1/n}$ is not only bounded; the limit

$$\lim_{n \to \infty} \gamma(n)^{1/n} = a$$

always exists (see [4] again). If $a > 1$, then, for all sufficiently large n,

$$\gamma(n) \geqq a^n,$$

and we say in this case that G is of *exponential growth*. If $a = 1$, two cases are possible. Either G is of *polynomial growth*, that is, $\gamma(n)$ is bounded above by a polynomial in n for all sufficiently large n, as follows:

$$\gamma(n) \leqq p(n);$$

or G is of *intermediate growth*; that is, $\gamma(n)$ grows slower than any function of the form a^n, $a > 1$ and faster than any polynomial in n.

Remark. The fact of being of a given type of growth is a property of the group; it does not depend on the choice of a set of generators [4]. Groups of all three types exist. Typical examples are the free group of rank r, for which $\gamma(n) = (r(2r - 1)^n - 1)/(r - 1)$, which has exponential growth. The free abelian group of rank r, for which $\gamma(n) = \sum_{i=0}^{n} 2^i \binom{r}{i}\binom{n}{i}$, has polynomial growth, with $p(n) = \gamma(n)$, of degree r. Groups of intermediate growth are much more difficult to find. Their existence was proved by Grigorchuk [2], who gave examples of groups whose growth function has the bounds

$$2^{n^{1/2}} \leqq \gamma(n) \leqq 2^{n^\alpha},$$

where $\alpha = \log_{32} 31$.

From Lemma 3 and Theorem 3, we have the main result of this paper.

THEOREM 4. *Let $\mathscr{G}$ be the Cayley graph of a finitely generated group whose growth is not exponential. Then, for a cellular automaton on $\mathscr{G}$, there exist* GOE *patterns if and only if there exist mutually erasable patterns.*

Proof. Let $\gamma(n)$ be the growth function of $\mathscr{G}$. Since the quantity $\gamma(n)/\gamma(n-1)$ is greater than or equal to 1, so is its lim inf as $n \to \infty$. If this limit is greater than 1, then $\gamma(n)$ is exponential. Moreover, $\mathscr{G}$ is tight. Thus all the hypotheses of Theorem 3 are satisfied, and the result follows. □

If the growth is exponential, neither Moore's theorem nor Myhill's theorem necessarily hold. This is shown in § 6.

5. The topology of $\mathscr{C}$. As in the classical case of Euclidean tessellations (see [3], for instance), we can introduce a topology on the set $\mathscr{C}$ of all configurations on the Cayley graph of a group G. Among other things, this topology can be used to show that the existence of GOE patterns is necessary and sufficient for the existence of GOE configurations. The latter result is known in the case of Euclidean tessellations (see [8, Lemma 1]). The purpose of this section is to show that it also holds for a Cayley graph of a group G, independently of the growth of G.

Let the set A of states be given the discrete topology. Then the set

$$\mathscr{C} = A^{\mathscr{G}} = \prod_{v \in \mathscr{G}} A_v$$

is endowed with the product topology. An element of the subbasis

$$p_v^{-1}(U), \quad v \in \mathscr{G}, \quad U \subseteq A,$$

where p_v is the projection on the vth component, is the set of all configurations for which the value at the vertex v is restricted to the elements of U. If $U = \{i, j, \ldots, k\}$, then

$$p_v^{-1}(U) = p_v^{-1}(i) \cup p_v^{-1}(j) \cup \cdots \cup p_v^{-1}(k).$$

The basic open sets are thus finite unions of sets of the form

$$(6) \qquad p_{v_1}^{-1}(i_1) \cap p_{v_2}^{-1}(i_2) \cap \cdots \cap p_{v_n}^{-1}(i_n), \qquad v_k \in \mathscr{G}, \quad i_k \in A.$$

If F is the finite array whose vertices are $v_1, v_2, \ldots, v_n$, then a basic open set of type (6) consists of all configurations that agree with a pattern on F. A basic open set is then determined by considering a finite set of patterns p_i on arrays F_i and taking all the configurations whose restriction to at least one of the F_i's is p_i. This topology is induced by the metric d defined as follows. If $c = c'$, then $d(c, c') = 0$. If $c \neq c'$, let n be the smallest integer such that the restrictions of c and c' to the ball $B_n \subseteq \mathscr{G}$ with center 1 and of radius n are different, and define $d(c, c') = 1/(n+1)$. If ρ is a nonnegative real number and c a configuration, then the ball of center c and radius ρ

$$B(c, \rho) = \{c' \mid d(c, c') \leqq \rho\}$$

consists of the configurations c' that agree with c at the vertices of $\mathscr{G}$ whose distance from the vertex 1 is at most n, where n is the smallest integer such that $1/(n+1) \leqq \rho$.

Here are a few properties of this topology. Let $c \in \mathscr{C}$, and let U be an open set containing c. Then $c \in U_1 \subseteq U$, where U_1 is a set of type (6) and U_1 contains an infinite number of configurations. Thus every point of $\mathscr{C}$ is an accumulation point, and $\mathscr{C}$ is perfect. Moreover, since A is a discrete space, $\mathscr{C}$ is totally disconnected, and since A is compact, by Tychonoff's theorem $\mathscr{C}$ is compact.

Consider now the transition map τ, a ball $B(c, \rho)$, and let c' be such that $\tau(c') \in B(c, \rho)$. Consider all configurations c^* such that c^* agrees with c' at $B(1, n)^+$, where n is the smallest integer such that $1/(n+1) \leq \rho$. All these c^* are such that $\tau(c^*) \in B(c, \rho)$, and they constitute the ball $B(c', 1/(n+2))$. Thus the inverse image of the ball $B(c, \rho)$ is open, and τ is continuous.

The next lemma characterizes the accumulation points of sequences.

LEMMA 4. *A configuration c is an accumulation point for the sequence $c_1, c_2, \cdots$ if and only if every pattern that agrees with c also agrees with c_i for infinitely many i's.*

Proof. Let c be an accumulation point of the sequence $c_1, c_2, \ldots$ and let p be a pattern agreeing with c. If $v_1, v_2, \ldots, v_n$ are the vertices of the support of p, and the value of p at v_k is i_k, then the open set (6) contains c, and so infinitely many c_i's. Therefore these c_i's agree with p. For the converse, let U be an open set containing c. Then U is a union of sets of the form (6), so that c belongs to a set of this form. Let p be the corresponding pattern; then c agrees with p, and by hypothesis so do infinitely many c_i's. Therefore the latter belong to the same set of type (6) as c, and so also to U. Thus every open set containing c contains infinitely many c_i's, and the result is proved. □

The above lemma leads us to the following theorem.

THEOREM 5. *Let $\mathscr{A}$ be a cellular automaton on a Cayley graph $\mathscr{G}$. Then the existence of* GOE *configurations is necessary and sufficient for the existence of* GOE *patterns.*

Proof. Necessity is clear. For the converse, assume that there are no GOE, and let c be a configuration. Let $F_1 \subset F_2 \subset \cdots$ be a sequence of subgraphs whose union is the whole graph $\mathscr{G}$. By assumption, each restriction $c|_{F_i}$ is in the image of some c_i under τ, as follows:

$$c|_{F_i} = \tau(c_i)|_{F_i}.$$

Let F be a finite subgraph of $\mathscr{G}$. Then $F \subseteq F_k$, some k, and therefore, for $i \geqq k$,

$$\tau(c_i)|_F = c|_F.$$

Thus every pattern that agrees with c also agrees with $\tau(c_i)$ for infinitely many i's. By Lemma 4, c is an accumulation point for the sequence $\tau(c_i)$, $i = 1, 2, \cdots$. Therefore a subsequence of it converges to c as follows:

$$\tau(c_{i_1}), \tau(c_{i_2}), \cdots \to c.$$

Since τ is continuous, the sequence of the c_{i_k}'s also converges, to c', say, and the limit being unique (the space $\mathscr{C}$ is Hausdorff), we have $\tau(c') = c$. Thus no configuration is GOE, and the theorem is proved. □

Remark. The existence of GOE configurations means that τ is not surjective. The above theorem says that nonsurjectivity of τ is equivalent to the existence of GOE patterns. The noninjectivity of τ, however, is not equivalent to the existence of mutually erasable patterns; this is easily seen. See [1] for a discussion about the various connections between surjectivity and injectivity of τ and the existence of GOE configurations and of mutually erasable patterns in the case of Euclidean tessellations.

6. Cayley graphs of exponential growth. This section exhibits cellular automata on graphs of exponential growth for which the theorems of Moore and Myhill do not hold. Examples of such graphs were first given by Muller [6]. Our counterexample to Moore's theorem will use a Cayley graph of the modular group. The counterexample to Myhill's theorem will be that of Muller. The Cayley graph of the modular group Γ with presentation

$$\Gamma = \langle x, y \mid x^2 = y^3 = 1 \rangle, \tag{7}$$

(the free product of the cyclic groups of order 2 and 3) is given in Fig. 3, below.

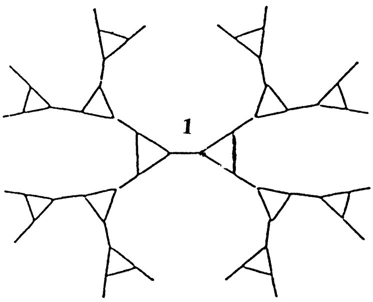


FIG. 3

Let $\delta(k)$ be the number of points whose distance from 1 is exactly k. We have

$$\delta(k)=\begin{cases}3\cdot 2^{(k-1)/2} & \text{if } k \text{ is odd,}\\ 2^{(k+2)/2} & \text{if } k \text{ is even, } k>0.\end{cases}$$

Now $\gamma(n) = 1 + \sum_{k=1}^{n} \delta(k)$, so that

$$\gamma(n)=\begin{cases}10\cdot 2^{(n-1)/2}-6 & \text{if } n \text{ is odd,}\\ 7\cdot 2^{n/2}-6 & \text{if } n \text{ is even.}\end{cases}$$

Clearly, $\gamma(n)$ is exponential.

Remark. In this example the limit of $\gamma(n)/\gamma(n-1)$ does not exist. For n even, this ratio tends to 7/5; for n odd, to 10/7. However, for other presentations of Γ (for instance, $\Gamma = \langle x, y \mid x^2 = (xy)^3 = 1\rangle$), this limit exists. Thus the existence of the limit if not a property of the group but of (the Cayley graph relative to) a given presentation of the group.

Consider now the automaton $\mathscr{A} = (A, \mathscr{G}, N, f)$, where

(i) The state set $A = \{0, 1\}$;

(ii) $\mathscr{G}$ is the Cayley graph of Γ for the presentation (7);

(iii) $N(v)$ is the set that contains v and the three vertices connected to it;

(iv) The state transition function f is defined as follows: (a) If the value at v is 1, then $f(N(v)) = 1$ if exactly two of the vertices connected to v have value 1; otherwise $f(N(v)) = 0$; (b) If the value at v is zero, then $f(N(v)) = 0$ if the three vertices around v all have value zero or all have value 1; otherwise, $f(N(v)) = 1$.

For this automaton there are two mutually erasable patterns; these are shown in Fig. 4, below.

That no GOE patterns exist can be seen as follows. Consider the array of Fig. 5. For any choice of a and b, and of b', c', and d' in $\{0, 1\}$, there exists a pattern p on the above array such that $p(A) = a$, $p(B) = b$ and $\tau(p)(B) = b'$, $\tau(p)(C) = c'$, and $\tau(p)(D) = d'$. This is shown in Fig. 6.

Now let p be a pattern on an array X. We prove by ınduction on the number of vertices of X that p is not GOE. If $|X| = 1$, there is nothing to prove. Let $|X| > 1$ and let v be an element of X at maximum distance from 1. By induction, there exists a configuration c such that $\tau(c)|_{X\setminus\{v\}} = p|_{X\setminus\{v\}}$. Now v can be considered one of the vertices B, C, D of Fig. 6. Assign an arbitrary value to a vertex B, C, or D not belonging to X, and the value it has in p if it belongs to X. Now the configuration c^* agreeing with c except possibly at C, D, E, and F, and having at these points the values given according to Fig. 6 is such that $c^*|_X = p$.

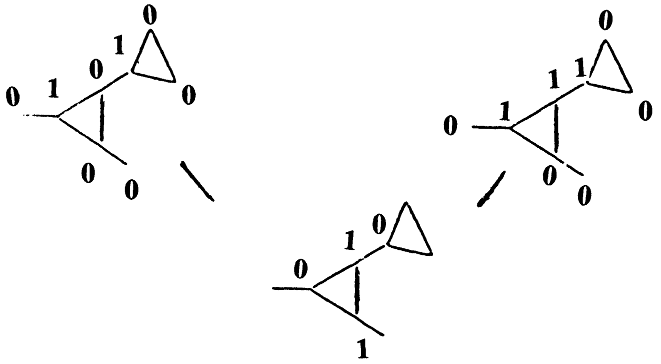


FIG. 4. *Two mutually erasable patterns for the automaton 𝒜.*

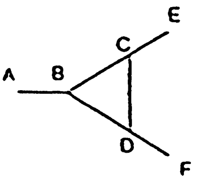


FIG. 5

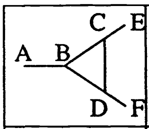
	b'=1 c'=1 d'=1	b'=1 c'=1 d'=0	b'=0 c'=1 d'=1	b'=0 c'=1 d'=0	b'=1 c'=0 d'=0	b'=0 c'=0 d'=0
a=0 b=0	1 1 0 0 1 1	1 1 0 0 1 0	0 1 0 0 0 1	0 1 0 0 0 1	0 1 0 0 0 0	0 0 0 0 0 0
a=1 b=0	0 1 1 0 0 1	0 1 1 0 0 0	1 1 1 0 1 1	1 1 1 0 1 0	0 0 1 0 0 0	1 0 1 0 1 0
a=0 b=1	1 0 0 1 1 0	1 0 0 1 1 1	0 0 0 1 0 0	0 0 0 1 1 0	1 1 0 1 1 1	1 0 0 1 0 1
a=1 b=1	1 1 1 1 0 0	1 1 1 1 0 1	0 1 1 1 0 1	1 0 1 1 1 1	1 0 1 1 0 1	1 1 1 1 1 1

FIG. 6. *A pattern at the entry (i, j) gives rise to the values at B, C, and D that are on top of column j, given the values at A and B that are at the left of row i.*

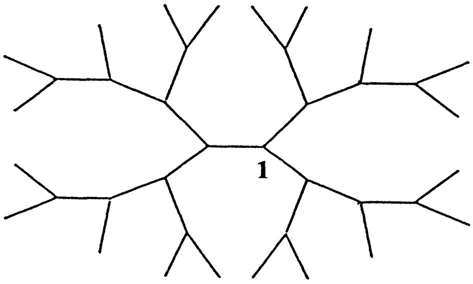


FIG. 7

We now describe Muller's counterexample to Myhill's theorem. Consider the group

$$G = \langle x, y, z \,|\, x^2 = y^2 = z^2 = 1 \rangle,$$

the free product of three copies of the cyclic group of order 2. The growth function of G is $\gamma(0) = 1$, $\gamma(n) = 3 \cdot 2^{n-1}$ if $n > 0$. Part of its Cayley graph is depicted below in Fig. 7.

In this picture, we use the convention of drawing only one edge for a generator if this element has order 2. With this convention, the graph of G is a tree. The automaton $\mathscr{A}$ has $A = \{0, 1, 2, 3\}$ as set of states. Define an operation on A by $i + i = 0$ and $i + j = k$ if i, j, and k are not zero and are all different. (This makes A a Klein group.) For a neighbourhood $N(v) = \{v, vx, vy, vz\}$ (in this order), the local map is defined as follows:

(i) $f(0, 1, 0, 0) = f(0, 2, 0, 0) = f(0, 0, 1, 0) = f(0, 0, 3, 0) = f(0, 0, 0, 2) = f(0, 0, 0, 3) = 1$;

(ii) $f(0, 3, 0, 0) = f(0, 0, 2, 0) = f(0, 0, 0, 1) = 0, f(i, 0, 0, 0) = 0$, all i's;

(iii) $f(l, i, j, k) = f(l, 0, 0, 0) + f(0, i, 0, 0) + f(0, 0, j, 0) + f(0, 0, 0, k)$.

Since the range of f is $\{0, 1\}$ any pattern for which the value at a point is 2 or 3 is GOE. We now prove that there are no mutually erasable patterns. First, the addition of A allows us to define an addition between configurations as follows:

$$(c_1 + c_2)(v) = c_1(v) + c_2(v).$$

Thus, if $(c_1 + c_2)(v) = 0$, then $c_1(v) = c_2(v)$. It follows from this definition that the parallel map is additive, shown below:

$$\tau(c_1 + c_2) = \tau(c_1) + \tau(c_2).$$

Assume that p_1 and p_2 are two mutually erasable patterns on an array X and let c_1 and c_2 be two configurations such that c_i agrees with p_i, $i = 1, 2$, on X, they agree outside X, and $\tau(c_1) = \tau(c_2)$. Thus $(c_1 + c_2)(v) = 0$ for $v \notin X$, and $\tau(c_1 + c_2) = \tau(c_1) + \tau(c_2) = 0$. Now if $c_1 + c_2$ is zero at all points of X, then $c_1 = c_2$, contrary to assumption. Thus there exists a point v at maximum distance d from 1 such that $(c_1 + c_2)(v) \neq 0$. Now, of the three vertices vx, vy, and vz, two are at a distance $d + 1$ from 1, and therefore $c_1 + c_2$ is zero at these points; let h and k be these vertices. Both have a neighbour, namely, v, at which $c_1 + c_2$ is not zero, the two other neighbours being zero. A case-by-case check using the definition of f shows that either $\tau(c_1 + c_2)(h)$ or $\tau(c_1 + c_2)(k)$ is not zero, a contradiction.

Acknowledgment. The authors are grateful to the referee for his or her helpful suggestions.

REFERENCES

[1] S. Amoroso, G. Cooper, and Y. Patt, *Some clarifications of the concept of a Garden-of-Eden configuration*, J. Comput. System Sci., 10 (1975), pp. 77–82.

[2] R. I. Grigorchuk, *Degrees of growth of finitely generated groups and the theory of invariant means*, Math USSR Izvestiya, 25 (1985), pp. 259–300.

[3] G. A. Hedlund, *Endomorphisms and automorphisms of the shift dynamical system*, Math. System Theory, 3 (1969), pp. 320–375.

[4] J. Milnor, *A note on curvature and fundamental groups*, J. Differential Geometry, 2 (1968), pp. 1–7.

[5] E. F. Moore, *Machine models of self-reproduction*, in Essays on Cellular Automata, Arthur B. Burks, ed., University of Illinois Press, Urbana, Chicago, London, 1970.

[6] D. E. Muller, class notes, University of Illinois, Urbana, IL, 1976.

[7] J. Myhill, *The converse of Moore's Garden of Eden Theorem*, Proc. Amer. Math. Soc., 14 (1963), pp. 685–686.

[8] D. Richardson, *Tessellations with local transformations*, J. Comput. System Sci., 6 (1972), pp. 373–388.

[9] P. E. Schupp, *Arrays, automata and groups—Some interconnections*, in Proceedings of the LITP Spring School on Theoretical Computer Science, May 1986, Argelès-Village, C. Choffrut, ed., Lecture Notes in Computer Science, 316, Springer-Verlag, Berlin, New York, 1988.

[10] J. von Neumann, *The Theory of Self-Reproducing Automata*, A. Burks, ed., University of Illinois Press, Urbana, IL, 1966.

SIAM J. DISC. MATH.
Vol. 6, No. 1, pp. 57–69, February 1993

005

THE WEIGHTED SPARSITY PROBLEM: COMPLEXITY AND ALGORITHMS*

S. THOMAS MCCORMICK† AND S. FRANK CHANG‡

Abstract. Many optimization algorithms involve repeated processing of a fixed set of linear constraints. If the constraint matrix A is preprocessed to make it sparser, algebraic operations should become faster. In many applications there is a priori information about the likelihood that each column will appear in a basis, which can be expressed as weights on the columns. This leads to considering the weighted sparsity problem (WSP): Find a row-equivalent constraint matrix with as small a weight of nonzeros as possible. The WSP is shown to be NP-hard even with a nondegeneracy assumption, and even if restricted to instances with at most three nonzeros per both row and column. WSP is shown to have a polynomial algorithm when the number of nonzeros per either row or column is limited to at most two. This contrasts with previous results that, assuming only nondegeneracy, the unweighted version of WSP does have a polynomial algorithm (this has proven to be practically useful in tests on real data). The polynomial algorithm for WSP with at most two nonzeros per row or column is based on solving one-row problems via minimum cut calculations, together with a sufficient condition for piecing these one-row solutions together into a global solution.

Key words. computational complexity, sparse matrices, bipartite matching

AMS(MOS) subject classifications. 65F50, 05C70, 68Q25

Introduction. Many optimization algorithms involve repeated processing of a fixed set of linear constraints. When the constraint matrix is *sparse* (has a very small proportion of nonzero entries), as is often the case in practice, algebraic operations become much faster, and consequently very large problems can be solved. Because the speed of such algorithms depends strongly on the degree of sparsity in the input, it is natural to ask whether a constraint matrix can be preprocessed in order to make it sparser.

A typical application for such a procedure is to the Simplex Algorithm for linear programming (projective algorithms require a somewhat different view of sparsity, which we plan to address in the future; see also Adler et al. [1]). In Simplex, the processing consists of maintaining a factorization of a changing basis matrix B chosen from the columns of the constraint matrix A. If A is sparser, then on average the various B's will be sparser.

In many applications, however, the modeler will have some a priori idea of which columns of A are more likely to appear in a basis than others. For example, some activities may seem economically more attractive than other activities. Alternatively, imagine that we keep a log of actual residence time in the basis for each column in a long optimization. We could then use the residence times as empirical predictions of the likelihood of each column appearing in a basis. (In a similar vein, Freund [9] considers solving central trajectory problems with weights on constraints, which measure their likelihood of being active.) This leads to consideration of the following problem.

Weighted sparsity problem (WSP). Given $A \in \mathscr{R}^{m \times n}$, $b \in \mathscr{R}^{m}$, and $w \in \mathscr{R}^{n}$, which define constraints $Ax = b$ with weight w_j on column j, find a nonsingular $T \in \mathscr{R}^{m \times m}$

* Received by the editors April 18, 1990; accepted for publication (in revised form) March 9, 1992. Both authors were partially supported by National Science Foundation grants ECS-84-04350 and CDR-84-21402 and by Office of Naval Research contract N0014-87-K0214. Most of this work was done while both authors were in the IE/OR Department at Columbia University.

† Faculty of Commerce and Business Administration, University of British Columbia, Vancouver, BC V6T 1Z2 Canada. This author's research was partially supported by a Natural Sciences and Engineering Research Council of Canada operating grant.

‡ GTE Laboratories, Inc., Waltham, Massachusetts 02254.

such that $\hat{A} \equiv TA$ minimizes

$$\sum_{j=1}^{n} w_j \cdot (\text{number of nonzeros in column } j \text{ of } \hat{A}).$$

Note that since T is nonsingular, defining $\hat{b} = Tb$ yields that $\{x \in \mathscr{R}^n \mid Ax = b\} = \{x \in \mathscr{R}^n \mid \hat{A}x = \hat{b}\}$. Thus such an $\hat{A}x = \hat{b}$ would be an optimally sparse equivalent set of constraints with respect to weights w. We assume that all w_j are strictly positive (columns with zero weight can be ignored, and negative weights do not make sense in this application). We also assume for convenience that A has full row rank. Theorem 3.4.1 from McCormick [18] implies that the results herein still hold without the full rank assumption.

Of course, in many applications there is no information about the columns, so the best that we can do is to make all column weights equal and consider the (unweighted) problem.

Sparsity problem (SP). Solve WSP with all $w_j = 1$.

For WSP, note that if we can make every row i of $\hat{A}$ individually as sparse as possible among all linear combinations of rows that include row i, then certainly $\hat{A}$ as a whole will be as sparse as possible. This suggests that we initially restrict our attention to the following problem.

One-row weighted sparsity problem (ORWSP). For a fixed row i, find multipliers t_{ik}, $k \neq i$, such that

$$\hat{A}_{i\cdot} \equiv A_{i\cdot} + \sum_{k \neq i} t_{ik} A_{k\cdot}$$

has a minimum weight of nonzeros.

(We have normalized the multipliers so that the coefficient on row i is one.) When all weights are one, we analogously have the *one-row sparsity problem* (ORSP).

The main results of this paper concern the complexity of the weighted problems. We review previous results on the unweighted versions of the problem in § 1 where we present a necessary nondegeneracy assumption, called the *matching property* (MP) (MP is necessary since all four problems are NP-hard without some "nondegeneracy" assumption on the numerical values of the nonzero entries of A). We also recall some of the techniques that were used to get a polynomial algorithm assuming MP in the unweighted case because they will be used again for the weighted case. These techniques are used to cast the objective of WSP into a more combinatorial form in § 2. Section 3 then uses this form to prove that WSP is NP-hard even assuming both MP and that every row and every column of A has at most three nonzeros. Section 4 then answers the natural question of whether a polynomial algorithm exists for WSP if we restrict either rows or columns to have at most two nonzeros each by constructing polynomial algorithms in those cases. The complexity results in this work and quoted from previous work (Hoffman and McCormick [15]) are summarized in the table below (it turns out that ORWSP and WSP always have the same complexity, as do ORSP and SP). Table 1 lists the theorems establishing the most restrictive NP-hardness results and most general polynomial algorithm results. Finally, § 5 has some concluding thoughts on heuristic algorithms for WSP.

We note that variants of the polynomial algorithm for SP have been implemented in McCormick [19] and Chang and McCormick [3], [4] and produced very good results on the real-life linear programming problems in NETLIB (see Gay [12] or Lustig [17]).

TABLE 1

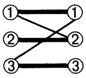
	Assumptions			
	None	MP	MP, at most 3 nz/row and col	MP, at most 2 nz/row or col
ORSP, SP ORWSP, WSP	NP-hard, Thm. 1.2 NP-hard	$\mathscr{P}$, Thm. 1.6 NP-hard	$\mathscr{P}$ NP-hard, Thm. 3.1	$\mathscr{P}$ $\mathscr{P}$, Thms. 4.1, 4.2

1. Review of previous results.

The methods that we use to analyze WSP involve bipartite matching theory and network flows (see, e.g., Lawler [16] or Ford and Fulkerson [7]). There is a simple correspondence between bipartite graphs and sparsity patterns of rectangular matrices. Given the sparse matrix A, define the bipartite graph $\mathscr{B}$ by setting the left nodes of $\mathscr{B} = \{\text{rows of } A\}$, the right nodes of $\mathscr{B} = \{\text{columns of } A\}$, and the edges of $\mathscr{B} = \{i - j \mid a_{ij} \neq 0\}$. This correspondence allows us to refer to sparsity patterns and bipartite graphs interchangeably. We display sparsity patterns as matrices, but use the language of bipartite graphs to describe them.

A subset P of the nonzeros of A, such that no two elements of P lie in the same row or column, is classically known as a *partial transversal* (see Welsh [25, § 7.1]). A partial transversal corresponds to a (not necessarily maximum) matching (see, e.g., Lawler [16, Chap. 5]) in a bipartite graph (i.e., a subset of edges with no common vertices). In the example below, the circled transversal corresponds to the heavy matching in the bipartite graph $\mathscr{B}$:

$$A=\begin{pmatrix} \otimes & \times & 0 \\ 0 & \otimes & 0 \\ \times & 0 & \otimes \end{pmatrix}, \quad \mathscr{B} =$$



(When we write a sparsity pattern, zero is represented by "0" or a blank and a nonzero by "×".) We favor the term *matching* even though it is historically inappropriate for matrices.

A matching in A is called *row-perfect* if all rows of A are matched; *column-perfect* is defined similarly. A matching is *perfect* if it is both row- and column-perfect. A *maximum matching* is one with a maximum number of nonzeros. If $R \subseteq \{1, 2, \ldots, m\}$ and $C \subseteq \{1, 2, \ldots, n\}$ then A_{RC} denotes the submatrix of A indexed by rows in R and columns in C and $M(A_{RC})$ denotes the size of a maximum matching in A_{RC}; $M(A_{RC})$ is sometimes called the *term rank* of A_{RC} (see Ryser [23, Chap. 5]).

We will solve one case of WSP with minimum cuts in maximum flow networks. Recall that, in a maximum flow network $G = (N, A)$ with source s and sink t, a cut is defined with respect to a node partition $N = S \cup T$ with $S \cap T = \emptyset$, $s \in S$, and $t \in T$. We call S the *s-side* of the cut. If x^* is a maximum flow in G, the s-side of the *standard min cut* is defined by $S^* = \{i \in N \mid \text{there is an augmenting path from } s \text{ to } i \text{ with respect to } x^*\}$ (this is the usual way to compute a min cut from x^*). We shall assume that all of our minimum cuts are standard minimum cuts, and we record the following fact, which we will need in Lemma 4.5.

PROPOSITION 1.1 (see [7, p. 13]). *S^* defines the unique minimum cardinality minimum cut.*

It turns out that even SP is too hard to solve in general because it is too hard to predict where the numerical values in A might cancel each other out during row arithmetic (we call this *unexpected cancellation*). This is formalized below.

THEOREM 1.2. SP *is* NP-*hard to solve in general.*

Proof. This was proved by Stockmeyer [24]; his proof is quoted as Theorem 1 in Hoffman and McCormick [15]. □

COROLLARY 1.3. WSP *is also* NP-*hard to solve in general.*

Because predicting unexpected cancellation makes analyzing sparse matrix problems very difficult, sparse matrix workers have traditionally assumed that unexpected cancellation will not happen, a "nondegeneracy" assumption (see Coleman [6]). A typical justification is that numerical entries are subject to measurement errors, which are equivalent to independent infinitesimal perturbations. This is similar to perturbation schemes for resolving degeneracy in linear programming (see Charnes [5]) and does rule out unexpected cancellation.

Unfortunately, this justification is invalid in practice because real matrices have many entries that are small integers (which are *not* subject to measurement error) and can lead to lots of unexpected cancellation. (See Murota and Iri [21] for an approach to matrices that treats "small integers" differently from "real numbers.") However, algorithms that are developed under such an assumption work very well in practice despite the failure of the assumption to hold. Thus, making such an assumption can be seen as a heuristic device for deriving good algorithms.

Corollary 1.3 forces us to make some sort of assumption to try to get a polynomial algorithm. We will use the same assumption that was developed in Hoffman and McCormick [15], which has indeed worked well in practice. (See Chang and McCormick [3], [4], or McCormick [19]; see also Adler et al. [2] for a heuristic approach to SP that does *not* assume nondegeneracy and in fact tries to take advantage of unexpected cancellation. Chang and McCormick [4] show that the present approach does not work as well on practical problems as the nonassumption approach of Adler et al. [2].) Our assumption is motivated by the expectation that if a submatrix can be permuted so that it has a nonzero diagonal, then it should have full rank. More formally, we assume that A has the following property.

Matching Property (MP). A has MP if rank $A_{RC} = M(A_{RC})$ for all row subsets R and column subsets C.

In classical terminology, MP states that term rank and numerical rank are the same for every submatrix of A. We shall assume that MP holds in the rest of this paper.

The Matching Property now allows us to make some strong statements about the structure of a solution to ORWSP. For a solution $t_{i\cdot}$ (with $t_{ii} = 1$) of ORWSP define $U = \{k \neq i \mid t_{ik} \neq 0\}$ the set of rows of A *used* by $t_{i\cdot}$, and $G = \{j \mid \hat{a}_{ij} = 0$ and $a_{kj} \neq 0$, some $k \in U \cup \{i\}\}$, the set of columns of A affected in a *good* way by $t_{i\cdot}$. That is, if $a_{ij} \neq 0$ but $\hat{a}_{ij} = 0$, then we say that a_{ij} was *hit*, and then $j \in G$. If $a_{ij} = 0$ and $a_{kj} \neq 0$ for some $k \in U$, then we would expect that $\hat{a}_{ij} \neq 0$; if instead $\hat{a}_{ij} = 0$, then a_{ij} is *avoided fill-in*, and again $j \in G$.

Intuitively, lack of unexpected cancellation means that if we use $|U|$ rows, then we cannot affect more than $|U|$ columns in a good way; i.e., $|G| \leqq |U|$. It does not pay to choose $|U| > |G|$, so $|U| = |G|$. The following theorem formalizes this argument.

THEOREM 1.4 (McCormick [18, Cor. 3.2.3]. *When* MP *holds, there is an optimal solution to* ORWSP *where* A_{UG} *is a square, nonsingular matrix.*

Note that $t_{i\cdot}$ determines U and G, and conversely, given a U, G with A_{UG} square and nonsingular, we could compute $t_{i\cdot}$ via

$$\begin{aligned} t_{iU} &= -A_{iG}A_{UG}^{-1}, \\ t_{ik} &= 0,\ k \notin U \cup \{i\}, \qquad (1.1) \\ t_{ii} &= 1. \end{aligned}$$

Thus we can equivalently search for an optimal U and G in solving ORWSP.

This structure is already almost enough to show that a polynomial algorithm for solving ORSP exists, and indeed Hoffman and McCormick [15] give such an algorithm based on computing minimum cuts.

However, if we solve the m ORWSPs, we still face the question of whether we can put all these local solutions together into a global solution to WSP. Minimizing per-row sparsity minimizes global sparsity, but it is not clear that pasting together the one-row solutions from (1.1) would result in a nonsingular T.

In fact, ORWSP solutions often yield a singular matrix T when pasted together. Consider the example with $w_1 = 10$, $w_2 = 1$, and

$$A = \begin{pmatrix} \times & \times \\ \times & \times \end{pmatrix}.$$

The optimal one-row solutions are $U_1 = \{2\}$, $U_2 = \{1\}$, and $G_1 = G_2 = \{1\}$, leading to

$$A = \begin{pmatrix} 0 & \times \\ 0 & \times \end{pmatrix}. \tag{1.2}$$

Because T causes $\hat{A}$ to lose rank, T is singular. It is too tempting for the individual ORWSP solutions to zero out high-weight columns like the first one in (1.2), although we globally need to retain at least one of the high-weight nonzeros to maintain rank.

This quandary was resolved when solving SP by finding a sufficient condition on the one-row solutions, which guarantees that they *can* be pasted together into a global solution, which we shall need in § 4. Suppose that U_i, G_i, $i = 1, 2, \dots, m$ are a set of feasible solutions to ORWSP for each row i and define $\bar{U}_i = U_i \cup \{i\}$. We say that $\{U_i, G_i\}$ are *transitive* if:

1. There is a fixed, row-perfect matching $\mathcal{M}$ in A such that G_i equals the columns matched to rows U_i under $\mathcal{M}$ for all i, and
2. If $j \in \bar{U}_i$, then $\bar{U}_j \subseteq \bar{U}_i$.

THEOREM 1.5 (Hoffman and McCormick [15, Thm. 7]). *Assuming* MP, *if* $\{U_i, G_i\}$ *are transitive, then the* T *defined by* (1.1) *is nonsingular.*

For SP, the ORSP solutions from Theorem 1.4 as sewn together by Theorem 1.5 yield the following theorem.

THEOREM 1.6 (Hoffman and McCormick [15, Thm. 8]). *There is a polynomial-time algorithm that solves* SP *under* MP.

2. Combinatorializing the one-row objective. To find a more combinatorial expression for the objective of ORWSP for row i under MP, consider choosing an optimal U, G pair as a two-stage process. First we choose the set of rows U that we shall use, then we choose the best G with respect to U. In choosing U, we are allowing *potential fill-in* in the columns $P(U) \equiv \{j \mid a_{ij} = 0 \text{ and } a_{kj} \neq 0, \text{ some } k \in U\}$. That is, given U, the nonzeros in $\hat{a}_{i\cdot}$ can potentially appear in either $Y \equiv \{j \mid a_{ij} \neq 0\}$ or in $P(U)$. We want to choose G as the subset of the $|U|$ heaviest columns from $Y \cup P(U)$, subject to G having to perfectly match to U (so that A_{UG} will be nonsingular, by MP). However, any G perfectly matching to U must be a subset of $Y \cup P(U)$, so it must be optimal to choose G as a set of columns matching to U that maximizes $\sum_{j \in G} w_j$ (since weights are only on columns and are positive, $|G|$ will always equal $|U|$).Thus, if we define $M(U)$ as such a set of maximum-weight, perfectly matchable columns, ORWSP is equivalent to

$$\min_{U} w(P(U)) - w(M(U)). \tag{2.1}$$

Note that for a given U, $P(U)$ and $M(U)$ are easy to compute. In fact, since weights are only on columns, $M(U)$ can be computed by the *Greedy Algorithm* (because subsets of columns hit by a matching form a *transversal matroid*, see Welsh [25]).

Because $P(U \cup V) \subseteq P(U) \cup P(V)$ we have that $w(P(U))$ is submodular. Less trivially, it can be shown that $w(M(U))$ is submodular (see Chang [2], Thm. 2.3.1). Unfortunately the difference between two submodular functions is in general neither sub- nor supermodular, and it is easy to construct examples of each in this case, even if we restrict ourselves to at most two nonzeros per row and column:

$$\begin{array}{lllcccc}
w_j & & & 1 & 100 & 100 & 1 \\
\text{row } i & & & \times & 0 & 0 & \times \\
S\{ & & & \times & \times & 0 & 0 \\
 & & T\{ & 0 & \times & \times & 0 \\
 & & & 0 & 0 & \times & 0
\end{array}
\qquad
\begin{array}{lccc}
w_j & 100 & 100 & 1 \\
\text{row } i & \times & 0 & \times \\
S\{ & \times & \times & 0 \\
T\{ & 0 & \times & 0
\end{array}$$

The example on the left shows that $w(P(U)) - w(M(U))$ is not submodular, while the example on the right shows that it is not supermodular.

When minimizing a set function $f : 2^E \rightarrow \mathscr{R}$, if f is submodular a polynomial algorithm must exist (see Grötschel, Lovász, and Schrijver [14]). If f is not submodular, the problem is often NP-hard. Indeed, we shall prove that WSP is NP-hard in the next section.

However, note that for (the unweighted) SP, $w(\mathscr{M}(U)) = |U|$, a modular function, and the one-row objective is now

$$\min_U |P(U)| - |U|.$$

It is still true that $|P(U)|$ is submodular, but since $|U|$ is modular, the difference is submodular. This explains why there is a polynomial algorithm for SP but not WSP.

3. WSP is NP-hard, ORWSP is NP-complete. In this section we prove that WSP is NP-hard, even assuming that MP holds and that ORWSP is NP-complete. We do not know how to prove that WSP is in NP because it is difficult to bound the size of the entries in the transformation matrix T.

THEOREM 3.1. WSP *is strongly* NP-*hard even with* MP, *and even if we restrict to instances with at most three nonzeros per row and per column.*

The restriction to at most three nonzeros per row and per column is not surprising. There is a simple folklore method (see Megiddo [20]) to transform any $Ax = b$ into an equivalent (but larger) $A'x' = b'$ in polynomial time, where A' has at most three nonzeros per row and per column.

We reduce the following known NP-complete problem to WSP.

Cubic Node Cover (CNC).

Instance: Undirected graph $\mathscr{G} = (N, E)$ with each node having degree exactly three (a *cubic* graph), and an integer $k \leqq |N|$.

Question: Is there an $H \subseteq N$ such that every edge $i\text{-}j \in E$ has $i \in H$ or $j \in H$ (H is a *node cover*) with $|H| < k$?

A proof of CNC's NP-completeness is in Garey and Johnson [11].

Proof of Theorem 3.1. Given instance $\mathscr{G}$, k of CNC, set $n = |N|$, $m = |E|$, and construct an instance of WSP as follows: A has $n + 4m + 1$ columns divided into n *node* columns indexed by nodes, each with weight 1; m *edge* columns indexed by E, each with weight $2m(n + 1) + 1$; $2m$ *incidence* columns indexed by pairs $(i, j\text{-}k)$ where $i = j$ or $i = k$, each with weight $n + 1$; m *setup* columns indexed by E, each with weight $m(2m(n + 1) + 1) + 1$; and one *enforcer* column with weight $m(m(2m(n + 1) + 1) + 1) + 1 \equiv M$. A has $3m + 1$ rows divided into row 1, $2m$ *incidence* rows indexed like incidence columns, and m *setup* rows indexed by E.

TABLE 2

	Node k	Edge k-l	Incidence k, k-l	Setup k-l	Enforcer
Row 1	0 always	0 always	0 always	× in first entry only	×
Incidence i, i-j	× iff $k = i$	× iff i-j = k-l	× iff $i = k$, i-j = k-l	0 always	0 always
Setup i-j	0 always	× iff i-j = k-l	0 always	× if on diagonal or first super diagonal	0 always

The nonzeros in A are as in Table 2.

Thus the incidence-node submatrix is the arc-node incidence matrix of $\mathcal{G}$ with every row split into two parts, the incidence-edge submatrix is a row-doubled identity, incidence-incidence is an identity, setup-edge is an identity, setup-setup is a diagonal with one superdiagonal matrix, and all other entries are zero except for the row 1-enforcer and first row 1-setup entries. Note that every row and column does have at most three nonzeros, because $\mathcal{G}$ is cubic. Also, all weights are polynomial in n and m, so this is a strongly polynomial reduction.

Now we claim that WSP has a solution in which $\hat{A}_{1\cdot}$ has weight at most $M + m(n + 1) + K$ if and only if CNC has a node cover H of size at most K. Consider any optimal solution to WSP on A. Because M is so large, row 1 will not be used by any other row. Thus we can process row 1 without worrying about the nonsingularity of T (solving WSP on A must solve ORWSP on row 1). Note that it always pays to use all setup rows for row 1 to hit the first setup nonzero in row 1 and to keep any other setup columns in row 1 from filling in. However, then all edge columns fill in, and it will pay to use m of the $2m$ incidence rows, exactly one from each pair (i, i-j), (j, i-j), to hit all the fill-in in the edge columns. Define $H = \{i \in N | i$ indexes a used incidence row$\}$; note that H is a node cover and that the weight of fill-in in the node columns is $|H|$. It will never pay to hit any of the fill-in in node columns, because doing so would cause much greater weighted fill-in in the incidence columns, fill-in that cannot be subsequently removed.

Because m of the incidence columns get filled in while hitting the edge column fill-in, the final weight of $\hat{A}_{i\cdot}$ in any optimal WSP solution is $M + m(n + 1) + |H|$ for node cover H. Given a node cover H, the construction can be reversed to obtain such a solution to ORWSP for row 1. Thus no polynomial algorithm for WSP with at most three nonzeros per row and per column can exist unless there is also a polynomial algorithm for the NP-complete problem CNC. □

COROLLARY 3.2. ORWSP *is strongly* NP-*complete even with* MP *and at most three nonzeros per row and column.*

Proof. For the A constructed in the proof of Theorem 3.1, WSP reduces to ORWSP for row 1, so ORWSP is NP-hard. Theorem 3.1 and (2.1) show that U is a (polynomial-length) *certificate* for ORWSP (see Garey and Johnson [10] for the definition of certificate), so ORWSP is NP-complete. □

Note that the proofs of Theorem 3.1 and Corollary 3.2 also happen to include the restriction that there are at most five distinct weights.

4. WSP is polynomial for at most two nonzeros per row or per column. Theorem 3.1 naturally raises the question of whether WSP with MP is still NP-hard if we further restrict to at most two nonzeros per row *or* column. (Other linear programming problems become easier with this restriction; see, e.g., Megiddo [20].) Despite the lack of submodularity in this case exhibited in the examples in § 2, we will prove that WSP with MP will now have a polynomial algorithm.

Matrices with at most two nonzeros per column are called *generalized network* matrices (see Orlin [22]), and they occur fairly often in practice. Thus it may seem that the column result would have some practical applications. However, the following example shows that minimizing weighted sparsity can destroy the generalized network structure of the matrix (the weights are given above the columns):

$$\begin{array}{c} \begin{array}{ccc} 1 & 0 & 1 \end{array} \\ \begin{pmatrix} \times & 0 & \times \\ \times & \times & 0 \\ 0 & \times & \times \end{pmatrix} \end{array} \to \begin{pmatrix} 0 & \times & 0 \\ \times & \times & 0 \\ 0 & \times & \times \end{pmatrix}.$$

Good, special-purpose generalized network algorithms exist (see Goldberg, Plotkin, and Tardos [13]), so it is probably better in practice to leave these matrices alone. Matrices with at most two nonzeros per row have such a special structure (as we will see) that they can be dealt with better by ad hoc methods.

THEOREM 4.1. *Assuming* MP, WSP $\in \mathscr{P}$ *when restricted to at most two nonzeros per row.*

Proof. We consider A as the sparsity pattern of an edge-node incidence matrix of a graph $\mathscr{G}$, with each singleton row representing a self-loop. The row-perfect matching of A assures us that there exists no more than one self-loop for each node in $\mathscr{G}$. We can assume without loss of generality that $\mathscr{G}$ is connected, with m edges and n nodes.

Because A has a row-perfect matching, $m \leqq n$, and because $\mathscr{G}$ is connected, we have $m = n$ or $m = n - 1$.

Case 1 ($m = n$). Then $\mathscr{G}$ is a 1-*tree* (i.e., a tree plus either an edge or a self-loop), and A is square and nonsingular (by MP). Thus A can be reduced to an identity via its inverse; and that is optimal.

Case 2 ($m = n - 1$). Now the graph $\mathscr{G}$ is a tree. Let B be the $m \times m$ submatrix consisting of the $n - 1$ heaviest columns of A. B is nonsingular since $\mathscr{G}$ is a tree, and by MP. We multiply A by B^{-1} to get the matrix in Fig. 1, which has the lightest total weight among all matrices equivalent to A. See Fig. 1.

The major work involved in Case 1 or Case 2 is simply Gaussian elimination. Therefore, the total time in processing the whole matrix A is bounded by $O(m^3)$. □

THEOREM 4.2. *Assuming* MP, WSP $\in \mathscr{P}$ *when restricted to at most two nonzeros per column.*

To prove the theorem we need to prove four lemmas. We consider the sparsity pattern of A as the node-edge incidence matrix of a graph $\mathscr{G}$, with each singleton column

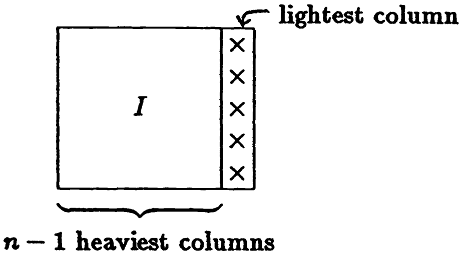


FIG. 1

representing a *partial* edge whose other end is dummy node 0. We can again assume without loss of generality that the nonpartial edges of $\mathcal{G}$ form a connected graph. For row set U and column set G, let $\mathcal{G}(U, G)$ denote the subgraph induced by the nodes in U and edges in G.

LEMMA 4.3. *There is an optimal solution* (U, G) *to* ORWSP *for row* 1 *with* $\mathcal{G}(\bar{U}, G)$ *a tree.*

Proof. Let $\bar{U}$ be the smallest cardinality optimal solution. Suppose that $j \in G$ has only one nonzero in $A_{\bar{U}_j}$, say in row i. If $i = 1$, then $A_{U_j} = 0$, contradicting the nonsingularity of A_{UG}. If $i \neq 1$, then $(U\backslash\{i\}, G\backslash\{j\})$ is a better solution than (U, G). Thus $\mathcal{G}(\bar{U}, G)$ induces no partial edges. Note that $|\bar{U}| = |G| + 1$, so all that is left to show is that $\mathcal{G}(\bar{U}, G)$ is connected.

Suppose $\mathcal{G}(\bar{U}, G)$ is not connected. Then there exists at least one connected component not able to reach node 1, say on node set V and edge set X. Therefore X must be a subset of the zero columns in row 1 (else X *could* reach row 1). Moreover, $|X|$ must be equal to $|V|$, for if not, then A_{VX} and $A_{U\backslash V, G\backslash X}$ cannot both have complete matchings, implying that A_{UG} cannot have a complete matching, a contradiction. Figure 2 below shows our situation. Now $(U\backslash V, G\backslash X)$ has ORWSP objective value at least as good as (U, G) (it might cause less fill in than (U, G)) and has smaller size. □

Using this lemma, we can give a graphical interpretation of the objective function (2.1) for the class of matrices we are dealing with. The optimal solution to ORWSP for row 1 is a tree $\mathcal{T}$ rooted at node 1. We have that $\bar{U}$ is the "nodes" or "set of nodes" spanned by $\mathcal{T}$, and $M(U)$ is a maximum spanning tree of $\bar{U}$. By the optimality of $\mathcal{T}$, the arcs of $\mathcal{T}$ form $M(U)$. Finally, $Y \cup P(U)$ is the set of arcs incident to $\bar{U}$. Thus $\mathcal{T}$ minimizes

$$\text{obj}\,(\mathcal{T}) \equiv \sum_{\substack{i-j \text{ hitting nodes in } \mathcal{T}\\ \text{(including edges in } \mathcal{T})}} w_{ij} - \sum_{i-j \in \mathcal{T}} w_{ij}.$$

Because the optimal solution to ORWSP for row 1 uses a maximum weight spanning tree $\mathcal{T}$ for a subset of nodes (rows) containing 1, it is natural to ask about the relationship between $\mathcal{T}$ and a fixed maximum spanning tree $\mathcal{T}^*$ for the entire graph $\mathcal{G}$.

LEMMA 4.4. *There is an optimal tree* $\mathcal{T}$ *for* ORWSP *that is a subtree of* $\mathcal{T}^*$.

Proof. Let $\mathcal{T}$ be an optimal ORWSP tree with a minimum number of non-$\mathcal{T}^*$ edges. If there is an edge $i\text{-}j \in \mathcal{T}\backslash\mathcal{T}^*$, let $\mathcal{S}'$ and $\mathcal{T}'$ be the two subtrees of $\mathcal{T}$ we get when $i\text{-}j$ is removed, where nodes $1, i \in \mathcal{T}'$, and $j \in \mathcal{S}'$. Let $(\mathcal{S}', \mathcal{T}')$ denote the cut

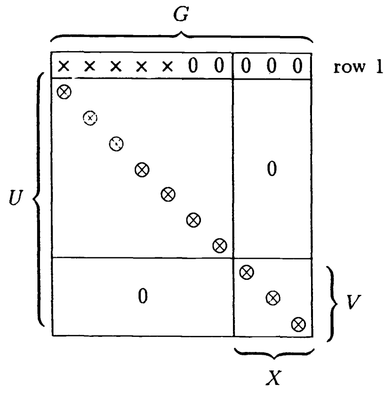


FIG. 2

$\{u-v \in \mathcal{G} \mid u \in \mathcal{S}', v \in \mathcal{T}'\}$. Now $i-j \notin \mathcal{T}^*$ implies that there is a unique cycle $Q \subseteq \mathcal{T}^* \cup i-j$. Because Q has already crossed the cut $(\mathcal{S}', \mathcal{T}')$ once (at $i-j$), it must cross again in some edge $k-l$ with $k \in \mathcal{S}'$, $l \notin \mathcal{S}'$. By optimality of $\mathcal{T}^*$ we must have that

$$w_{kl} \geqq w_{ij}. \tag{4.1}$$

Case 1. ($l \in \mathcal{T}'$) See Fig. 3. Note that $\bar{\mathcal{T}} \equiv \mathcal{T} \cup k-l \setminus i-j$ is again a tree spanning all the nodes in $\mathcal{T}$. Now (4.1) implies that obj $(\bar{\mathcal{T}}) \leqq$ obj $(\mathcal{T})$; but $\bar{\mathcal{T}}$ has fewer non-$\mathcal{T}^*$ edges than $\mathcal{T}$, which is a contradiction.

Case 2. ($l \notin \mathcal{T}'$) See Fig. 4. Now consider tree $\mathcal{T}'$ versus tree $\mathcal{T}$; obj $(\mathcal{T})$ includes w_{kl} but not w_{ij}, whereas obj $(\mathcal{T}')$ includes w_{ij} but not w_{kl}. Any other edge included in obj $(\mathcal{T}')$ is also included in obj $(\mathcal{T})$, so by (4.1) obj $(\mathcal{T}) \geqq$ obj $(\mathcal{T}')$. But $\mathcal{T}'$ has fewer non-$\mathcal{T}^*$ edges than $\mathcal{T}$, which again contradicts the assumption on $\mathcal{T}$. □

Call a node subset U not containing node 1 *connected* to 1 if $U \cup \{1\}$ induces a (connected) subtree of $\mathcal{T}^*$, and define $\mathcal{C}_1$ to be the family of node sets connected to 1. Now $\mathcal{C}_1$ is closed under union and intersection and so is a *ring family* (see Frank and Tardos [8]). For node $j \neq 1$ set $w(j)$ equal to the weight of the first edge on the $\mathcal{T}^*$ path from j to 1 (the *predecessor arc* of j). Then for $U \in \mathcal{C}_1$ the edge subset $M(U)$ forms a subtree of $\mathcal{T}^*$ and $w(M(U)) = \sum_{j \in U} w(j) \equiv w(U)$, a modular function. Thus ORWSP in this case becomes

$$\min_{U \in \mathcal{C}_1} w(P(U)) - w(U), \tag{4.2}$$

a submodular function defined on a ring family. Thus, although (as we saw in § 2) the objective is not submodular on all sets, it *is* submodular on $\mathcal{C}_1$, which is enough to imply that ORWSP with at most two nonzeros per column is in $\mathcal{P}$ (see Grötschel, Lovász, and Schrijver [14]).

To get a more reasonable polynomial algorithm than the ellipsoid-based one in Grötschel, Lovász, and Schrijver [14], we solve (4.2) via a minimum cut calculation. For row 1 define a maximum flow network $\mathcal{N}_1$ as follows. Make node 1 the source, node 0 the sink, and direct all edges $j-k$ of $\mathcal{T}^*$ away from 1 with capacity $c_{jk} = w_{jk}$. Replace edge $j-k \notin \mathcal{T}^*$ by a new node e_{jk} with arcs $j \rightarrow e_{jk}$ and $k \rightarrow e_{jk}$ with capacity ∞, and arc $e_{jk} \rightarrow 0$ with capacity w_{jk}. Finally, direct each partial arc $j-0$ from j to 0 with capacity $c_{j0} = w_{j0}$.

Suppose that (S, T) is the standard min cut in $\mathcal{N}_1$ with $1 \in S$ and $0 \in T$. We claim that the row nodes $\bar{U}_1$ in S induce a subtree $\mathcal{T}(S)$ of $\mathcal{T}^*$. If not, then there would be a

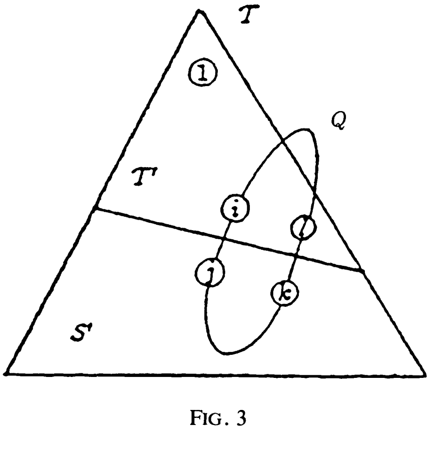


FIG. 3

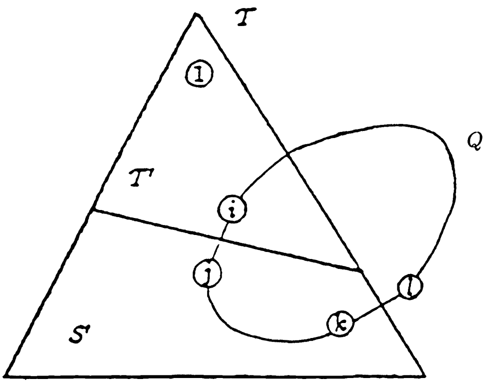


FIG. 4

connected component $\hat{C}$ of the subgraph induced by $\bar{U}_1$ different from the component C_1 containing 1. There are no arcs in $\mathcal{N}_1$ connecting a row node in C_1 to a row node in $\hat{C}$, so that deleting the row nodes in $\hat{C}$ from $\bar{U}_1$ could only improve the min cut, contradicting that (S, T) is the minimal min cut.

Now define $U_1 = \bar{U}_1 \setminus \{1\}$ and G_1 as the set of predecessor arcs for all $i \in U_1$, so $|G_1| = |U_1|$. If $i \in \bar{U}_1$ and i–j is a non-$\mathcal{T}^*$ edge incident to i with $j \neq 0$, e_{ij} must be in S (because $c_{i,e_{ij}} = \infty$). Thus if i–j is a non-$\mathcal{T}^*$ edge with $j \neq 0$, arc $e_{ij} \rightarrow 0$ is cut by (S, T), whereas if $j = 0$, then arc $j \rightarrow 0$ is cut by (S, T). Thus $\text{cap}(S, T) = \text{obj}(\mathcal{T}(S))$. This construction is reversible, which shows that this *tree algorithm* solves ORWSP in this special case.

To get around the difficulty in pasting together these one-row solutions illustrated in (1.2), let $\mathcal{M}^*$ be a fixed, maximum weight row-perfect matching in A with row i matched to column $\mathcal{M}^*(i)$, and call the column set it hits M. (Note that $\mathcal{M}^*$ matches row node i to edge $\mathcal{M}^*(i)$ incident to i in $\mathcal{G}$.) Then MP implies that $A_{\cdot M}$ is nonsingular, and the nonsingularity of a WSP solution T implies that $\hat{A}_{\cdot M} \equiv TA_{\cdot M}$ is also nonsingular. By permuting the rows of T we can assume that $\hat{A}$ has matching $\mathcal{M}^*$ in exactly the same positions as in A. To ensure that $\mathcal{M}^*$ is preserved in $\hat{A}$ we consider the following problem.

Global ORWSP (GORWSP). For row i, find U and G with $\mathcal{M}^*(i) \notin G$, which minimize the weight of the nonzeros in $A_{i\cdot}$. That is, solve ORWSP with the extra constraint that column $\mathcal{M}^*(i)$ does not get hit.

GORWSP is easy to solve: just temporarily delete column $\mathcal{M}^*(i)$ from the matrix and use the Tree Algorithm to solve the modified ORWSP. We call this the *global tree algorithm*.

We want to prove that the resulting (U_i, G_i) pairs are transitive in the sense of Theorem 1.5 to get global optimality. To do this we need to analyze the relation between our globally fixed $\mathcal{M}^*$ and the maximum weight-spanning tree $\mathcal{T}_i^*$, which we fix for solving row i's GORWSP. Define $\mathcal{G}_i$ as $\mathcal{G}$ with edge $\mathcal{M}^*(i)$ removed.

Consider the subgraph of $\mathcal{G}$ induced by the nonpartial edges in $\mathcal{M}^*$ and let $\{C_k\}$ be the set of its connected components. Note that each C_k must have either one fewer edge than its number of nodes (so that C_k is a tree, and one node in C_k is matched to a partial edge under $\mathcal{M}^*$; we consider this partial edge to be part of C_k), or the same number of edges and nodes (so that C_k is a 1-tree). Suppose that row node i is in C_l. Then there exists a maximum spanning tree $\mathcal{T}_i^*$ for $\mathcal{G}_i$, which contains every edge in $\mathcal{M}^*$ except for $\mathcal{M}^*(i)$ from C_l and a lightest weight edge from the unique cycle in each C_k, which is a 1-tree (possibly including $C_l \setminus \{\mathcal{M}^*(i)\}$), else $\mathcal{M}^*$ would not be optimal. This is the $\mathcal{T}_i^*$ that we will use to solve GORWSP for row i with the global tree algorithm.

LEMMA 4.5. *The $\bar{U}_i$ resulting from the global tree algorithm using $\mathscr{T}_i^*$ is a subset of C_l.*

Proof. Denote the cut induced by subtree $\mathscr{T}$ by cut $(\mathscr{T})$. Lemma 4.4 says that the global tree algorithm will produce an optimal $\bar{U}_i$, which is a subtree $\mathscr{T}_{\text{big}}$ of $\mathscr{T}_i^*$. If $\mathscr{T}_{\text{big}}$ is not contained in C_l, then there is an edge $u-v \in \mathscr{T}_{\text{big}} \backslash \mathscr{M}^*$ with, say, i, $u \in C_l$ and $v \in C_r$, $r \neq l$. Define $\mathscr{T}_{\text{small}}$ as $\mathscr{T}_{\text{big}}$ minus the subtree rooted at u. Edge $u-v$ contributes w_{uv} to cap (cut $(\mathscr{T}_{\text{small}})$), which is missing from cap (cut $(\mathscr{T}_{\text{big}})$). Now cut $(\mathscr{T}_{\text{big}})$ must cut some edge a–b in C_q, for if all nodes of C_q are in $\mathscr{T}_{\text{big}}$ and C_q is a 1-tree, then cut $(\mathscr{T}_{\text{big}})$ cuts the lightest edge in C_q's cycle; if C_q is a tree, cut $(\mathscr{T}_{\text{big}})$ cuts C_q's partial edge. This edge a–b does not contribute to cap (cut $(\mathscr{T}_{\text{small}})$). Any other edges contributing to cap (cut $(\mathscr{T}_{\text{small}})$) also contribute to cap (cut $(\mathscr{T}_{\text{big}})$).

But it must be true that $w_{uv} \leqq w_{ab}$ for all a–b in C_q (including the partial edge, if any), otherwise we could swap $u-v$ for a–b in $\mathscr{M}^*$, contradicting its optimality. Thus, cap (cut $(\mathscr{T}_{\text{small}})$) $\leqq$ cap (cut $(\mathscr{T}_{\text{big}})$); by Proposition 1.1, the Global Tree Algorithm prefers $\mathscr{T}_{\text{small}}$ to $\mathscr{T}_{\text{big}}$. □

LEMMA 4.6. *The (U_i, G_i) resulting from the global tree algorithm are transitive in the sense of Theorem* 1.5.

Proof. Because $U_i \subseteq C_l$ by Lemma 4.5, the predecessor arc of each $j \in U_i$ is edge $\mathscr{M}^*(j)$ (directed into j). Thus, G_i is indeed picked as the set of columns that match into U_i with respect to $\mathscr{M}^*$, which is condition (1) of transitivity.

For condition 2 of transitivity, Lemma 4.5 says that $j \in U_i$ implies that i and j are in the same C_l. Suppose that $\bar{U}_j \backslash \bar{U}_i \neq \varnothing$. Note that $\bar{U}_i \cap \bar{U}_j$ is a candidate for j's min cut, but that the global tree algorithm picked $\bar{U}_j = (\bar{U}_i \cap \bar{U}_j) \cup (\bar{U}_j \backslash \bar{U}_i)$ instead, so that adding $\bar{U}_j \backslash \bar{U}_i$ must have made the objective value for $\bar{U}_i \cap \bar{U}_j$ strictly decrease. But then adding $\bar{U}_j \backslash \bar{U}_i$ to $\bar{U}_i$ to get $\bar{U}_i \cup \bar{U}_j$ would also strictly decrease the objective value for $\bar{U}_i$, contradicting its optimality. Thus $\bar{U}_j \subseteq \bar{U}_i$. □

Proof of Theorem 4.2. By Lemma 4.6 the $\{U_i, G_i\}$ from the Global Tree Algorithm are transitive, so by Theorem 1.5, the transformation matrix T is nonsingular. We showed above that any nonsingular T must preserve $\mathscr{M}^*$, and the global tree algorithm ensures that the GORWSP solutions attain the minimum weight of nonzeros in every row subject to this constraint. Thus T solves WSP. □

5. Conclusions and extensions. We have seen that the weighted sparsity problem is generally very difficult, even for quite restricted instances, even if we make the nondegeneracy assumption MP. It is easy if we further restrict to at most two nonzeros per row or column, but this is apparently of mostly theoretical interest. Of greater practical interest is that WSP does become easy if all weights are equal and MP holds. Practical algorithms exist for SP that have been applied to real LPs that violate MP with good results. We report on two implementations in McCormick [18], [19] and Chang and McCormick [4]. The computational results so far indicate that solving SP can significantly speed up solving linear programs. Note that SP can in fact handle any set of 0,1 weights, because columns of weight zero can be deleted from consideration.

The polynomial algorithm for WSP with at most two nonzeros per column in § 4 suggests a reasonable heuristic for (unrestricted) WSP: Let $\mathscr{M}^*$ be a maximum weight matching in A, and extend w to rows via $w(i) \equiv w_{\mathscr{M}^*(i)}$. Now construct bipartite network $\mathscr{B}_i$ for A as was done in solving SP via the *parallel algorithm* (PA) in Hoffman and McCormick [15], except that $c_{sk} = w(k)$ and $c_{jt} = w_j$ if $a_{ij} = 0$, $c_{jt} = 0$ otherwise. Selecting U_i as the rows in a minimum cut in $\mathscr{B}_i$ and G_i with respect to $\mathscr{M}^*$ should produce a reasonable answer to WSP in a *weighted* PA algorithm. A more realistic heuristic would be to apply the same idea to Hoffman and McCormick's [15] *sequential algorithm* or Chang and McCormick's [3] *hierarchical algorithm*, which have more desirable properties in practice. However, $\mathscr{M}^*$ might have to be recalculated at some rows if a matched entry gets hit. We must also be careful to select G_i as a high-weight subset with desirable numerical properties.

Acknowledgment. These problems were originally suggested by Walter Murray. We thank him for his help.

REFERENCES

[1] I. ADLER, N. KARMARKAR, M. G. C. RESENDE, AND G. VEIGA, *Data structures and programming techniques for the implementation of Karmarkar's algorithm*, ORSA J. Comput. 1 (1989), pp. 84–106.

[2] S. F. CHANG, *Increasing sparsity in matrices for large-scale optimization—theoretical properties and implementation aspects*, Ph.D. thesis, Columbia University, New York, 1989.

[3] S. F. CHANG AND S. T. MCCORMICK, *A hierarchical algorithm for making sparse matrices sparser*, UBC Faculty of Commerce working paper 90-MSC-012, Vancouver, BC, 1990a; Math. Programming, 56 (1992), pp. 1–30.

[4] ———, *Computational results for the hierarchical algorithm for making sparse matrices sparser*, UBC Faculty of Commerce working paper 90-MSC-013, Vancouver, BC, 1990b; ACM Trans. Math. Software, to appear.

[5] A. CHARNES, *Optimality and degeneracy in linear programming*, Econometrica, 20 (1952), pp. 160–170.

[6] T. F. COLEMAN, *Large sparse numerical optimization*, Lecture Notes in Computer Science 165, Springer-Verlag, Berlin, 1984.

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

[8] A. FRANK AND É. TARDOS, *Generalized polymatroids and submodular flows*, Math. Programming B, 42 (1988), pp. 489–563.

[9] R. M. FREUND, *Projective transformations for interior point methods, Part* II: *Analysis of an algorithm for finding the weighted center of a polyhedral system*, Tech. Report OR 180-88, MIT, Cambridge, MA, 1988.

[10] M. R. GAREY AND D. S. JOHNSON, *The rectilinear steiner tree problem is* NP-*complete*, SIAM J. Appl. Math., 32 (1977), pp. 826–834.

[11] ———, *Computers and Intractability*, W. H. Freeman, New York, 1979.

[12] D. M. GAY, *Electronic mail distribution of linear programming test problems*, Mathematical Programming Society Committee on Algorithms Newsletter 13, 1985, pp. 10–12.

[13] A. V. GOLDBERG, S. A. PLOTKIN, AND É. TARDOS, *Combinatorial algorithms for the generalized circulation problem*, Tech. Report MIT/LCS/TM-358, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 1988.

[14] M. GRÖTSCHEL, L. LOVÁSZ, AND A. SCHRIJVER, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica, 1 (1981), pp. 169–197.

[15] A. J. HOFFMAN AND S. T. MCCORMICK, *A fast algorithm that makes matrices optimally sparse*, in Progress in Combinatorial Optimization, W. R. Pulleyblank, ed., Academic Press, New York, 1984, pp. 185–196.

[16] E. L. LAWLER, *Combinatorial Optimization*: *Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.

[17] I. J. LUSTIG, *An analysis of an available set of linear programming test problems*, Comp. Oper. Res., 16 (1987), pp. 173–184.

[18] S. T. MCCORMICK, *A combinatorial approach to some sparse matrix problems*, Ph.D. thesis, Stanford University, Stanford, CA, 1983.

[19] ———, *Making sparse matrices sparser*: *Computational results*, Math. Programming, 49 (1990), pp. 91–111.

[20] N. MEGIDDO, *Towards a genuinely polynomial algorithm for linear programming*, SIAM J. Comput., 12 (1983), pp. 347–353.

[21] K. MUROTA AND M. IRI, *Structural solvability of systems of equations—a mathematical formulation for distinguishing accurate and inaccurate numbers in structural analysis of systems*, Japan J. Appl. Math., 2 (1985), pp. 247–271.

[22] J. B. ORLIN, *On the simplex algorithm for networks and generalized networks*, Math. Programming Studies, 24 (1985), pp. 166–178.

[23] H. J. RYSER, *Combinatorial mathematics*, MAA Camus Mathematical Monograph 14, Providence, RI, 1963.

[24] L. J. STOCKMEYER, personal communication, 1982.

[25] D. J. A. WELSH, *Matroid Theory*, Academic Press, New York, London, 1976.

SIAM J. DISC. MATH.
Vol. 6, No. 1, pp. 70–86, February 1993

006

EXACT FORMULAS FOR MULTITYPE RUN STATISTICS IN A RANDOM ORDERING*

MACDONALD MORRIS†‡, GABRIEL SCHACHTEL†§, AND SAMUEL KARLIN†¶

Abstract. Exact formulas are developed for the probability that a random ordering of a fixed collection of letters contains any specified collection of runs. These results are particularly useful for short sequences and other cases where asymptotic formulas cannot be trusted. Computer programs to evaluate these formulas are available.

Key words. run tests, clusters, arrangements, orderings, sequences

AMS(MOS) subject classifications. 05A05, 05A15, 05A19

Introduction. A variety of run tests are presently used to identify nonrandomness in sequences. Many of these tests are based on models in which the letters are generated independently or with Markov dependence. Tests based on a shuffling model in which the pool of sequence elements is fixed include results on the total number of runs (of any length) and on the length of the longest run. In this paper, we present exact formulas for the probability that a random ordering of a fixed pool of letters contains any minimum collection of runs of specified lengths and letter types.

A multinomial model that has been used to investigate the significance of runs in sequences allows each letter type L_i to be chosen with probability p_i independently at each position in the sequence. This model and more general models in which the composition of the sequence is not fixed in advance are known as *unconditional* models. The probability of success runs in the independently and identically distributed (i.i.d.) model was investigated early by Mood (1940) among others. Asymptotics for this model have been studied by Erdös and Rényi (1970), Guibas and Odlyzko (1980), and Deheuvels and Devroye (1987), among others. A model in which the letters are generated as a Markov sequence has been studied by Samarova (1981) and Foulser and Karlin (1987), who obtained asymptotics for the longest success runs of a single letter, multiple letters, and certain repeated patterns of letters. The related problem of clusters is the subject of a large body of literature (see, for example, Glaz (1989), Leung (1989), Wallenstein and Neff (1987), and the bibliography of Naus (1979)).

Models in which the numbers of letters of each type are fixed are known as *conditional* models. Although some researchers have considered the case in which arrangements of the fixed pool of letters have different likelihoods of occurrence (see, for example, Bateman (1948)), we will be considering only the model in which all arrangements are equally likely (the shuffling model). This model was studied by Mood (1940), who developed formulas for the distribution of the total number of runs (of any length) and for the expectation and moments of the number of runs of a given exact length. Bateman developed a formula for the probability of at least one run exceeding a given length. The

* Received by the editors August 23, 1991; accepted for publication (in revised form) March 12, 1992.

† Department of Mathematics, Stanford University, Stanford, California 94305.

‡ Present address, Department of Pharmaceutical Chemistry, University of California–San Francisco, San Francisco, California 94143-0446. This author's work was supported by a National Science Foundation Graduate Fellowship.

§ Present address, Mathematics Division, University of Texas, San Antonio, Texas 78285. This author's work was partially supported by Deutsche Forschungsgemeinschaft DFGScha441/3-1.

¶ This author's work was partially supported by National Institutes of Health grants HG00335-04 and GM10452-29 and by National Science Foundation grant DMS86-06244.

elegant proof of this result presented in Bradley (1968) was the point of departure for the derivations herein. This paper derives a formula for the number of arrangements of a given pool of letters having any specified set of runs of any number of different letter types and lengths (see § 3). To aid in computation and illustrate the methods of proof, we derive two simpler formulas, which are restricted to runs of only one letter type: multiple runs of one minimum length (see § 1) and any number of minimum lengths (see § 2).

When analyzing data generated by an unknown process for a tendency to form runs, the conditional model is preferable because it makes fewer assumptions about the process. If the unconditional model is used with probabilities p_i taken from the observed frequencies, there is a tendency to underestimate the significance of uncommon events, even when the underlying process is, in fact, independent. (This results from the facts that the observed frequencies vary about the actual generating frequencies and the probability of an uncommon run is a convex function of the frequency.) Tests based on the unconditional model are true to their significance threshold, provided only that the underlying process is order-independent. Although the conditional and unconditional models are identical in the appropriate asymptotics, differences between them can be quite pronounced for shorter sequences, and for more uncommon events. The results presented herein, because they are exact and conditioned on the sequence composition, are thus particularly useful for problems involving relatively short sequences and other cases where the asymptotics cannot be relied upon. A computer program to calculate run probabilities based on these formulas has been implemented in the language C and is available from the authors.

The probabilities we are investigating can be stated in terms of the order statistics giving the lengths of the runs of each letter type in order of size (the length of the longest run, the length of the second longest, and so forth). For example, the probability of obtaining at least s_1 separate runs of length at least r_1, and s_2 additional separate runs of at least r_2 A's ($r_2 \leqq r_1$), is $\Pr\{R_{s_1} \geqq r_1,\ R_{s_1+s_2} \geqq r_2\}$, where R_j is the run-order statistic giving the length of the jth longest run of A's. Problems involving runs of two or more letter types involve the multidimensional order statistics R_j^L, where R_j^L is the jth longest run length of letter type L. The probability of observing at least s_{ij} separate runs of length r_{ij} or greater for each letter type i, and length index j, is $\Pr\{R_{\sigma_{ij}}^{L_i} \geqq r_{ij}\colon i = 1, 2, \ldots, p;\ j = 1, 2, \ldots, q_i\}$, where $\sigma_{ij} = \sum_{\nu=1}^{j} s_{i\nu}$. Determining the probability of observing the specified runs and determining the number of arrangements containing those runs are, of course, equivalent problems, since the total number of arrangements is known.

The original motivation for this work comes from the study of nucleic and amino acid sequences, especially the study of runs of charged amino acids in protein sequences. Statistically significant runs, clusters, and patterns of charged amino acids have been shown to be associated with several classes of regulatory proteins and to be useful in identifying sequence features of biological interest (Karlin et al. (1989), Karlin (1990)). Many sequences, while having no single run of statistically significant length, have multiple runs (either of one charge or of both positive and negative charge), which, taken together, appear highly nonrandom. The formulas presented below provide a statistical basis for this distinction. We can, for example, compute the probability that a random ordering of 400 amino acids, of which 50 are positively charged and 40 are negatively charged, contains a run of at least four positively charged amino acids and an additional run of at least five negatively charged amino acids. When applying these formulas to a collection of sequences, it is important to remember that the criterion for significance of each sequence depends on its individual length and composition. As a result, in some sequences, relatively short runs may qualify as statistically significant, while in other sequences much longer runs will not.

The results we obtain can be applied equally well to the problem of spacings, which are regarded as runs of the complementary type. For example, the question of whether the occurrences of a particular letter or set of letters in a sequence are unusually evenly or unevenly spaced can be approached by investigating whether there are unusually many or unusually few long spacings. Applications of spacing formulas to molecular biology include evaluating the nature of inhomogeneity in DNA and protein sequences and assessing the distribution of sites in physical and genetic maps (e.g., Karlin and Macken, 1991).

Results. For convenience, all the results are listed here together. The function Δ, which appears in the results, is defined as

$$\Delta(i, s) = \begin{cases} (-1)^{i+s}\binom{i-1}{s-1} & \text{if } s > 0, \\ 1 & \text{if } s \leqq 0 \text{ and } i = 0, \cdot, \\ 0 & \text{otherwise.} \end{cases}$$

(This is a natural value for the negative binomial.) Proofs are given in the following sections.

Result 1. Given N letters, n_1 A's, and $n_2 = (N - n_1)$ B's, the number of orderings with at least s separate runs of at least r A's is

$$\sum_{h \geqq s} \Delta(h, s)\binom{n_2 + 1}{h}\binom{N - hr}{n_2}.$$

Result 2. Given N letters, n_1 A's, and $n_2 = (N - n_1)$ B's, the number of orderings with at least s_i separate runs of at least r_i A's for $i = 1, 2, \ldots, q$, where $\vec{s} = (s_1, s_2, \ldots, s_q)$ and $\vec{r} = (r_1, r_2, \ldots, r_q)$ with $r_1 > r_2 > \cdots > r_q$, is

$$\sum_{\mathbf{H} \geqq \mathbf{S}} H_q!\binom{n_2 + 1}{H_q}\binom{n_2 + t}{n_2}\prod_{i=1}^{q}\left(\frac{\Delta(h_i, S_i - H_{i-1})}{h_i!}\right),$$

where H_i and S_i are the partial sums of $\vec{h}$ and $\vec{s}$ (e.g., $H_j = \sum_{i=1}^{j} h_i$, $H_0 = 0$), $\mathbf{H} \geqq \mathbf{S}$: $\Leftrightarrow H_i \geqq S_i$ for all i, and $t = n_1 - \sum_{i=1}^{q} h_i r_i$.

Result 3. Given $N = \sum_{i=1}^{p} n_i$ letters, n_i of each letter type L_i, the number of orderings that have at least s_{ij} separate runs of at least r_{ij} letters L_i, for $i = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, q_i$, where s_{ij} and r_{ij} are positive integers and $r_{i1} > r_{i2} > \cdots > r_{iq_i}$, is

$$\sum_{\mathbf{H} \geqq \mathbf{S}} \sum_{\substack{\vec{b} \leqq \vec{d} \leqq \vec{H} \\ 0 \leqq \vec{\delta} \leqq \vec{d}}} (-1)^{\vartheta(\vec{d}) + \varepsilon(\vec{\delta})}\left(\sum_{i=1}^{p} (d_i + t_i)\right)!$$

$$\prod_{i=1}^{p}\left(\frac{H^{(i)}!}{d_i!t_i!}\binom{H^{(i)} - 1}{H^{(i)} - d_i}\binom{d_i}{\delta_i}\prod_{j=1}^{q_i}\left(\frac{\Delta(h_{ij}, S_j^{(i)} - H_{j-1}^{(i)})}{(h_{ij})!}\right)\right),$$

where $t_i = n_i - \delta_i - \sum_{j=1}^{q_i} h_{ij}r_{ij}$, $H^{(i)} = \sum_{j=1}^{q_i} h_{ij}$, $\varepsilon(\vec{\delta}) := \sum_{i=1}^{p} \delta_i$, $\vartheta(\vec{d}) = \sum_{i=1}^{p} (H^{(i)} - d_i)$, $\vec{b}$ is the vector whose ith component is 1 if $H^{(i)} \geqq 1$ and 0 if $H^{(i)} = 0$, $S_j^{(i)}$ and $H_j^{(i)}$ are partial sums (e.g., $S_j^{(i)} = \sum_{\nu=1}^{j} s_{i\nu}$, $S_0^{(i)} = 0$), and $\mathbf{H} \geqq \mathbf{S} \Leftrightarrow H_j^{(i)} \geqq S_j^{(i)}$ for all i and j.

COROLLARY TO RESULT 3. *The number of arrangements of n_1 A's, n_2 B's, and n_3 C's that contain a run of at least r_1 A's and a run of at least r_2 B's is given by*

$$\sum_{\vec{u}_1, \vec{v}_1, \vec{u}_2, \vec{v}_2} (-1)^{\langle \vec{e}, \vec{u}_1 + \vec{u}_2 \rangle} \frac{(n_3 + t_1 + t_2 + z)!}{[\prod_{i=1}^{\infty} u_1(i)!u_2(i)!v_1(i)!v_2(i)!]t_1!t_2!n_3!},$$

where

$$t_1 = n_1 - \sum_{i=1}^{\infty} ir_1u_1(i) + (ir_1 + 1)v_1(i),$$

$$t_2 = n_2 - \sum_{i=1}^{\infty} ir_2u_2(i) + (ir_2 + 1)v_2(i),$$

$$z = \sum_{i=1}^{\infty} u_1(i) + u_2(i) + v_1(i) + v_2(i),$$

and the summation is over all combinations of nonnegative integer vectors $\vec{u}_1$, $\vec{u}_2$, $\vec{v}_1$, $\vec{v}_2$ *such that* $n_1 > t_1 \geqq 0$ *and* $n_2 > t_2 \geqq 0$.

This corollary with proof can be found as Result 4 in Morris (1990).

Preliminaries. Let $\mathscr{L}$ be a pool of N letters from a p-letter alphabet: n_1 of letter L_1, n_2 of letter $L_2, \ldots, n_p$ of letter $L_p(N = \sum_{i=1}^{p} n_i, p \geqq 2)$. Define the *orderings* of $\mathscr{L}$ to be the distinguishable arrangements of all members of letterpool $\mathscr{L}$, and let the sequence $Z = (z_1, z_2, \ldots, z_N)$ be one such ordering. We define a run within Z as a subsequence 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 runs of *B*'s only of lengths 1 and 3. We will sometimes refer to runs of the same letter type as *separate* to emphasize that they must be separated by intervening letters of a different type.

Let $\mathscr{F}$ 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 $\mathscr{F}$*-acceptable*) if, for each run specified in $\mathscr{F}$, 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 $\mathscr{F}$, 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 $\mathscr{F}$*-pool*. For each run specified in $\mathscr{F}$, 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 or $\mathscr{F}$-pool. Arrangements of the $\mathscr{F}$-pool are called *configurations* to distinguish them from orderings of the full letterpool, $\mathscr{L}$. 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, $\mathscr{L}$ 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 $\mathscr{F} = \{$at least two runs$(A) \geqq 3\}$. We use six of the A's to construct two chunks of three A's each, denoted by x's. Our $\mathscr{F}$-pool thus consists of two x chunks, one remaining A, and three B's, and each arrangement of this $\mathscr{F}$-pool 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 $\mathscr{F}$-acceptable since it has runs of lengths 3 and 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 $\mathscr{F}$-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 $\mathscr{F}$-pool, 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: (1) multiple ways of *positioning* a chunk within a longer run, (2) multiple *occupancy* of two or more chunks within a single long run, and (3) 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 (see also Table 1).

(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 $\mathscr{O}_1 = (A, A, A, A, B)$ is counted by the configurations $\mathscr{C}_1 = (A, x, B)$ and $\mathscr{C}_2 = (x, A, B)$. 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, $\mathscr{C}_1$ is endplaced; $\mathscr{C}_2$ 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 $\mathscr{O}_2 = (A, A, A, A, A, A, A, B)$ is not acceptable because it has only one separate run of length 3 or greater. However, $\mathscr{O}_2$ is counted by the configurations $\mathscr{C}_3 = (x, x, A, B)$, $\mathscr{C}_4 = (x, A, x, B)$, and $\mathscr{C}_5 = (A, x, x, B)$. 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 $\mathscr{F}$-pool and is therefore acceptable. Furthermore, an acceptable ordering, because it has a separate run for each chunk in the $\mathscr{F}$-pool, 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 $\mathscr{O}_2$, for example, has no disjoint orderings ($\mathscr{C}_3$ and $\mathscr{C}_4$ are not endplaced, and $\mathscr{C}_5$ 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 $\mathscr{C}_6 = (x, B, x, B, A, A, A)$, $\mathscr{C}_7 = (x, B, A, A, A, B, x)$, and $\mathscr{C}_8 = (A, A, A, B, x, B, x)$. A judiciously weighted sum of the numbers of disjoint configurations eliminates overcounting of type (3) and gives the number of acceptable orderings. In the construction of this sum, we make use of the following standard equality:

$$\sum_{i \geqq s} \Delta(i, s)\binom{k}{i} = 1 \quad \text{for } k \geqq s \text{ and } k \geqq 0, \tag{1}$$

TABLE 1
The three sources of overcounting.

	A	A	A	A	A	C	G	G	A	A	C	G	G	A	A	A	A	G	G	G	A	A	A	G	T	A	T	"solution"
positioning	$5a$					C	$2g$		A	A	C	$2g$		A	A	$2a$		G	$2g$		A	$2a$		G	T	A	T	endplaced
	$5a$					C	$2g$		A	A	C	$2g$		A	A	$2a$		G	$2g$		$2a$		A	G	T	A	T	
occupancy	$5a$					C	$2g$		A	A	C	$2g$		A	A	$2a$		G	$2g$		A	$2a$		G	T	A	T	separate (and
	$5a$					C	$2g$		A	A	C	$2g$		$2a$		$2a$		G	$2g$		A	A	A	G	T	A	T	endplaced) disjoint
assignment	$5a$					C	$2g$		A	A	C	$2g$		A	A	$2a$		G	$2g$		A	$2a$		G	T	A	T	weight fct. cancels
	$5a$					C	$2g$		$2a$		C	$2g$		A	A	A	A	G	$2g$		A	$2a$		G	T	A	T	overcounting by
	$5a$					C	$2g$		$2a$		C	$2g$		A	A	$2a$		G	$2g$		A	A	A	G	T	A	T	assignment

For each type of overcounting, the middle column shows two or more configurations corresponding to the same ordering (shown at top). Chunks are designated by their length and letter type (e.g., $5a$ is a chunk of 5 A's). The right column describes the method by which this overcounting is resolved.

$$\text{where} \quad \Delta(i, s) = \begin{cases} (-1)^{i+s}\binom{i-1}{s-1} & \text{if } s > 0, \\ 1 & \text{if } s \leqq 0 \text{ and } i = 0, \\ 0 & \text{otherwise.} \end{cases}$$

In some of the derivations below (see §§ 1 and 2), it is possible to eliminate all nondisjoint configurations in a single step. In the general case, however, we will first discard all nonendplaced configurations, and then in a second step we will discard the remaining nonseparate configurations.

1. Multiple runs of one letter with one minimum length. In this section, we wish to determine the number of arrangements of N letters, n_1 A's, and $n_2 = (N - n_1)$ B's that contain at least s separate runs of A's, each of length $\geqq r$ (where $sr \leqq n_1$, and $s \leqq n_2 + 1$). In other words, given a letterpool $\mathscr{L}_1$ with n_1 A's, n_2 B's, and a condition $\mathscr{F}_1 = \{\text{at least } s \text{ runs}(A) \geqq r\}$, what is the number of $\mathscr{F}_1$-acceptable orderings?

We will proceed in two steps, first counting the number DC(h, r) of disjoint configurations of a generalized $\mathscr{F}$-pool containing h chunks and then determining the number AO(s, r) of acceptable orderings through a weighted sum of the form $\sum_{h \geqq s} w(h, s)\text{DC}(h, r)$.

1.1. The number of disjoint configurations (DC). DC(h, r) is the number of disjoint configurations of a pool of letters and chunks corresponding to the letterpool $\mathscr{L}_1$ and containing h chunks(A) of length r, $(n_1 - hr)$ remaining A's, and n_2 B's. It is called the generalized $\mathscr{F}_1$-pool because it is a generalized version of the $\mathscr{F}_1$-pool: While the length and letter type of the chunks remains the same, their number h is allowed to vary over the range $h \geqq s$.

The number of disjoint configurations is the number of ways of carrying out the following two-step construction process:

(i) Order the individual letters (excluding the chunks) from the pool. Since there are a total of $N - hr$ individual letters, of which n_2 are B's, there are a total of

$$\binom{N - hr}{n_2}$$

ways to do this;

(ii) Insert the h chunks into this ordering. For the configuration to be endplaced, it is necessary and sufficient that no chunk precede an A, leaving as possible insertion points n_2 locations preceding B's and one location at the right end. For the configuration to be separate, it is necessary and sufficient that no more than one chunk be inserted at any of these locations. This step can be carried out in

$$\binom{n_2 + 1}{h}$$

ways.

The product from steps (i) and (ii) gives us the following total number of disjoint configurations with h chunks:

$$DC(h, r) = \binom{n_2 + 1}{h}\binom{N - hr}{n_2}. \tag{2}$$

Remark. The restriction $hr \leqq n_1$ simply specifies that A's required for the chunks cannot exceed the number available. Clearly, there are no configurations for larger h, since the $\mathscr{F}$-pool cannot be constructed; the purpose of the limit is to prevent (2), which can have nonzero and even negative values for $hr \geqq N$, from being evaluated in these cases.

1.2. The number of acceptable orderings (AO). Following an argument similar to that used by Bradley (1968), we now determine the number AO(s, r) of acceptable orderings as a weighted sum of the numbers of disjoint configurations.

Let G_m denote the number of orderings with *exactly* m runs$(A) \geqq r$. Then AO(s, r) can be rewritten as the sum over all G_m with $m \geqq s$ as follows:

$$\text{AO}(s, r) = G_s + G_{s+1} + G_{s+2} + \cdots . \tag{3}$$

As noted previously, DC, the number of disjoint configurations, overcounts the number of acceptable orderings by different assignments of chunks to runs (see Preliminaries). Recall the process of "placing" chunks on runs of the ordering to generate the different corresponding configurations. When the resulting configuration must be disjoint, there is no choice of how many chunks may occupy each run (no more than one), nor is there any choice of where the chunk may be positioned in the run (it must be placed at the right-hand end). The only choice is of which runs will be assigned chunks. Therefore, the number of disjoint configurations of h chunks corresponding to an ordering with exactly m runs$(A) \geqq r$ is the number of ways of choosing h of the m runs into which to place the chunks, or $\binom{m}{h}$.

For each m, there are G_m orderings with exactly m runs$(A) \geqq r$ and $\binom{m}{h} G_m$ corresponding disjoint configurations of h chunks. Recalling that each disjoint configuration corresponds to exactly one acceptable ordering, we can write

$$\text{DC}(h, r) = \sum_{m \geqq h} \binom{m}{h} G_m. \tag{4}$$

The desired number of acceptable orderings can now be calculated as a weighted sum of the DC(h, r) values of (2) for $h \geqq s$.

Result 1. Given N letters, n_1 A's, and $n_2 = (N - n_1)$ B's, the number AO(s, r) of orderings with at least s separate runs of at least r A's is

$$\text{AO}(s, r) = \sum_{h \geqq s} \Delta(h, s) \binom{n_2 + 1}{h} \binom{N - hr}{n_2},$$

where the function $\Delta(h, s)$ is defined in (1).

Proof. Using (1)–(4) leads to the following stated result:

$$\text{AO}(s, r) = \sum_{m \geqq s} 1 \cdot G_m = \sum_{m \geqq s} \left[\sum_{h \geqq s} \Delta(h, s) \binom{m}{h} \right] G_m = \sum_{h \geqq s} \Delta(h, s) \sum_{m \geqq s} \binom{m}{h} G_m$$

$$= \sum_{h \geqq s} \Delta(h, s) \text{DC}(h, r) = \sum_{h \geqq s} \Delta(h, s) \binom{n_2 + 1}{h} \binom{N - hr}{n_2}.$$

Remarks. If the number of A's in required runs exceeds the number of available A's ($sr > n_1$), or if there are not enough B's to separate the s chunks ($s > n_2 + 1$), the sum is zero, as it should be.

2. Multiple runs of one letter with multiple minimum lengths. We now generalize Result 1, to allow for more than one minimum length; i.e., the letterpool $\mathcal{L}_2$ with n_1 A's and $n_2 = (N - n_1)$ B's will remain the same as $\mathcal{L}_1$, but the condition $\mathcal{F}_1$ will be replaced by $\mathcal{F}_2 = \{$at least s_1 runs$(A) \geqq r_1$, at least s_2 additional runs$(A) \geqq r_2, \cdots$, and at least s_q additional runs$(A) \geqq r_q$; where $r_1 > r_2 > \cdots > r_q\}$. In vector notation, $\mathcal{F}_2 = \{$at least $\vec{s}$ runs$(A) \geqq \vec{r}\}$, where $\vec{s} = (s_1, s_2, \ldots, s_q)$ and $\vec{r} = (r_1, r_2, \ldots, r_q)$.

An approach analogous to § 1 will provide us first with the number $\mathrm{DC}(\vec{h}, \vec{r})$ of disjoint configurations of a generalized $\mathscr{F}_2$-pool with $\vec{h}$ chunks and then, by a weighted sum, with the desired number $\mathrm{AO}(\vec{s}, \vec{r})$ of $\mathscr{F}_2$-acceptable orderings.

2.1. The number of disjoint configurations. Use the notation H_j for the jth partial sum of the vector $\vec{h}$ ($H_j := \sum_{i=1}^{j} h_i$, $H_0 := 0$), M_j for the partial sum of $\vec{m}$, and so forth. The generalized $\mathscr{F}_2$-pool contains h_1 chunks(A) of length r_1, h_2 chunks(A) of length $r_2, \ldots, h_q$ chunks(A) of length r_q, $t = (n_1 - \sum_{i=1}^{q} h_i r_i)$ remaining A's, and n_2 B's.

We determine $\mathrm{DC}(\vec{h}, \vec{r})$ by applying steps (i) and (ii) of the construction process in § 1. In step (ii), however, since we are now dealing with chunks of q different sizes, we have not one but $H_q!/\prod_{i=1}^{q}(h_i)!$ ways to order the H_q chunks before inserting them in the chosen locations, giving us

$$\mathrm{DC}(\vec{h}, \vec{r}) = \frac{H_q!}{\prod_{i=1}^{q}(h_i)!}\binom{n_2+1}{H_q}\binom{n_2+t}{n_2}. \tag{5}$$

2.2. The number of acceptable orderings. Consider now an ordering of the A's and B's with *exactly* m_1 runs$(A) \geqq r_1$, m_2 additional runs$(A) \geqq r_2, \ldots, m_q$ additional runs(A) of length $\geqq r_q$. By analogy to § 1, we may now ask what is the number of disjoint configurations with $\vec{h}$ chunks(A) of lengths $\vec{r}$ that correspond to each ordering with exactly $\vec{m}$ runs$(A) \geqq \vec{r}$. We will call this number $\mathrm{RC}(\vec{m}, \vec{h})$. Since the h_1 chunks(A) of length r_1 can be assigned to the $M_1 = m_1$ runs$(A) \geqq r_1$ of each of the orderings in $\binom{M_1}{h_1}$ ways, and the h_2 chunks(A) of length r_2 can then be assigned to the $M_2 - H_1 = m_1 + m_2 - h_1$ remaining runs$(A) \geqq r_2$ in $\binom{M_2 - H_1}{h_2}$ ways, and so forth, we obtain

$$\mathrm{RC}(\vec{m}, \vec{h}) = \prod_{i=1}^{q}\binom{M_i - H_{i-1}}{h_i}. \tag{6}$$

It is clear from this derivation that an ordering corresponds to one or more disjoint configurations with $\vec{h}$ chunks if and only if $M_i \geqq S_i$ for all i, which we will denote by $\mathbf{M} \geqq \mathbf{S}$.

An ordering is $\mathscr{F}_2$-acceptable if and only if it corresponds to one or more disjoint configurations of $\vec{s}$ chunks (see Preliminaries). Therefore an ordering with $\vec{m}$ runs is acceptable if and only if $\mathbf{M} \geqq \mathbf{S}$. If $G_{\vec{m}}$ is the number of different orderings with exactly $\vec{m}$ runs$(A) \geqq \vec{r}$, then we can rewrite $\mathrm{AO}(\vec{s}, \vec{r})$ by analogy to (3) as

$$\mathrm{AO}(\vec{s}, \vec{r}) = \sum_{\mathbf{M} \geqq \mathbf{S}} G_{\vec{m}}. \tag{7}$$

Since there are $\mathrm{RC}(\vec{m}, \vec{h})$ different disjoint configurations for each of the $G_{\vec{m}}$ orderings with exactly $\vec{m}$ runs$(A) \geqq \vec{r}$, and since each disjoint configuration corresponds to exactly one ordering, which must be acceptable,

$$\mathrm{DC}(\vec{h}, \vec{r}) = \sum_{\mathbf{M} \geqq \mathbf{H}} \mathrm{RC}(\vec{m}, \vec{h}) G_{\vec{m}} = \sum_{\mathbf{M} \geqq \mathbf{H}} \prod_{i=1}^{q}\binom{M_i - H_{i-1}}{h_i} G_{\vec{m}}. \tag{8}$$

The desired number of acceptable orderings can now be calculated as a weighted sum of the $\mathrm{DC}(\vec{h}, \vec{r})$ values of (5).

Result 2. Given N letters, n_1 A's, and $n_2 = (N - n_1)$ B's, the number $\mathrm{AO}(\vec{s}, \vec{r})$ of orderings with at least s_i separate runs of at least r_i A's for $i = 1, 2, \ldots, q$, where $\vec{s} = (s_1, s_2, \ldots, s_q)$ and $\vec{r} = (r_1, r_2, \ldots, r_q)$ are positive integer vectors with $r_1 > r_2 > \cdots > r_q$, is

$$\mathrm{AO}(\vec{s}, \vec{r}) = \sum_{\mathbf{H} \geqq \mathbf{S}} H_q!\binom{n_2+1}{H_q}\binom{n_2+t}{n_2}\prod_{i=1}^{q}\left(\frac{\Delta(h_i, S_i - H_{i-1})}{h_i!}\right),$$

where H_i and S_i are the partial sums of $\vec{h}$ and $\vec{s}$ (e.g., $H_j = \sum_{i=1}^{j} h_i$, $H_0 = 0$), and $\mathbf{M} \geqq \mathbf{S}$: $\Leftrightarrow M_i \geqq S_i$ for all i, and $t = n_1 - \sum_{i=1}^{q} h_i r_i$. The function Δ is defined in (1).

Proof. For ease of notation, define

$$\sum_{h_i} f(i) := \sum_{h_i = S_i - H_{i-1}}^{M_i - H_{i-1}} \Delta(h_i, S_i - H_{i-1}) \binom{M_i - H_{i-1}}{h_i}.$$

Recall that $M_i - H_{i-1} \geqq h_i \geqq S_i - H_{i-1} \Leftrightarrow M_i \geqq H_i \geqq S_i$. From (1), we know that $\sum_{h_i} f(i) = 1$ for $\mathbf{M} \geqq \mathbf{H} \geqq \mathbf{S} \geqq \mathbf{0}$. Using this, along with (7), (8), and then (5), leads to the following result:

$$\begin{aligned}
\mathrm{AO}(\vec{s}, \vec{r}) &= \sum_{\mathbf{M} \geqq \mathbf{S}} 1 \cdot G_{\vec{m}} \\
&= \sum_{\mathbf{M} \geqq \mathbf{S}} \left[\sum_{h_1} \left(f(1) \sum_{h_2} \left(f(2) \cdots \sum_{h_{q-1}} \left(f(q-1) \sum_{h_q} f(q) \right) \cdots \right) \right) \right] G_{\vec{m}} \\
&= \sum_{\mathbf{M} \geqq \mathbf{S}} \left[\sum_{\mathbf{M} \geqq \mathbf{H} \geqq \mathbf{S}} \prod_{i=1}^{q} \Delta(h_i, S_i - H_{i-1}) \binom{M_i - H_{i-1}}{h_i} \right] G_{\vec{m}} \\
&= \sum_{\mathbf{H} \geqq \mathbf{S}} \left(\prod_{i=1}^{q} \Delta(h_i, S_i - H_{i-1}) \right) \sum_{\mathbf{M} \geqq \mathbf{H}} \prod_{i=1}^{q} \binom{M_i - H_{i-1}}{h_i} G_{\vec{m}} \\
&= \sum_{\mathbf{H} \geqq \mathbf{S}} \left(\prod_{i=1}^{q} \Delta(h_i, S_i - H_{i-1}) \right) \cdot \mathrm{DC}(\vec{h}, \vec{r}) \\
&= \sum_{\mathbf{H} \geqq \mathbf{S}} \prod_{i=1}^{q} \left(\frac{\Delta(h_i, S_i - H_{i-1})}{h_i!} \right) \cdot H_q! \binom{n_2 + 1}{H_q} \binom{n_2 + t}{n_2}.
\end{aligned}$$

Remarks. Terms for which $S_i - H_{i-1} \leqq 0$ and $h_i \neq 0$ can be ignored, since $\Delta = 0$ for these values; e.g., if $\vec{s} = (2, 1, 2)$ and $h_1 \geqq 5$, then we can ignore terms where h_2 and h_3 are nonzero. If there are insufficient A's to construct the desired runs ($n_1 < \sum_{i=1}^{q} s_i r_i$), or insufficient B's to separate the desired runs of A's ($n_2 < S_q - 1$), the formula gives zero, as it should.

3. Runs of multiple lengths and letter types. The problems treated in the previous sections are restricted to the case of runs of only one letter type. We now drop this restriction and allow runs of any number of different letter types. Formally, we state the problem as follows: Given a letter pool $\mathscr{L}_3$ with n_i letters of types L_i, ($i = 1, 2, \ldots, p$) and a condition $\mathscr{F}_3 = \{$at least s_{ij} runs$(L_i) \geqq r_{ij}$; $r_{i1} > r_{i2} > r_{i3} > \cdots > r_{iq_i}$; $i = 1, 2, \ldots, p$; $j = 1, 2, \ldots, q_i\}$, find the number of $\mathscr{F}_3$-acceptable orderings of $\mathscr{L}_3$, $\mathrm{AO}(\|s_{ij}\|, \|r_{ij}\|, \vec{n})$. (Here $\|s_{ij}\|$ and $\|r_{ij}\|$ are ragged arrays. The vector $\vec{n}$ gives the total number of each letter type in $\mathscr{L}_3$.)

Where runs of only one letter were desired, we used a two-step construction process to determine the number of disjoint configurations. When runs of more than one letter are desired, this method is no longer feasible. Instead, we calculate the number of unrestricted configurations $\mathrm{UC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d}, \vec{\delta})$ of a generalized $\mathscr{F}_3$-pool and use this value to compute first the number of endplaced configurations $\mathrm{EC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d})$ and then the number of disjoint configurations $\mathrm{DC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n})$.

To determine the numbers of endplaced and disjoint configurations, a given set $\mathscr{D}$ of unrestricted (respectively, endplaced) configurations is first considered as a subset of a much larger set of configurations of generalized $\mathscr{F}$-pools. This larger set is then partitioned into three subsets, $\mathscr{D}^0$, $\mathscr{D}^+$, and $\mathscr{D}^-$, where $\mathscr{D}^0$ is the set of endplaced (respectively, disjoint) configurations whose cardinality we wish to know. Through a properly defined bijection, we show that $\mathscr{D}^+$ and $\mathscr{D}^-$ contain equal numbers of elements, and hence by counting $\mathscr{D}^+$ positively and $\mathscr{D}^-$ negatively, we will be left with the cardinality of $\mathscr{D}^0$. The number of acceptable orderings is then determined by a weighted sum as in §§ 1 and 2. The following arguments involving extended chunks and megachunks are intended to familiarize the reader with the methods that will be used in the proof.

3.1. The method of extension. The method of extension, which is central to the calculation of the number of endplaced configurations in the general context, is best illustrated by a simple example: Consider an $\mathscr{F}$-pool with a certain number of individual letters and with two chunks, x, of the same length and letter type. What is the number $\mathrm{EC}(x, x)$ of endplaced configurations of this $\mathscr{F}$-pool? We define an *extendable* chunk as one that is adjacent on its right to an individual letter of the same type. The total number $\mathrm{UC}(x, x)$ of unrestricted configurations can then be rewritten as the sum of the number (EE) of configurations where both chunks are extendable (e.g., $xAxAB$), the number (EN) of configurations in which one chunk is extendable and the other is nonextendable (e.g., $xABAx$), and the number (NN) of configurations in which both chunks are nonextendable (e.g., $AABxx$):

$$\mathrm{UC}(x, x) = \mathrm{EE} + \mathrm{EN} + \mathrm{NN}.$$

Now consider an $\mathscr{F}'$-pool that is identical to the original $\mathscr{F}$-pool, except that one of the two chunks has been lengthened by one letter, and the pool of remaining letters has been likewise reduced. A chunk that has been lengthened by the addition of a single letter is called an *extended* chunk. The $\mathscr{F}'$-pool thus contains, in addition to the individual letters, one extended chunk x' and one unextended chunk x. We claim that the number $\mathrm{UC}(x, x')$ of unrestricted configurations of the $\mathscr{F}'$-pool is

$$\mathrm{UC}(x, x') = 2\mathrm{EE} + \mathrm{EN}.$$

Note that each configuration of the $\mathscr{F}'$-pool can be "converted" to exactly one configuration of the $\mathscr{F}$-pool by disconnecting the rightmost letter of the extended chunk. Each nonendplaced configuration of the $\mathscr{F}$-pool can be converted to one or more configurations of the $\mathscr{F}'$-pool by connecting an extendable chunk with the individual letter to its right. (By arbitrary convention, such conversions always extend chunks to the right and not the left.) $\mathscr{F}$-pool configurations with only one nonextendable chunk (of which there are EN) can each be converted to one configuration of the $\mathscr{F}'$-pool, whereas those with two extendable chunks (of which there are EE) can each be converted into two $\mathscr{F}'$ configurations, depending on which of the chunks is extended. The claim follows.

Lengthening both chunks to obtain an $\mathscr{F}''$-pool containing two extended chunks x' and using the conversion argument given above, we find that

$$\mathrm{UC}(x', x') = \mathrm{EE}.$$

The configurations we wish to count (i.e., those we call endplaced) are configurations of unextended, nonextendable chunks. Hence the number of endplaced configurations is

$$\mathrm{EC}(x, x) = \mathrm{NN} = \mathrm{UC}(x, x) - \mathrm{UC}(x, x') + \mathrm{UC}(x', x').$$

As will be seen later, this approach can be generalized to obtain the number of endplaced configurations of any number of chunks of any number of letter types and lengths.

3.2. The rightmost extension site. An *extension site* in a configuration is a boundary between letters that can be connected to extend a chunk or disconnected to make it normal length. The boundary between an extendable chunk of normal length and the letter to its right is an unconnected extension site; the boundary between the rightmost two letters of an extended chunk is a connected extension site. By this definition, a configuration is endplaced if and only if it has no extension sites.

Each nonendplaced configuration u has a unique *rightmost extension site* $\mathrm{RES}(u)$. Consider two nonendplaced configurations u and $\hat{u}$, which are identical, except that u has an unconnected RES, whereas in $\hat{u}$ it is connected (i.e., a normal-length chunk followed by a single letter of the same type in u has been replaced by a single extended chunk in $\hat{u}$). We call u and $\hat{u}$ an RES-*conjugated pair*, and each is the RES-conjugated partner of the other. For example, if A and B are individual letters, if a and b are unextended chunks of A's and B's, and if a' and b' are the corresponding extended chunks, then $u = (b'BAaaA\mathbf{a}\mathbf{A}AaBb)$ and $\hat{u} = (b'BAaaA\mathbf{a}'AaBb)$ are partners. There are three extension sites in u, of which one (within b') is connected and two (aA) are disconnected; the rightmost extension site is an aA, which has been connected to an a' in $\hat{u}$. Every nonendplaced configuration has a uniquely defined RES-conjugated partner. Let $\varepsilon(u)$ be the number of connected extension sites (the number of extended chunks) in configuration u. Since the number of extended chunks differs between partners by one, $\varepsilon(u)$ is always even in one partner and odd in the other (this will be important in the proof).

3.3. The method of fusion. The method of fusion, which is central to the calculation of the number of disjoint configurations, is also well illustrated by the following example: Consider an $\mathscr{F}$-pool with three elementary chunks x of the same length and letter type, and some set of individual letters. What is the number of disjoint configurations of this $\mathscr{F}$-pool? Recall that a configuration is disjoint if it is endplaced and if the chunks are separate (not adjacent). The total number $\mathrm{EC}(x, x, x)$ of endplaced configurations with three chunks can be rewritten as the sum of the number $X\,|\,X\,|\,X$ of endplaced configurations in which all three chunks are separate, the number $XX\,|\,X$ of endplaced configurations in which two chunks are adjacent and one is separate, and the number XXX of endplaced configurations in which all three chunks are adjacent

$$\mathrm{EC}(x, x, x) = X\,|\,X\,|\,X + XX\,|\,X + XXX.$$

A *megachunk* is a "chunk of chunks"—a group of chunks that is treated as an indivisible unit. To distinguish them from megachunks, regular chunks will sometimes be called *elementary chunks*. Consider an $\mathscr{F}^*$-pool, which is identical to the original $\mathscr{F}$-pool, except that two of the three elementary chunks have been fused to form a single megachunk, denoted $2x$. We claim that $\mathrm{EC}(2x, x)$, the number of endplaced configurations of the $\mathscr{F}^*$-pool, is

$$\mathrm{EC}(2x, x) = XX\,|\,X + 2XXX.$$

Indeed, endplaced configurations of the original pool can be "converted" to endplaced configurations of the $\mathscr{F}^*$-pool: in this case, by connecting two adjacent chunks of an $\mathscr{F}$ configuration to form a single megachunk. Configurations of the original pool with two adjacent chunks (of which there are $XX\,|\,X$) can each be converted to megachunk configurations in one way, whereas the XXX configurations with three adjacent chunks can be converted in two ways. The claim follows.

Joining all three elementary chunks to obtain an $\mathscr{F}^{**}$-pool with one large megachunk $3x$, we find by similar reasoning that $\mathrm{EC}(3x) = XXX$.

We call a configuration disjoint if it is an endplaced and separate configuration of the elementary chunks. Hence the number of disjoint configurations is

$$\mathrm{DC}(x, x, x) = X\,|\,X\,|\,X = \mathrm{EC}(x, x, x) - \mathrm{EC}(2x, x) + \mathrm{EC}(3x).$$

As will be seen later, this approach can be generalized to obtain the number of disjoint configurations of any number of chunks of any number of letter types and lengths. In this more general situation, the chunks comprising a megachunk must be of one letter type, but they may vary in length, and their order is significant: Two megachunks are identical if and only if they have the same letter type and are composed of the same length component chunks in the same order.

3.4. The rightmost fusion site. A *fusion site* is the boundary between two elementary chunks of the same letter type. If the two elementary chunks are part of a megachunk, the fusion site is said to be *connected*, otherwise it is *unconnected*. By this definition, the disjoint configurations are those endplaced configurations with no fusion sites.

Each endplaced nondisjoint configuration u has a unique *rightmost fusion site* RFS(u). Consider two nondisjoint configurations u and $\tilde{u}$, which are identical, except that u has an unconnected RFS, whereas in $\tilde{u}$ it is connected (i.e., an unconnected pair of elementary chunks in u has been replaced in $\tilde{u}$ by an equivalent megachunk). We call u and $\tilde{u}$ an RFS-conjugated pair, and each is the RFS-conjugated partner of the other. For example, if A and B are individual letters, if a and b are elementary chunks of A's and B's, and if $2a$ is a megachunk composed of two a chunks, then $u = (bbAaaB\mathbf{aa}Bb)$ and $\tilde{u} = (bbAaaB\mathbf{2a}Bb)$ are partners. Every nondisjoint configuration has a uniquely defined RFS-conjugated partner. Let $\varphi(u)$ be the number of connected fusion sites in configuration u. Since the number of connected fusion sites differs between partners by one, $\varphi(u)$ is always even in one partner and odd in the other (this will be important in the proof).

3.5. The number of unrestricted configurations. Condition $\mathscr{F}_3$, as stated in the beginning of this section, requires s_{ij} runs of minimum lengths r_{ij} and letter types L_i, for $i = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, q_i$. The corresponding $\mathscr{F}_3$-pool consists for each letter type L_i of s_{ij} elementary chunks of lengths r_{ij} and of $t_i = n_i - \sum_{j=1}^{q_i} s_{ij} r_{ij}$ remaining letters, and it is denoted by $(\|s_{ij}\|, \|r_{ij}\|, \vec{n})$.

We will generalize this $\mathscr{F}_3$-pool in three ways: by allowing the number $\|h_{ij}\|$ of elementary chunks to vary, by fusing some chunks to form megachunks, and by extending some of these chunks and megachunks with the addition of a single letter. Those elementary chunks that are not fused will be called megachunks also for simplicity; thus, generalized $\mathscr{F}_3$-pools consist only of megachunks and individual letters. Define $H^{(i)} := \sum_{j=1}^{q_i} h_{ij}$ to be the total number of elementary chunks of letter type L_i. Let $\vec{d} = (d_1, d_2, \ldots, d_p)$ and $\vec{\delta} = (\delta_1, \delta_2, \ldots, \delta_p)$ describe a generalized $\mathscr{F}_3$-pool in which there are exactly d_i megachunks of letter type L_i, of which exactly δ_i are extended. $\vec{d}$ and $\vec{\delta}$ are constrained by $b_i \leqq d_i \leqq H^{(i)}$ and $0 \leqq \delta_i \leqq d_i$, where $\vec{b}$ is the vector whose ith component is 1 if $H^{(i)} \geqq 1$ and zero if $H^{(i)} = 0$ (i.e., when there are no desired runs of this color). Note that a single choice of $\vec{d}$ and $\vec{\delta}$ can specify more than one generalized $\mathscr{F}_3$-pool, since it is usually possible to arrange the elementary chunks into $\vec{d}$ megachunks in more than one way. The following lemma gives the total number UC$(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d}, \vec{\delta})$ of unrestricted configurations generated from generalized $\mathscr{F}$-pools characterized by $\|h_{ij}\|$, $\vec{d}$, and $\vec{\delta}$.

LEMMA 1. *The total number of unrestricted configurations generated from generalized $\mathscr{F}_3$-pools with $\|h_{ij}\|$ elementary chunks that have been fused to form δ_i extended and $(d_i - \delta_i)$ unextended megachunks of letter type L_i $(i = 1, 2, \ldots, p)$ is*

$$(9)\qquad \mathrm{UC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d}, \vec{\delta}) = \left(\sum_{i=1}^{p} (d_i + t_i)\right)! \prod_{i=1}^{p} \frac{H^{(i)}!\dbinom{H^{(i)} - 1}{H^{(i)} - d_i}\dbinom{d_i}{\delta_i}}{d_i!\,t_i!\prod_{j=1}^{q_i} (h_{ij})!}.$$

This equality applies to only those values of $\|h_{ij}\|$ and $\vec{\delta}$ for which the number $t_i = n_i - \delta_i - \sum_{j=1}^{q_i} h_{ij} r_{ij}$ of remaining letters of letter type L_i is nonnegative.

Proof. For letter type L_i determine first the number of orderings of the available elementary chunks: this is $H^{(i)}!/\prod_j (h_{ij})!$. Next, choose for each of these orderings $H^{(i)} - d_i$ fusion sites out of the $H^{(i)} - 1$ possible ones: this can be done in

$$\binom{H^{(i)} - 1}{H^{(i)} - d_i}$$

ways and creates d_i megachunks. Then choose δ_i of the d_i megachunks for extension, which can be done in

$$\binom{d_i}{\delta_i}$$

ways. This results altogether in

$$\binom{H^{(i)} - 1}{H^{(i)} - d_i}\binom{d_i}{\delta_i}H^{(i)}! \Big/ \prod_j (h_{ij})!$$

ways to obtain the required megachunks and to order the megachunks of each letter type. We then arrange the megachunks and remaining letters of all types together, noting that the order of the d_i megachunks of each type has already been established. This can be done in $(\sum (d_i + t_i))!/\prod_i (d_i!t_i!)$ ways.

3.6. The number of endplaced configurations. We wish to count the number of endplaced configurations $\mathrm{EC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d})$ of the generalized $\mathscr{F}_3$-pools containing d_i megachunks of letter type L_i, none of which are extended. To do this, we must consider a much larger set of configurations: In particular, we consider the set $\Lambda(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d})$ of all configurations generated from generalized $\mathscr{F}_3$-pools in which any number of the $\vec{d}$ megachunks of each type have been extended. Let $\mathrm{UC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d}, \vec{\delta})$ be the *set* of all unrestricted configurations in which exactly $\vec{\delta}$ of the $\vec{d}$ megachunks have been extended. We define

$$\Lambda(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d}) := \bigcup_{0 \leq \vec{\delta} \leq \vec{d}} \mathrm{UC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d}, \vec{\delta}).$$

The next lemma provides us with a partition of Λ, which will be useful in deriving a formula for the number $\mathrm{EC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d})$ of endplaced configurations.

LEMMA 2. *The set* $\Lambda = \Lambda(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d})$ *can be divided into three disjoint subsets,* Λ^0, Λ^+, *and* Λ^-, *such that* (i) $\Lambda = \Lambda^0 \cup \Lambda^+ \cup \Lambda^-$, (ii) *$u$ is endplaced if and only if* $u \in \Lambda^0$, *and* (iii) $\Lambda^+ \cong \Lambda^-$ (*i.e., there exists a bijection* α: $\Lambda^+ \to \Lambda^-$).

Proof. Recall that $\varepsilon(u) := \sum_{i=1}^{p} \delta_i$ is the number of extended chunks in u. Define $\Lambda^0 := \{u \in \Lambda : u \text{ is an endplaced configuration}\}$, $\Lambda^+ := \{u \in \Lambda \backslash \Lambda^0 : \varepsilon(u) \text{ is even}\}$, and $\Lambda^- := \{u \in \Lambda \backslash \Lambda^0 : \varepsilon(u) \text{ is odd}\}$. Clearly, this definition satisfies conditions (i) and (ii). Consider the bijection that takes each nonendplaced configuration to its RES-conjugated partner. As previously noted in § 3.2, this is well defined and one-to-one on the set of nonendplaced configurations and maps Λ^+ to Λ^-, and vice versa, thus satisfying the conditions of the lemma.

LEMMA 3. *The number* $\mathrm{EC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d})$ *of endplaced configurations of generalized $\mathscr{F}$-pools in which there are exactly $\vec{d}$ megachunks is*

$$(10) \qquad \mathrm{EC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d}) = \sum_{0 \leq \vec{\delta} \leq \vec{d}} (-1)^{\varepsilon(\vec{\delta})}\mathrm{UC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d}, \vec{\delta}),$$

where $\varepsilon(\vec{\delta}) = \sum_{i=1}^{p} \delta_i$ *is the number of extended megachunks and* $\mathrm{UC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d}, \vec{\delta})$ *is the number of unrestricted configurations as given in Lemma* 1.

Proof. From the previous lemma, we conclude that the sets Λ^+ and Λ^- have equal numbers of elements ($|\Lambda^+| = |\Lambda^-|$). Noting that $\varepsilon(u) = 0$ for $u \in \Lambda^0$ (an endplaced configuration has no extended chunks), we obtain

$$\begin{aligned} \mathrm{EC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d}) &= |\Lambda^0| = |\Lambda^0| + |\Lambda^+| - |\Lambda^-| \\ &= |\{u \in \Lambda : \varepsilon(u) \text{ is even}\}| - |\{u \in \Lambda : \varepsilon(u) \text{ is odd}\}| \\ &= \sum_{0 \leqq \vec{\delta} \leqq \vec{d}} (-1)^{\varepsilon(\vec{\delta})} \mathrm{UC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d}, \vec{\delta}). \end{aligned}$$

3.7. The number of disjoint configurations. Now we wish to count the number $\mathrm{DC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n})$ of disjoint configurations of generalized $\mathscr{F}_3$-pools in which $\|h_{ij}\|$ is allowed to vary, but where no chunks are fused or extended. Since disjoint configurations are by definition endplaced, we may use the sets Λ^0 of endplaced configurations as a starting point. Let

$$\Omega(\|h_{ij}\|, \|r_{ij}\|, \vec{n}) := \bigcup_{\vec{1} \leqq \vec{d} \leqq \vec{H}} \Lambda^0(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d}).$$

The set Ω includes all the endplaced (disjoint as well as nondisjoint) configurations in which no chunks have been fused (when $\vec{d} = \vec{H}$), and a whole class of nondisjoint endplaced configurations in which some of the chunks have been fused to form larger megachunks (when $\vec{d} \neq \vec{H}$). Lemma 4 uses an argument parallel to that used in Lemmas 2 and 3 to determine the number of disjoint configurations of elementary chunks.

LEMMA 4. *The number* $\mathrm{DC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n})$ *of disjoint configurations of the elementary chunks with the remaining letters is*

$$\mathrm{DC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n}) = \sum_{\vec{1} \leqq \vec{d} \leqq \vec{H}} (-1)^{\varphi(\vec{d})} \mathrm{EC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d}), \tag{11}$$

where $\varphi(\vec{d}) = \sum_{i=1}^{p} (H^{(i)} - d_i)$ *is the total number of connected fusion sites within all the megachunks*, $\vec{H}$ *is the total number of elementary chunks of each letter type, and* $\mathrm{EC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d})$ *is the number of endplaced configurations as given in Lemma* 3.

Proof. Partition $\Omega = \Omega(\|h_{ij}\|, \|r_{ij}\|, \vec{n}, \vec{d})$ into the following three subsets: $\Omega^0 := \{u \in \Omega : u \text{ is a separate configuration}\}$, $\Omega^+ := \{u \in \Omega \backslash \Omega^0 : \varphi(u) \text{ is even}\}$, and $\Omega^- := \{u \in \Omega \backslash \Omega^0 : \varphi(u) \text{ is odd}\}$. Since a configuration is disjoint if and only if it is endplaced and separate, Ω^0 is the set of disjoint configurations. Consider the transformation that takes each nondisjoint configuration to its RFS-conjugated partner. As previously noted in § 3.4, this is well defined and one-to-one on the set of nondisjoint configurations, and, since it either increases or decreases the number of connected fusion sites by exactly one, maps Ω^+ to Ω^-, and vice versa, establishing the equal cardinality of these two sets. Noting that the value of $\varphi(\vec{d})$ is even for configurations in Ω^0 and Ω^+ and odd for those in Ω^-, the proof proceeds as in Lemma 3.

3.8. The number of acceptable orderings. We determine the number $\mathrm{AO}(\|s_{ij}\|, \|r_{ij}\|, \vec{n})$ of $\mathscr{F}_3$-acceptable orderings in a manner similar to § 2, by summing the numbers $\mathrm{DC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n})$, of disjoint configurations and adjusting for overcounting by proper weights. The approach is somewhat more tedious than in § 2 because of the increased dimension of the problem.

Use the notation $\vec{s}_i := (s_{i1}, s_{i2}, \ldots, s_{iq_i})$ for the ith row of matrix $\|s_{ij}\|$, and $S_j^{(i)}$ for the partial sum of the first j elements of $\vec{s}_i$: $S_j^{(i)} = \sum_{\nu=1}^{j} s_{i\nu}$, $S_0^{(i)} = 0$. If, for two matrices $\|h_{ij}\|$ and $\|s_{ij}\|$, the inequality $H_j^{(i)} \geqq S_j^{(i)}$ holds for all i, j, we write $\mathbf{H} \geqq \mathbf{S}$. Previously, we introduced $G_{\vec{m}}$ as the number of orderings containing exactly $\vec{m}$ runs$(A) \geqq \vec{r}$; here we generalize this, taking $G_{\|m_{ij}\|}$ as the number of orderings of the letterpool $\mathscr{L}_3$ with exactly $\vec{m}_i$ runs$(L_i) \geqq \vec{r}_i$ for all i.

Let $\mathrm{RC}(\|m_{ij}\|, \|h_{ij}\|)$ be the number of different disjoint configurations with $\|h_{ij}\|$ chunks$(L_i) \geqq \|r_{ij}\|$ that correspond to a single ordering with exactly $\|m_{ij}\|$ runs$(L_i) \geqq \|r_{ij}\|$. $\mathrm{RC}(\|m_{ij}\|, \|h_{ij}\|)$ can be generalized from (6) by simply taking the product over the different letter types as follows:

$$\mathrm{RC}(\|m_{ij}\|, \|h_{ij}\|) = \prod_{i=1}^{p} \mathrm{RC}[\vec{m}_i, \vec{h}_i] = \prod_{i=1}^{p} \prod_{j=1}^{q_i} \binom{M_j^{(i)} - H_{j-1}^{(i)}}{h_{ij}}.$$

Now $\mathrm{DC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n})$ can be expressed as a linear combination of all $G_{\|m_{ij}\|}$ with $\mathbf{M} \geqq \mathbf{H}$ as follows:

$$\begin{aligned} \mathrm{DC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n}) &= \sum_{\mathbf{M} \geqq \mathbf{H}} \mathrm{RC}(\|m_{ij}\|, \|h_{ij}\|) G_{\|m_{ij}\|} \\ &= \sum_{\mathbf{M} \geqq \mathbf{H}} \left(\prod_{i=1}^{p} \prod_{j=1}^{q_i} \binom{M_j^{(i)} - H_{j-1}^{(i)}}{h_{ij}} \right) G_{\|m_{ij}\|}. \end{aligned} \tag{12}$$

We are now able to express $\mathrm{AO}(\|s_{ij}\|, \|r_{ij}\|, \vec{n})$ as a weighted sum of the known $\mathrm{DC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n})$ values from (11).

Result 3. Given $N = \sum_{i=1}^{p} n_i$ letters (n_i of each letter type L_i), the number $\mathrm{AO}(\|s_{ij}\|, \|r_{ij}\|, \vec{n})$ of orderings that have at least s_{ij} separate runs of at least r_{ij} letters L_i (for $i = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, q_i$), where s_{ij} and r_{ij} are positive integers and $r_{i1} > r_{i2} > \cdots > r_{iq_i}$, is

$$\mathrm{AO}(\|s_{ij}\|, \|r_{ij}\|, \vec{n}) = \sum_{\mathbf{H} \geqq \mathbf{S}} \sum_{\substack{\vec{b} \leqq \vec{d} \leqq \vec{H} \\ 0 \leqq \vec{\delta} \leqq \vec{d}}} (-1)^{\varphi(\vec{d}) + \varepsilon(\vec{\delta})} \left(\sum_{i=1}^{p} (d_i + t_i) \right)! \prod_{i=1}^{p} \left(\frac{H^{(i)}!}{d_i! t_i!} \binom{H^{(i)} - 1}{H^{(i)} - d_i} \binom{d_i}{\delta_i} \prod_{j=1}^{q_i} \left(\frac{\Delta(h_{ij}, S_j^{(i)} - H_{j-1}^{(i)})}{(h_{ij})!} \right) \right),$$

where $S_j^{(i)}$ and $H_j^{(i)}$ are partial sums (e.g., $S_j^{(i)} = \sum_{\nu=1}^{j} s_{i\nu}$, $S_0^{(i)} = 0$); $\mathbf{H} \geqq \mathbf{S} \Leftrightarrow H_j^{(i)} \geqq S_j^{(i)}$ for all i and j; $H^{(i)} = \sum_{j=1}^{q} h_{ij}$ is the total number of elementary chunks of letter type L_i; $\varepsilon(\vec{\delta}) := \sum_{i=1}^{p} \delta_i$ is the total number of extended chunks (connected extension sites); $\varphi(\vec{d}) = \sum_{i=1}^{p} (H^{(i)} - d_i)$ is the total number of connected fusion sites; $\vec{b}$ is the vector whose ith component is 1 if $H^{(i)} \geqq 1$ and 0 if $H^{(i)} = 0$; and $t_i = n_i - \delta_i - \sum_{j=1}^{q_i} h_{ij} r_{ij}$ is the number of remaining letters of type L_i. The function Δ is defined in (1).

Proof. Define

$$\sum_{h_{ij}} f(i,j) := \sum_{h_{ij} = S_j^{(i)} - H_{j-1}^{(i)}}^{M_j^{(i)} - H_{j-1}^{(i)}} \Delta(h_{ij}, S_j^{(i)} - H_{j-1}^{(i)}) \binom{M_j^{(i)} - H_{j-1}^{(i)}}{h_{ij}}.$$

We know from (1) that $\sum_{h_{ij}} f(i,j) = 1$. Using this, along with (12) and the multidimensional equivalent to (7), $\mathrm{AO}(\|s_{ij}\|, \|r_{ij}\|, \vec{n}) = \sum_{\mathbf{M} \geqq \mathbf{S}} G_{\|m_{ij}\|}$, we obtain

$$\begin{aligned} &\mathrm{AO}(\|s_{ij}\|, \|r_{ij}\|, \vec{n}) \\ &\quad = \sum_{\mathbf{M} \geqq \mathbf{S}} 1 \cdot G_{\|m_{ij}\|} \\ &\quad = \sum_{\mathbf{M} \geqq \mathbf{S}} \left[\sum_{h_{11}} \left(f(1,1) \sum_{h_{12}} \left(f(1,2) \cdots \sum_{h_{pq_{p-1}}} \left(f(p, q_{p-1}) \sum_{h_{pq_p}} f(p, q_p) \right) \cdots \right) \right) \right] G_{\|m_{ij}\|} \\ &\quad = \sum_{\mathbf{M} \geqq \mathbf{S}} \left[\sum_{\mathbf{M} \geqq \mathbf{H} \geqq \mathbf{S}} \prod_{i=1}^{p} \prod_{j=1}^{q_i} \Delta(h_{ij}, S_j^{(i)} - H_{j-1}^{(i)}) \binom{M_j^{(i)} - H_{j-1}^{(i)}}{h_{ij}} \right] G_{\|m_{ij}\|} \end{aligned}$$

$$= \sum_{\mathbf{H} \geqq \mathbf{S}} \left(\prod_{i=1}^{p} \prod_{j=1}^{q_i} \Delta(h_{ij}, S_j^{(i)} - H_{j-1}^{(i)}) \right) \left(\sum_{\mathbf{M} \geqq \mathbf{H}} \prod_{i=1}^{p} \prod_{j=1}^{q_i} \binom{M_j^{(i)} - H_{j-1}^{(i)}}{h_{ij}} G_{\|m_{ij}\|} \right)$$

$$= \sum_{\mathbf{H} \geqq \mathbf{S}} \left(\prod_{i=1}^{p} \prod_{j=1}^{q_i} \Delta(h_{ij}, S_j^{(i)} - H_{j-1}^{(i)}) \right) \cdot \mathrm{DC}(\|h_{ij}\|, \|r_{ij}\|, \vec{n}).$$

Applying (9)–(11) leads to the result.

Remarks. For computational purposes, **H** is bounded above by the constraint $t_i \geqq 0$ for all i (the summand is zero if the chunks require more letters than are available), and the number of terms is additionally limited by the fact that $\Delta = 0$ when $h_{ij} \neq 0$ and $S_j^{(i)} - H_{j-1}^{(i)} \leqq 0$. If there are insufficient letters of type L_i to construct the desired chunks or if there are insufficient elements of other types to separate the desired chunks of type L_i, the formula of Result 3 gives zero, as it should. Note that, if there are no required runs of letter type L_i,

$$\binom{H^{(i)} - 1}{H^{(i)} - d_i} = \binom{-1}{0} = 1.$$

REFERENCES

G. Bateman (1948), *On the power function of the longest run as a test for randomness in a sequence of alternatives*, Biometrika, 35, pp. 97–112.

J. V. Bradley (1968), *Distribution Free Statistical Tests*, Prentice–Hall, Englewood Cliffs, NJ.

V. Brendel and S. Karlin (1989), *Association of charge clusters with functional domains of cellular transcription factors*, Proc. Nat. Acad. Sci., 86, pp. 5698–5702.

P. Deheuvels and L. Devroye (1987), *Limit laws of Erdos-Renyi-Shepp type*, Ann. Probab., 15, pp. 1363–1386.

P. Erdös and A. Rényi (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 a semi-Markov process*, Stochastic Process. Appl., 24, pp. 203–224.

J. Glaz (1989), *Approximations and bounds for the distribution of the scan statistic*, J. Amer. Statist. Assoc., 84, pp. 560–566.

L. J. Guibas and A. M. Odlyzko (1980), *Long repetitive patterns in random sequences*, Z. Wahrsch. Verw. Geb., 53, pp. 241–262.

S. Karlin (1990), *Distribution of clusters of charged amino acid in protein sequences*, in DNA Protein Complexes and Proteins, Vol. 2, Ramaswamy H. Sarma and Mukti H. Sarma, eds., Adenine Press, Schenectady, NY.

S. Karlin, B. E. Blaisdell, and V. Brendel (1990), *Identification of significant sequence patterns in proteins*, Meth. Enzymology, 183, pp. 388–402.

S. Karlin, B. E. Blaisdell, E. S. Mocarski, and V. Brendel (1989), *A method to identify distinctive charge configurations in protein sequences, with application to human herpesvirus polypeptides*, J. Molecular Biol., 205, pp. 165–177.

S. Karlin and C. Macken (1991), *Some statistical problems in the assessment of inhomogeneities of* DNA *sequence data*, J. Amer. Statist. Assoc., 86, pp. 27–35.

M. Y. Leung (1989), *Probabilistic Models and Computational Algorithms for Some Problems from Molecular Sequence Analysis*, Ph.D. thesis, Stanford Univ. Math. Dept., Stanford, CA.

A. M. Mood (1940), *The distribution theory of runs*, Ann. Math. Statist., 11, pp. 367–392.

M. S. Morris (1990), *Mathematical Methods for Molecular Sequence Analysis and Genome Map Assembly*, Ph.D. thesis, Stanford Univ. Math. Dept., Stanford, CA.

J. I. Naus (1979), *An indexed bibliography of clusters, clumps and coincidences*, Internat. Statist. Rev., 47, pp. 47–78.

S. S. Samarova (1981), *On the length of the longest head-run for a Markov chain with two states*, Theory Probab. Appl., 26(3), pp. 498–509.

S. Wallenstein and N. Neff (1987), *An approximation for the distribution of the scan statistic*, Statist. Med., 6, pp. 197–207.

SIAM J. DISC. MATH.
Vol. 6, No. 1, pp. 87-99, February 1993

007

CHARACTERIZATION OF THE HOMOMORPHIC PREIMAGES OF CERTAIN ORIENTED CYCLES*

HUISHAN ZHOU†

Abstract. The classes of digraphs that can be homomorphically mapped to certain oriented cycles are characterized by the forbidden homomorphic preimages. This characterization can be used to prove the membership of the corresponding decision problems in the class NP $\cap$ coNP.

Key words. digraph, basic cycle, homomorphism, homomorphic preimage, NP, coNP

AMS(MOS) subject classification. 05C

1. Introduction. In this paper, we will only be concerned with digraphs with neither loops nor multiple arcs. Let G be a digraph. Then the vertex set and the arc set are denoted by $V(G)$ and $E(G)$, respectively. A *homomorphism* f of a digraph G to a digraph H is a mapping of $V(G)$ to $V(H)$ for which $f(u)f(v) \in E(H)$ whenever $uv \in E(G)$. G is called the *homomorphic preimage* of H under f. If f is onto, then H is called the *homomorphic image* of G under f. The existence, respectively, nonexistence, of a homomorphism from G to H will be denoted by $G \twoheadrightarrow H$, respectively, $G \not\twoheadrightarrow H$.

Homomorphism is a very useful concept. There is an interesting link between language and graph theory by homomorphisms and language interpretations [10], [18]. Many classes of graphs can be described by means of homomorphisms [8], [9], [16], [21], [22], [24], [28]. Therefore, the problem of the existence of graph homomorphisms has attracted considerable attention [1]-[4], [7], [9], [11]-[13], [17], [19], most of which also concentrated on the computational complexity of recognizing the homomorphism of any graph to a fixed-target graph. Another source of interest in homomorphism is Hedetniemi's conjecture [5], [6], [8], [10], [20], [23], which states that the chromatic number of the categorical product of two n-chromatic graphs is n. This led to the definition of a multiplicative directed or undirected graph W [8], [21], [24], [29]. W is multiplicative if the categorical product of two graphs G and H cannot be homomorphically mapped to W whenever neither G nor H can be homomorphically mapped to W. Hedetniemi's conjecture simply states that complete graphs are multiplicative. Some multiplicative or nonmultiplicative graphs and digraphs were given in [8], [21], [22], [24]-[27], [29]. In particular, oriented paths (cycles) have been completely characterized with respect to multiplicativity in [25] (see also [14], [15], [27]). A major step in proving the multiplicativity or nonmultiplicativity of a graph W is to analyze the forbidden homomorphic primages of W, either completely in the case of multiplicativity, or at least partially in the case of nonmultiplicativity.

In [28] we listed the graphs we knew at that point, the homomorphic preimages of which can be characterized by the forbidden homomorphic preimages. We also characterized the classes of graphs that can be homomorphically mapped to certain special basic paths by the forbidden homomorphic preimages.

For further consideration, we are naturally concerned about oriented cycles. The simplest oriented cycle (except directed cycles) in this regard are basic cycles. A *basic cycle* [15], [26] is a closed cycle generated by concatenating some number of directed paths one by one, alternatively, forward and backward, with all the backward-directed

* Received by the editors July 8, 1991; accepted for publication (in revised form) October 18, 1991.

† Department of Mathematics and Computer Sciences, Georgia State University, University Plaza, Atlanta, Georgia 30303-3083.

paths having a fixed length. The smallest basic cycle is an oriented cycle of three arcs, with two arcs directed forward and one arc directed backward, which is also the smallest transitive tournament. The class of digraphs that can be homomorphically mapped to it is characterized by the digraphs with the directed path of length 3 as the forbidden homomorphic preimage [8]. In this paper, we consider two kinds of basic cycles "next" to the above-mentioned smallest basic cycle, i.e., $C_{2,1,2,1}$ and $C_{3,1}$ as illustrated in Figs. 1 and 12, state the results for them, and give the proof only for $C_{2,1,2,1}$. Surprisingly, the proof is not easy.

The *directed path* $\vec{P}_n$ has a sequence of different vertices $v_0, v_1, \cdots, v_n$, and arcs $v_0v_1, v_1v_2, \cdots, v_{n-1}v_n$. An *oriented path* $P[x_0, x_1, \cdots, x_n]$ has a sequence of distinct vertices $x_0, x_1, \cdots, x_n$ and arcs $(x_0, x_1), (x_1, x_2), \cdots, (x_{n-1}, x_n)$, where (x_i, x_{i+1}) denotes either x_ix_{i+1} or $x_{i+1}x_i$. We often use the same notation $P[x_0, x_1, \cdots, x_n]$ to denote the vertex set of $P[x_0, x_1, \cdots, x_n]$. In the notation $P[x_0, x_1, \cdots, x_n]$, we usually take the order of traversal from x_0 to x_n. The subpath of an oriented path $P[x_0, x_1, \cdots, x_n]$ induced by the vertices $x_i, x_{i+1}, \cdots, x_{j-1}$, and x_j is denoted by $P[x_i, x_{i+1}, \cdots, x_{j-1}, x_j]$ or simply $P[x_i, x_j]$. Let P_1 and P_2 be two oriented paths with the specified orders of traversal. Then the *concatenation* of P_1 and P_2, denoted by P_1P_2 (or P_1vP_2), is the oriented path obtained by identifying the last vertex of P_1 and the first vertex of P_2 (the vertex v is used to denote both the last vertex of P_1 and the first vertex of P_2). An oriented cycle C is an oriented path P with the first vertex and the last vertex identified. We denote $C = vPv$, where v is the first as well as the last vertex of P. Thus $\vec{C}_n = v\vec{P}_nv$ is a directed cycle of length n.

The *level* of a vertex x in an oriented path P (or cycle) with respect to a chosen vertex a of P, denoted by $L_{P,a}(x)$ (or simply $L(x)$ with $L(a) \equiv 0$ if no confusion will result) is the difference between the number of edges directed forward and the number of edges directed backward on the subpath from the chosen vertex a to x. In the case of an oriented cycle, we should also specify the order of traversal.

2. Characterization of the homomorphic preimages of $C_{2,1,2,1}$.

Let $C_{2,1,2,1}$ be the oriented cycle given in Fig. 1.

Let $\mathscr{D}$ be the class of all digraphs, $\mathscr{C}$ the class of all oriented cycles, and $\mathscr{P}$ the class of all oriented paths.

Let θ_1 and θ be the classes of digraphs defined as follows:

$$\theta_1 = \{C \in \mathscr{C}\colon C \text{ contains no } \vec{P}_3, C \text{ contains odd number of } \vec{P}_2\text{'s}\},$$

$$\theta = \{\vec{P}_3\} \cup \theta_1.$$

Then we can characterize the homomorphic preimages of $C_{2,1,2,1}$ as stated in the following theorem.

THEOREM 2.1. *For any digraph G, the following statements are equivalent:*

(1) *G can be homomorphically mapped to $C_{2,1,2,1}$;*

(2) *for any digraph P in θ, P cannot be homomorphically mapped to G;*

and

(3) *G contains no subgraph that is isomorphic to $\vec{C}_2$, $\vec{C}_3$, or any digraph P in θ.*

Remark. Equivalently, we can state this theorem as follows:

$$\{G \in \mathscr{D}\colon G \rightarrow C_{2,1,2,1}\} = \{G \in \mathscr{D}\colon \text{for any } P \in \theta, P \not\rightarrow G\}$$
$$= \{G \in \mathscr{D}\colon G \text{ contains no subgraph}$$
$$\text{that is isomorphic to } \vec{C}_2, \vec{C}_3 \text{ or any digraph } P \text{ in } \theta\}; \qquad \text{or}$$

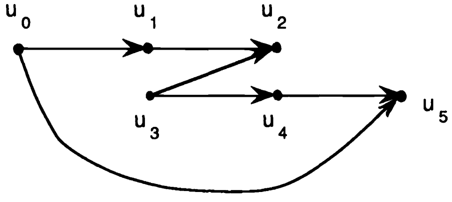


FIG. 1

$$\begin{aligned}\{G \in \mathcal{D}\colon G \not\to C_{2,1,2,1}\} &= \{G \in \mathcal{D}\colon \text{for some } P \in \theta,\ P \to G\}\\ &= \{G \in \mathcal{D}\colon G \text{ contains some subgraph}\\ &\qquad \text{that is isomorphic to } \vec{C}_2,\ \vec{C}_3 \text{ or a digraph } P \text{ in } \theta\}.\end{aligned}$$

In proving the nonmultiplicativity of $C_{2,1,2,1}$ [26], we did not perform the exhaustive search of the complete obstructions of $C_{2,1,2,1}$ as stated above. We simply chose $G = \overrightarrow{P}_3$ and $H = v\overrightarrow{P}_2\overleftarrow{P}_1 v$ from θ, then $G \not\to C_{2,1,2,1}$, $H \not\to C_{2,1,2,1}$, but $G \times H \to C_{2,1,2,1}$.

Before giving the proof, we introduce the notion of cluster and some related results.

An oriented path P is called a *cluster* if

(1) $\max\{L(v)\colon v \in P\} - \min\{L(v)\colon v \in P\} = 2$; and

(2) P starts and ends with a $\overrightarrow{P}_2$.

In general, a cluster can have the following forms:

(a) $\overrightarrow{P}_2 P^{e1} \overleftarrow{P}_2 P^{e2} \overrightarrow{P}_2 \cdots P^{e(2k-1)} \overleftarrow{P}_2 P^{e(2k)} \overrightarrow{P}_2$ (for an illustration, see $P[b, l]$ in Fig. 2);

(b) $\overleftarrow{P}_2 P^{e1} \overrightarrow{P}_2 P^{e2} \overleftarrow{P}_2 \cdots P^{e(2k-1)} \overrightarrow{P}_2 P^{e(2k)} \overleftarrow{P}_2$ (for an illustration, see $P[f, i]$ in Fig. 2);

(c) $\overrightarrow{P}_2 P^{e1} \overleftarrow{P}_2 P^{e2} \overrightarrow{P}_2 \cdots P_2 P^{e(2k-1)} \overleftarrow{P}_2$ (for an illustration, see $P[b, i]$ in Fig. 2); and

(d) $\overleftarrow{P}_2 P^{e1} \overrightarrow{P}_2 P^{e2} \overleftarrow{P}_2 \cdots \overleftarrow{P}_2 P^{e(2k-1)} \overrightarrow{P}_2$ (for an illustration, see $P[f, l]$ in Fig. 2),

where each P^{ei} ($i = 1, 2, \cdots, 2k - 1$ or $2k$) is an oriented path with an even number of alternating backward and forward (or forward and backward) arcs (may be empty). In the following, P^e and P^{ei} ($i = 1, 2, \cdots$) always represent oriented paths consisting of an even number of alternating backward and forward (or forward and backward) arcs; P^o and P^{oi} ($i = 1, 2, \cdots$) always represent oriented paths consisting of an odd number of alternating backward and forward (or forward and backward) arcs. Usually, we use $\overrightarrow{P}_n$ to denote a directed path of length n and do not care about the order of traversal. However, when we want to specify a directed path of length n as a subpath of an oriented path in the representation of concatenation, we use $\overrightarrow{P}_n$ to denote a directed path of length n directed forward, and $\overleftarrow{P}_n$ to denote a directed path of length n directed backward, as we have seen in (a)–(d) above.

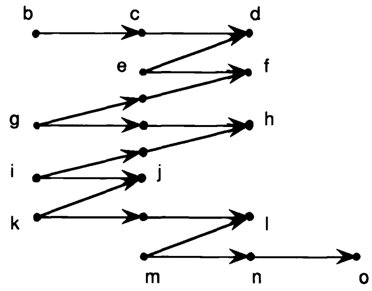


FIG. 2

A subpath Q of an oriented path P (or an oriented cycle C) is called a *maximal cluster* in P (or C) if Q is a cluster and if Q is not contained in a cluster with more vertices.

In Fig. 2, $P[d, k]$ is not a cluster; $P[b, c, d]$ and $P[f, l]$ are clusters, but not maximal clusters. $P[b, l]$ and $P[m, o]$ are maximal clusters in the oriented path $P[b, o]$. For our convenience, when we speak of a cluster in an oriented path (or an oriented cycle) we always mean a maximal cluster, unless otherwise specified. We say that the cluster $P[b, l]$ is neighboring to the cluster $P[m, o]$ with joining arc ml. In general, when we say that two different clusters A and B are *neighboring clusters* in an oriented path P, we mean that there are no clusters between A and B. Obviously, we have the following observation.

LEMMA 2.2. *Let A and B be two neighboring clusters in an oriented path P with no $\vec{P}_3$. Then there is an odd number of alternating forward and backward (or backward and forward) arcs between A and B.*

The oriented path of an odd number of alternating forward and backward (or backward and forward) arcs between A and B is called the *connection* of the two clusters A and B.

We also make the following convention that the oriented cycle

$$C = v\vec{P}_2P^{e1}\overleftarrow{P}_2P^{e2}\vec{P}_2 \cdots \vec{P}_2P^{e(2k-1)}\overleftarrow{P}_2P^{e(2k)}v,$$

contains *zero clusters*. Note that

$$C = v\vec{P}_2P^{e1}\overleftarrow{P}_2P^{e2}\vec{P}_2 \cdots \vec{P}_2P^{o(2k-1)}v$$

contains one cluster. See Fig. 3 for oriented cycles containing zero, one, and two clusters. The reader may note that the digraph in Fig. 3(a) is not a core; the digraphs in Figs. 3(b) and 3(c) are cores. *A core is a (di)graph that cannot be homomorphically mapped to its proper subgraph.*

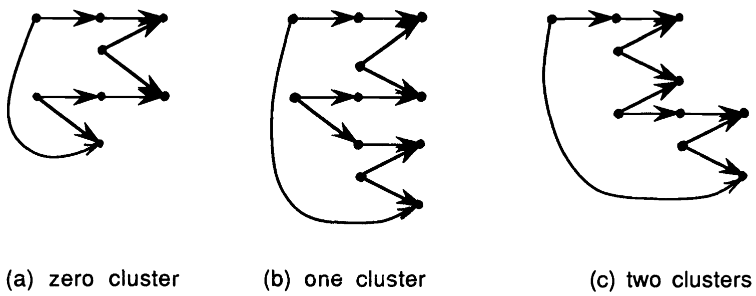
LEMMA 2.3. *Let $C \neq \vec{C}_2, \vec{C}_3$ be an oriented cycle containing no $\vec{P}_3$. Assume that C contains n $\vec{P}_2$'s and k clusters. Then n and k have the same parity.*

Proof. We prove the lemma by induction on n. If $n = 1$, then $C = v\vec{P}_2P^ov$. Obviously, C has one cluster. If $n = 2$, then either $C = v\vec{P}_2P^{e1}\overleftarrow{P}_2P^{e2}v$, which has zero clusters, or $C = v\vec{P}_2P^{o1}\vec{P}_2P^{o2}v$, which has two clusters. Now suppose that the lemma is true for smaller n. We prove that the lemma is true for n according to the following three cases.

Case 1. There exists a cluster B in C such that B contains more than two $\vec{P}_2$'s,

$$C = vC_1BC_2v \quad \text{and} \quad B = B_1\vec{P}_2P^e\overleftarrow{P}_2B_2 \quad (\text{or } B = B_1\overleftarrow{P}_2P^e\vec{P}_2B_2).$$

Then, by deleting $\vec{P}_2P^e\overleftarrow{P}_2$ (or $\overleftarrow{P}_2P^e\vec{P}_2$) from B and identifying the two end vertices in C,



(a) zero cluster (b) one cluster (c) two clusters

FIG. 3

we obtain that

$$C^* = vC_1B^*C_2v \quad \text{and} \quad B^* = B_1B_2 \quad (B^* \neq \emptyset \text{ is a cluster}).$$

C^* has k clusters, which is the same number as C has, and has $n - 2$ $\vec{P}_2$'s, which is two less than the number that C has. By the induction hypothesis, $n - 2$ and k have the same parity, so n and k have the same parity.

Case 2. There exists a cluster B in C such that B contains exactly two $\vec{P}_2$'s,

$$C = vC_1P^{o1}BP^{o3}C_2v \quad \text{and} \quad B = \vec{P}_2P^e\overleftarrow{P}_2 \quad (\text{or } B = \overleftarrow{P}_2P^e\vec{P}_2).$$

Then, by deleting all of B and identifying the two end vertices in C, we obtain that $C^* = vC_1P^{o1}P^{o3}C_2v$, where $P^{o1}P^{o3}$ is an oriented path with an even number of alternating backward and forward (or forward and backward) arcs. If C_1 and C_2 are in one cluster, then C^* contains zero clusters and C contains two clusters. Since C^* contains $n - 2$ $\vec{P}_2$'s, which is even by the induction hypothesis, C contains even (n) $\vec{P}_2$'s. If C_1 and C_2 are not in one cluster, then the last cluster of C_1 and the first cluster of C_2 are distinct; they will be combined to become one cluster in C^*. C^* has $n - 2$ $\vec{P}_2$'s and $k - 2$ clusters. Applying the induction hypothesis to C^*, n and k have the same parity.

Case 3. Every cluster in C contains exactly one $\vec{P}_2$; then all $\vec{P}_2$ are in one direction along the order of traversal of C. We have that

$$C = vC_1BC_2v \quad \text{and} \quad B = P^{o1}\vec{P}_2P^{o2}\vec{P}_2P^{o3} \quad (\text{or } B = P^{o1}\overleftarrow{P}_2P^{o2}\overleftarrow{P}_2P^{o3}).$$

By deleting $\vec{P}_2P^{o2}\vec{P}_2P^{o3}$ (or $\overleftarrow{P}_2P^{o2}\overleftarrow{P}_2P^{o3}$) from B and identifying the two end vertices in C, we obtain $C^* = vC_1P^{o1}C_2v$. C^* will have $n - 2$ $\vec{P}_2$'s and $k - 2$ clusters. Applying the induction hypothesis to C^*, n and k have the same parity.

LEMMA 2.4. *Let $C \neq \vec{C}_2, \vec{C}_3$ be an oriented cycle containing no $\vec{P}_3$. Then $C \to C_{2,1,2,1}$ if and only if C contains an even number of $\vec{P}_2$'s.*

Proof. If C contains no $\vec{P}_2$, then $C \to \vec{P}_1$; hence $C \to C_{2,1,2,1}$. If C contains some $\vec{P}_2$'s, but C contains zero clusters, then $C \to \vec{P}_2$; hence $C \to C_{2,1,2,1}$. (This is the important reason why we define $C = v\vec{P}_2P^{e1}\overleftarrow{P}_2P^{e2}\vec{P}_2 \cdots \vec{P}_2P^{e(2k-1)}\overleftarrow{P}_2P^{e(2k)}v$ to have zero clusters.) If C contains one cluster, then $C \not\to C_{2,1,2,1}$.

Next, we consider the general case. Let us call the subpath $[u_0, u_1, u_2]$ *back* and the subpath $[u_3, u_4, u_5]$ *front* in the oriented cycle $C_{2,1,2,1}$. When we try to map an oriented cycle C to $C_{2,1,2,1}$, any subpath $\vec{P}_2$ of C should either be mapped to the back or to the front; accordingly, all the $\vec{P}_2$'s in one cluster should be mapped either to the back or to the front. It is impossible that some $\vec{P}_2$'s of the cluster are mapped to the front, and some $\vec{P}_2$'s of the same cluster are mapped to the back. A cluster is said to be mapped to the front (back) if all the subpaths $\vec{P}_2$ of that cluster are mapped to the front (back). If a cluster is mapped to the front (back), then the neighboring clusters must be mapped to the back (front). Therefore, $C \to C_{2,1,2,1}$ if and only if C contains an even number of clusters if and only if C contains an even number of $\vec{P}_2$'s. □

Proof of Theorem 2.1. Let $P \in \theta$. Obviously, $P \not\to C_{2,1,2,1}$ for $P = \vec{P}_3$. If $P \in \theta_1$, then $P \not\to C_{2,1,2,1}$ by Lemma 2.4.

(1) *implies* (2). Let $G \in \mathscr{D}$ be such that $G \to C_{2,1,2,1}$. We prove that for any $P \in \theta$, $P \not\to G$. Suppose, on the contrary, that there exists $P \in \theta$, $P \to G$. Then the same $P \in \theta$ will satisfy $P \to C_{2,1,2,1}$ by a composite mapping, a contradiction.

(2) *implies* (3). The proof is obvious.

(3) *implies* (1). Let $G \in \mathscr{D}$ be such that G contains no subgraph that is isomorphic to $\vec{C}_2$, $\vec{C}_3$ or any digraphs P in θ. We prove that $G \to C_{2,1,2,1}$. Obviously, G contains no $\vec{P}_3$, and for any oriented cycle C (C contains no $\vec{P}_3$ either) in G, $C \neq \vec{C}_2$, $C \neq \vec{C}_3$, C contains an even number of $\vec{P}_2$'s. Hence any oriented cycle in G can be mapped to $C_{2,1,2,1}$ by Lemma 2.4.

If G contains no $\vec{P}_2$, then any vertex in G is either a source or a sink; hence we can homomorphically map G to $C_{2,1,2,1}$ by mapping all the sources to u_0 and all the sinks to u_1.

If G contains only one $\vec{P}_2$, say $\vec{P}_2 = [a_0, a_1, a_2]$, then $G \setminus \{a_1\}$ has two components G_1 and G_2, where G_1 contains a_0 and G_2 contains a_2. Otherwise, there will exist an oriented cycle containing one $\vec{P}_2$, a contradiction. Thus we can map a_i to u_i for $i = 0$, 1, 2; map sources in G_1 to u_0, sinks in G_1 to u_1 (or u_5), sinks in G_2 to u_2, and sources in G_2 to u_1 (or u_3). It is easy to see that this map is a homomorphic map.

Now we assume that G contains more than one $\vec{P}_2$. For any vertex $v \in V(G)$, let

$$O(v) = \{x \in V(G)\colon vx \in E(G)\} \text{ and } I(v) = \{x \in V(G)\colon xv \in E(G)\}.$$

Obviously, for any vertex v such that $O(v), I(v) \neq \emptyset$, any vertex in $O(v)$ is a sink and any vertex in $I(v)$ is a source, since there is no $\vec{P}_3$ in G. There is no arc between the vertices of $O(v) \cup I(v)$ since there is neither $\vec{C}_3$ nor cycle containing one $\vec{P}_2$.

Now we construct the graph G^* from G by the following subgraph replacement operation. For any $v \in V(G)$ with neither $O(v) = \emptyset$ nor $I(v) = \emptyset$, let $I(v) = \{x_1, x_2, \cdots, x_{|I(v)|}\}$ and $O(v) = \{z_1, z_2, \cdots, z_{|O(v)|}\}$. Let the subgraph induced by $\{v\} \cup O(v) \cup I(v)$ be replaced by the following subgraph B_v (see Fig. 4 for the illustration):

$$V(B_v) = O(v) \cup I(v) \cup \{y_{11}, y_{12}, \cdots, y_{1|O(v)|}; y_{21}, y_{22}, \cdots, y_{2|O(v)|}; \cdots$$
$$y_{|I(v)|1}, y_{|I(v)|2}, \cdots, y_{|I(v)||O(v)|}\};$$

$$E(B_v) = \{x_1y_{11}, y_{11}z_1, x_1y_{12}, y_{12}z_2, \cdots, x_1y_{1|O(v)|}, y_{1|O(v)|}z_{|O(v)|}, x_2y_{21}, y_{21}z_1, x_2y_{22},$$
$$y_{22}z_2, \cdots, x_2y_{2|O(v)|}, y_{2|O(v)|}z_{|O(v)|}\,x_{|I(v)|}, y_{|I(v)|1}, y_{|I(v)|1}z_1,$$
$$x_{|I(v)|}y_{|I(v)|2}, y_{|I(v)|2}z_2, \cdots, x_{|I(v)|}y_{|I(v)||O(v)|}, y_{|I(v)||O(v)|}z_{|O(v)|}\}.$$

Obviously, we have the following observations.

FACT 1. *$G \rightarrow C_{2,1,2,1}$ if and only if $G^* \rightarrow C_{2,1,2,1}$.*

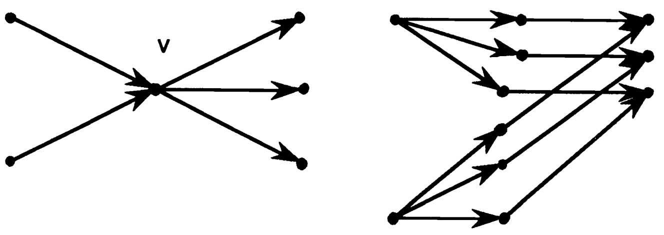
FACT 2. *Any oriented cycle in G^* contains an even number of $\vec{P}_2$'s, as well as an even number of clusters.*

FACT 3. *For any vertex $v \in V(G^*)$, if v is neither a source nor a sink, then $|O(v)| = |I(v)| = 1$.*

We now prove that $G^* \rightarrow C_{2,1,2,1}$. Take one $\vec{P}_2$ from G^*, say $\vec{P}_2 = [a_0, a_1, a_2]$. We map $[a_0, a_1, a_2]$ to the back, i.e., $f(a_i) = u_i$ $(i = 0, 1, 2)$.

For any $x \in V(G^*)$, there exists an oriented path P connecting x with $[a_0, a_1, a_2]$. P is either

$$P = C_1J_1C_2J_2 \cdots J_{k-1}C_k \quad \text{or} \quad P = C_1J_1C_2J_2 \cdots J_{k-1}C_kJ_k,$$



(a) v and its neighborhood (b) B_v

FIG. 4

where C_i ($i = 1, 2, \cdots, k$) are clusters, J_i ($i = 1, 2, \cdots, k-1$) are connections between two clusters. $[a_0, a_1, a_2]$ is contained in C_1, since $|O(a_1)| = |I(a_1)| = 1$ in G^* by Fact 3, and x is the end vertex of C_k (in the first case) or the end vertex of J_k (in the second case). J_k is an oriented path consisting of alternating forward and backward (or backward and forward) arcs. Let L be the level function of C_k (or C_kJ_k) with $0 \leqq L \leqq 2$. Now we define the following map (which depends on P).

If k is odd, then

$$f_P(x) = \begin{cases} u_0 & \text{if } L(x) = 0 \text{ and } x \text{ is a source,} \\ u_4 & \text{if } L(x) = 0 \text{ and } x \text{ is not a source,} \\ u_2 & \text{if } L(x) = 2 \text{ and } x \text{ is a sink,} \\ u_4 & \text{if } L(x) = 2 \text{ and } x \text{ is not a sink,} \\ u_3 & \text{if } L(x) = 1 \text{ and } x \text{ is a source,} \\ u_5 & \text{if } L(x) = 1 \text{ and } x \text{ is a sink,} \\ u_1 & \text{if } L(x) = 1 \text{ and } x \text{ is neither a source nor a sink,} \end{cases}$$

where when we say that x is a source (sink), x is a source (sink) in G^* (the same remark holds for k being even). If k is even, then

$$f_P(x) = \begin{cases} u_3 & \text{if } L(x) = 0 \text{ and } x \text{ is a source,} \\ u_1 & \text{if } L(x) = 0 \text{ and } x \text{ is not a source,} \\ u_5 & \text{if } L(x) = 2 \text{ and } x \text{ is a sink,} \\ u_1 & \text{if } L(x) = 2 \text{ and } x \text{ is not a sink,} \\ u_0 & \text{if } L(x) = 1 \text{ and } x \text{ is a source,} \\ u_2 & \text{if } L(x) = 1 \text{ and } x \text{ is a sink,} \\ u_4 & \text{if } L(x) = 1 \text{ and } x \text{ is neither a source nor a sink;} \end{cases}$$

see Figs. 5 and 6 for the illustrations.

In Fig. 5, we have that

i	0	1	2	3	4	5	6	7	8	9	10	11
$f_P(x_i)$	u_3	u_2	u_1	u_0	u_5	u_0	u_1	u_2	u_3	u_2	u_3	u_4

.

In Fig. 6, we have that

i	0	1	2	3	4	5	6	7	8	9	10
$f_P(x_i)$	u_3	u_4	u_5	u_0	u_5	u_4	u_3	u_2	u_3	u_2	u_1

.

If we can prove that, for arbitrary two oriented paths P_1 and P_2 from $[a_0, a_1, a_2]$ to x, $f_{P_1}(x) = f_{P_2}(x)$, then we have uniquely defined a map from G^* to $C_{2,1,2,1}$.

Let P_1 and P_2 be two oriented paths from $[a_0, a_1, a_2]$ to x. We prove $f_{P_1}(x) = f_{P_2}(x)$ by the induction on the number of common parts shared by P_1 and P_2. Suppose first that P_1 and P_2 have two common parts: one part is the singleton x, and the other

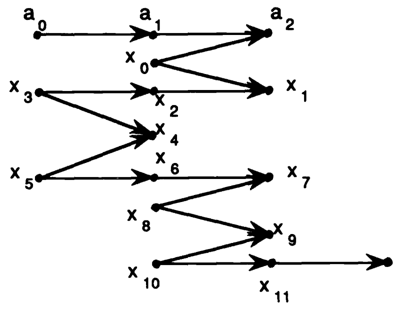


FIG. 5

part is an oriented path containing $[a_0, a_1, a_2]$. Then P_1 and P_2 will form an oriented cycle C in three possible ways, as described in Fig. 7.

We count the number of clusters starting from the cluster to which $[a_0, a_1, a_2]$ belongs. For each cluster C^* in the cycle C, we have two numbers n_1 and n_2; n_i is the number of clusters that is obtained by counting along the P_i side first $(i = 1, 2)$.

CLAIM. *n_1 and n_2 have the same parity.*

Proof. For the case in Fig. 7(a), in which $[a_0, a_1, a_2]$ is part of the cycle C, the proof is obvious, since C contains an even number of clusters by Fact 2. For the case in Fig. 7(b), $[a_0, a_1, a_2]$ is not part of the cycle. We write

$$P_1 = [a_0, a_1, a_2]Q_0A_0B_0yB_1A_1Q_1x$$

and

$$P_2 = [a_0, a_1, a_2]Q_0A_0B_0yB_2A_2Q_2x,$$

where each B_i $(i = 0, 1, 2)$ is the oriented path consisting of alternating backward and forward arcs (may be empty), each A_i $(i = 0, 1, 2)$ is a $\vec{P}_2$, each Q_i $(i = 0, 1, 2)$ is an oriented path (may be empty), and y is either a source or a sink in P_i $(i = 1, 2)$. Let β_i be the number of arcs in B_i (for $i = 0, 1, 2$). Then

(1) A_1 and A_2 are in the same cluster along $A_1B_1yB_2A_2$
$\Leftrightarrow \beta_1 + \beta_2 =$ even
$\Leftrightarrow \beta_1$ and β_2 have the same parity.

Under this circumstance, by the parity argument, we have
A_0 and A_1 are in the same (different) cluster(s) along P_1
$\Leftrightarrow \beta_0$ and β_1 have the same (different) parity

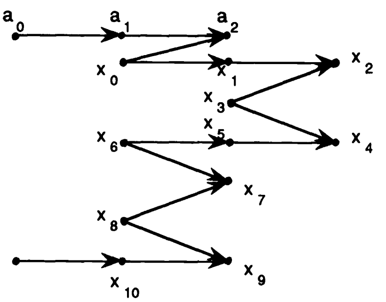
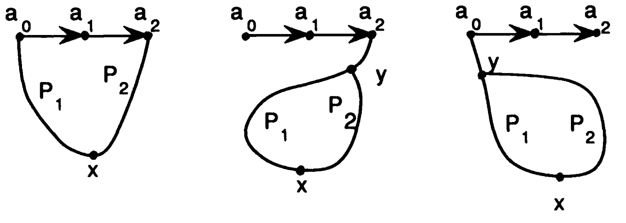


FIG. 6



(a) (b) (c)

FIG. 7

$\Leftrightarrow$ β_0 and β_2 have the same (different) parity

$\Leftrightarrow$ A_0 and A_2 are in the same (different) cluster along P_2,

which is consistent with the following direct argument: If A_1 and A_2 are in the same cluster, and if A_0 and A_1 are in the same (different) cluster(s), then A_0 and A_2 are in the same (different) cluster(s). This also implies that $n_1 = n_2$ by applying the fact that there are an even number of clusters in C.

(2) A_1 and A_1 are in different clusters along $A_1B_1yB_2A_2$

$\Leftrightarrow$ $\beta_1 + \beta_2 =$ odd

$\Leftrightarrow$ β_1 and β_2 have different parity.

Under this circumstance, by the parity argument, we have

A_0 and A_1 are in the same (different) cluster(s) along P_1

$\Leftrightarrow$ β_0 and β_1 have the same (different) parity

$\Leftrightarrow$ β_0 and β_2 have different (the same) parity

$\Leftrightarrow$ A_0 and A_2 are in different clusters (the same cluster) along P_2,

which is consistent with the following direct argument: If A_1 and A_2 are in different clusters, and if A_0 and A_1 are in the same (different) cluster(s), then A_0 and A_2 are in different clusters (the same cluster). This again implies that $n_1 = n_2$.

Similar analysis can be applied to Fig. 3(c). The proof of the claim is completed.

Assume first that x is in a cluster C_x of C. If C_x is counted in the n_1th cluster along P_1, and the n_2th cluster along P_2, then n_1 and n_2 have the same parity as proved above. Therefore, $f_{P_1}(x) = f_{P_2}(x)$. Assume now that x is not in a cluster of C. Then x is in a connection that connects two clusters C_1 of P_1 and C_2 of P_2. If C_1 is the n_1th cluster in P_1, and C_2 is the n_2th cluster in P_2, then n_1 and n_2 have different parity. Assume that n_1 is odd and n_2 is even; see Fig. 8(a) for the illustration. We have, by the definition of f_{P_i} $(i = 1, 2)$,

$$f_{P_1}(x) = f_{P_2}(x) = u_3 \quad \text{if } x \text{ is a source,}$$

$$f_{P_1}(x) = f_{P_2}(x) = u_1 \quad \text{if } x \text{ is not a source,}$$

$$f_{P_1}(y) = f_{P_2}(y) = u_2 \quad \text{if } y \text{ is a sink,}$$

$$f_{P_1}(y) = f_{P_2}(y) = u_4 \quad \text{if } y \text{ is not a sink.}$$

Therefore, $f_{P_1}(x) = f_{P_2}(x)$ for any vertex x in the connection between C_1 and C_2 for the case illustrated in Fig. 8(a). Similarly, we can check that $f_{P_1}(x) = f_{P_2}(x)$ is true by the definition of f_{P_i} $(i = 1, 2)$ for the remaining three cases: In Fig. 8(b), n_1 is odd and n_2 is even, as is the case in Fig. 8(a), but the relative location of the connection with respect to C_1 and C_2 is changed. In Figs. 8(c) and 8(d), n_1 is even and n_2 is odd. In Fig. 8, a rectangle represents a cluster.

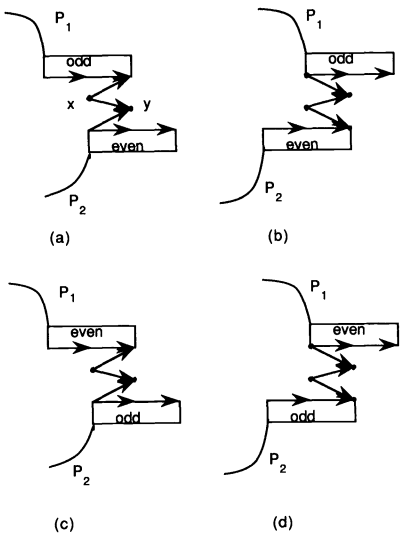


FIG. 8

We still assume that P_1 and P_2 have two common parts but the part containing x is not a singleton; see Fig. 9. We use the same analysis for z as we used for y in Fig. 7(b). The location of $\vec{P}_2$'s around z with respect to the odd or even number of clusters must be consistent without contradiction. We again obtain that $f_{P_1}(x) = f_{P_2}(x)$.

Now assume that P_1 and P_2 are two oriented paths connecting $[a_0, a_1, a_2]$ and y; P_3 and P_4 are two oriented paths connecting y and x; P_i $(i = 1, 2)$ have no common vertex except $[a_0, a_1, a_2]$ and y at their two ends; P_i $(i = 3, 4)$ have no common vertex except y and x at their two ends; and P_i $(i = 1, 2)$ and P_j $(j = 3, 4)$ have one common vertex y at their end. We prove that $f_{P_1P_3}(x) = f_{P_2P_4}(x)$.

By Fact 3, y must be either a source or a sink. Let

$$P_1 = \cdots A_1B_1y, \quad P_2 = \cdots A_2B_2y, \quad P_3 = yB_3A_3\cdots, \quad P_4 = yB_4A_4\cdots,$$

where each A_i $(i = 1, 2, 3, 4)$ is a $\vec{P}_2$ and each B_i $(i = 1, 2, 3, 4)$ is an oriented path consisting of alternating forward and backward (or backward and forward) arcs. Let β_i be the number of arcs in B_i $(i = 1, 2, 3, 4)$. If β_1 and β_2 are odd, then A_1 and A_2 are in one cluster. If β_3 is odd, then A_1 and A_3 are in one cluster along the path $A_1B_1yB_3A_3$ and A_2 and A_3 are also in one cluster along the path $A_2B_2yB_3A_3$. If β_4 is odd, then A_1 and A_4 are in one cluster along the path $A_1B_1yB_4A_4$; A_2 and A_4 are also in one cluster along the path $A_2B_2yB_4A_4$; and A_3 and A_4 are also in one cluster along the path $A_3B_3yB_4A_4$. This case is illustrated in Fig. 10(a). The detailed practical illustrations are in Figs. 10(b) and 10(c). There are altogether $2^4 = 16$ cases. Four other cases are illustrated in Fig. 11. The

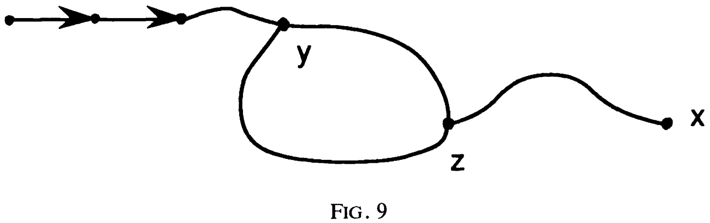
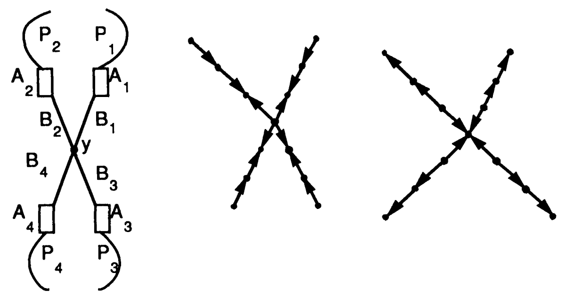


FIG. 9



(a) (b) (c)

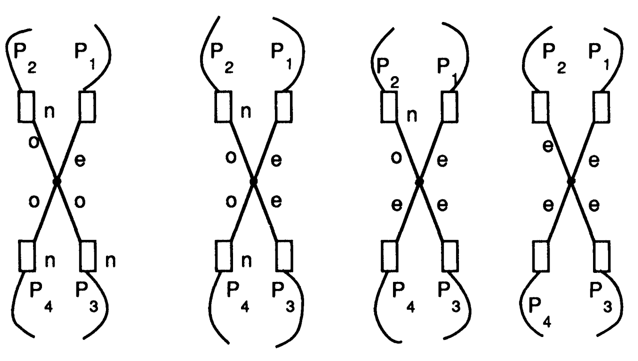
FIG. 10

notation is explained as follows: Each rectangle represents a $\vec{P}_2$. Each straight line represents an oriented path consisting of alternating forward and backward (or backward and forward) arcs, the letter o (respectively, e) near the line means that the number of arcs in this oriented path is odd (respectively, even). Let A_1 be in the specified cluster. Then the letter n near the rectangle means that the $\vec{P}_2$ represented by the rectangle is in the neighbouring cluster. Otherwise, they are in the same cluster as A_1. Therefore, everything is consistent here. We will never have the following problem: A_3 is in the same cluster as A_1 along $A_1B_1yB_3A_3$, but in the neighbouring cluster when analysed along $A_1B_1yB_2A_2$ followed by along $A_2B_2yB_3A_3$. If A_1 is mapped to the back (front), then A_i ($i = 2, 3, 4$), marked n, must be mapped to the front (back); otherwise, A_i ($i = 2, 3, 4$) is mapped to the back (front). Applying the same argument as before to the oriented cycle circled by P_3 and P_4, we can then obtain that $f_{P_1P_3}(x) = f_{P_2P_4}(x)$.

In the more general case, that is, for arbitrary two oriented paths P and Q from $[a_0, a_1, a_2]$ to x, P and Q may intersect in several different places, we can use induction on the number of maximal common subpaths of P and Q and, applying similar analysis as above from the nth common part to $(n + 1)$th common part, obtain that $f_P(x) = f_Q(x)$.

Therefore, we have uniquely defined a map f from G^* to $C_{2,1,2,1}$.

Now let xy be an arc of G^*. We must prove that $f(x)f(y)$ is an arc of $C_{2,1,2,1}$. We must only prove that for some oriented path P connecting $[a_0, a_1, a_2]$ and xy, $f_P(x)f_P(y)$ is an arc of $C_{2,1,2,1}$. It is a routine work to check by the definition of f_P. □



(a) (b) (c) (d)

FIG. 11

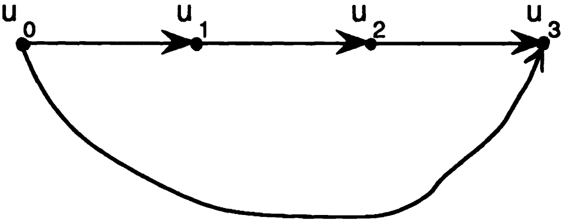


FIG. 12

3. The homomorphic preimages of some other oriented cycles. Next, we may think of the oriented cycle $C_{3,1}$ given in Fig. 12.

Let

$\theta_2 = \{P \in \mathscr{P}$: P contains no $\vec{P}_4$. P begins and ends with $\vec{P}_3$, and contains no $\vec{P}_3$ anywhere else. There are odd number of $\vec{P}_2$'s in $P\}$,

$\omega = \{\vec{P}_4\} \cup \theta_1 \cup \theta_2$.

Then we can characterise the homomorphic preimages of $C_{3,1}$ as stated in the following theorem.

THEOREM 3.1. *For any digraph G, the following are equivalent:*

(1) *G can be homomorphically mapped to $C_{3,1}$; and*

(2) *For any digraph $P \in \omega$, P cannot be homomorphically mapped to G.*

The proofs of this theorem will be the subject of another paper. Therefore, we omit the proofs here.

4. Computational consideration. As applications of our results, we prove that the following two decision problems are in NP ∩ coNP.

Instance: A digraph G.

Question: Is G homomorphic to $C_{2,1,2,1}$ (respectively, $C_{3,1}$)?

We only prove the result for $C_{2,1,2,1}$. Similar arguments can be applied to $C_{3,1}$.

It is easy to see that the problem is in NP. In fact, we may guess a partition of $V(G)$ into six parts $V_0, V_1, \cdots, V_5$, and then check the mapping of the vertices in V_i to u_i ($i = 0, 1, \cdots, 5$) to see if it is a homomorphism. To prove that it also belongs to coNP, we note the equivalent version of Theorem 2.1: For any digraph G, G cannot be homomorphically mapped to $C_{2,1,2,1}$ if and only if G contains some subgraph that is isomorphic to $\vec{C}_2$, $\vec{C}_3$ or an oriented cycle C containing an odd number of $\vec{P}_2$'s. Checking if any given subgraph of G is isomorphic to $\vec{C}_2$, $\vec{C}_3$, or an oriented cycle C containing odd number of $\vec{P}_2$'s, can be done in polynomial time.

REFERENCES

[1] J. BANG-JENSEN, P. HELL, AND G. MACGILLIVRAY, *The complexity of coloring by semicomplete digraphs*, SIAM J. Discrete Math., 1 (1988), pp. 281-298.

[2] J. BANG-JENSEN AND P. HELL, *On the effect of two cycles on the complexity of coloring*, Discrete Appl. Math., 26 (1990), pp. 1-23.

[3] J. BANG-JENSEN, P. HELL, AND G. MACGILLIVRAY, *On the complexity of coloring by superdigraphs of bipartite graphs*, Discrete Math., to appear.

[4] ———, *Hereditarily hard coloring problems*, submitted

[5] D. Duffus, B. Sands, and R. Woodrow, *On the chromatic number of the product of graphs*, J. Graph Theory, 9 (1985), pp. 487–495

[6] H. El-Zahar and N. Sauer, *The chromatic number of the product of two 4-chromatic graphs is 4*, Combinatorica, 5 (1985), pp. 121–126.

[7] W. Gutjahr, E. Welzl, and G. Woeginger, *Polynomial graph colorings*, J. Graph Theory, to appear.

[8] R. Haggkvist, P. Hell, D. J. Miller, and N. Lara, *On multiplicative graphs and the product conjecture*, Combinatorica, 8 (1988), pp. 71–81.

[9] R. Haggkvist and P. Hell, *On A-mote universal graphs*, submitted.

[10] S. T. Hedetniemi, *Homomorphisms of graphs and automata*, Tech. Report 03105-44-T, University of Michigan, 1966.

[11] P. Hell and J. Nesetril, *Homomorphisms of graphs and their orientations*, Monatsh. Math., 85 (1978), pp. 39–48.

[12] P. Hell, *An introduction to the category of graphs*, Ann. New York Acad. Sci., 328 (1979), pp. 120–136.

[13] P. Hell and J. Nesetril, *On the complexity of H-coloring*, J. Combin. Theory Ser. B., 48 (1990), pp. 92–110.

[14] P. Hell, H. Zhou, and X. Zhu, *Homomorphisms to oriented cycles*, Combinatorica, to appear.

[15] ———, *Multiplicativity of oriented cycles*, submitted.

[16] P. Komarek, *Some new good characterizations for directed graph*, Casopis. Pĕst. Mat., 51 (1984), pp. 348–354.

[17] G. MacGillivray, *On the complexity of colouring by vertex-transitive and arc-transitive digraphs*, SIAM J Discrete Math., 4 (1991), pp. 397–408.

[18] H. A. Maurer, A. Salomaa, and D. Wood, *Colorings and interpretations: A connection between graphs and grammar forms*, Discrete Appl. Math., 3 (1981), pp. 119–135.

[19] H. A. Maurer, J. H. Sudborough, and E. Welzl, *On the complexity of the general coloring problem*, Inform. and Control, 51 (1981), pp. 123–145.

[20] D. J. Miller, *The categorical product of graphs*, Canadian J. Math., 20 (1968), pp. 1511–1521.

[21] J. Nesetril and A. Pultr, *On classes of relations and graphs determined by subobject and factorobjects*, Discrete Math., 22 (1978), pp. 287–300.

[22] A. Pultr and J. Vinarek, *Productive classes and subdirect irreducibility in particular for graphs*, Discrete Math., 20 (1977), pp. 159–176.

[23] N. Sauer and X. Zhu, *An approach to Hedetniemi's conjecture*, unpublished manuscript, 1990.

[24] H. Zhou, *Multiplicativity (part* I)*: Variations, multiplicative graphs and digraphs*, J. Graph Theory, 15 (1991), pp. 469–488.

[25] ———, *Multiplicativity (part* II)*: Nonmultiplicative digraphs and characterization of oriented paths*, J. Graph Theory, 15 (1991), pp. 489–509.

[26] ———, *On the nonmultiplicativity of oriented cycles*, SIAM J. Discrete Math., 5 (1992), pp. 207–218.

[27] ———, *Multiplicativity (part* III)*: On weak-multiplicativity of oriented paths*, submitted.

[28] ———, *Characterization of the homomorphic preimages of certain oriented paths*, submitted.

[29] X. Zhu, *Multiplicative structures*, Ph.D. thesis, Department of Mathematics, The University of Calgary, Calgary, Alberta, Canada, 1990.

SIAM J. DISC. MATH.
Vol. 6, No. 1, pp. 100–109, February 1993

008

MATCHINGS IN THE PARTITION LATTICE*

E. RODNEY CANFIELD†

Abstract. Within the lattice of partitions of a finite set, the kth *level* consists of partitions having k blocks. A *matching* between levels k_1 and k_2 is a one-to-one function assigning to each partition in the smaller level another in the larger level, which is related to the first by refinement. It is shown that matchings between adjacent levels of the partition lattice fail to exist precisely for k in an interval. The endpoints of the matchingless interval are shown to equal asymptotically $n \log 2/\log n$ and $n \log 4/\log n$.

Key words. Sperner, partition, matching, lattice

AMS(MOS) subject classifications. 05A18, 05D15, 06A07

1. Introduction. The subject of this paper is the existence of matchings between consecutive levels of the partition lattice Π_n. (See §2 for formal definitions.) We solve a problem posed by Kung [8, §7] and obtain an analogous result for the "other half" of Π_n not considered in the latter work. We describe the main results of [8] in connection with our Theorem 3.4 later in this introduction, but the reader is urged to consult the original preprint for the interesting details concerning the Sperner property in geometric lattices. The work [4] is also highly recommended as an excellent survey of the various Sperner properties, especially in the context of partition lattices, including further background and a large bibliography.

Recall that a ranked partially ordered set is *Sperner* when its width and largest Whitney number agree. The Whitney numbers of the partition lattice are the Stirling numbers of the second kind, and the location K_n of the largest, $S(n, K_n)$, is given asymptotically by $K_n \sim n/\log n$ [6]. Letting $\Pi_{n,k}$ denote the class of partitions having k blocks, our main results are summarized in the following statement. All logarithms are natural.

THEOREM. *There are monotone increasing sequences L_n and R_n, defined for $n \geq 3$, such that* (1) *for $k < L_n$ there is a matching of $\Pi_{n,k}$ into $\Pi_{n,k+1}$*; (2) *for $k > R_n$ there is a matching of $\Pi_{n,k}$ into $\Pi_{n,k-1}$*; (3) *there are no other matchings between consecutive levels of Π_n. Moreover, as $n \to \infty$, $L_n/K_n \to \log 2$, and $R_n/K_n \to \log 4$.*

The role played by previous research in establishing the above results is an interesting story. In [1] it is shown that Π_{n,K_n-1} cannot be matched into Π_{n,K_n}, and it is stated that "with virtually no change" the same method demonstrates that no matchings exist for $|k - K_n| = o\big(n/(\log n)^{3/2}\big)$. In [10] it is stated, without details, that no matching of $\Pi_{n,k}$ into $\Pi_{n,k-1}$ is possible even for k as large as $(1 - \delta)K_n \log 4$. Essentially, two methods, [1] and [11], have been known for proving the impossibility of matchings in Π_n. It is a surprise that pushing these two techniques to their apparent limit, the one being used for $k < K_n$ and the other for $k > K_n$, yields results that are asymptotically the best possible. That is, for those values of k for which the methods fail, matchings, in fact, exist.

The information of the previous paragraph is asymptotic, the precise statement being that, given $\delta > 0$ and large n, matchings from $\Pi_{n,k}$ into $\Pi_{n,k+1}$ exist when $k \leq (1-\delta)K_n \log 2$, and do not exist when $k \geq (1+\delta)K_n \log 2$ (similarly for $k > K_n$ with $\log 4$ replacing $\log 2$). Given only this asymptotic information, we could imagine the transition to be chaotic; that is, for some range of k having length $o(n/\log n)$, located around $K_n \log 2$ and $K_n \log 4$, we find matchings existing and not existing in an unpredictable

*Received by the editors January 21, 1992; accepted for publication (in revised form) March 12, 1992. This research was supported by National Science Foundation grant DMS-9021113.

†Department of Computer Science, University of Georgia, Athens, Georgia 30602.

manner. The result, however, is that the transition is abrupt: we have the sequences L_n and R_n described in the above theorem. The major part of the existence proof for L_n and R_n is implicit in the work of Mullin [9], one of the earliest papers dealing with matchings in the partition lattice.

The rest of the present paper is organized as follows. In §2 we give definitions and some preliminary results. Section 3 contains the main proofs. The theorem stated above is broken down into three parts. The existence of L_n and R_n is shown in Theorem 3.1, the two limits in Theorems 3.2 and 3.3.

Our Theorem 3.4 bears a curious, but possibly only coincidental, relationship to the main theorem of [8]. In the latter, Kung has shown that the incidence matrix of the refinement relation between $\Pi_{n,k}$ and $\Pi_{n,k-1}$ is nonsingular when $k > n/2$. This implies the existence of matchings in that region. Theorem 3.4 states that a particularly appealing method of finding matchings in the partition lattice has the same natural limit of applicability ($k > n/2$) as that found in [8] for the Radon transform method. It would be interesting to know if any incidence matrices are singular at a level for which matchings exist.

A condition that implies the Hall criteria, less obvious than the one presented in Lemma 2.2, was originally used to prove Theorems 3.2 and 3.3. Although this condition has proved in the end to be unnecessary for this work, it may be useful elsewhere and is recorded in Theorem 3.5.

2. Notation and preliminaries. The set $\{1, 2, \ldots, n\}$ is denoted $[n]$. A *partition* π is a set of nonempty blocks, pairwise disjoint, whose union is all of $[n]$. We say that π_1 *refines* π_2, written $\pi_1 \leq \pi_2$, provided that each block of π_1 is contained in some block of π_2. The set of all partitions of n, ordered by refinement, forms a lattice Π_n called the *partition lattice*. We define the kth *level* of Π_n to be those partitions π having k blocks and denote it by $\Pi_{n,k}$; that is,

$$\Pi_{n,k} = \{\pi \in \Pi_n : |\pi| = k\}.$$

The cardinality of $\Pi_{n,k}$ is the Stirling number of the second kind $S(n, k)$. Note that as a lattice the discrete partition $\{\{1\}, \{2\}, \ldots, \{n\}\}$ is the bottom element, and the coarse partition $\{\{1, 2, \ldots, n\}\}$ is the top. Hence the elements of rank $n - k$ are those that we have decided to call the kth level, and the $(n - k)$th Whitney number of the lattice is $S(n, k)$.

Let (X, Y, E) be a triple in which X and Y are disjoint finite sets, and E is a subset of $X \times Y$. For $S \subseteq X$, we define the *E-degree of* S, $d_E(S)$, to be $|\{y \in Y : (x, y) \in E$ for some $x \in S\}|$, and we write $d_E(x)$ for $d_E(\{x\})$. For $T \subseteq Y$, $d_E(T)$ is defined similarly. We say that E is a *matching*, provided that $d_E(z) \leq 1$ for $z \in X \cup Y$, and $d_E(z)$ is identically 1 on at least one of X or Y. If d_E is identically 1 on X, then we say that X *is matched into* Y by E. We say that E *contains a matching*, provided that there exists a matching $E' \subseteq E$. Our notion of matching is sometimes called a complete matching. Let us recall the famous condition of Hall [5] for the existence of a matching.

THEOREM 2.1 (see Hall [5]). *Let $E \subseteq X \times Y$. Then the set E contains a matching of X into Y if and only if, for all subsets $S \subseteq X$,*

$$|S| \leq d_E(S). \tag{2.1}$$

As is well known, it follows from Theorem 2.1 that, if $|X| \leq |Y|$ and if $E \subseteq X \times Y$ is *regular*, that is, $d_E(x)$ is constant for $x \in X$ and $d_E(y)$ is constant for $y \in Y$, then there is a matching of X into Y. Equally easy is the following result, which we prove as a lemma.

LEMMA 2.2. *Let $E \subseteq X \times Y$. If $d_E(x) > 0$ for all $x \in X$ and*

$$\min\{d_E(x) : x \in X\} \geq \max\{d_E(y) : y \in Y\}, \tag{2.2}$$

then there is a matching of X into Y.

Proof. Let $S \subseteq X$. Since

$$\begin{aligned} |S| \min\{d_E(x) : x \in X\} &\leq |\{(x,y) \in E : x \in S\}| \\ &\leq d_E(S) \max\{d_E(y) : y \in Y\}, \end{aligned}$$

the desired conclusion follows, after division, from Theorem 2.1. □

When we speak of the matching problem between two levels k_1 and k_2 of the partition lattice, we mean the triple (X, Y, E), where X is Π_{n,k_1}, Y is Π_{n,k_2}, and E is the set of ordered pairs $(\pi_1, \pi_2) \in X \times Y$ such that π_1 and π_2 are related by refinement. Another convenient notation is $\Pi_{n,k_1} \hookrightarrow \Pi_{n,k_2}$, meaning that Π_{n,k_1} can be matched into Π_{n,k_2}. We also use "$\not\hookrightarrow$" to indicate that a matching is not possible. Unless specified otherwise, when we use the function $d_E()$ for a matching problem in the partition lattice, the set E consists of all pairs determined by the refinement relation. In the second half of Theorem 3.3, and in Theorem 3.4, the set E is a proper subset of the refinement relation.

The second lemma of this section is a special case of [2, Thm. C] adequate for our present application. First, we provide some notation. If $g(x)$ is a nonzero polynomial with nonnegative coefficients,

$$g(x) = \sum_j c_j x^j,$$

then, for each positive real number r, we associate with $g(x)$ a random variable X_r by declaring that X_r equals j with probability $c_j r^j / g(r)$. The mean, variance, and absolute third moment about the mean of X_r are μ_g, σ_g^2, and t_g, respectively. The latter notation has the advantage of conciseness, but it suggests, misleadingly, that μ_g depends only on $g(x)$, omitting reference to the parameter r. This hopefully will cause no confusion. Even when two polynomials $g(x)$ and $h(x)$ are involved, the notations μ_g, σ_h^2, and so forth imply an underlying real parameter; this parameter will be the same for both g and h, and it will be denoted always r. The degree of $g(x)$ is $\deg(g)$, and the smallest integer j for which the coefficient c_j of $g(x)$ is strictly positive is called the *order* of g, $\operatorname{ord}(g)$. The coefficients of $g(x)$ are *properly log concave*, provided that, for $\operatorname{ord}(g) \leq j \leq \deg(g)$, $c_j > 0$ and $c_j^2 > c_{j-1}c_{j+1}$.

There is an obvious relation between the coefficients of the power $g(x)^k$ and the probability, call it $p(n,k)$, that the sum Z of k independent copies of X_r equals n. Sums of independent variables are the object of interest in the central limit theorem, and, in particular, the remarkable Berry–Esseen inequality [3, p. 521] gives a universal bound on the difference of Z's distribution function from the normal distribution, in terms of the third absolute moment about the mean t_g. Information about this distribution, the values of which are sums $\sum_{n' \leq n} p(n',k)$, can be converted to information about a particular $p(n,k)$, (that is, a local limit theorem), provided the $p(n,k)$ have adequate "smoothness." An example of inadequate smoothness is the situation of every other $p(n,k)$ being zero; no local limit theorem is possible in such a situation, although a central limit theorem may hold. On the other hand, an example of adequate smoothness is proper log concavity. Proper log concavity, moreover, is preserved under the taking of products

[7, Chap. 8]. The derivation of the local from the central limit theorem in the presence of log concavity is a routine, albeit somewhat tedious, calculation. We simply state the needed result and refer any reader interested in the proof to [2]. As usual, the operator $[x^n]$ applied to a power series $f(x)$ yields the coefficient of x^n in $f(x)$.

LEMMA 2.3. *There are absolute constants c_1, c_2, and c_3 such that if polynomials g, h have properly log concave coefficients and the integers l_1, l_2, n, and the real parameter r satisfy the two relations*

$$\sigma \geq c_1, \qquad \left|\frac{n-\mu}{\sigma}\right| \leq 0.01, \tag{2.3}$$

where

$$\mu = l_1\mu_g + l_2\mu_h, \qquad \sigma^2 = l_1\sigma_g^2 + l_2\sigma_h^2,$$

then

$$[x^n]g(x)^{l_1}h(x)^{l_2} = g(r)^{l_1}h(r)^{l_2}\frac{\exp(-\frac{1}{2}(\frac{n-\mu}{\sigma})^2)}{\sigma r^n(2\pi)^{1/2}}(1+\epsilon),$$

with

$$|\epsilon| \leq \frac{c_2\max(t_g/\sigma_g^2, t_h/\sigma_h^2) + c_3}{\sigma^{1/2}}.$$

The final lemma of this section is a technical calculation, which will be needed later.

LEMMA 2.4. *Let m be a positive integer, $r = m/2$, $g(x) = \sum_{j=1}^{m} x^j/j!$, and $h(x) = \sum_{j=m+1}^{2m} x^j/j!$. Then, in the notation defined above, $\mu_g \sim r$, $\sigma_g^2 \sim r$, $\mu_h \sim 2r$, and $\sigma_h^2 \to 2$, as $m \to \infty$.*

Proof. All the $O(\ )$ bounds in this proof are as $m \to \infty$. Using a geometric series as an upper bound, $\sum_{j=m+1}^{\infty} r^j/j! = O(r^m/m!)$. Estimating $r^m/m!$ by Stirling's formula, for some constant $c > 1$, we have that

$$g(r) = \sum_{j=1}^{m} r^j/j! = e^r(1 + O(c^{-r}))$$

$$\sum_{j=1}^{m} jr^j/j! = re^r(1 + O(c^{-r}))$$

$$\sum_{j=1}^{m} j^2r^j/j! = (r^2 + r)e^r(1 + O(c^{-r})).$$

From these three relations, we obtain the desired formulas for $\mu_g$and σ_g^2.

Now we proceed to the polynomial $h(x)$. Again using a geometric series and Stirling's formula, we find the bound (since $5\log 2 > 3$)

$$\sum_{j>m+5\log m} r^j/j! = \frac{r^m}{m!}O(m^{-3}). \tag{2.4}$$

Let s_1 and s_2 be the functions of an integer argument h defined by $s_1 = 1 + 2 + \cdots h$ and $s_2 = 1^2 + 2^2 + \cdots h^2$; then, for $|z| < 1$,

$$\sum_{h=1}^{\infty} s_1 z^h = z(1-z)^{-3},$$

$$\sum_{h=1}^{\infty} (s_1^2 + s_2) z^h = 2z(1+2z)(1-z)^{-5}.$$

Taking $z = \frac{1}{2}$ and truncating the sums, we find that

$$\sum_{1 \le h \le 5 \log m} s_1/2^h = 4 + O(m^{-3}), \tag{2.5}$$

$$\sum_{1 \le h \le 5 \log m} (s_1^2 + s_2)/2^h = 64 + O(m^{-3}). \tag{2.6}$$

As $m \to \infty$ we have, uniformly for $h = O(\log m)$,

$$(m+1)(m+2)\cdots(m+h) = m^h\bigl(1 + s_1/m + (s_1^2 - s_2)/2m^2 + O(h^5/m^3)\bigr),$$

and so

$$r^h/(m+1)(m+2)\cdots(m+h) = (r/m)^h(1 - s_1/m + (s_1^2 + s_2)/2m^2 + O(h^6/m^3)). \tag{2.7}$$

Combining (2.4) through (2.7),

$$\begin{aligned}\sum_{j=m+1}^{2m} r^j/j! &= \frac{r^m}{m!} \sum_{1 \le h \le 5 \log m} r^h/(m+1)(m+2)\cdots(m+h) + \sum_{m+5 \log m < j \le 2m} r^j/j! \\ &= \frac{r^m}{m!}\bigl(1 - 4/m + 32/m^2 + O(m^{-3})\bigr).\end{aligned} \tag{2.8}$$

Since $r^{2m}/(2m)!$ is exponentially small in comparison to $r^m/m!$, we can deduce from (2.8) that

$$\begin{aligned}\sum_{j=m+1}^{2m} j r^j/j! &= r\Bigl(\sum_{j=m+1}^{2m} r^j/j! + \frac{r^m}{m!} - \frac{r^{2m}}{(2m)!}\Bigr) \\ &= 2r\frac{r^m}{m!}\bigl(1 - 2/m + 16/m^2 + O(m^{-3})\bigr),\end{aligned} \tag{2.9}$$

and similarly, after first calculating $\sum_{j=m+1}^{2m} j(j-1)r^j/j!$,

$$\sum_{j=m+1}^{2m} j^2 r^j/j! = 4r^2\frac{r^m}{m!}\bigl(1 + 6/m^2 + O(m^{-3})\bigr). \tag{2.10}$$

From (2.8)–(2.10), the desired assertions about $\mu_h$and σ_h^2 follow. □

Remark. A referee noted that a bound of the form $\sigma_h^2 = O((\log r)^4)$ is obtainable by similar, but less detailed, analysis and suffices for what follows.

3. Proofs.

THEOREM 3.1. *There are monotone increasing sequences L_n and R_n defined for $n \geq 3$ such that, for $k < L_n$, we have $\Pi_{n,k} \hookrightarrow \Pi_{n,k+1}$; for $k > R_n$, we have $\Pi_{n,k} \hookrightarrow \Pi_{n,k-1}$; and, for all other k, $\Pi_{n,k}$ can be matched into neither of $\Pi_{n,k\pm 1}$. Moreover, both L_n and R_n grow by at most* 1 *when n increases by* 1.

Proof. The proof is by induction on n. For $n = 3$, both L_n and R_n are equal to K_n. The inductive step requires four facts, and the more difficult two have been given by Mullin; see our remark below. First, by [9], if $\Pi_{n,k} \hookrightarrow \Pi_{n,k+1}$ for $k < L_n$, then $\Pi_{n+1,k} \hookrightarrow \Pi_{n+1,k+1}$ for $k < L_n$. Second, by [9], if $\Pi_{m,k} \hookrightarrow \Pi_{m,k-1}$ for $k > R_n$ and $m \leq n$, then $\Pi_{n+1,k} \hookrightarrow \Pi_{n+1,k-1}$ for $k > R_n + 1$. To these two facts we add two additional observations of a more elementary nature. Third, if $\Pi_{n,k} \not\hookrightarrow \Pi_{n,k+1}$, then $\Pi_{n+1,k+1} \not\hookrightarrow \Pi_{n+1,k+2}$, because any set $S \subseteq \Pi_{n,k}$ whose refinements in $\Pi_{n,k+1}$ constitute a smaller set T gives rise to a set $S' \subseteq \Pi_{n+1,k+1}$ with the same property: simply take S' to be all the partitions of S with the singleton block $\{n+1\}$ added on. Indeed, the refinements T' of S' are obtained from T by the same process of adding on a singleton block. Fourth, if $\Pi_{n,k} \not\hookrightarrow \Pi_{n,k-1}$, then $\Pi_{n+1,k} \not\hookrightarrow \Pi_{n+1,k-1}$. For now, any set $U \subseteq \Pi_{n,k}$ whose co-refinements in $\Pi_{n,k-1}$ constitute a smaller set V gives rise to a set $U' \subseteq \Pi_{n+1,k}$ with the same property: U' contains all partitions that can be obtained from some partition in U by adding the element $n+1$ to some block. Thus $|U'| = k|U|$. The co-refinements V' of U' are obtained from V by the same process of adding element $n+1$ to each block. Hence $|V'| = (k-1)|V|$, and our fourth assertion is established.

The net effect of the preceding four facts is this: Given that L_n and R_n have the property within Π_n claimed by the theorem, then the status of where matchings exist within Π_{n+1} is completely determined, except for $k = L_n$ and $k = R_n + 1$. Whatever may be the status of these two values of k, however, L_{n+1} and R_{n+1} exist and have the stated growth condition. This completes the induction. □

Remark. Mullin [9] states his main theorem this way: If for all n it is the case that $\Pi_{n,K_n-1} \hookrightarrow \Pi_{n,K_n}$ and also that $\Pi_{n,K_n+1} \hookrightarrow \Pi_{n,K_n}$, then, in fact, for all n it is the case that matchings exist between every pair of consecutive levels in Π_n. Despite this "all or none" statement, his proof is an explicit construction of matchings leading to the first two of the four key assertions above.

THEOREM 3.2. *For $n \geq 5$ and $k \leq n \log 2/\log n$, there is a matching of $\Pi_{n,k}$ into $\Pi_{n,k+1}$. Moreover, the constant* $\log 2$ *is the best possible in that, for each $\delta > 0$, there is an integer n_0 depending on δ such that, for $n \geq n_0$ and $k \geq (1+\delta)n\log 2/\log n$, there is no matching of $\Pi_{n,k}$ into $\Pi_{n,k+1}$.*

Proof. We will show that condition (2.2) for a matching is satisfied. Let $x \in \Pi_{n,k}$; then

$$\begin{aligned} d_E(x) &= \sum_{B\in x}(2^{|B|-1}-1) \\ &\geq k(2^{n/k-1}-1), \end{aligned}$$

since $\sum 2^{|B|}$ is minimized when all $|B|$ are equal. However, $n/k \geq \log n/\log 2$, and so $2^{n/k} \geq n$. On the other hand, for each $y \in \Pi_{n,k+1}$, we have $d_E(y) = \binom{k+1}{2}$; hence (2.2) follows, provided only that n is so large that $n/2 - 1 \geq (k+1)/2$. The latter is true for $n \geq 6$, and for $n = 5$ the assertion of the theorem is easily checked by inspection.

Next, we want to see that $\log 2$ is the best possible constant. Fix $\delta > 0$, and consider a sequence of pairs (n,k) for which $n \to \infty$ and $k \sim \beta n/\log n$ with $\beta > (1+\delta)\log 2$. We may assume that $k \leq K_n$, since otherwise, of course, $\Pi_{n,k} \not\hookrightarrow \Pi_{n,k+1}$. Hence, without loss, $\beta < 1+\delta$. For each pair (n,k), we will define a collection of partitions $A \subseteq \Pi_{n,k}$ and show that, for all sufficiently large n, it fails the Hall condition (2.1). The collection A will depend on two parameters l and m and specifically will consist of all partitions $\pi \in \Pi_{n,k}$ that have exactly l blocks whose sizes lie in the range 1 to m, and $k-l$ blocks whose sizes lie in the range $m+1$ to $2m$. Let $g(x)$ and $h(x)$ be the polynomials $\sum_1^m x^j/j!$ and $\sum_{m+1}^{2m} x^j/j!$, respectively. With $a(n,k,l,m) = |A|$, we have by a standard generating function argument (see, for instance, [12, §3.6]) the identity

$$\sum_n a(n,k,l,m)x^n/n! = \frac{g(x)^l h(x)^{k-l}}{l!(k-l)!}.$$

Our plan is to estimate $|A|$ with Lemma 2.3. We start with the following three definitions:

$$\begin{aligned} m &= \left\lfloor \frac{\log n}{(1+\delta)\log 2} \right\rfloor, \\ r &= m/2, \\ l &= \left\lfloor \frac{k\mu_h - n}{\mu_h - \mu_g} \right\rfloor. \end{aligned} \tag{3.1}$$

By Lemma 2.4, we have the following four relations: $\mu_g \sim r$, $\sigma_g^2 \sim r$, $\mu_h \sim 2r$, and $\sigma_h^2 \sim 2$. Were the floor function "$\lfloor\ \rfloor$" omitted from definition (3.1) of l, the left side of (2.3) would be exactly zero. Note that $l/k \to 2(\beta - (1+\delta)\log 2)/\beta$, and, by our assumption that $(1+\delta)\log 2 < \beta < (1+\delta)$, the limit of l/k lies strictly between 0 and 1. This implies that $\sigma^2 = \Omega(kr)$, and so we find that $(n-\mu)/\sigma = O((r/k)^{1/2})$. Thus, for all large n, condition (2.3) holds. By a crude estimate, $\max(t_g/\sigma_g^2, t_h/\sigma_h^2) = O(m)$, and so $\epsilon = O(r/k^{1/4})$. From Lemma 2.3, then, as $n \to \infty$, with $k \sim \beta n/\log n$ and m,l chosen by (3.1), we have that

$$|A| = a(n,k,l,m) \sim \frac{n!}{l!(k-l)!} \frac{g(r)^l h(r)^{k-l}}{\sigma r^n (2\pi)^{1/2}}.$$

How large is the collection of all refinements of partitions in A ? Define C_1 and $C_2 \subseteq \Pi_{n,k+1}$ in the same manner as A, but with parameters $(l+1, m)$ and $(l+2, m)$, respectively. Certainly, all refinements of A are found among $C_1 \cup C_2$. A key point in our argument is that the *same* parameter r used in estimating $|A|$ will serve to estimate $|C_1| = a(n, k+1, l+1, m)$ and $|C_2| = a(n, k+1, l+2, m)$; that is, condition (2.3) will hold for both C_1 and C_2. Moreover, the two values of σ arising in the estimation of $|C_1|$ and $|C_2|$ are asymptotic to that arising in the estimation of $|A|$, and the two values of ϵ again tend to zero; in short, we arrive at the rather simple equation

$$\frac{|C_1| + |C_2|}{|A|} \sim \frac{g(r)}{l} + \left(\frac{g(r)}{l}\right)^2 \frac{k-l}{h(r)}. \tag{3.2}$$

Recall that both l and $k-l$ are $\Omega(k)$. Because $g(r) \le e^r$ and $h(r) \ge r^{m+1}/(m+1)!$, it is easy to see that both fractions on the right side of (3.2) tend to zero. In fact, using Stirling's formula for $m!$, $g(r)/l = O\big((\log n)/n^{p_1}\big)$ and $(g(r)/l)^2((k-l)/h(r)) = O\big((\log n)^{3/2}/n^{p_2}\big)$ with p_1 and p_2 equal to $1-1/(1+\delta)\log 4$ and $\delta/(1+\delta)$. This concludes our proof. $\square$

THEOREM 3.3. *Fix $\delta > 0$. There is an integer n_0 depending on δ such that, for all $n \ge n_0$, (1) there is no matching of $\Pi_{n,k}$ into $\Pi_{n,k-1}$ for $k \le (1-\delta)n\log 4/\log n$, (2) and there is a matching of $\Pi_{n,k}$ into $\Pi_{n,k-1}$ for $k \ge (1+\delta)n\log 4/\log n$.*

Proof. Let $m = \lfloor(1+\delta)\log n/\log 4\rfloor$ and consider the set S of $\pi \in \Pi_{n,k}$ at least $n^{1-\delta/2}$ of whose block sizes equal $2m$, at least $n^{1-\delta/2}$ of whose block sizes equal or exceed $3m$, and all of whose block sizes belong to the set $\{m, 2m, 3m, 3m+1, \cdots\}$. Note that, for sufficiently large n, the set S is not empty. Some reflection shows that each $y \in \Pi_{n,k-1}$ refined by at least one $\pi \in S$ is, in fact, refined by many; precisely, for each $y \in \Pi_{n,k-1}$,

$$\{\pi \in S : \pi \leq y\} \neq \emptyset \;\Rightarrow\; |\{\pi \in S : \pi \leq y\}| \geq \frac{1}{2}\binom{2m}{m} n^{1-\delta/2}.$$

For n sufficiently large, $\frac{1}{2}\binom{2m}{m} n^{1-\delta/2} \geq \frac{1}{2}(n\log 4/\log n)^2$, implying that the Hall condition (2.1) fails for this set S, and there can be no matching of $\Pi_{n,k}$ into $\Pi_{n,k-1}$.

Proceeding now to assertion (2), we will show that the triple $(\Pi_{n,k}, \Pi_{n,k-1}, E)$, with E a suitably defined proper subset of the refinement relation, satisfies condition (2.2) for a matching. Assume that $k \geq (1+\delta)n\log 4/\log n$, and define ϵ and b by the equations

$$\begin{aligned} \epsilon &= (1+\delta)\log 4/\log n, \\ b &= (1-\delta/3)\log n/\log 4. \end{aligned}$$

Without loss, we take $\delta \leq 1/2$. Let E be the set of pairs (π_1, π_2) such that π_1 refines π_2 by splitting a block B whose size is no more than $2b$ into two blocks B_1 and B_2 each of whose sizes is no more than b. We now check condition (2.2).

Suppose that $x \in \Pi_{n,k}$. Since $k \geq \epsilon n$, we have $(1/b\epsilon)kb \geq n$, which implies that the fraction of blocks in x whose size exceeds b is no more than $1/b\epsilon$. Hence

$$|\{B \in x : |B| \leq b\}| \geq \left(1 - \frac{1}{b\epsilon}\right)k.$$

Let n be sufficiently large that $\delta k \geq 30$. Since $b\epsilon \geq 1 + \delta/2 \geq (1-\delta/3)^{-1}$,

$$\begin{aligned} d_E(x) &\geq \binom{\delta k/3}{2} \\ &\geq (\delta k)^2/20. \end{aligned} \tag{3.3}$$

Next, suppose that $y \in \Pi_{n,k-1}$. Since it must be a block of size $2b$ or less that is split to create an E-relation,

$$k2^{2b} \geq d_E(y). \tag{3.4}$$

Let n be sufficiently large that $\delta^2\epsilon \geq 20n^{-\delta/3}$; then, since $2^{2b} = n^{1-\delta/3}$, conditions (3.3) and (3.4) imply that

$$d_E(x) \geq d_E(y),$$

and the proof is complete. □

Remark. The construction of set S used to prove assertion (1), above, is a trivial modification of the one used by Shearer [11]. Shearer has stated [10, p. 15] that, for all n sufficiently large, something even stronger than (1) is true: namely, that every maximum-sized antichain in Π_n includes partitions π the number of whose blocks satisfies $|\pi| > (1-\delta)n\log 4/\log n$.

The next theorem states that a particular sublattice of the partition lattice contains a matching for k in the range covered by [8].

THEOREM 3.4. *Let E_1 be the set of all pairs $(\pi_1, \pi_2) \in \Pi_n \times \Pi_n$ such that π_1 refines π_2 by creating one new block of size 1. Then, E_1 contains a matching of $\Pi_{n,k}$ into $\Pi_{n,k-1}$ for $k \geq n/2+1$.*

Proof. We rely on Lemma 2.2 once again. Let $x \in \Pi_{n,k}$ and s be the number of singleton blocks in x. Then $d_{E_1}(x) = \binom{s}{2} + s(k-s)$, and the latter is an increasing function for $0 \leq s \leq k - 1/2$. If $k = n$, the theorem is trivial, and for $k < n$ we must have $s \leq k-1$. Because $k \geq n/2+1$, we have $s \geq 2$. Taking $s = 2$,

$$d_{E_1}(x) \geq 2k-3.$$

Now let $y \in \Pi_{n,k-1}$. Clearly, $d_{E_1}(y)$ is the number of blocks in y of size 2 plus the number of elements belonging to blocks of size 3 or more in y. It is easy to show that $d_{E_1}(y)$ can be maximized when $n-k$ is even by a partition having blocks of sizes 1 and 3 only, and when $n-k$ is odd by a partition having one block of size 2 and all others of size 1 and 3. These need not be the only maxima, but at least we know that

$$d_{E_1}(y) \leq 3(n-k)/2.$$

For k in the stated range, we see that $d_{E_1}(x) \geq d_{E_1}(y)$, and this completes the proof. □

Our last theorem says that the condition of Lemma 2.2 for verifying the Hall criteria may be relaxed to the extent that $d_E(x) \geq d_E(y)$ is not required of all pairs (x, y), but only those pairs belonging to E. The proof given here is due to Griggs.

THEOREM 3.5. *Let $E \subseteq X \times Y$. If $d_E(x) > 0$ for all $x \in X$ and*

$$(x,y) \in E \Rightarrow d_E(x) \geq d_E(y),$$

then E contains a matching of X into Y.

Proof. The proof is by induction on $|X|$, the case where $|X| = 1$ being trivial. Assume that $|X| = n+1$, and that the theorem holds for smaller sets X. Note that whenever E satisfies the hypothesis of the theorem, so does the restriction of E to any subset of X. By induction, for any $S \subset X$, $S \neq X$, E contains a matching of S into Y, so that $d_E(S) \geq |S|$. Thus it suffices to show that $d_E(X) \geq |X|$. Let $x_0 \in X$, and, using induction, find $M \subseteq E$ a matching of $X - x_0$ into Y. We can extend M to a matching of all of X into Y, provided only that x_0 is adjacent to some point of Y not in M. If this is not the case, since $d_E(x_0) > 0$,

$$\begin{aligned}\sum_{y \in Y} d_E(y) &= |E| \\ &> \sum_{x \in X - x_0} d_E(x) \\ &\geq \sum_{y:(x,y) \in M} d_E(y),\end{aligned}$$

which implies some other vertex in Y is adjacent to X; that is, $d_E(X) \geq |M| + 1 = |X|$. As noted above, this completes the induction. □

Acknowledgments. The author thanks Jerry Griggs for several valuable comments and for the above proof of Theorem 3.5. The author also thanks an anonymous referee for several corrections and helpful suggestions.

REFERENCES

[1] E. R. CANFIELD, *On a problem of Rota*, Adv. in Math., 5 (1977), pp. 1–10.

[2] ———, *Application of the Berry–Esséen inequality to combinatorial estimates*, J. Combin. Theory Ser. A, 28 (1980), pp, 17–25.

[3] W. FELLER, *An Introduction to Probability Theory and Its Applications*, Vol. II, John Wiley, New York, 1966.

[4] J. R. GRIGGS, *The Sperner property in geometric and partition lattices*, in The Dilworth Theorems, K. P. Bogart, R. Freese, and J. P. S. Kung, eds., Birkhäuser, Boston, 1990, pp. 298–304.

[5] P. HALL, *On representatives of subsets*, J. London Math. Soc., 10 (1935), pp. 26–30.

[6] L. H. HARPER, *Stirling behavior is asymptotically normal*, Ann. Math. Statist., 38 (1967), pp. 410–414.

[7] S. KARLIN, *Total Positivity*, Stanford University Press, Stanford, CA, 1968.

[8] J. P. S. KUNG, *The Radon transforms of a combinatorial geometry*. II. *Partition lattices*, Adv. in Math., to appear.

[9] R. MULLIN, *On Rota's problem concerning partitions*, Aequationes Math., 2 (1969), pp. 98–104.

[10] J. B. SHEARER, *Some problems in combinatorics*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 1980.

[11] ———, *A simple counterexample to a conjecture of Rota*, Discrete Math., 28 (1979), pp. 327–330.

[12] H. S. WILF, *generatingfunctionology*, Academic Press, New York, 1990.

SIAM J. DISC. MATH
Vol. 6, No. 1, pp. 110–123, February 1993

009

COMMUNICATION COMPLEXITY AND QUASI RANDOMNESS*

FAN R. K. CHUNG[†] AND PRASAD TETALI[‡]

Abstract. The multiparty communication complexity concerns the least number of bits that must be exchanged among a number of players to collaboratively compute a Boolean function $f(x_1, \dots, x_k)$, while each player knows at most t inputs for some fixed $t < k$. The relation of the multiparty communication complexity to various hypergraph properties is investigated. Many of these properties are satisfied by random hypergraphs and can be classified by the framework of quasi randomness. Namely, many disparate properties of hypergraphs are shown to be mutually equivalent, and, furthermore, various equivalence classes form a natural hierarchy. In this paper, it is proved that the multiparty communication complexity problems are equivalent to certain hypergraph properties and thereby establish the connections among a large number of combinatorial and computational aspects of hypergraphs or Boolean functions.

Key words. multiparty protocols, discrepancy, lower bounds

AMS(MOS) subject classifications. 05, 68, 94

1. Introduction.

Many problems arising in interactive and distributive computation share the general framework that a number of processors wish to collaboratively evaluate a Boolean function while each processor has only partial information. We are interested in determining the minimum amount of information transfer required, under the assumption that each processor has unlimited computational power and that the messages are transferred by a "blackboard," viewed by all processors.

One of the most interesting examples is the *round-table* model, proposed by Chandra, Furst, and Lipton [CFL], involving k players each having a number X_i on his/her forehead (so that the ith player knows all numbers except for X_i). For $k = 3$, they proved a tight lower bound for the minimum number of bits to be exchanged to compute the sum of X_i's. For general k, the lower bounds were further improved by Babai, Nisan, and Szegedy [BNS], who gave a lower bound of $\Omega(m2^{-k})$ for computing some explicit functions on k strings m-bits each.

When only two players are involved, it is just the usual model for communication complexity, which was first proposed by Yao [Y] and has been studied extensively by many researchers [BFS], [HMT], [L], [LS], [MS], [PS], [Th]. In this paper, we consider the following model generalizing both the round-table model and Yao's model: A number of players wish to cooperatively determine a Boolean function $f(x_1, \dots, x_k)$, which accepts k inputs each m bits long. Suppose that each player knows at most t inputs. We are interested in minimizing the number of bits $C_{k,t}(f)$ to be exchanged to compute f.

Determining the communication complexity $C_{k,t}(f)$ could be a difficult problem for a general function f. The main purpose of this paper is to demonstrate the relation of communication complexity to several hypergraph properties. Consequently, lower bounds for $C_{k,t}$ can then be established. These hypergraph properties arise in the study of random-like graph properties, called *quasi-random*.

Quasi randomness was first introduced in [CGW] by showing that a large number of disparate graph properties are mutually equivalent, in the sense that any graph satisfying one of the properties must of necessity satisfy all of them. More recently, in [C] it was shown that several equivalence classes $\mathcal{A}_i$ form a hierarchy of classes of properties for k-uniform hypergraphs (or k-graphs, for short) and for Boolean functions with k input

*Received by the editors October 19, 1990; accepted for publication (in revised form) November 14, 1991.

[†] Bell Communications Research, Morristown, New Jersey 07960.

[‡] Visitor, DIMACS Center, Rutgers University, P.O. Box 1179, Busch Campus, Piscataway, New Jersey 08855. Present address, AT&T Bell Laboratories, Room 2C-169, Murray Hill, New Jersey 07974-0636.

arguments (also called k-functions). The quasi-random class $\mathcal{A}_k$, introduced in [CG1], consists of graph properties such as: "For any fixed $s \geq 2k$ all k-graphs on s vertices appear almost equally often as induced subgraphs of G." On the other hand, in $\mathcal{A}_0$ there is the property that the number of edges in G is approximately the same as the number of nonedges in G. The detailed description of the equivalence classes $\mathcal{A}_i$ and the hierarchy $\mathcal{A}_0 \supset \mathcal{A}_1 \supset \cdots \supset \mathcal{A}_k$ are described in §2.

Among various properties in the equivalence class $\mathcal{A}_i$, there are two interesting invariants—the i-discrepancy and the i-deviation (see §2 for definitions). Intuitively, the i-deviation provides a quantitative indication as to how much the graph deviates from random graphs. *Discrepancy* is useful in various contexts, in particular, corresponding to various statistical tests arising in complexity analysis. Roughly speaking, *discrepancy* is a "global" property that is often hard to compute, while *deviation* is a "local" property that is easy to compute. The quasi randomness results imply that the i-discrepancy of a function is small if and only if its i-deviation is small. Furthermore, the i-discrepancy can be used to characterize the communication complexity $C_{k,i}$. Using the results of [BNS], this further leads to explicit construction of functions $f_{k,t}$ with communication complexity $C_{k,t}$ lower bounded by $\Omega(mc^{-t})$. One of the consequences is a simple proof of the lower bound of $\Omega(m2^{-k})$ on the communication complexity of the "generalized inner product" function, as described in §3.

The communication complexity $C_{k,t}$ corresponds in a natural way to the complexity of a t-head Turing machine that computes Boolean functions with k inputs (as discussed in §3). As an immediate consequence, lower bounds for time-space trade-offs can be obtained. We prove that, for any fixed t, any $(t-1)$-head TM computing the function $f_{k,t}$ on m-bit strings requires a time-space trade-off of $TS \geq \Omega(m^2)$.

Discrepancy can also be interpreted in terms of a game of *switches* and *lights* (also discussed in §3). Apart from being interesting, this interpretation yields a short proof that the communication complexity $C_{k,i}$ of a *random* k-function f is at least $((k-i+1)/2)m$.

In §4 we conclude with some open problems and remarks about the relations of communication complexity to other complexity issues. The *quantitative* quasi-random classes for k-graphs with edge density α and various *expansion* properties are also mentioned.

2. Quasi-random classes.

2.1. Notation. We use $\binom{X}{k}$ to denote the set of k-element subsets of a set X of cardinality $\geq k$. A k-graph $G = (V, E)$ consists of a set $V = V(G)$, called the *vertices* of G, and a subset $E = E(G)$ of the set $\binom{V}{k}$, called the *edges* of G. Throughout this paper, G denotes a k-graph on n vertices unless otherwise specified.

For $X \subseteq V$, $G[X]$ denotes the subgraph of G induced by X, i.e., $G[X] = \left(X, E \cap \binom{X}{k}\right)$. Let H denote an l-graph, where $l < k$ and $V(H) = V(G)$. The set $E(G, H)$ of edges of G induced by H is defined as

$$E(G,H) = \left\{x \in E(G) : \binom{x}{l} \subseteq E(H)\right\}.$$

For $l = 1$, the edge set of H is just a subset of $V(G)$ and $E(G, H) = E(G[H])$. We denote $e(G) =| E(G) |$ and $e(G, H) =| E(G, H) |$.

Discrepancy. For $i \geq 2$, the *i-discrepancy* of G, denoted by $\mathrm{disc}_i(G)$, is defined as follows:

$$\mathrm{disc}_i(G) = \max_{H:(i-1)-\text{graph}} \frac{| e(G,H) - e(\bar{G},H) |}{| V(G) |^k},$$

where $\bar{G}$ denotes the complement of G with edge set $\left\{x \in \binom{V}{k} : x \notin E(G)\right\}$.

We remark that disc$_2$ is often called *discrepancy* in the literature. disc$_i$ can be viewed as a natural generalization of *discrepancy*.

We let $\mu_G : \binom{V}{k} \to \{-1, 1\}$ denote the edge function of G; i.e., for $x \in \binom{V}{k}$,

$$\mu_G(x) = \begin{cases} -1 & \text{if } x \in E, \\ 1 & \text{otherwise.} \end{cases}$$

Let V^k denote the set of k-tuples $(v_1, \ldots, v_k)$, $v_i \in V$, where the v's are not necessarily distinct. Let $\prod_G^{(i)} : V^{k+i} \to \{-1, 1\}$ denote the following function of G:

$$\prod_G^{(i)}(u_1, \ldots, u_{2i}, v_{i+1}, \ldots, v_k) = \prod_{\epsilon_1} \cdots \prod_{\epsilon_i} \mu_G(\epsilon_1, \ldots, \epsilon_i, v_{i+1}, \ldots, v_k),$$

where $\epsilon_j \in \{u_{2j-1}, u_{2j}\}$ for $j \le i$. Note that $\prod_G^{(i)}$ is a product of 2^i terms, each of which is an edge function. For $i = 0$, we define $\prod_G^0 = \mu_G$.

Deviation. The *i-deviation* of G, denoted by dev$_i(G)$, is defined as follows:

$$\text{dev}_i(G) = \frac{1}{n^{k+i}} \sum_{u_1, \cdots, u_{k+i}} \prod_G^{(i)}(u_1, \ldots, u_{k+i}).$$

Thus dev$_i(G)$ assumes a value between -1 and 1. (Another interpretation is that n^{k+i} dev$_i$ is the difference of the number of "even partial (squashed) octahedrons" and the "odd partial (squashed) octahedrons," as described in [CG1] and [CG2].)

2.2. Quasi randomness. We use the following convention. Suppose that we have two classes $P = P(o(1))$ and $P' = P'(o(1))$, each with occurrences of the asymptotic $o(1)$ notation. By the implication "$P \Rightarrow P'$," we mean that, for each $\epsilon > 0$, there is a $\delta > 0$ (a function of ϵ and k but independent of n) such that, if $G(n)$ satisfies $P(\delta)$, then it also satisfies $P'(\epsilon)$, provided that $n > n_0(\epsilon)$. Two properties P and P' are said to be equivalent if $P \Rightarrow P'$ and $P' \Rightarrow P$.

Here we define several classes of properties for k-graphs.

For $i = 0$ and 1, define the properties

R_0 : $e(G) - e(\bar{G}) = o(n^k)$, where $\bar{G}$ denotes the complement of G,

R_1 : G is almost regular. That is,

$$\sum_{u_1, \cdots, u_{k-1}} \left(d^+(u_1, \ldots, u_{k-1}) - d^-(u_1, \ldots, u_{k-1})\right)^2 = o(n^{k+1}),$$

where

$$d^+(u_1, \ldots, u_{k-1}) = \mid \{v \in V : \{u_1, \ldots, u_{k-1}, v\} \in E(G)\} \mid$$

and

$$d^-(u_1, \ldots, u_{k-1}) = \mid \{v \in V : \{u_1, \ldots, u_{k-1}, v\} \notin E(G)\} \mid .$$

For $i \geq 2$, define

$$R_i : \text{For every}(i-1)\text{-graph } H, \quad e(G,H) - e(\bar{G},H) = o(n^k).$$

In [CG1] it was shown that the property $\mathrm{dev}_k(G) = o(1)$ for a hypergraph G is equivalent to a number of properties, among which are

Q : For all k-graphs G' on $2k$ vertices, the number of (labelled) occurrences of G' in G as an induced subgraph is $(1+o(1))n^{2k}2^{-\binom{2k}{k}}$.

Let s denote a fixed integer and $s \geq 2k$.

$Q(s)$: For all k-graphs $G'(s)$ on s vertices, the number of (labelled) occurrences of G' in G as an induced subgraph is $(1+o(1))n^{s}2^{-\binom{s}{k}}$.

In [C] the deviation property is further generalized to the following property (denoted P_i). For $i \geq 0$, P_i : $\mathrm{dev}_i(G) = o(1)$. The main results of [C] can be summarized in the following two theorems.

THEOREM 1. *Properties P_i and R_i are equivalent for $i = 0, \ldots, k$. In particular, for $i \geq 2$, we have that*

$$\text{(i)} \quad \mathrm{disc}_i(G) = \max_{H:(i-1)-graph} \frac{|\, e(G,H) - e(\bar{G},H)\,|}{|\, V(G)\,|^k} < (\mathrm{dev}_i(G))^{1/2^i},$$

$$\text{(ii)} \quad \mathrm{dev}_i(G) < 4^i\,(\mathrm{disc}_i(G))^{1/2^i}.$$

Theorem 1, in fact, has interesting computational implications. It is easy to see that computing disc_i for general G (naively) takes time $O(2^{n^i} \cdot n^k)$, since the number of i-graphs is $O(2^{n^i})$ and, for each i-graph H, computing $|\, e(G,H) - e(\bar{G},H)\,|$ takes $O(n^k)$ time. On the other hand, dev_i can be computed in time $O(n^{k+i})$, since dev_i is a sum of n_{k+i} terms, each term, respectively, is a product of 2^i subterms, each of which is an edge function. Thus Theorem 1 leads to the following conclusion: Although it takes exponential time to compute disc_i exactly, an approximation can be obtained by using dev_i in only polynomial time. We remark that it would be of interest if the power $1/2^i$ on the right-hand sides of the inequalities could be improved.

THEOREM 2. *Let $\mathcal{A}_i$ denote the equivalence class of k-graphs for which P_i holds. Then*

$$\mathcal{A}_0 \supset \mathcal{A}_1 \supset \mathcal{A}_2 \cdots \supset \mathcal{A}_k.$$

The family $\mathcal{A}_i = \mathcal{A}_i^{(k)}$ of k-graphs is said to be (k,i)-quasi-random or, sometimes, i-quasi-random if there is no confusion. The term "k-quasi-random" for k-graphs is the same as "quasi-random" in previous papers.

Here we describe the constructions of k-graphs G_i, separating class $\mathcal{A}_i$ from $\mathcal{A}_{i+1}$. It is used in a later section on lower bounds for communication complexity. Since $P_i \Rightarrow P_{i+1}$ for any i, we have that $\mathcal{A}_i \supseteq \mathcal{A}_{i+1}$. To show that $\mathcal{A}_i \supset \mathcal{A}_{i+1}$, for $i = 0, \ldots, k-1$, the idea is to construct k-graphs G_i with the property that $G_i \in \mathcal{A}_i$ and $G_i \notin \mathcal{A}_{i+1}$ using quasi-random graphs as the basic building blocks. In [CG1] two families of quasi-random k-graphs are given, one of which is the Paley k-graph P_k with $V(P_k) = \{1, 2, \ldots, n\}$ (n is a prime) and $\mu_{P_k}(u_1, \ldots, u_k) = 1$ if and only if $u_1 + \cdots + u_k$ is a quadratic residue modulo n.

For each i, we define the k-graph G_i as follows:

$$\begin{aligned} V(G_i) &= V(P_i) = V, \\ E(G_i) &= \{x \in \binom{V}{k} : \mid \binom{x}{i} \cap E(P_i) \mid \equiv 0 \pmod 2\}. \end{aligned}$$

Claim. It holds that $G_i \in \mathcal{A}_i \setminus \mathcal{A}_{i+1}$.

Proof. The proof is divided into two parts.

Part 1 ($G_i \in \mathcal{A}_i$). It is shown in [C] (by using the character sum inequality of Burgess [B]) that

$$\text{dev}_i(G_i) = O(n^{-1/2}).$$

Therefore G_i satisfies Property P_i, and hence is in $\mathcal{A}_i$.

Part 2 ($G_i \notin \mathcal{A}_{i+1}$). Consider the set $E(G_i, P_i)$ of edges of G_i induced by the Paley graph P_i. That an edge x is in $E(G_i, P_i)$ means that every i-subset of x has a sum that is a quadratic nonresidue. By definition, x contains an even number of i-sets, each of which has a sum that is quadratic nonresidue. This can happen only when $\binom{k}{i} \equiv 0 \pmod 2$. Therefore either $E(G_i, P_i)$ is empty or $E(\bar{G}_i, P_i)$ is empty. Since k and i are all fixed integers,

$$\begin{aligned} \mid E(G_i, P_i) - E(\bar{G}_i, P_i) \mid &= \left| E\left(\binom{V}{k}, P_i\right) \right| \\ &= (1 + o(1)) \frac{n^k}{2^{\binom{k}{i}}} \\ &\neq o(n^k). \end{aligned}$$

Thus $G_i \notin \mathcal{A}_{i+1}$.

We now describe a more general construction of k-functions G_i using any quasi-random graph in $\mathcal{A}_k$ as the basic building block.

General construction for $G_i \in \mathcal{A}_i \setminus \mathcal{A}_{i+1}$. Note that the proof of Part 2 is quite general; it does not use the fact that the basic building block was the Paley k-graph P_k. We show here that, in fact, any quasi-random graph in $\mathcal{A}_k$ serves the purpose, as well. (For example, the family of "even intersection" k-graphs defined in [CG1] is an equally good choice.) First, we need the following definition of the "neighborhood graph" of a k-graph. Given a k-graph G, the *neighborhood graph* $G(v)$ of a vertex v is the graph having vertex set $G \setminus \{v\}$ and edge set $E(G(v)) = \left\{x \in \binom{V}{k-1} : x \cup \{v\} \in E(G)\right\}$.

Let H_i be a quasi-random i-graph on n vertices. Then we define the k-graph G_i as follows:

$$\begin{aligned} V(G_i) &= V(H_i) = V, \\ E(G_i) &= \left\{x \in \binom{V}{k} : \left|\binom{x}{i} \cap E(H_i)\right| \equiv 0 \pmod 2\right\} \end{aligned}$$

We outline the proof of $G_{k-1} \in \mathcal{A}_{k-1} \setminus \mathcal{A}_k$.

Part 1 ($G_{k-1} \in \mathcal{A}_{k-1}$). As a direct consequence of the definition of a neighborhood graph, we have that

$$\text{dev}_i(G) = \frac{1}{n} \sum_{v \in V} \text{dev}_i(G(v)).$$

For a fixed vertex v, consider the neighborhood graph $G_{k-1}(v)$ of the k-graph G_{k-1}. The edge set of $G_{k-1}(v)$ can be characterized as follows:

$$E\left(G_{k-1}(v)\right) = E_1 \cup E_2,$$

where

$$E_1 = \left\{ y \in \binom{V}{k-1} : y \in H_{k-1} \text{ and } E(H_{k-1}(v)) \cap \binom{y}{k-2} \equiv 0 \pmod 2 \right\}$$

and

$$E_2 = \left\{ y \in \binom{V}{k-1} : y \notin H_{k-1} \text{ and } E(H_{k-1}(v)) \cap \binom{y}{k-2} \equiv 1 \pmod 2 \right\}$$

Thus

$$(*) \qquad \mu_{G_{k-1}(v)} = \mu_{H_{k-1}} \cdot \mu_{\delta(H_{k-1}(v))},$$

where $\delta(H_{k-1}(v))$ is defined to be

$$\delta(H_{k-1}(v)) = \left\{ y \in \binom{V}{k-1} : E(H_{k-1}(v)) \cap \binom{y}{k-2} \equiv 0 \pmod 2 \right\}.$$

It is easy to verify that $(*)$ implies that

$$\operatorname{dev}_{k-1}(G_{k-1}(v)) = \operatorname{dev}_{k-1}(H_{k-1}).$$

Thus

$$\operatorname{dev}_{k-1}(G_{k-1}) = \sum_v \frac{\operatorname{dev}_{k-1}(G_{k-1}(v))}{n} n = o(1), \quad \text{since} \quad H_{k-1} \in \mathcal{A}_{k-1}.$$

This shows that $G_{k-1} \in \mathcal{A}_{k-1}$.

Part 2. The proof of $G_{k-1} \notin A_k$ is identical to the proof of Part 2 with the Paley graph construction, above.

3. Communication complexity.

3.1. Quasi-random classes of functions. A k-function is a function f from V^k to $\{-1, 1\}$. We note that k-functions can be viewed as ordered k-graphs, and k-graphs can be regarded as symmetric k-functions. In fact, most known lower bound constructions for k-functions are symmetric and thus can be reduced to hypergraphs. We see in the following that the notions of discrepancy and deviation extend to k-functions, as well. For convenience, we use the same notation (disc and dev) for discrepancy and deviation of k-functions. Thus, for example, disc(f) refers to the deviation of a k-function f, whereas disc(G) denotes that of a k-graph G.

Let I denote a subset of size i of $\{1, \ldots, k\} = [k]$. For a k-tuple $x = (x_1, \ldots, x_k)$, we define x_I to be an i-tuple $(x_{a_1}, \ldots, x_{a_i})$, where $a_1 < \cdots < a_i$ and $a_i \in I$.

Discrepancy. Let $\mathcal{H}_i$ denote a family of i-functions, where $i < k$ and the members of $\mathcal{H}_i$ are indexed by $\binom{[k]}{i}$, denoted by h_I. We define $E(f, \mathcal{H}_i)$ as follows:

$$E(f, \mathcal{H}_i) = \left\{ x \in V^k : f(x) = -1 \text{ and for every } h_I \in \mathcal{H}_i,\ h_I(x_I) = -1 \right\}.$$

We denote the cardinality of $E(f,\mathcal{H}_i)$ by $e(f,\mathcal{H}_i)$. The i-discrepancy of f is defined as follows:

$$\operatorname{disc}_i(f) = \max_{\mathcal{H}_{i-1}} \frac{\mid e(f,\mathcal{H}_{i-1}) - e(-f,\mathcal{H}_{i-1}) \mid}{\mid V \mid^k}.$$

Deviation. Define $\prod_{f,I}^{(i)} : V^{k+i} \to \{-1,1\}$ by

$$\prod_{f,I}^{(i)}(x_1,\dots,x_{k+i}) = \prod_{\epsilon_1}\cdots\prod_{\epsilon_k} f(\epsilon_1,\dots,\epsilon_k),$$

where $\epsilon_j \in \{x_{j+m-1}, x_{j+m}\}$ if $j \in I$ and $m =\mid I \cap [1,j] \mid$, and $\epsilon_j = x_{i+m}$ if $j \notin I$. The i-deviation of f is defined to be

$$\operatorname{dev}_i(f) = \max_I \frac{1}{n^{k+i}} \sum_{x_1,\cdots,x_{k+i}} \prod_{f,I}^{(i)}(x_1,\dots,x_{k+i}),$$

where I ranges over all subsets of $[k]$ of size i.

For fixed i, we consider the following properties for a k-function:

$$\begin{aligned} \tilde{R}_i \ &: \ \text{For } i \geq 2, \text{for every family } \mathcal{H}_{i-1} \text{ of } (i-1)\text{-functions,} \\ &\quad\ e(f,\mathcal{H}_{i-1}) - e(-f,\mathcal{H}_{i-1}) = o(n^k), \\ \tilde{P}_i \ &: \ \operatorname{dev}_i(f) = o(1). \end{aligned}$$

It can be shown that properties $\tilde{R}_i$ and $\tilde{P}_i$ are equivalent. In fact, the analogues of Theorems 1 and 2 for k-functions also hold (see [C]).

3.2. Multiparty communication games. In [BNS], Babai, Nisan, and Szegedy considered the communication complexity for k-functions, where each of the k players knows exactly $k-1$ inputs. Let $x = (x_1,\cdots,x_k)$ denote an input chosen uniformly over all k-tuples. Then the communication complexity is bounded by $\log 1/\Gamma(f)$, where

$$\Gamma(f) = \max_S (\Pr[x \in S \text{ and } f(x) = -1] - \Pr[x \in S \text{ and } f(x) = 1]),$$

where S ranges over so-called "cylinder intersections." The theorem below generalizes the result of [BNS].

We first extend the notion of "cylinders" and "cylinder intersections" for functions in class A_i. A subset of $S^{(i-1)}$ of k-tuples is called a *cylinder* if membership in $S^{(i-1)}$ depends only on $i-1$ coordinates. Thus, based on which $i-1$ of the coordinates the k-tuple depends, there are $\binom{k}{i-1}$ types of $S^{(i-1)}$ in A_i. Furthermore, a subset of k-tuples is a *cylinder intersection* if it can be represented as an intersection of cylinders. Let $\cap S^{(i-1)}$ represent a subset that is an intersection of all $\binom{k}{i-1}$ types of cylinders. We define $\Gamma_i(f)$ of f to be

$$\Gamma_i(f) = \max_{\cap S^{(i-1)}} \left(\Pr[x \in \cap S^{(i-1)} \text{ and } f(x) = -1] - \Pr[x \in \cap S^{(i-1)} \text{ and } f(x) = 1]\right).$$

Let I denote the subset of i coordinates on which $S^{(i)}$ depends. Then we have the following natural correspondence between cylinders $S^{(i)}$ and i-functions h_I, for $i = 1,\dots,k-1$:

$$x \in S^{(i)} \quad \Leftrightarrow \quad h_I(x_I) = -1$$

and

$$x \in \cap S^{(i)} \quad \Leftrightarrow \quad \text{for every } h_I \in \mathcal{H}_i,\ h_I(x_I) = -1.$$

This enables us to prove the following.

THEOREM 3. *For* $i = 2, \ldots, k$,

$$\begin{aligned} \Gamma_i(f) &= \text{disc}_i(f), \\ C_i(f) &\geq \log \frac{1}{\text{disc}_i(f)}, \end{aligned}$$

where $C_i(f)$ *denotes the communication complexity of* f *in class* A_i.

Proof. Since x is chosen uniformly over all 2^{mk} possible k-tuples, we have that

$$\begin{aligned} \Gamma_i(f) &= \max_{\cap S^{(i-1)}} \left(\Pr[x \in \cap S^{(i-1)} \text{ and } f(x) = -1] - \Pr[x \in \cap S^{(i-1)} \text{ and } f(x) = 1]\right) \\ &= \max_{\cap S^{(i-1)}} \frac{1}{2^{mk}} \Big[\mid \{x : x \in \cap S^{(i-1)} \text{ and } f(x) = -1\} \mid \\ &\qquad - \mid \{x : x \in \cap S^{(i-1)} \text{ and } f(x) = 1\} \mid \Big] \\ &= \max_{\mathcal{H}_{i-1}} \frac{1}{n^k} \left[e(f, \mathcal{H}_{i-1}) - e(-f, \mathcal{H}_{i-1})\right] \\ &= \text{disc}_i(f). \end{aligned}$$

The second part of this proof is similar to that of Lemma 2.2 in [BNS]; we include it here for completeness. Let P be any valid protocol for the given function f. We denote by $P(x)$ the value of $f(x)$ as computed by the protocol P. Let N be the number of different possible strings that may be written on the board by P. We want to prove that $N \geq 1/\Gamma_i(f)$. With each string s, we associate $X_{P,s}$, the set of inputs for which s gets written on the board by P. It is easy to see that $X_{P,s}$ is a *cylinder intersection* $\cap S^{(i-1)}$.

Let x be chosen uniformly over all k-tuples. Since P is a valid protocol,

$$\mid \Pr[P(x) = f(x)] - \Pr[P(x) \neq f(x)] \mid = 1.$$

We can estimate the same by summing over different $X_{P,s}$ as follows:

$$\begin{aligned} &\mid \Pr[P(x) = f(x)] - \Pr[P(x) \neq f(x)] \mid \\ &\qquad \leq \sum_s \mid \Pr\left[P(x) = f(x) \text{ and } x \in X_{P,s}\right] - \Pr\left[P(x) \neq f(x) \text{ and } x \in X_{P,s}\right], \end{aligned}$$

where s ranges over all possible strings that may be written. Thus

$$\begin{aligned} 1 &\leq \sum_s \mid \Pr\left[P(x) = f(x) \text{ and } x \in X_{P,s}\right] - \Pr\left[P(x) \neq f(x) \text{ and } x \in X_{P,s}\right] \\ &= \sum_s \Pr\left[f(x) = 1 \text{ and } x \in X_{P,s}\right] - \Pr\left[f(x) = -1 \text{ and } x \in X_{P,s}\right] \\ &\leq \sum_s \Gamma_i(f), \text{ since } X_{P,s} \text{is a cylinder intersection} \\ &= N\Gamma_i(f). \end{aligned}$$

This proves that

$$C_i(f) = \log N \geq \log \left[\frac{1}{\Gamma_i(f)}\right]. \qquad \square$$

We note that we do not restrict the number of players. Suppose that we consider the minimum number $C_{k,i}(p)$ of bits required to be exchanged for some p players, each knowing at most $i-1$ inputs of a k-function. It is easy to see that $C_{k,i}(p) = C_{k,i}(p')$ if $p' > p$. Moreover, $C_{k,i}(p'') > C_{k,i}(p)$ if $p'' < p$.

Fact. For any k-function f, $C_i(f) \leq (k-i+1)m$.

Proof. If $(k-i+1)$ inputs get written on the board, then *some* player would know all k inputs. This could be done, trivially, if a player always writes an input that is not already present on the board.

THEOREM 4. *For a random k-function* $f, C_i(f) \geq ((k-i+1)/2)m$.

Proof. For a random k-function f, it is easy to verify that, with probability approaching 1, we have that $|e(f,H) - e(-f,H)| = O(n^{(k+i-1)/2})$ for every $(i-1)$-function H, and this is the best possible. Using similar methods as in [ESp], this implies that

$$\begin{aligned} \operatorname{disc}_i(f) &= \max_{\mathcal{H}_{i-1}} \frac{|e(f,\mathcal{H}_{i-1}) - e(-f,\mathcal{H}_{i-1})|}{n^k} \\ &= O(n^{(-k+i-1)/2}) \\ &= O(2^{(-k+i-1)m/2}). \end{aligned}$$

Hence

$$C_i(f) = \Omega\left(\frac{(k-i+1)}{2}m\right).$$

In [BNS] examples of functions f with $C_k(f) = \Omega(m/2^k)$ are given. Here we give a short proof for the following "box-product" of functions.

Box-product of k-functions and deviation. Let $f : V^k \to \{-1,1\}$ and $g : W^k \to \{-1,1\}$ be two k-functions. We define $f\square g : (V \times W)^k \to \{-1,1\}$ to be the following k-function:

$$f\square g\,((x_1,y_1),\dots,(x_k,y_k)) = f(x_1,\dots,x_k)\cdot g(y_1,\dots,y_k).$$

It can be shown that (see also [CG2])

$$\operatorname{dev}_i(f\square g) = \operatorname{dev}_i(f)\cdot \operatorname{dev}_i(g).$$

Example 1. Consider the graph G on three vertices v_1, v_2, v_3, with the edges $\{v_1,v_2\}$ and $\{v_2,v_3\}$; let $V = \{v_1,v_2,v_3\}$ and f denote the edge function of G. It is easy to check that $\operatorname{dev}_0(f) = \operatorname{dev}_1(f) = 1/9$. Taking the box-product of f with itself gives us the function $f' = f\square f$ with the properties $\operatorname{dev}_0(f') = \operatorname{dev}_1(f') = 1/81$.

Example 2. Consider the following "generalized inner product function" f_m, defined on subsets S_i of a set of size m:

$$f_m(S_1,\dots,S_k) = \begin{cases} 1 & \textit{if } S_1 \cap \cdots \cap S_k \text{ is even,} \\ -1 & \text{otherwise.} \end{cases}$$

For the special case where $m = 1$, f_1, each S_i is a singleton or empty. It is easy to verify, by induction on m, that

$$f_m = f_1 \square \cdots \square f_1 \qquad (m \text{ times}).$$

Since $\mathrm{dev}_i(f_1) = 1 - 2^{-k-i+1} = c < 1$, we readily obtain that $\mathrm{dev}_i(f_m) < c^m$. In particular, $\mathrm{dev}_k(f_m) < c^m$, where $c < 1$. This implies that $\mathrm{disc}_k(f_m) < c^{m/2^k}$. By Theorem 3,

$$C_k(f_m) \geq \log \frac{1}{\mathrm{disc}_k(f_m)} = \Omega\left(\frac{m}{2^k}\right).$$

Therefore we prove the following theorem.

THEOREM 5. *The generalized inner product function f_m has $C_k(f_m) = \Omega(m/2^k)$.*

One of the main results in [BNS] is to establish an upper bound for $\mathrm{disc}_k f_m$, and thereby obtain a lower bound for $C_k(f_m)$. Independently, an upper bound for $\mathrm{disc}_k f_m$ is also proved in [CG1]. However, both the proofs are more complicated in comparison to the one described above. The significance of the box-product is thus apparent. Starting with a function with $\mathrm{dev}_i < 1$, we can construct functions with exponentially small dev_i by repeatedly considering the box-product of the original function with itself.

The following result shows that Theorem 5 is an instance in a more general setting.

THEOREM 6. *There are explicit k-functions f satisfying*

$$C_i(f) = \Omega\left(\frac{m}{2^i}\right).$$

Proof. Recall from §2.2 that we constructed k-graphs $G_i \in \mathcal{A}_i \setminus \mathcal{A}_{i+1}$ for which

$$\mathrm{dev}_i(G_i) = O(n^{-1}).$$

In terms of k-functions, this implies that $\mathrm{dev}_i(f_{k,i}) = O(2^{-m})$. So

$$\begin{aligned} \mathrm{disc}_i(f_{k,i}) &\leq (\mathrm{dev}_i)^{1/2^i} \\ &= O(2^{-m/2^i}). \end{aligned}$$

This implies that $C_i(f_{k,i}) = \Omega(m/2^i)$.

Remark. One of the important tasks is to find communication complexity lower bounds that do not decrease exponentially in k for some explicit k-function. This would improve results [BNS] on pseudorandom sequences, time-space trade-offs for multihead Turing machines, and length-width trade-offs for oblivious branching programs. Improving the relation (Theorem 1) between disc_i and dev_i would be significant for the same reason.

3.3. Application to Turing machines. Let f be a k-function. Under our general communication model, we have the following analogue of the result of Babai, Nisan, and Szegedy [BNS] for the time-space trade-off of Turing machines, and we omit the proofs here.

LEMMA 1. *Any i-head Turing machine that computes a k-function f from the following input*:

$$< x_1 > * * ** < x_2 > * * * * \cdots * * * * < x_k >$$

(where $* * **$ *means l spaces on the input tape) requires a time-space trade-off of* $TS \geq lC_{i+1}(f)/i$.

Hence we have the following result.

THEOREM 7. *For any fixed i, any i-head Turing machine computing the k-function* $f_{k,i}$ *requires a time-space trade-off of* $TS \geq \Omega(m^2)$.

3.4. Discrepancy and the switching lights model. There is yet another interpretation for disc_i in terms of the *switching lights* model, first described in [Sp] for the two-dimensional case. The game consists of an $n \times n$ array A of lights and $2n$ switches, one for each row x_i and column y_j. Each switch, when thrown, changes each light in its line from *off* to *on*, or from *on* to *off*. The *difference* is defined as the absolute value of the number of lights that are on *minus* the number of lights that are off, ranging over all possible settings of the switches. Given an initial configuration, the goal is to maximize the difference. Mathematical formulation of this problem shows that maximizing this difference corresponds to computing the discrepancy (the Γ function) in the multiparty communication model in a sense made precise in the theorem below.

Consider a k-dimensional array of n^k lights. Imagine each switch controlling an i-dimensional hyperplane of n^i lights; i.e., each switch, when thrown, changes each light in the particular hyperplane from *off* to *on*, or from *on* to *off*. There are $(i+1)n^{k-i}$ such switches, and the aim is to maximize the difference between the number of lights *on* and *off*. We denote this by D_k^i. Thus, in k-dimensions, we formulate $k-1$ discrepancy problems associated with the switching game.

In three-dimensions, we have two problems: D_3^2 and D_3^1. The distinction is that each switch controls a plane of lights in one case, and a line of lights in the other. Intuitively, we would expect D_3^1 to be higher than D_3^2, and the intuition is right. The mathematical formulation of this case (D_3^2) is as follows.

Let the array of n^3 lights be represented by $A(ijk) = \pm 1$, for $i, j, k = 1, \ldots, n$. Thus 1 represents a light *on* and -1 a light *off*. Furthermore, we let x_i, y_j, z_k represent the $3n$ switches. "Throwing" a switch x_i corresponds to setting $x_i = -1$. Given an initial setting of $A(i,j,k) = \pm 1$, we define the *discrepancy* of A to be

$$D(A) = \max_{x_i, y_j, z_k = \pm 1} A(i,j,k) \cdot x_i y_j z_k,$$

i.e., the maximum difference between the number of lights *on* and *off* that we can obtain by throwing the switches. Furthermore, we define

$$D_3^2 = \min_A \max_{x_i y_j z_k} A(i,j,k) x_i y_j z_k$$

to be the maximum ranging over all possible initial configurations of A. The case of D_k^i for general i has a similar mathematical formulation.

The following theorem establishes the equivalence between D_k^i and the "discrepancy" Γ_i in the context of multiparty communication complexity. First, we associate with a given k-input function f, the k-dimensional array A_f of size $2^m \times \cdots \times 2^m$, where

$$A_f(i_1, \ldots, i_k) = f(x_1 = i_1, \ldots, x_k = i_k).$$

Thus we are assuming (without loss of generality) that each input x_j ranges from 1 to 2^m. We then have the following theorem.

THEOREM 8. *We have*

$$\Gamma_i(f(m)) = \frac{1}{2^{mk}} D_k^{k-i}(A_f).$$

Proof. Basically, the number of inputs each player knows corresponds to the number of coordinates required to specify a switch, and the possible bit sequences by the players correspond to the switch settings. We describe the proof for $i = k - 1$. The general case is quite similar and will be omitted. It is easy to see that Γ_{k-1} can be rewritten as follows (see [BNS]):

$$\Gamma_{k-1}(f(m)) = \max_{\phi_1,\dots,\phi_k} |E\,[f(x_1,\dots,x_k)\phi_1(x_1,\dots,x_k)\dots\phi_k(x_1,\dots,x_k)]\,|,$$

where the expectation is over all possible 2^{mk} choices of $x_1,\cdots,x_k$, and the maximum is taken over all functions $\phi_j : (\{0,1\}^m)^k \to \{0,1\}$ such that ϕ_j does not depend on x_j. (Intuitively, ϕ_j corresponds to possible messages communicated by player P_i.) Thus

$$\Gamma_{k-1}(f(m)) = \frac{1}{2^{mk}} \max_{\phi_1,\dots,\phi_k} |\sum_{x_1}\cdots\sum_{x_k}[f(x_1,\dots,x_k)\phi_1(x_1,\dots,x_k)\dots\phi_k(x_1,\dots,x_k)]\,|,$$

whereas *discrepancy* of A_f in the switching game is defined as

$$D_k^1(A_f) = \max_{s_{i_1},\dots,s_{i_k}} \sum_{i_1=1}^{2^m}\cdots\sum_{i_k=1}^{2^m} A(i_1,\dots,i_k)s_{i_1}\dots s_{i_k},$$

where the switch $s_{i_j} : \{1,\dots,2^m\}^k \to \{0,1\}$ depends on all but index i_j. It is now easy to see that the functions ϕ_j correspond to the switches s_{i_j}. Thus $\Gamma_{k-1}(f(m)) = (1/2^{mk})D_k^1(A_f)$. □

The following theorem appears in [ESp] in the form of a result on a hypergraph-coloring problem.

THEOREM 9. *There exist arrays A of n^k lights such that*

$$D_k^i(A) \le c(k,i)n^{(k+i-1)/2},$$

where $c(k,i)$ is an explicit constant depending on k and i.

Proof. The proof is straightforward using the probabilistic method and can be found in [T].

Remark 1. Theorem 7 shows that, for a random k-function f, $\text{disc}_i(f) = O\left(n^{(k+i-1)/2}\right)$. Thus this yields a simple proof of

$$\begin{aligned} C_i(f) &\ge \log\left(n^{(k+i-1)/2}\right) \\ &= \log\left(2^{(k+i-1)m/2}\right) \\ &= \frac{(k+i-1)}{2}m. \end{aligned}$$

Remark 2. Note that Theorem 9 guarantees the *existence* of an array A such that $D_k^i(A) \le cn^{(2k-i)/2}$. Can we, in fact, construct such an array? The question is open for $k > 2$. For $k = 2$, it is known that an $n \times n$ *Hadamard matrix* H works! That is,

$$D_2^1(H) \le n^{3/2}.$$

However, it is not clear how to generalize the notion of Hadamard matrices for the case of $k > 2$. Apart from being an interesting derandomization question, this has the following implications. In view of Theorem 8, upper bounds on D_k^i yield, in turn, upper bounds on Γ_i, and, furthermore, give lower bounds on the communication complexity of multiparty protocols. Thus, making Theorem 5.1 constructive seems to be an interesting open problem.

Remark 3. The inequality in Theorem 7 is the best possible. That is, given any arbitrary initial configuration for the array of lights, we can set the switches such that the maximum difference is $\Omega(n^{(k+i-1)/2})$. In fact, the random configuration achieves the bound that can be proved by generalizing the result in [ESp]. The method of conditional expectations can be used in derandomizing the algorithm and a sequential as well as a parallel algorithm is described in [T] to achieve the optimal setting of the switches.

4. Problems and remarks. In addition to various problems that were mentioned in previous sections, many problems and directions remain to be explored. It would be of interest to establish relations and connections with other complexity problems. For example, an interesting relation between circuit complexity and quasi randomness has been demonstrated through some recent work of Hastad and Goldmann [HG]. Using the results of [BNS], Hastad and Goldmann show that (among other things) evaluating the generalized inner product function on $k+1$ inputs by a depth 3 unweighted threshold circuit with bottom fanin at most k would require size $2^{\Omega(n/k4^k)}$. One way to improve these lower bounds is to derive explicit hypergraphs or k-functions with smaller discrepancy or higher communication complexity.

Although we deal with hypergraphs with the edge density 1/2, the results can easily be generalized to hypergraphs or functions with any fixed edge density α, for $0 < \alpha < 1$. For a function f from V^k to $\{-1, 1\}$, we define $f_\alpha(x) = 1 - \alpha$ if $f(x) = -1$ and $f_\alpha(x) = -\alpha$ if $f(x) = 1$. In [C], $\text{dev}_i f_\alpha$, $\text{disc}_i f_\alpha$, and the class $\mathcal{A}_{i,\alpha}$ are defined analogous to dev_i, disc_i, and $\mathcal{A}_i$. In particular, the 2-discrepancy $\text{disc}_{2,\alpha}$ is described as follows:

$$\text{disc}_{2,\alpha}(f) = \max_{X \subseteq V} \frac{e(f, X) - \alpha \mid X \mid^k}{\mid X \mid^k},$$

where $e(f, X) = \mid \{x \in \binom{X}{k} : f(x) = -1\} \mid$. Suppose that we choose α to be $e(f, X) = \mid \{x \in V^k : f(x) = -1\} \mid / \mid V \mid^k$ (which can be viewed as the density of "ordered" hyperedges). Then $\text{disc}_{2,\alpha}(f)$ associates with the maximum quantity that the number of ordered-edges in a subset X can differ from the average. If we can use $\text{dev}_{2,\alpha}$ to (upper) bound $\text{disc}_{2,\alpha}(f)$, then we can (lower) bound the number of edges leaving X from every $X \subseteq V$ and thus assert the expanding property of the hypergraphs.

REFERENCES

[B] D. A. Burgess, *On character sums and primitive roots*, Proc. London Math. Soc., 12 (1962), pp. 179–192.

[BFS] L. Babai, P. Frankl, and J. Simon, *Complexity classes in communication complexity theory*, in Proc. 27th IEE Sympos. on the Foundation of Computer Science, Toronto, Ontario, Canada, 1986, pp. 337–347.

[BNS] L. Babai, N. Nisan, and M. Szegedy, *Multiparty protocols and logspace-hard pseudorandom sequences*, in Proc. 21st Annual ACM Sympos. on the Theory of Computing, Seattle, WA, 1989, pp. 1–11.

[C] F. R. K. Chung, *Quasi-random classes of hypergraphs*, Random Structures Algorithms, 1 (1990), pp. 363–382.

[CFL] A. K. Chandra, M. L. Furst, and R. J. Lipton, *Multiparty protocols*, in Proc. 24th IEE Sympos. on the Foundation of Computer Science, Tucson, AZ, 1983, pp. 94–99.

[CGW] F. R. K. Chung, R. L. Graham, and R. M. Wilson, *Quasi-random graphs*, Combinatorica, 9 (1989), pp. 345–362.

[CG1] F. R. K. Chung and R. L. Graham, *Quasi-random hypergraphs*, Random Structures Algorithms, 1 (1990), pp. 105–124.

[CG2] ———, *Quasi-random set systems*, J. Amer. Math. Soc., 4 (1991), pp. 151–196.

[ESp] P. Erdos and J. Spencer, *Imbalances in k-colorations*, Networks, 1 (1971), pp. 379–385.

[HG] J. Hastad and M. Goldmann, *On the power of small-depth threshold circuits*, in Proc. 31st IEE Sympos. on the Foundation of Computer Science, St. Louis, MO, 1990, pp. 610–618.

[HMT] A. Hajnal, W. Maass, and G. Turán, *On the communication complexity of graph properties*, in Proc. 20th Annual ACM Sympos. on the Theory of Computing, Chicago, IL, 1988, pp. 186–191.

[MS] K. Melhorn and E. M. Schmidt, *Las Vegas is better than determinism in* VLSI *and distributed computing*, in Proc. 14th Annual ACM Sympos. on the Theory of Computing, San Francisco, CA, 1982, pp. 330–337.

[L] L. Lovász, *Computational Complexity : A Survey, in Paths, Flows, and* VLSI-*Layout*, B. Korte et al., eds., Springer-Verlag, Berlin, New York, 1990, pp. 235–266.

[LS] L. Lovász and M. Sals, *Lattices, Möbius functions and communication complexity*, in Proc. 29th Sympos. on the Foundation of Computer Science, White Plains, NY, 1988, pp. 81–90.

[PS] C. H. Papadimitriou and M. Sipser, *Communication Complexity*, in Proc. 14th Annual ACM Sympos. on the Theory of Computing, Boston, MA, 1983, pp. 196–200.

[T] P. Tetali, *Analysis and applications of probabilistic techniques*, Ph.D. thesis, New York University, New York, 1991.

[SP] J. Spencer, *Ten Lectures on the Probabilistic Method*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1987.

[Th] C. D. Thompson, *Area-time complexity for* VLSI, in Proc. 11th Annual ACM Sympos. on the Theory of Computing, Atlanta, GA, 1979, pp. 81–88.

[Y] A. C. C. Yao, *Some complexity questions related to distributive computing*, in Proc. 11th Annual ACM Sympos. on the Theory of Computing, Atlanta, GA, 1979, pp. 209–213.

SIAM J. DISC. MATH
Vol. 6, No. 1, pp. 124–138, February 1993

DISCRETE LOGARITHMS IN $GF(P)$ USING THE NUMBER FIELD SIEVE*

DANIEL M. GORDON†

Abstract. Recently, several algorithms using number field sieves have been given to factor a number n in heuristic expected time $L_n[1/3; c]$, where

$$L_n[v; c] = \exp\{(c + o(1))(\log n)^v(\log\log n)^{1-v}\}$$

for $n \to \infty$.

This paper presents an algorithm to solve the discrete logarithm problem for $GF(p)$ with heuristic expected running time $L_p[1/3; 3^{2/3}]$. For numbers of a special form, there is an asymptotically slower but more practical version of the algorithm.

Key words. discrete logarithms, number field sieve

AMS(MOS) subject classification. 11Y16

1. Introduction. Given a prime p and integers a and b, the discrete logarithm problem in $GF(p)$ is to find an integer x (if any exists) such that

$$a^x \equiv b \pmod p. \tag{1}$$

The difficulty of computing discrete logarithms has been used in the construction of several cryptographic systems (see, for example, [15]). The most successful implementation of a discrete logarithm algorithm for $GF(p)$ to date is by LaMacchia and Odlyzko [11], who solved the discrete logarithm problem modulo primes of 58 and 67 digits using the Gaussian integers method. This method, introduced by Coppersmith, Odlyzko, and Schroeppel in [8], uses a complex quadratic field to aid the sieving process.

Define

$$L_x[v; c] = \exp\{(c + o(1))(\log x)^v(\log\log x)^{1-v}\}, \tag{2}$$

for $x \to \infty$. The Gaussian integers method, as well as several other methods described in [8], find discrete logarithms for $GF(p)$ in expected time $L_p[1/2; 1]$.

The idea of using number field sieves has been used recently for factoring. Lenstra et al. [13] have used a number field sieve to obtain rapid factorizations of numbers of the form $r^e \pm s$, for small r and s. Buhler, Lenstra, and Pomerance [5] have generalized this method to factor general numbers n in time $L_n[1/3; c]$. Adleman [1] and Coppersmith [7] have suggested further improvements.

Some necessary facts and heuristic assumptions about algebraic number theory and linear algebra computations are discussed in §2. In §3 an overview of an algorithm for computing discrete logarithms in $GF(p)$ using the number field sieve is given. Using these results and assumptions, §4 shows that the algorithm works in expected time $L_p[1/3; 3^{2/3}]$. Another version for special numbers, which is asymptotically slower but more practical, is given in §5.

2. Computational background. There are a number of specialized algorithms and heuristic assumptions that are needed to give a good running time for finding discrete logarithms with the number field sieve. Similar assumptions are used in [13] for estimating the time needed to factor with the number field sieve.

*Received by the editors April 9, 1990; accepted for publication (in revised form) February 26, 1992.

†Department of Computer Science, University of Georgia, Athens, Georgia 30602.

2.1. Smoothness. Call an integer y-smooth if all of its prime factors are at most y. Let $\psi(x, y)$ be the number of integers $\leq x$ that are y-smooth. We need results on the probabilities of various rational and algebraic integers being smooth. The following special case of a theorem of Canfield, Erdős, and Pomerance [6] gives an estimate for the probability of a number in a given range being smooth.

THEOREM 1. *Suppose that* $0 < w < v \leq 1, \gamma > 0$, *and* $\delta > 0$ *are fixed. Let* x *and* y *be functions of* p *such that* $x = L_p[v; \gamma]$ *and* $y = L_p[w; \delta]$ *for* $p \to \infty$. *Then*

$$\frac{\psi(x,y)}{x} = L_p[v-w; -\frac{\gamma}{\delta}(v-w)] \qquad \textit{for } p \to \infty.$$

The ratio $\psi(x, y)/x$ is the probability that a random number in $(0, x]$ is y-smooth. In this paper, we deal with numbers near x that are not random, but we use the heuristic assumption that their probability of being smooth is also given by Theorem 1. For example, we assume that numbers of the form $c + dm$, for c and d running through a narrow range and m fixed, are smooth as often as random numbers of the same size.

The elliptic curve method (ECM) for factoring an integer n depends on finding an elliptic curve for which the order of the curve modulo a prime divisor of n is smooth (see [14]). The following conjecture implies that enough such curves exist so that the ECM can expect to find one in reasonable time.

CONJECTURE 1. *Given the conditions of Theorem* 1, *the probability that a random number in* $(x - \sqrt{x}, x + \sqrt{x})$ *is* y*-smooth is* $L_p[v - w; -\gamma/\delta(v - w)]$ *for* $p \to \infty$.

This conjecture implies the following special case of Conjecture 2.10 of [14].

CONJECTURE 2. *The expected time for the* ECM *to factor an* $L_p[v; c]$*-smooth integer in* $[0, p]$ *is* $L_p[v/2; \sqrt{2vc}]$ *for* $p \to \infty$.

2.2. Linear algebra. Another operation that will take a large part of the computation time is dealing with matrix equations over $\mathbb{Q}$. Given an $S \times T$ sparse integer matrix A, where $S > T$ and the entries in A are all at most T in absolute value, must find a linear relation over $\mathbb{Q}$ for the rows of A. This may be done by the following algorithm, due to Pomerance [17] (see [10] for an alternative algorithm).

ALGORITHM M. Let A be a $(T + 1) \times T$ matrix over $\mathbb{Z}$, with each row having at most E nonzero entries, each of absolute value at most T. This probabilistic algorithm returns a linear relation for the rows of A.

Step 1. Attempt to compute the rank r of A.

Choose a random prime $q_0 \leq ET \log T$. By using Gaussian elimination mod q_0, find the rank r_0 of A mod q_0. Rearrange the rows so that the first r_0 rows are linearly independent mod q_0. Call the rearranged rows $v_1, v_2, \cdots, v_{T+1}$. The result of the Gaussian elimination determines an $r_0 \times r_0$ submatrix $\hat{A}$ of the first r_0 rows of A such that $\hat{A}$ is nonsingular mod q_0.

Step 2. Attempt to express v_{r_0+1} as a linear combination of $v_1, \cdots, v_{r_0}$ mod q for each prime $q \leq ET \log T$.

We attempt this via Wiedemann's coordinate recurrence method [21]. Let $\mathbf{P}$ denote the product of the primes q for which we are successful, and let $\mathbf{P}'$ denote the product of the remaining primes up to $ET \log T$. If $\mathbf{P}' > (E^{1/2}T)^T$, then return to Step 1 and begin again.

Step 3. Attempt to compute the determinant D of $\hat{A}$.

For each prime $q|\mathbf{P}$, use Wiedemann's probabilistic determinant algorithm [21] to compute an integer $D_q \in \{0, 1, \cdots, q - 1\}$, which is the determinant of $\hat{A}$ mod q with probability at least $1 - (ET)^{-2}$. Use the Chinese remainder theorem to compute the integer D_0 closest to zero with $D_0 \equiv D_q \bmod q$ for each prime $q|\mathbf{P}$. Repeat this step until a value of D_0 is found with $0 < |D_0| \leq (E^{1/2}T)^T$.

Step 4. Attempt to produce a linear relation among the rows of A.

With the Chinese remainder theorem and the results of Steps 2 and 3, compute the integers $c_1, \cdots, c_{r_0}$ closest to zero such that

$$D_0 v_{r_0+1} \equiv \sum_{i=1}^{r_0} c_i v_i \pmod{\mathbf{P}}.$$

If any c_i has absolute value exceeding $(E^{1/2}T)^T$, return to Step 3. Otherwise, we have found the relation

$$D_0 v_{r_0+1} = \sum_{i=1}^{r_0} c_i v_i. \tag{3}$$

THEOREM 2. *Suppose that* $T \geq E \geq 12$. *If Algorithm* M *terminates, then* (3) *is a correct equation. The expected running time of Algorithm* M *is* $O(E^2T^3 \log^3 T)$.

Proof. By the assumptions on A, we have that $\| v_i \| \leq E^{1/2}T$ for each row v_i of A. Thus, by Hadamard's inequality, the absolute value of the determinant of any submatrix of A is at most $(E^{1/2}T)^T$. From results of Rosser and Schoenfeld [18], it follows that the number of distinct prime factors of any such nonzero determinant is less than $2T$. However, from the same reference, the number $\pi(ET \log T)$ of primes $q \leq ET \log T$ exceeds $ET/3$. We can thus conclude that for at least half of the primes $q \leq ET \log T$, the rank of A mod q is equal to the rank r of A over $\mathbb{Q}$. Thus, with probability at least $1/2$, the number r_0 returned in Step 1 is equal to r. The running time for one iteration of Step 1 is $O(T^3 \log^2 T)$ bit operations.

If $r_0 = r$, then v_{r_0+1} is a linear combination of $v_1, \cdots, v_{r_0}$ over $\mathbb{Q}$, and the least common denominator of the rational scalars involved divides the determinant D of $\hat{A}$. Thus, if $r_0 = r$, then $\mathbf{P}' \leq (E^{1/2}T)^T$. If v_{r_0+1} is a linear combination of $v_1, \cdots, v_{r_0}$ mod q, then Wiedemann's coordinate recurrence method will be able to express v_{r_0+1} as such a linear combination in $O(ET^2)$ operations mod q. Thus the running time for one iteration of Step 2 is $O(E^2T^3 \log^2 T)$ bit operations.

Wiedemann's determinant-finding algorithm can calculate the correct determinant with probability at least $1 - (ET)^{-2}$ in $O(ET^2 \log T)$ operations mod q. Among all the numbers D_q computed in Step 3, the probability that at least one such D_q is not congruent to D mod q is at most $\pi(ET \log T)(ET)^{-2}$. From [18] we have $\pi(ET \log T) < 2ET$. Thus the probability that the number D_0 computed in Step 3 is not D is at most $2(ET)^{-1}$. The time for the Chinese remainder theorem is $O(\log^2 \mathbf{P})$, which is $O((ET \log T)^2)$ by [18]. The total time for Step 3 is $O(E^2T^3 \log^3 T)$ bit operations.

If $D_0 = D$, then $D_0 v_{r_0+1}$ is an integral combination of $v_1, \cdots, v_{r_0}$, and the integer scalars $c_1, \cdots, c_{r_0}$ are all at most $(E^{1/2}T)^T$ in absolute value. Since $\mathbf{P} > 2(E^{1/2}T)^T$, knowing those scalars mod $\mathbf{P}$ is enough to determine them. Thus, if $D_0 = D$, then Step 4 will be successful; that is, we will not need to return to Step 3. Furthermore, (3) is a correct equation. The running time of Step 4 is $O(E^2T^3 \log^2 T)$. □

For the special number field sieve, we need only solve matrix equations modulo $p - 1$. This may be done using Wiedemann's algorithm in $O(ET^2 \log^2 T)$ bit operations for matrices satisfying the conditions specified in Algorithm M. If the factorization of $p - 1$ is known, a solution can be found modulo each prime factor, and a solution mod

$p-1$ can be obtained using the Chinese remainder theorem and Hensel's lemma. If not, then Wiedemann's algorithm may be used modulo $p-1$. Either the algorithm will work or it will discover a factor of $p-1$, and the algorithm may be repeated on each factor.

2.3. Algebraic number theory. Throughout this paper, p will be a prime for which we wish to solve the discrete logarithm problem in $GF(p)$. We represent $GF(p)$ by $\mathbb{Z}/p\mathbb{Z}$, where elements are identified with their least nonnegative residues.

We choose an integer m and $f(x) \in \mathbb{Z}[x]$ of degree k such that f is monic, irreducible over $\mathbb{Q}$, and $f(m) \equiv 0 \pmod{p}$. Such an f may be found by choosing an m of suitable size and finding the base m representation of p, say $p = \sum_{i=0}^{k} a_i m^i$. Then $f(x) = \sum_{i=0}^{k} a_i x^i$ satisfies $f(m) = p$ and is irreducible by a theorem of Brillhart, Filaseta, and Odlyzko [4].

We also require that p does not divide Δ_f, the discriminant of f. If this happens for a particular m, we may choose a different m, or alter f by adding m to some a_i and subtracting 1 from a_{i+1}. The irreducibility of the new f may be checked quickly; see [12]. Note that $\Delta_f = (-1)^{k(k-1)/2} R(f, f')$ may be calculated efficiently. $R(f,g)$ here denotes the resultant of f and g.

Let $\alpha \in \mathbb{C}$ denote a root of f, $K = \mathbb{Q}(\alpha)$, and $\mathcal{O}_K$ denote the ring of integers in K. If s is a prime number not dividing the index $[\mathcal{O}_K : \mathbb{Z}[\alpha]]$, then its factorization in $\mathcal{O}_K$ is given by the following proposition (see, for example, [22]).

PROPOSITION 1. *For a prime number s not dividing the index, suppose that f factors in $GF(s)[x]$ as*

$$f(x) \equiv \prod_i g_i(x)^{e_i} \bmod s, \tag{4}$$

with each g_i monic and irreducible mod s, *and $g_i \not\equiv g_j$ for $i \neq j$. Then $(s) = \prod_i \mathfrak{s}_i^{e_i}$, for different prime ideals $\mathfrak{s}_i = (s, g_i(\alpha))$ and $N(\mathfrak{s}_i) = s^{\deg(g_i)}$.*

In particular, since $(p, \Delta_f) = 1$, $\mathfrak{p} = (p, \alpha - m)$ is a first-degree prime factor of (p) in $\mathcal{O}_K$, and we have $\mathcal{O}_K/\mathfrak{p} \cong GF(p)$. We may define a homomorphism φ from $\mathbb{Z}[\alpha]$ to $\mathbb{Z}/p\mathbb{Z}$ as in other number field sieve algorithms, by sending α to m mod p.

We say a prime ideal of $\mathcal{O}_K$ is *bad* if its norm divides the index. All other prime ideals will be called *good*.

Prime numbers dividing the index can be recognized efficiently using a theorem of Dedekind (see [22]): Suppose that f factors mod s as in (4). Then the prime number s divides the index if and only if there is some j for which $e_j \geq 2$ and

$$(g_j \bmod s) \left| \left(s^{-1}(f - \prod_i g_i^{e_i}) \bmod s \right) \right.$$

as elements of $GF(s)[x]$.

For any $y \in \mathbb{Z}$, call an algebraic integer in $\mathbb{Z}[\alpha]$ y-smooth if it is divisible only by good prime ideals of $\mathcal{O}_K$ of norm at most y. We must find smooth numbers of the form $c + d\alpha$, for c and d rational, coprime integers of moderate size.

To do so, we start by attempting to factor

$$\begin{aligned} |N(c+d\alpha)| &= |(-d)^k f(-c/d)| \\ &= |c^k - a_{k-1}c^{k-1}d + \cdots + a_1 c(-d)^{k-1} + a_0(-d)^k| \\ &\leq (k+1) \cdot \max\{|c|, |d|\}^k \cdot \max_i\{|a_i|\}. \end{aligned} \tag{5}$$

PROPOSITION 2. *Suppose that $c, d \in \mathbb{Z}$ are coprime and $N(c+d\alpha)$ is relatively prime to the index $[\mathcal{O}_K : \mathbb{Z}[\alpha]]$. Then $(c + d\alpha)$ factors completely into good first-degree prime ideals in $\mathcal{O}_K$.*

Proof. For each rational prime s dividing $|N(c + d\alpha)|$, there is a unique ideal of norm s dividing $(c + d\alpha)$. This is because, if a prime ideal dividing s divides $(c + d\alpha)$, then $\alpha \equiv -c/d$ modulo the ideal, and since the right side is rational, the congruence holds mod s. Thus $c_s \equiv -c/d \pmod{s}$ is a root of f mod s and, by Proposition 1, determines the unique ideal $\mathfrak{s} = (s, \alpha - c_s)$ dividing $c + d\alpha$.

The norm $N(\mathfrak{s}) = |\mathcal{O}_K/\mathfrak{s}|$ is clearly a power of s. We have $|\mathbb{Z}[\alpha]/(\mathfrak{s} \cap \mathbb{Z}[\alpha])| = s$, since the representatives of classes in $\mathbb{Z}[\alpha]/(\mathfrak{s} \cap \mathbb{Z}[\alpha])$ are just $\alpha, \alpha + 1, \cdots, \alpha + (s - 1)$. Since $|\mathcal{O}_K/\mathbb{Z}[\alpha]|$ is relatively prime to s, $\mathcal{O}_K/\mathbb{Z}[\alpha]$ maps to the identity under reduction mod $\mathfrak{s}$, so $|\mathcal{O}_K/\mathfrak{s}| = s$ as well. Therefore the power of $\mathfrak{s}$ dividing $(c + d\alpha)$ is the same as the power of s dividing the norm. □

For the number fields K we deal with here, the discriminant will be huge, so most operations in K will be impractical. One operation we must be able to do is take a small set of units, given as products of a large number of algebraic integers, and find a multiplicative dependency among them.

Let r_1 be the number of real embeddings of K, let $2r_2$ be the number of complex embeddings, and let $r = r_1 + r_2$. Let $\sigma_1, \cdots, \sigma_{r_1}$ denote the real embeddings, and $\sigma_{r_1+1}, \overline{\sigma_{r_1+1}}, \cdots, \sigma_r, \overline{\sigma_r}$ the others. We define a mapping $l : K \to \mathbb{C}^{r_1+r_2}$ in the usual way, by

$$l(x) = (\log|\sigma_1(x)|, \cdots, \log|\sigma_{r_1}(x)|, 2\log|\sigma_{r_1+1}(x)|, \cdots, 2\log|\sigma_r(x)|).$$

This mapping sends the units in $\mathcal{O}_K$ into a lattice $\mathcal{L} \in \mathbb{R}^r$, with roots of unity mapped to the origin. The following theorem of Dobrowolski [9] shows that other units cannot be too close to the origin.

LEMMA 1. *Let γ be a nonzero algebraic integer in K, and denote by $\overline{|\gamma|}$ the maximal modulus of its conjugates. Then*

$$\overline{|\gamma|} < 1 + \frac{\log k}{6k^2}$$

only if γ is a root of unity.

This implies that, for any unit u that is not a root of unity, $\| l(u) \| > \log(1 + ((\log k)/6k^2)) > 1/(10k^2)$ for $k > 1$.

THEOREM 3. *Suppose that $M > 80rk^2$, and let $u_1, \cdots u_{2r}$ be units in $\mathcal{O}_K$, with $\| l(u_i) \| < M$ for $i = 1, \cdots, 2r$. Then there is a nontrivial linear relation*

$$\sum_{i=1}^{2r} c_i \cdot l(u_i) = \mathbf{0} \tag{6}$$

with each c_i an integer with $|c_i| < M^2$.

Proof. Consider the set S of all sums $\sum_{i=1}^{2r} c_i \cdot l(u_i)$ with $0 \leq c_i < M^2$. There are formally M^{4r} such sums, and it suffices to show that two of them are equal.

For all vectors $s \in S$, we have $\| s \| < 2rM^3$. Therefore all $s \in S$ are in an r-dimensional sphere of radius $2rM^3$, and, by the lemma, no two members of $\mathcal{L}$ are closer than $1/(10k^2)$ to each other. Let $V_r(x)$ denote the volume of an r-dimensional sphere of radius x. Then the number of lattice points in the sphere is at most

$$\frac{V_r(2rM^3 + 1/(20k^2))}{V_r(1/(20k^2))} < (80rk^2M^3)^r = M^{3r}(80rk^2)^r.$$

This is less than M^{4r}, however, and so, by the pigeonhole principle, there must be two equal vectors in S. □

This dependence does not cancel out the units completely, since the resulting unit $\prod u_i^{c_i}$ could be a root of unity. If an lth root of unity is in a field of degree $k \geq \phi(l)$, then we have $l < 6k \log\log k$ by [18]. The root of unity that it is can be determined by calculating the arguments of each $\sigma_r(u_i)$.

If the root of unity is not one, we will look at other vectors $\mathbf{c}'$ until one is found for which $\prod u_i^{c_i'} = 1$. In practice, an lth root of unity could be eliminated by raising the equation to the lth power. We will not do that here, to avoid dealing with the possibility of losing information when l and $p-1$ have a common divisor.

By the above, if $M > 80rk^2$ and we are given $2r$ units $u_1, \cdots, u_{2r}$ with $\| l(u_i) \| < M$ for $i = 1, \cdots, 2r$, then there is a nontrivial relation $\prod_{i=1}^{2r} u_i^{c_i} = 1$ with each c_i an integer with $|c_i| < 6k(\log\log k)M^2$.

Of course, existence is not enough. For the algorithm, we must find such a nontrivial relation. This can be done using an application of the Lenstra–Lenstra–Lovász (LLL) algorithm due to Babai [2]. For a lattice $\mathcal{L}$, let $\lambda(\mathcal{L})$ be the length of the shortest nonzero vector in $\mathcal{L}$.

THEOREM 4. *Let $b_1, \cdots, b_n$ be vectors in $\mathbb{Z}^n$ with Euclidean length less than N, and let $\mathcal{L}$ denote the lattice generated by $b_1, \cdots, b_n$. We can find a vector $v \in \mathcal{L}$ such that*

$$\| v \| \leq (3/\sqrt{2})^n \lambda(\mathcal{L})$$

in time $O\left(n^{5+\epsilon}(\log N)^{2+\epsilon}\right)$, for any $\epsilon > 0$.

This algorithm will be used to find the dependency of Theorem 3. The time estimate is the same as for the LLL algorithm [12], using fast multiplication.

THEOREM 5. *Suppose that $M > 80rk^2$, and let $u_1, \cdots, u_{2r}$ be units in $\mathcal{O}_K$, with $\| l(u_i) \| < M$ for $i = 1, \cdots, 2r$. A nontrivial relation $\prod_{i=1}^{2r} u_i^{c_i} = 1$ can be found in time $O(r^{5+\epsilon}(\log M)^{2+\epsilon})$, for any $\epsilon > 0$.*

Proof. Let $l_m(x)$ denote $l(x)$ with each coordinate l_i replaced by $\lfloor 2^m l_i \rfloor$, and let $\mathcal{L}_m$ be the lattice generated by $l_m(u_1), \cdots, l_m(u_{2r})$.

For $\mathbf{c} = (c_1, c_2, \cdots, c_{2r})$ as in Theorem 3,

$$\| \sum_{i=1}^{2r} c_i \cdot l_m(u_i) \| = \| \sum_{i=1}^{2r} c_i \cdot (2^m l(u_i) + \epsilon_i) \| = \| \mathbf{0} + \sum_{i=1}^{2r} c_i \cdot \epsilon_i \| < 2r^{3/2}M^2,$$

where each ϵ_i is a vector with all coordinates less than 1 in absolute value. We will show that such vectors $\mathbf{c}$ are short vectors in $\mathcal{L}_m$ and that they are sufficiently shorter than other vectors to guarantee that the algorithm of Theorem 4 will find one.

There is a (highly unlikely) possibility that $\sum_{i=1}^{2r} c_i \cdot l_m(u_i) = \mathbf{0}$ for all choices of $c_1, \cdots, c_{2r}$ in Theorem 3, so that the shortest nonzero vector could be longer than $2r^{3/2}M^2$. If the algorithm ever failed because of this, we could repeat it with a lattice $\mathcal{L}_m'$ where one coordinate l_j is replaced by $\lceil 2^m l_j \rceil$ instead of $\lfloor 2^m l_j \rfloor$. By the Gelfond–Schneider theorem (see, for example, [3]) the lattices are different, since $2^m l_j$ cannot be an integer. Therefore no vector $\mathbf{c}$ that is not a root of unity with $c_j \neq 0$ could be zero in both $\mathcal{L}_m$ and $\mathcal{L}_m'$, and at least one lattice (say $\mathcal{L}_m$) has $\lambda(\mathcal{L}_m) < 2r^{3/2}M^2$.

Any vector $\sum_{i=1}^{2r} c_i \cdot l_m(u_i)$ not corresponding to a relation of the form (6) will have one coordinate at least $\lfloor 2^m/10k^2 \rfloor$ in absolute value, by Lemma 1. Taking $2^m > 20k^2r^25^rM^2$, this implies that the vector has length greater than $2r^25^rM^2$.

By Theorem 4, we can find a vector in $\mathcal{L}_m$ of length at most $(3/\sqrt{2})^{2r}\lambda(\mathcal{L}_m)$. However, $2r^2 5^r M^2 > (3/\sqrt{2})^{2r}\lambda(\mathcal{L}_m)$, so the vector found must correspond to a relation (6). □

3. Discrete logarithms in $GF(p)$. The algorithm consists of two main parts. The first is finding the discrete logarithms of a factor base of small rational primes, which only must be done once for a given p. The second actually finds the logarithm of an individual $b \in GF(p)$ by finding the logs of a number of "medium-sized" primes and combining these to find the log of b. In addition, for each number field used (one for the precomputation and several for the individual logarithm calculations), the good degree-one prime ideals of small norm in that field must be determined using the method discussed in §2.

We will assume that a, the base for the discrete logarithm, is B-smooth, where B is a bound for the size of primes in the factor base. If a is not smooth, then we may choose a random number that is smooth over the factor base, call it a', and use it as the base for logarithms instead of a. Then find $\log_{a'} a$, and use the identity

$$\log_a b \equiv \log_{a'} b / \log_{a'} a \pmod{p-1}.$$

If a' is not a generator for $GF(p)^*$, then $\log_{a'} a$ and $\log_{a'} b$ may not exist. If this happens, we just choose another value of a' until we find one for which $\log_{a'} a$ exists. Alternatively, we could factor $p-1$ using the number field sieve factoring algorithm and then test if an a' is a generator by checking that $(a')^{(p-1)/q} \not\equiv 1 \pmod{p}$ for each prime q dividing $p-1$. There is no guarantee that a small generator exists, but Shoup [20] has shown that the extended Riemann hypothesis implies that there is a constant c such that for all primes p, $GF(p)^*$ has a generator less than $c\,\omega(p-1)^4(\log(\omega(p-1))+1)^4\log^2 p$. Here $\omega(n)$ is the number of distinct prime factors of n.

The reason for requiring a to be smooth is to have at least one inhomogeneous relation for the logs of the factor base, using the equation

$$\log_a a = 1 \equiv \sum_{q^t \| a} t \log_a q \pmod{p-1}. \tag{7}$$

3.1. Precomputation. Let p be a prime and a be a primitive element of $GF(p)$. As described in §2.3, choose an integer m and an irreducible monic polynomial $f(x) \in \mathbb{Z}[x]$ such that $(p, \Delta_f) = 1$ and $f(m) \equiv 0 \pmod{p}$. Let $\alpha \in \mathbb{C}$ denote a root of f, $K = \mathbb{Q}(\alpha)$, and $\mathcal{O}_K$ denote the ring of integers in K. Let $\mathfrak{p} = (p, \alpha - m)$, so we have $\mathcal{O}_K/\mathfrak{p} \cong GF(p)$.

The factor base $\mathcal{B}$ will consist of two parts: $\mathcal{B}_{\mathbb{Q}}$ will be rational primes $\leq B$, and $\mathcal{B}_K$ will be good prime ideals in $\mathcal{O}_K$ of degree one and norm $\leq B$. Let $\mathcal{B}'$ denote the subset of $\mathcal{B}_{\mathbb{Q}}$ consisting of the prime factors of a.

For the precomputation stage, we solve for the logarithms of the rational primes. We will do this by sieving through pairs of small integers c and d. A "hit" will be a coprime pair c, d for which $c + dm$ and $c + d\alpha$ are both smooth over $\mathcal{B}$. These can be searched for efficiently by sieving $c + dm$ and $N(c + d\alpha)$. Suppose that we find a c and d for which both are smooth, say

$$c + dm = \prod_{s \text{ prime}, s \leq B} s^{w_s(c,d)} \tag{8}$$

and

$$|N(c + d\alpha)| = \prod_{s \text{ prime}, s \leq B} s^{v_s(c,d)}, \tag{9}$$

for $v_s, w_s \in \mathbb{Z}_{\geq 0}$. By Proposition 2, for each s in (9) with $v_s > 0$ there is a unique ideal $\mathfrak{s}$ in $\mathcal{B}_K$ lying over s and dividing $c + d\alpha$. Let $v_{\mathfrak{s}}(c,d) = v_s(c,d)$ for this ideal, and be zero for other ideals in $\mathcal{B}_K$ of norm s. Thus we have

$$c + dm = \prod_{s \in \mathcal{B}_{\mathbb{Q}}} s^{w_s(c,d)} \tag{10}$$

and

$$(c + d\alpha) = \prod_{\mathfrak{s} \in \mathcal{B}_K} \mathfrak{s}^{v_{\mathfrak{s}}(c,d)}. \tag{11}$$

In the Gaussian integers method, where K is a complex quadratic field with class number one, the factorization into ideals in (11) can be rewritten as a product of algebraic integers in $\mathcal{O}_K$ and one of a few (at most six) units. Then the equations can be related using $\varphi(c + d\alpha) \equiv c + dm \pmod{p}$, and, from enough of these equations, a solution can be determined that gives the logs of every element of $\mathcal{B}$. A similar technique will be used for special p in §5. For the number fields K that we are dealing with here, we must use a different method.

We continue sieving through pairs (c, d) until we have collected more than $|\mathcal{B}|$ equations of the form (10) and (11). Then we form a matrix with the w_s's and $v_{\mathfrak{s}}$'s for each equation as its rows and apply Algorithm M to the submatrix of columns corresponding to elements of $\mathcal{B} - \mathcal{B}'$. In this way, we cancel out all those primes to find equations involving only primes in $\mathcal{B}'$ (the resulting equations could be trivial, but we will use the heuristic assumption that they will behave as if they were random equations). We then have a set $\mathcal{S}$ of pairs (c, d) and integers $x(c, d)$ for $(c, d) \in \mathcal{S}$ such that

$$\prod_{(c,d) \in \mathcal{S}} (c + dm)^{x(c,d)}$$

is divisible only by primes in $\mathcal{B}'$, and

$$\prod_{(c,d) \in \mathcal{S}} (c + d\alpha)^{x(c,d)} = U, \tag{12}$$

where U is a unit in $\mathcal{O}_K$.

After gathering $2r$ equations of the form (12), we may find a combination of these that cancels all the units, by Theorem 5. This results in an equation of the following form:

$$\prod_{c,d} (c + d\alpha)^{y(c,d)} = 1, \tag{13}$$

and so

$$\prod_{c,d} (c + dm)^{y(c,d)} \equiv \prod_{c,d} \varphi(c + d\alpha)^{y(c,d)} \equiv 1 \pmod{p}. \tag{14}$$

Using the factorizations in (10), this gives

$$\prod_{s \in \mathcal{B}'} s^{z_s} \equiv 1 \pmod{p}, \tag{15}$$

where $z_s = \sum_{c,d} w_s(c,d) y(c,d)$.

Taking logs, we have that

$$\sum_{s \in \mathcal{B}'} z_s \log_a s \equiv 0 \pmod{p-1}. \tag{16}$$

Once we have more than $|\mathcal{B}'|$ such equations, we can attempt to solve these homogeneous equations together with (7) and obtain the logs of every prime in $\mathcal{B}'$, using Gaussian elimination modulo $p-1$. If the matrix does not determine a unique solution, we may collect more equations until it does. Since $|\mathcal{B}'| < \log p$, the fact that we must have $|\mathcal{B}'|$ runs of Algorithm M will not affect the complexity analysis.

3.2. Finding individual logarithms. To compute the logarithm of b, we first convert the problem into finding logarithms of "medium-sized" primes. This is done by choosing random integers $l \in [1, p-1]$ until we find one for which

$$a^l b \equiv q_1 q_2 \cdots q_t \pmod{p}, \tag{17}$$

where each of the q_i are moderately sized (say $\leq p^{1/k}$). Then, by finding the discrete logarithms of each q_i, we will obtain the discrete logarithm of b.

For each i, take $m_i = q_i h_i$, where h_i is a number smooth over $\mathcal{B}$ chosen so that m_i is close to $p^{1/k}$. Let $f_i(x)$ be a monic polynomial of degree k such that $f_i(m_i) \equiv 0 \pmod{p}$ and define

$$f_{i,j}(x) = f_i(x) + j(m_i - x).$$

Then $f_{i,j}(m_i) \equiv 0 \pmod{p}$, and, if $f_{i,j}(x)$ is irreducible over $\mathbb{Q}$ and $\alpha_{i,j}$ is a root of $f_{i,j}(x)$, then in $\mathbb{Q}(\alpha_{i,j})$, $|N(\alpha_{i,j})| = |f_{i,j}(0)|$. We sieve through values of j to find ones for which $f_{i,j}(0)$ is B-smooth and continue until we find one with $f_{i,j}(x)$ irreducible, and $(pf_{i,j}(0), \Delta_{f_{i,j}}) = 1$. We will use this polynomial to find the logarithm of q_i.

Once a suitable value of j has been found, the factorization of α_i $(= \alpha_{i,j})$ in $K_i = \mathbb{Q}(\alpha_i)$ gives us the following equations:

$$m_i = q_i h_i \equiv \varphi(\alpha_i) \pmod{p} \tag{18}$$

and

$$(\alpha_i) = \prod_{\mathfrak{s} \in \mathcal{B}_{K_i}} \mathfrak{s}^{u_\mathfrak{s}}. \tag{19}$$

As in the precomputation stage, we will sieve through small c and d until we collect enough equations of the form (10) and (11) to cancel factors not in $\mathcal{B}'$ and obtain

$$q_i h_i \prod_{c,d} (c + dm_i)^{t(c,d)} \equiv \varphi(\alpha_i) \prod_{c,d} \varphi(c + d\alpha_i)^{t(c,d)} \equiv 1 \pmod{p}, \tag{20}$$

where the left product is divisible only by q_i and primes in $\mathcal{B}'$. Note that we only need one such equation, since the logs of primes in $\mathcal{B}'$ are known from the precomputation.

Thus we have

$$q_i \equiv \prod_{s \in \mathcal{B}'} s^{z'_s} \pmod{p},$$

and so

$$\log_a q_i \equiv \sum_{s \in \mathcal{B}'} z'_s \log_a s \pmod{p-1}. \tag{21}$$

We do this procedure once for each q_i and combine their logarithms to find $\log_a b$. The sieving and cancellation in this stage is the same as in the precomputation. The only difference is that we must keep (18) and (19) and find other equations with rank sufficient to cancel out the factors in those equations and the units that arise. It is a reasonable heuristic assumption that the equations will have full rank, and most discrete logarithm algorithms involve a similar assumption. An exception is the rigorous algorithm of Pomerance in [16], but we have no version of his Lemma 4.1 that works in this setting.

4. Runtime analysis. We will choose two parameters to optimize the performance: the size of B will be $L_p[1/3; \delta]$ and the size of m will be $L_p[2/3; \gamma]$, with δ and γ to be chosen later.

For the precomputation, take

$$k = \left\lceil \frac{1}{\gamma} \left(\frac{\log p}{\log \log p} \right)^{1/3} \right\rceil .$$

Then choose $m \in \mathbb{Z}$ less than $p^{1/k}$ and f irreducible of degree k as described earlier. Let α be a root of f, and $K = \mathbb{Q}(\alpha)$.

We will search through pairs of integers c, d that are relatively prime and at most $L_p[1/3; \lambda]$ in absolute value. There are thus $L_p[1/3, 2\lambda]$ pairs. We have

$$|c + dm| \leq L_p[2/3; \gamma] \quad \text{and} \quad |N(c + d\alpha)| \leq L_p[2/3; \gamma + \lambda/\gamma]$$

by (6).

Using the heuristic assumptions of §2.1, we expect to obtain enough hits to solve for the logs of $\mathcal{B}'$ after

$$L_p[1/3; \frac{\gamma}{3\delta} + \frac{\gamma + \lambda/\gamma}{3\delta} + \delta]$$

trials. Letting this equal $L_p[1/3; 2\lambda]$, we obtain

$$\lambda = \frac{2\gamma^2 + 3\delta^2\gamma}{6\delta\gamma - 1}. \tag{22}$$

The time necessary to sieve through all these values is $L_p[1/3; 2\lambda]$. Each use of Algorithm M to solve the matrix equations takes time $L_p[1/3; 3\delta]$, taking $T = L_p[1/3; \delta]$ and $E = O(\log p)$. To cancel the units as described in §2.3 takes time $L_p[1/3; 2\delta]$. This follows from Theorem 5, taking $M = \exp(L_p[1/3; \delta])$.

This is done $|\mathcal{B}'| < \log p$ times, so the total time is still $L_p[1/3; 3\delta]$. Altogether, the precomputation takes time $L_p[1/3; 3\delta]$.

To calculate the discrete log of a particular $b \in GF(p)$, we choose a random $l \in [1, p-1]$ and see if $a^l b \bmod p$ is $L_p[2/3; \gamma]$-smooth. Assuming Conjecture 2, the ECM can detect such smooth numbers with probability $1 - o(1)$ in time $L_p[1/3; 2\sqrt{\gamma/3}]$. If no factorization is found after that amount of time, another value of l can be tried. We expect to find an l for which $a^l \bmod p$ is smooth after $L_p[1/3; 1/(3\gamma)]$ trials, by Theorem 1.

Once such a value has been found, we have $a^l b \equiv q_1 q_2 \cdots q_t \pmod p$, and it suffices to find the discrete logarithm of each q_i.

Then we choose $m_i = q_i h_i$ of size $L_p[2/3;\gamma]$ for each q_i, and find an irreducible monic polynomial f of degree k for which $f(m_i) \equiv 0 \pmod p$ and $f_i(0)$ is B-smooth. The constant term of f is $L_p[2/3;\gamma]$, so finding a smooth value should take time $L_p[1/3;\gamma/(3\delta)]$.

The next step is to collect equations as in the precomputation. The parameters are the same, and so the time will be the same, unlike most discrete logarithm algorithms, for which the precomputation takes more time than finding individual logarithms.

The total time is $L_p[1/3;M]$, where

$$M = \max\left\{2\lambda, 3\delta, \frac{1}{3\gamma} + 2\sqrt{\frac{\gamma}{3}}, \frac{\gamma}{3\delta}\right\}.$$

By choosing $\gamma = (\frac{3}{8})^{1/3}$, $\delta = 3^{-1/3}$, and $\lambda = (\frac{9}{8})^{1/3}$, we note that (22) is satisfied, and we achieve an optimal time of $L_p[1/3;3^{2/3}]$.

5. Discrete logs for special p. As with the number field sieve factoring algorithm, it is possible to modify the discrete logarithm algorithm for numbers of a special form. The method we present here is a generalization of the Gaussian integer method to higher-degree fields. While asymptotically slower than the method of §3, it avoids the use of Algorithm M and so is more practical for numbers of a reasonable size.

In [15] McCurley offers $100 for breaking a Diffie–Hellman scheme (which is no harder than, and may be equivalent to, finding discrete logarithms) with the prime $p = 2 \cdot 739 \cdot q + 1$, where $q = (7^{149} - 1)/6$. For this number, the scheme given below would be faster than the method of §3, although, since p has 128 digits, even this method would require an exorbitant amount of computer time.

Let

$$k = \left\lceil \frac{1}{\gamma}\left(\frac{\log p}{\log\log p}\right)^{1/5}\right\rceil$$

for some $\gamma > 0$ to be chosen later. The special method will apply to primes p for which there exists an irreducible monic polynomial f of degree k and integer m near $p^{1/k}$ for which $f(m) \equiv 0 \pmod p$, and all the coefficients of f are small. "Small" is a flexible term, but can be taken to mean that the resulting field $K = \mathbb{Q}(\alpha)$ for α a root of f has small enough discriminant that the class group and unit group can be dealt with.

For instance, if $r^e - s \equiv 0 \pmod p$, for a small positive integer r and a nonzero integer s of small absolute value, let l be the smallest integer for which $kl > e$. Then $r^{kl} \equiv sr^{kl-e} \pmod p$, and so if we pick $m = r^l$ and $f(x) = x^k - sr^{kl-e}$, we have $f(m) \equiv 0 \pmod p$.

For the number q, above, we could take $k = 6$, $m = 7^{25}$, and $f(x) = x^6 - 7$. The number p is more difficult; with the same k and m, we would need to take $f(x) = 739x^6 - 5152$. Using a nonmonic polynomial would not cause major difficulties, but the larger coefficients would increase the difficulty of operations in $\mathcal{O}_K$ and reduce the hit rate for the sieving.

Let α be a root of f, and $K = \mathbb{Q}(\alpha)$. For simplicity, we will assume that $\mathcal{O}_K = \mathbb{Z}[\alpha]$ is a unique factorization domain.

Choose $B = L_p[2/5;\delta]$, where $\delta > 0$ is another parameter to be chosen later. Our factor base $\mathcal{B}$ will consist of rational primes $< B$ ($\mathcal{B}_{\mathbb{Q}}$), first-degree primes (algebraic integers, not ideals) in $\mathcal{O}_K$ with norm less than B and a fundamental set of units in $\mathcal{O}_K$ ($\mathcal{B}_K$). We will be dealing explicitly with the ideals and the units in K, and so it is necessary to calculate generators for the unit group and the ideals in $\mathcal{B}_K$. This may be done as in [13], by searching elements of the form $\sum_{i=0}^{k-1} a_i\alpha^i$, with a_i's of small absolute value, for ones of small norm, and combining these to obtain the necessary units and generators of the ideals.

The base for logarithms for algebraic numbers is not important; it may be a small prime that generates $(O_K/\mathfrak{p})^*$, for $\mathfrak{p}$ a prime ideal of norm p, or an algebraic number ρ with $a \equiv \varphi(\rho) \pmod{p}$.

The precomputation step will determine the discrete logs of the whole factor base, not just a subset of the rational part. As before, sieve through c and d less than $L_p[2/5; \lambda]$, looking for values with $c + dm$ and $N(c + d\alpha)$ both smooth. We have

$$c + dm = L_p[4/5; \gamma], \quad \text{and} \quad N(c + d\alpha) = L_p[3/5; \lambda/\gamma] = L_p[4/5; 0].$$

Therefore the probability of both being B-smooth is $L_p[2/5; -2\gamma/(5\delta)]$. Obtaining $L_p[2/5; \delta]$ hits will take expected time

$$L_p[2/5; 2\gamma/(5\delta) + \delta],$$

with $\lambda = \gamma/(5\delta) + \delta/2$.

Each hit gives us an equation involving logarithms of the factor base. Once we have more than $|\mathcal{B}| = L_p[2/5; \delta]$ hits, we solve the resulting matrix equation over $\mathbb{Z}/(p-1)\mathbb{Z}$ using Wiedemann's algorithm in time $L_p[2/5; 2\delta]$. Heuristically, we expect there to be a unique solution, which will give the logarithms of the factor base.

To find an individual logarithm, we again reduce the problem to finding the logs of medium-sized primes q_i by looking for $a^\ell b \pmod{p}$ smooth. Now it will be advantageous to take the q_i's much smaller than m, say of size $L_p[3/5; \theta]$. Assuming Conjecture 2, if $a^\ell b$ is this smooth, we expect the ECM to factor it with probability $1 - o(1)$ in time $L_p[3/10; \sqrt{6\theta/5}]$. We expect a smooth number to occur in about $L_p[2/5; 2/(5\theta)]$ trials, so the total time is $L_p[2/5; 2/(5\theta)]$.

For each q_i, we will sieve c and d for which $q_i | (c + dm)$, say fixing d and taking $c = c_0 + eq_i$, to find one value for which $(c + dm)/q_i$ and $N(c + d\alpha)$ are both B-smooth. Once this happens we are done, since, from the precomputation, we know the logs of the whole factor base.

We cannot change m as in the general method, since this would result in a field with large discriminant. Therefore at least one of c and d must be about as big as q_i, so $(c + dm)/q_i = L_p[4/5; \gamma]$, and $N(c + d\alpha) = L_p[4/5; \theta/\gamma]$. (Note that, for the general number field sieve method, $N(c + d\alpha)$ would be $L_p[1; 1]$, which is why multiple fields were needed.) The expected time to find both B- smooth is therefore

$$L_p\left[2/5; \frac{2(\gamma + \theta/\gamma)}{5\delta}\right].$$

Thus the time for the precomputation is $L_p[2/5; \mu]$, where

$$\mu = \max\left\{\frac{2\gamma}{5\delta} + \delta, 2\delta\right\}, \tag{23}$$

and the time for finding individual logarithms is $L_p[2/5; \nu]$, where

$$\nu = \max\left\{\frac{2}{5\theta}, \frac{2(\gamma + \theta/\gamma)}{5\delta}\right\}. \tag{24}$$

Since θ does not occur in the precomputation, we may choose it to make the two terms in (24) equal, as follows:

$$\theta = \frac{-\gamma^2 + \sqrt{\gamma^4 + 4\delta\gamma}}{2}.$$

The choices for γ and δ depend on how time is to be divided between the two stages. Enlarging δ reduces the time needed to find individual logarithms, but at the cost of increasing the precomputation time. If the times are to be equal (say if only one logarithm is desired for a given p), then the optimal values are

$$\gamma = 10^{-1/5} \quad \text{and} \quad \delta = \left(\frac{4}{125}\right)^{1/5},$$

giving a time of $L_p[2/5; \mu] = L_p[2/5; \nu]$, where

$$\mu = \nu = \left(\frac{128}{125}\right)^{1/5} \approx 1.00475.$$

If many instances are to be done for one p, more time could be spent on the precomputation. For $\mu \geq (128/125)^{1/5}$, if we spend $L_p[2/5; \mu]$ time on the precomputation, each logarithm can be found in time

$$L_p\left[2/5; \left(\frac{128}{125\mu^2}\right)^{1/3}\right].$$

For any $c \geq 1$, the Gaussian integer method can find logarithms in time $L_p[1/2; 1/(2c)]$ if $L_p[1/2; c]$ is spent on the precomputation. Where the above method becomes faster than the Gaussian integer method depends largely on the $o(1)$ terms and the choice of f, but for a good f it is well under 100 digits. More research is needed to say for which size primes and polynomials the special number field sieve algorithm is a practical improvement.

The general number field sieve algorithm is definitely not practical for any reasonable numbers. The crossover point for $L_p[1/2; 1]$ and $L_p[1/3; 3^{2/3}]$ (the times for the Gaussian integer method and the general number field sieve) is 218 digits. The crossover point for $L_p[2/5; 1.00475]$ and $L_p[1/3; 3^{2/3}]$ (the times for the special and general number field sieves) is above 320,000 digits.

If $\mathcal{O}_K$ has class number $h > 1$, then we must cancel the nonprincipal ideals that occur in (11). If we have calculated h, then the algorithm may proceed as in §3, with Algorithm M replaced by Wiedemann's algorithm modulo h, to obtain an equation involving only principal ideals.

Finally, it should be noted that the special number field sieve can also be applied to primes that are values of homogeneous forms in two variables, as well as polynomials. Let f be a polynomial of degree k, and X and Y be integers near $p^{1/k}$, such that

$$Y^k f(X/Y) = X^k + a_{k-1}X^{k-1}Y + \cdots + a_0 Y^k \equiv 0 \pmod{p}.$$

Then the above method may still be used, with the homomorphism $\varphi(c + d\alpha) = c + dX/Y$. Then the sieving phase searches for values of c and d for which $c + d\alpha$ and $cY + dX$ are both smooth. The analysis is the same as given above.

6. Recent developments. The general number field sieve algorithm is still impractical for large numbers, largely because of the need for Gaussian elimination over $\mathbb{Q}$. Methods to avoid this problem have been suggested by Adleman [1] for number field sieve factoring and by Schirokauer [19] for discrete logarithms over $GF(p)$. Coppersmith very recently has suggested using multiple fields to factor n in time $L_n[1/3; c]$ with $c \approx 1.902$, an improvement over $c \approx 2.08$ for the original algorithm of Buhler, Lenstra, and Pomerance, and $c \approx 1.92$ for the methods of Lenstra and Adleman. The resulting algorithms, while faster, are still impractical for numbers within reach of modern computers. Use of the number field sieve in number-theoretic algorithms is a rapidly-developing area. These developments, and the improvements of the constants above, are likely to continue.

The practicality of the special number field sieve is of interest for discrete log-based cryptosystems. By choosing a prime p with a good f and m (as in §5) as the base for such a system, its security would be weakened. A person with knowledge of f might be able to use it as a "trapdoor" to break the system. More study is needed to say how much of an advantage this would actually be.

Acknowledgments. The author thanks Carl Pomerance for allowing the presentation of his Algorithm M here and for many suggestions that greatly improved the form and content of this paper. Thanks also to Andrew Odlyzko for several e-mail discussions about discrete logarithms, and to Hendrik Lenstra for helpful comments.

REFERENCES

[1] L. M. ADLEMAN, *Factoring numbers using singular integers*, in Proc. 23rd ACM Symposium on Theory of Computing, New Orleans, LA, 1991, pp. 64–71.

[2] L. BABAI, *On Lovász's lattice reduction and the nearest lattice point problem*, in Proc. 2nd Annual Symposium on the Theoretical Aspects of Computing, Paris, France, K. Mehlhorn, ed., Springer, Berlin, pp. 13–20.

[3] A. BAKER, *Transcendental Number Theory*, Cambridge University Press, Cambridge, UK, 1975.

[4] J. BRILLHART, M. FILASETA, AND A. ODYLZKO, *On an irreducibility theorem of A. Cohn*, Canad. J. Math, 33 (1981), pp. 1055–1059.

[5] J. BUHLER, H. W. LENSTRA, JR., AND C. POMERANCE, *Factoring integers with the number field sieve*, preprint.

[6] E. R. CANFIELD, P. ERDŐS, AND C. POMERANCE, *On a problem of Oppenheim concerning "Factorisatio Numerorum,"* J. Number Theory, 17 (1983), pp. 1–28.

[7] D. COPPERSMITH, *Modifications to the number field sieve*, J. Cryptology, to appear.

[8] D. COPPERSMITH, A. M. ODLYZKO, AND R. SCHROEPPEL, *Discrete logarithms in GF(p)*, Algorithmica, 1 (1986), pp. 1–15.

[9] E. DOBROWOLSKI, *On the maximal modulus of conjugates of an algebraic integer*, Bull. Acad. Polon. Sci. Ser. Sci. Math. Astronom. Phys., 26 (1978), pp. 291–292.

[10] E. KALTOFEN AND B. D. SAUNDERS, *On Wiedemann's method for solving sparse linear systems*, Proceedings AAECC-5 SLNCS 536 (1991), pp. 29–38.

[11] B. LAMACCHIA AND A. M. ODLYZKO, *Computation of discrete logarithms in prime fields*, Designs, Codes and Cryptography, 1 (1991), pp. 47–62.

[12] A. K. LENSTRA, H. W. LENSTRA, JR., AND L. LOVÁSZ, *Factoring polynomials with rational coefficients*, Math. Ann., 261 (1982), pp. 515–534.

[13] A. K. LENSTRA, H. W. LENSTRA, JR., M. S. MANASSE, AND J. M. POLLARD, *The number field sieve*, in Proc. 22nd ACM Symposium on Theory of Computing, Baltimore, MD, 1990, pp. 564–572.

[14] H. W. LENSTRA, JR., *Factoring integers with elliptic curves*, Ann. Math., 126 (1987), pp. 649–673.

[15] K. MCCURLEY, *The discrete logarithm problem*, in Cryptology and Computational Number Theory, Proceedings of Symposia in Applied Mathematics, American Mathematical Society, Providence, RI, 1990.

[16] C. Pomerance, *Fast, rigorous factorization and discrete logarithm algorithms*, in Discrete Algorithms and Complexity, D. S. Johnson et al., eds., Academic Press, Orlando, 1987, pp. 119–143.

[17] ———, personal communication, 1990.

[18] J. B. Rosser and L. Schoenfeld, *Approximate formulas for some functions of prime numbers*, Illinois J. Math., 6 (1962), pp. 64–94.

[19] O. Schirokauer, *On Pro-Finite Groups and on Discrete Logarithms*, Ph.D. thesis, Univ. of California, Berkeley, CA, May 1992.

[20] V. Shoup, *Searching for primitive roots in finite fields*, Math. Comput., 58 (1992), pp. 369–380.

[21] D. H. Wiedemann, *Solving sparse linear equations over finite fields*, IEEE Trans. Inform. Theory, 32 (1986), pp. 54–62.

[22] H. Zantema, *Class numbers and units*, in Computational Methods in Number Theory, Vol. II, H. W. Lenstra, Jr. and R. Tijdeman, eds., Mathematisch Centrum, Amsterdam, 1982, pp. 213–234.

SIAM J. DISC. MATH.
Vol. 6, No. 1, pp. 139–151, February 1993

011

POLYHEDRAL PROPERTIES OF CLUTTER AMALGAM[*]

P. NOBILI[†] AND A. SASSANO[‡]

Abstract. A *clutter* $\mathcal{L}$ is a collection of subsets of a ground set $E(\mathcal{L})$ with the property that, for every pair $A_i, A_j \in \mathcal{L}$, A_i is neither contained in nor contains A_j. A *cover* of $\mathcal{L}$ is a subset of E intersecting every member of $\mathcal{L}$. The *covering polytope* $Q(\mathcal{L})$, associated with a clutter $\mathcal{L}$, is the convex hull of the incidence vectors of the covers of $\mathcal{L}$. The polytope $Q(\mathcal{L})$ provides a common generalization for several polytopes associated with combinatorial optimization problems (stable set, knapsack, acyclic subdigraph, bipartite subgraph, etc.) that can be formulated as covering problems with respect to suitably defined clutters.

In this paper, a binary composition operation is described, the *clutter amalgam*, that combines two clutters $\mathcal{L}_1$ and $\mathcal{L}_2$ to produce a new clutter $\mathcal{L}$ called amalgam of $\mathcal{L}_1$ and $\mathcal{L}_2$. Furthermore, an isomorphic polyhedral composition operation is introduced that combines the linear descriptions of the polytopes $Q(\mathcal{L}_1)$ and $Q(\mathcal{L}_2)$ and produces a linear description of the polytope $Q(\mathcal{L})$.

The clutter amalgam operation has the crucial property that if the clutters $\mathcal{L}_1$ and $\mathcal{L}_2$ are *ideal* then the amalgam $\mathcal{L}$ is also ideal. Finally, the restriction of the clutter amalgam to graphs properly generalizes the *graph amalgam* introduced by Burlet and Fonlupt and defines a new perfection-preserving operation.

Key words. clutters, polyhedra, perfect graphs

AMS(MOS) subject classifications. 05C70, 52B99

1. Introduction. A *subset system* $\mathcal{L}$ such that no two *members* $A_i, A_j \in \mathcal{L}$ satisfy $A_i \subset A_j$ is called a *clutter*. The *ground set* of $\mathcal{L}$ is denoted by $E(\mathcal{L})$. A set $C \subseteq E(\mathcal{L})$ that has nonempty intersection with every member of $\mathcal{L}$ is said to be a *cover* of $\mathcal{L}$.

In this paper, we study the polytope $Q(\mathcal{L}) = \text{conv}\{x^C \in \Re^{E(\mathcal{L})} : \text{C is a cover of } \mathcal{L}\}$, which is called the *covering polytope* associated with $\mathcal{L}$. We can write

$$Q(\mathcal{L}) = \text{conv}\left\{x \in \{0,1\}^{E(\mathcal{L})} : \sum_{e \in A} x_e \geq 1, A \in \mathcal{L}\right\}.$$

A possible way to describe the structure of the polytope $Q(\mathcal{L})$ is to define composition/decomposition operations for clutters and to provide isomorphic polyhedral operations that produce the linear description of the polytope associated with the composed clutter in terms of the linear descriptions of the polytopes associated with the components. More specifically, consider the following composition scheme for clutters.

Let $\mathcal{S}$ be a class of clutters constituted by the following conditions:

(i) A finite number of simple clutters for which the linear descriptions of the associated polytopes are known ("building blocks");

(ii) The clutters that are obtainable from the building blocks by applying a finite number of times some composition rule "$\diamond$";

(iii) Nothing else.

Assume, without loss of generality, that the composition rule "$\diamond$" is a binary operation, i.e., composes two clutters $\mathcal{L}_1$ and $\mathcal{L}_2$ to give a clutter $\mathcal{L}$. Furthermore, suppose that the linear descriptions of $Q(\mathcal{L}_1)$ and $Q(\mathcal{L}_2)$ are known and that there exists a binary operation "$\star$," which combines (linear descriptions of) polytopes and is isomorphic to the operation "$\diamond$" in the sense that the relation

$$Q(\mathcal{L}_1 \diamond \mathcal{L}_2) = Q(\mathcal{L}_1) \star Q(\mathcal{L}_2)$$

[*] Received by the editors June 5, 1989; accepted for publication (in revised form) April 1, 1992.

[†] Istituto di Analisi dei Sistemi ed Informatica del CNR, Viale Manzoni 30, 00185 Rome, Italy.

[‡] Dipartimento di Informatica e Sistemistica, Universitá di Roma "La Sapienza," via Buonarroti 12, 00185 Rome, Italy.

holds for any pair of clutters $\mathcal{L}_1$ and $\mathcal{L}_2$ in $\mathcal{S}$.

It follows that we can obtain the linear description of the polytope $Q(\mathcal{L})$ for each clutter $\mathcal{L}$ in $\mathcal{S}$ by simply knowing the linear descriptions of the polytopes associated with the building blocks and the series of compositions that produces $\mathcal{L}$.

Following this general approach, we introduce in this paper two operations to compose (and decompose) clutters. The first operation (§2) is called *weak cutset identification* and generalizes the clique cutset identificaton introduced by Chvàtal [4] for graphs. We prove that, if a clutter $\mathcal{L}$ is obtained as weak cutset identification of two clutters $\mathcal{L}_1$ and $\mathcal{L}_2$, then the linear description of $Q(\mathcal{L})$ is given by the *union* of the linear descriptions of $Q(\mathcal{L}_1)$ and $Q(\mathcal{L}_2)$.

In §3 we introduce the *clutter amalgam* operation. This operation is an extension of the *join* operation defined by Cunningham [5] for independence systems and of the *graph amalgam* operation defined by Burlet and Fonlupt for the stable set problem on graphs [2]. It is interesting to also note that the restriction to graphs (*edge-clutters*) of the clutter amalgam constitutes a proper generalization (Remark 3.12) of the graph amalgam.

For the clutter amalgam operation, we describe how the linear descriptions of the polytopes $Q(\mathcal{L}_1)$ and $Q(\mathcal{L}_2)$ must be combined to obtain a complete linear description of $Q(\mathcal{L})$.

Finally, we prove that both the weak cutset identification and the clutter amalgam preserve *ideality* and that, restricted to edge-clutters (graphs), they define two new composition operations that preserve *perfection*.

In the remainder of this section, we give the main definitions and notation used throughout the paper.

Let E be any finite set. We consider the linear vector space $\Re^E$, whose vectors have components indexed by the elements of E. If $C \subseteq E$, then $x^C \in \Re^E$ denotes the *incidence vector* of C; that is, $x_e^C = 1$ if $e \in C$, $x_e^C = 0$ otherwise. The vectors of $\Re^E$ whose components are all equal to zero or all equal to 1 are, respectively, denoted by **0** and **1**.

Let $\mathcal{L}$ be a clutter and $Z \subseteq E(\mathcal{L})$ be any set. The clutter of all the members of $\mathcal{L}$ that have empty intersection with Z is said to be obtained from $\mathcal{L}$ by *deletion* of the set Z and is denoted by $\mathcal{L}\backslash Z$. The clutter of all the minimal members of $\{A_i - Z : A_i \in \mathcal{L}\}$ is said to be obtained from $\mathcal{L}$ by *contraction* of Z and denoted by $\mathcal{L}/Z$. Any clutter $\mathcal{L}'$ obtained from $\mathcal{L}$ by a sequence of deletions and contractions is said to be a *minor* of $\mathcal{L}$.

The family of all the covers of $\mathcal{L}$ that are minimal (with respect to set inclusion) is called the *blocker* of $\mathcal{L}$ and denoted by $b(\mathcal{L})$. Evidently, $b(\mathcal{L})$ is a clutter. Moreover, the relation between clutters and their blockers is symmetric; that is, if two clutters $\mathcal{L}_1$ and $\mathcal{L}_2$ satisfy $\mathcal{L}_1 = b(\mathcal{L}_2)$, then they also satisfy $\mathcal{L}_2 = b(\mathcal{L}_1)$. Two clutters in such a relation are said to be a *blocking pair* of clutters.

In the following proposition, we recall some basic properties of $Q(\mathcal{L})$ (see, for instance, [6]).

PROPOSITION 1.1. *We have the following properties*:

(i) *$Q(\mathcal{L})$ is nonempty if and only if, for each member $A \in \mathcal{L}$, $|A| \geq 1$*;

(ii) *$Q(\mathcal{L})$ is full-dimensional (i.e., $\dim(Q(\mathcal{L})) = |E(\mathcal{L})|$) if and only if $|A| \geq 2$ for each $A \in \mathcal{L}$*;

if $Q(\mathcal{L})$ is full-dimensional, then, for each $e \in E(\mathcal{L})$,

(iii) *The inequality $x_e \geq 0$ defines a (trivial) facet of $Q(\mathcal{L})$ if and only if $|A - \{e\}| \geq 2$ for each $A \in (\mathcal{L})$*;

(iv) *The inequality $x_e \leq 1$ defines a (trivial) facet of $Q(\mathcal{L})$*;

(v) *Every nontrivial facet of $Q(\mathcal{L})$ is defined by an inequality of the form $\sum_{e \in E(\mathcal{L})} a_e x_e \geq a_0$ where all the coefficients are nonnegative*;

(vi) *The hyperplanes supporting nontrivial facets of $Q(\mathcal{L})$ do not contain the vectors* **0** *and* **1**.

In the following, we assume that all the covering polytopes we consider are full-dimensional. Such an assumption can be made without loss of generality. In fact, by Proposition 1.1 (ii), a polytope $Q(\mathcal{L})$ is not full-dimensional if and only if there exists a singleton member $\{f\} \in \mathcal{L}$. In such a case, every cover of $\mathcal{L}$ contains the element f, and we have

$$Q(\mathcal{L}) = \left\{(x, x_f) \in \Re^{E(\mathcal{L})} : x \in Q(\mathcal{L}\backslash\{f\}), x_f = 1\right\}.$$

Hence we can consider the polytope $Q(\mathcal{L}\backslash\{f\})$ instead of $Q(\mathcal{L})$.

The following proposition, whose straightforward proof is omitted, relates the structure of a polytope $Q(\mathcal{L}')$ to the structure of $Q(\mathcal{L})$, where $\mathcal{L}'$ is some minor of $\mathcal{L}$ obtained by deletion.

PROPOSITION 1.2. *Let $\mathcal{L}'$ be a nonempty minor of $\mathcal{L}$ obtained by deleting a set $Z \subseteq E(\mathcal{L})$. Then the polytope $Q(\mathcal{L}')$ is the projection of the polytope $Q(\mathcal{L})$ onto the subspace $\Re^{E(\mathcal{L})-Z} \subseteq \Re^{E(\mathcal{L})}$. Consequently, an inequality $\sum_{e\in E(\mathcal{L})-Z} a_e x_e \geq a_0$ is valid for $Q(\mathcal{L})$ if and only if it is valid for $Q(\mathcal{L}')$; moreover, if it defines a facet of $Q(\mathcal{L})$, then it also defines a facet of $Q(\mathcal{L}')$.*

A clutter $\mathcal{L}$ is said to be *ideal* if and only if every nontrivial facet of $Q(\mathcal{L})$ is defined by an inequality $\sum_{e\in A} x_e \geq 1$ for some $A \in \mathcal{L}$. Consequently, a clutter is ideal if and only if

$$Q(\mathcal{L}) = Q_{LP}(\mathcal{L}) = \left\{x \in \Re^{E(\mathcal{L})} : \sum_{e\in A} x_e \geq 1, A \in \mathcal{L}; \quad 0 \leq x_e \leq 1, e \in E(\mathcal{L})\right\}.$$

If the clutter $\mathcal{L}$ is not ideal, we have that $Q(\mathcal{L}) \subset Q_{LP}(\mathcal{L})$ and hence that other facet-defining inequalities are needed to describe $Q(\mathcal{L})$. A basic family of facet-defining inequalities of $Q(\mathcal{L})$ is associated with a class of subsets of $E(\mathcal{L})$ introduced by the following definition.

DEFINITION 1.3. Let $\mathcal{L}$ be a clutter; a subset R of $E(\mathcal{L})$ is a 2-clique of $\mathcal{L}$ if, for each pair of elements $e_i, e_j \in R$, the set $\{e_i, e_j\}$ is a member of $\mathcal{L}$. A 2-clique is said to be *maximal* if there does not exist a 2-clique R' of $\mathcal{L}$ such that $R \subset R'$.

PROPOSITION 1.4 (see [6]). *Let $\mathcal{L}$ be a clutter and R a 2-clique of $\mathcal{L}$, then the inequality*

$$\sum_{e\in R} x_e \geq |R| - 1$$

defines a facet of $Q(\mathcal{L})$ if and only if R is maximal.

A clutter $\mathcal{L}$ is said to be an *edge-clutter* if $|A| = 2$ for each $A \in \mathcal{L}$. It follows that each edge-clutter is associated with a graph $G_{\mathcal{L}} = (V, E)$ such that $V = E(\mathcal{L})$ and $A \in \mathcal{L}$ if and only if $A = \{e_i, e_j\}$ and $e_ie_j \in E$.

Evidently, each cover of $\mathcal{L}$ corresponds to a *vertex cover* of $G_{\mathcal{L}}$, and the polytope $Q(\mathcal{L})$ corresponds to the *vertex cover polytope* $Q(G_{\mathcal{L}})$. Moreover, the complement of a cover of $\mathcal{L}$ (vertex cover of $G_{\mathcal{L}}$) is a *stable set* of $G_{\mathcal{L}}$. It follows that the stable set polytope $P(G_{\mathcal{L}})$ is the image of $Q(G_{\mathcal{L}})$ under the affine tranformation $y = \mathbf{1} - x$.

Consequently, an inequality $\sum_{e\in V} a_e x_e \leq a_0$ defines a facet of $P(G_{\mathcal{L}})$ if and only if the inequality $\sum_{e\in V} a_e y_e \geq -a_0 + \sum_{e\in V} a_e$ defines a facet of $Q(G_{\mathcal{L}})$.

Furthermore, we have that a maximal 2-clique of an edge-clutter $\mathcal{L}$ corresponds to a clique of $G_{\mathcal{L}}$ and that the facet-defining inequality $\sum_{e\in R} x_e \geq |R| - 1$ of $Q(\mathcal{L}) \equiv Q(G_{\mathcal{L}})$

corresponds, modulo the affine transformation $y = \mathbf{1} - x$, to the facet-defining inequality $\sum_{e \in R} x_e \le 1$ of $P(G_{\mathcal{L}})$ (clique inequality).

A well-known result due to Chvàtal [4] asserts that a graph G is *perfect* if and only if each nontrivial facet of $P(G)$ is defined by a clique inequality. It follows that if $\mathcal{L}$ is an edge-clutter and $G_{\mathcal{L}}$ is perfect, then each nontrivial facet of $Q(\mathcal{L})$ is defined by a 2-clique inequality $\sum_{e \in R} x_e \ge |R| - 1$.

2. Weak cutsets. Let $\mathcal{L}$ be a clutter; a proper subset C of $E(\mathcal{L})$ is a *cutset* of $\mathcal{L}$ if there exists a partition (E_1, E_2) of $E(\mathcal{L}) - C$ with $E_1, E_2 \neq \emptyset$ and such that every member of $\mathcal{L}$ is either a subset of $E_1 \cup C$ or a subset of $E_2 \cup C$. The sets E_1 and E_2 are said to be the *shores* of the cutset C. Evidently, if C is a cutset, then every subset C' of C is a cutset of the clutter $\mathcal{L} \backslash (C - C')$.

DEFINITION 2.1. A cutset W of a clutter $\mathcal{L}$ is said to be a *weak cutset* if

$$|\{A \cap W' : A \in b(\mathcal{L})\}| \le |W'| + 1, \qquad \text{for each } W' \subseteq W.$$

If W is a weak cutset of a clutter $\mathcal{L}$, and if E_1 and E_2 are the shores of W, then say that $\mathcal{L}$ arises as *weak cutset identification* of the clutters $\mathcal{L}_1 = \mathcal{L} \backslash E_2$ and $\mathcal{L}_2 = \mathcal{L} \backslash E_1$. The clutters $\mathcal{L}_1$ and $\mathcal{L}_2$ are called the *components* of the clutter $\mathcal{L}$.

In the following theorem, we describe the polyhedral structure of the polytope $Q(\mathcal{L})$ in terms of the structure of the polytopes $Q(\mathcal{L}_1)$ and $Q(\mathcal{L}_2)$. We denote by $S(a) = \{e \in E(\mathcal{L}) : a_e > 0\}$ the *support* of an inequality $\sum_{e \in E(\mathcal{L})} a_e x_e \ge a_0$.

THEOREM 2.2. *Let $\mathcal{L}$ be a clutter and let $W \subseteq E(\mathcal{L})$ be a weak cutset of $\mathcal{L}$. Let E_1 and E_2 be the shores of W and let $\mathcal{L}_1 = \mathcal{L} \backslash E_2$ and $\mathcal{L}_2 = \mathcal{L} \backslash E_1$. Then the linear description of the polytope $Q(\mathcal{L})$ is given by the union of the linear descriptions of the polytopes $Q(\mathcal{L}_1)$ and $Q(\mathcal{L}_2)$.*

Proof. If every facet of $Q(\mathcal{L})$ is defined by an inequality whose support is either a subset of $E_1 \cup W$ or a subset of $E_2 \cup W$, then the theorem follows by Proposition 1.2. Hence we assume that there exists a facet F of $Q(\mathcal{L})$ defined by an inequality $a^T x \ge a_0$ whose support $S(a)$ has nonempty intersection with both E_1 and E_2.

Let $E_1' = E_1 \cap S(a)$, $E_2' = E_2 \cap S(a)$, $W' = W \cap S(a)$, and $\mathcal{L}' = \mathcal{L} \backslash (E(\mathcal{L}) - S(a))$. We can write the inequality $a^T x \ge a_0$ as

$$\sum_{e \in E_1'} a_e x_e + \sum_{e \in E_2'} a_e x_e + \sum_{e \in W'} a_e x_e \ge a_0, \tag{2.1}$$

where all the coefficients are strictly positive. By Proposition 1.2, inequality (2.1) defines a facet of $Q(\mathcal{L}')$.

Let $\mathcal{C}'$ be the family of covers of $\mathcal{L}'$ whose incidence vectors satisfy (2.1) as an equality. Since inequality (2.1) has full support in $E(\mathcal{L}')$, we have that each member of $\mathcal{C}'$ belongs to $b(\mathcal{L}')$. Let C_i and C_j be any two members of $\mathcal{C}'$ such that $C_i \cap W' = C_j \cap W'$. We claim that the following equations are satisfied:

$$\begin{aligned} \sum_{e \in C_i \cap E_1'} a_e &= \sum_{e \in C_j \cap E_1'} a_e, \\ \sum_{e \in C_i \cap E_2'} a_e &= \sum_{e \in C_j \cap E_2'} a_e. \end{aligned} \tag{2.2}$$

In fact, suppose that one of the above equalities does not hold and let, without loss of generality, $\sum_{e \in C_i \cap E_1'} a_e < \sum_{e \in C_j \cap E_1'} a_e$. Since W' is a cutset of $\mathcal{L}'$ and $C_i \cap W' =$

$C_j \cap W'$, we have that the set $(C_i \cap E_1') \cup (C_j - E_1')$ is a cover of $\mathcal{L}'$ and that its incidence vector does not satisfy inequality (2.1), which is a contradiction.

Now let M be the matrix whose rows are the incidence vectors of the covers in $\mathcal{C}'$. Let y^e be the column of M associated with the element $e \in E(\mathcal{L}')$ and let M' be the matrix with $|W'| + 2$ columns defined as follows: The first $|W'|$ columns of M' coincide with the columns of M associated with the elements in W' the last two columns of M', say y^1 and y^2, are defined as

$$y^1 = \sum_{e \in E_1'} a_e y^e, \qquad y^2 = \sum_{e \in E_2'} a_e y^e.$$

Since inequality (2.1) is facet-defining for $Q(\mathcal{L}')$, we have that the matrix M has full column rank. It follows, by construction, that also the matrix M' has full column rank. Furthermore, (2.2) implies that, for each pair of covers $C_i, C_j \in \mathcal{C}'$ such that $C_i \cap W' = C_j \cap W'$, we have that $y_i^1 = y_j^1$ and $y_i^2 = y_j^2$. It follows that the rows of M' corresponding to the covers C_i and C_j are equal. As a consequence, the matrix M' contains at most $|\{A \cap W' : A \in \mathcal{C}'\}|$ different rows. Moreover, since $\mathcal{C}' \subseteq b(\mathcal{L}')$, we have that $|\{A \cap W' : A \in \mathcal{C}'\}| \leq |\{A \cap W' : A \in b(\mathcal{L}')\}|$.

Consequently, since $|\{A \cap W' : A \in b(\mathcal{L}')\}| \leq |\{A \cap W' : A \in b(\mathcal{L})\}|$, we have, by definition of weak cutset, that the number of different rows of M' is at most $|W'| + 1$ and hence that M' cannot have full column rank. This contradicts the assumption that inequality (2.1) defines a facet of $Q(\mathcal{L}')$ and completes the proof of the theorem. □

As a first example of weak cutset identification, consider a cutset W of a clutter $\mathcal{L}$, which is a 2-clique of $\mathcal{L}$. It follows that every minimal cover of $\mathcal{L}$ either contains W or misses at most one of its elements. Moreover, the same is true for any subset W' of W. Hence W is a weak cutset of $\mathcal{L}$, and Theorem 2.3 implies the following corollary, which is a generalization to arbitrary clutters of the well-known result of Chvàtal [4] relative to the *clique-cutset identification* and the stable set polytope.

COROLLARY 2.3. *Let $\mathcal{L}$ be a clutter and let W be a cutset that is a 2-clique of $\mathcal{L}$. Let E_1 and E_2 be the shores of W and let $\mathcal{L}_1 = \mathcal{L}\backslash E_2$ and $\mathcal{L}_2 = \mathcal{L}\backslash E_1$ be the components of $\mathcal{L}$. Then $\mathcal{L}$ is a weak cutset identification of $\mathcal{L}_1$ and $\mathcal{L}_2$, and the linear description of the polytope $Q(\mathcal{L})$ is given by the union of the linear descriptions of the polytopes $Q(\mathcal{L}_1)$ and $Q(\mathcal{L}_2)$.*

As another example, consider a cutset W of $\mathcal{L}$ that is totally ordered by a relation $\prec$ such that $e_i \prec e_j$ $(e_i, e_j \in W)$ implies that every member of $\mathcal{L}$ containing e_i contains also e_j. Every minimal cover of $\mathcal{L}$ contains at most one element from W, otherwise some element is redundant; it follows that W is a weak cutset.

COROLLARY 2.4. *Let $\mathcal{L}$ be a clutter and let W be a cutset whose elements are totally ordered by a relation $\prec$ such that, for every pair of elements $e_i, e_j \in W$ with $e_i \prec e_j$ and for every member A of $\mathcal{L}$, $e_i \in A$ implies $e_j \in A$. Let E_1 and E_2 be the shores of W and let $\mathcal{L}_1 = \mathcal{L}\backslash E_2$ and $\mathcal{L}_2 = \mathcal{L}\backslash E_1$ be the components of $\mathcal{L}$. Then $\mathcal{L}$ is a weak cutset identification of $\mathcal{L}_1$ and $\mathcal{L}_2$ and the linear description of the polytope $Q(\mathcal{L})$ is given by the union of the linear descriptions of the polytopes $Q(\mathcal{L}_1)$ and $Q(\mathcal{L}_2)$.*

3. Clutter amalgam. In this section, we introduce a new binary composition operation for clutters, called *clutter amalgam*. This operation has the property that its restriction to the family of edge-clutters provides a proper generalization of the *graph amalgam* operation as defined by Burlet and Fonlupt [2].

In the previous section, we defined a clutter $\mathcal{L}$ containing a weak cutset $W \subseteq E(\mathcal{L})$ to be the weak cutset identification of the clutters (proper minors of $\mathcal{L}$) $\mathcal{L}_1 = \mathcal{L}\backslash E_2$ and

$\mathcal{L}_2 = \mathcal{L}\backslash E_1$. Furthermore, we proved that the linear description of the polytope $Q(\mathcal{L})$ is obtained as the union of the linear descriptions of the polytopes $Q(\mathcal{L}_1)$ and $Q(\mathcal{L}_2)$.

In the following definition, we characterize a two-element member of the minor $\mathcal{L}^W = \mathcal{L}\backslash(E(\mathcal{L}) - W)$, which will play a basic role in the definition of clutter amalgam.

DEFINITION 3.1. Let W be a weak cutset of a clutter $\mathcal{L}$ and let E_1 and E_2 be the shores of W. A pair of elements e_1 and e_2 of W such that $\{e_1, e_2\} \in \mathcal{L}$ is a W-pair of $\mathcal{L}$ if there exists a maximal 2-clique $R = \{e_1, e_2, \dots, e_r\}$ of $\mathcal{L}$ such that the following properties hold:

(i) If $A \in \mathcal{L}$ and $e_i \in A$, then $A \subseteq E_i \cup W$ $(i = 1, 2)$;

(ii) For each $A \in \mathcal{L}$ such that $A \cap R = \{e_i\}$ $(i = 1, 2)$ and each $e_h \in R - \{e_1, e_2\}$, there exists $A' \in \mathcal{L}$ such that $A' - \{e_h\} \subseteq A - \{e_i\}$;

(iii) For each pair $A_1, A_2 \in \mathcal{L}^W$ with $A_1 \cap R = \{e_1\}$ and $A_2 \cap R = \{e_2\}$, there exists $A \in \mathcal{L}^W$ such that $A \subseteq (A_1 \cup A_2) - \{e_1, e_2\}$.

By (i) of Definition 3.1, we have that $R \subseteq W$, and, by (ii) and (iii), we have that, if $\{e_h, e_1\} \in \mathcal{L}$ and $\{e_h, e_2\} \in \mathcal{L}$, then $e_h \in R$. It follows that R is the *unique* maximal 2-clique containing e_1 and e_2.

The main properties of a W-pair are given by the following lemma.

LEMMA 3.2. *Let W be a weak cutset of a clutter $\mathcal{L}$, let $\{e_1, e_2\} \subseteq W$ be a W-pair of $\mathcal{L}$, and let $R = \{e_1, e_2, \dots, e_r\}$ be the maximal 2-clique containing $\{e_1, e_2\}$. Then every nontrivial facet-defining inequality $a^T x \geq a_0$ of the polytope $Q(\mathcal{L})$ satisfies the following conditions*:

(i) *$a_{e_h} \geq a_{e_i}$ for $e_h \in R - \{e_1, e_2\}$ and $i = 1, 2$;*

(ii) *If $a_{e_i} > 0$, then $S(a) \subseteq E_i \cup W$, $i = 1, 2$;*

(iii) *If $a_{e_1} > 0$ and $a_{e_2} > 0$, then, for some positive $\alpha \in \Re$, we have that*

$$\left(a^T x \geq a_0\right) = \alpha \left(\sum_{f \in R} x_f \geq r - 1 \right).$$

Proof. (i) Let $e_i \in \{e_1, e_2\}$ and $e_h \in R - \{e_1, e_2\}$. Since the inequality $a^T x \geq a_0$ defines a nontrivial facet of $Q(\mathcal{L})$, we have that there exists a cover C of $\mathcal{L}$ whose incidence vector satisfies the inequality $a^T x \geq a_0$ as an equality and such that $e_h \notin C$. Moreover, since $A_{ih} = \{e_i, e_h\} \in \mathcal{L}$, we have that $e_i \in C$.

Now we claim that the set $C - \{e_i\}$ is a cover of $\mathcal{L} - \{A_{ih}\}$. In fact, suppose that there exists $A \in \mathcal{L} - \{A_{ih}\}$ with the property that $A \cap (C - \{e_i\}) = \emptyset$. Since $e_h \notin C$ and thus $R - \{e_h\} \subseteq C$, it follows that $A \cap R = \{e_i\}$, and, by (ii) of Definition 3.1, there exists $A' \in \mathcal{L}$ such that $A' - \{e_h\} \subseteq A - \{e_i\}$. Consequently, we have that $C \cap A' = C \cap (A' - \{e_h\}) \subseteq C \cap (A - \{e_i\}) = \emptyset$, contradicting the assumption that C is a cover of $\mathcal{L}$.

The above claim implies that the set $C' = C - \{e_i\} \cup \{e_h\}$ is a cover of $\mathcal{L}$, and hence we have that $a_{e_h} \geq a_{e_i}$.

(ii) By (i) of Definition 3.1, we have that the sets $W_1 = W - \{e_1\}$ and $W_2 = W - \{e_2\}$ are cutsets of $\mathcal{L}$, and hence, by Definition 2.1, both W_1 and W_2 are weak cutsets of $\mathcal{L}$. Let us consider the cutset W_1; we have that the shores of W_1 are $E_1 \cup \{e_1\}$ and E_2 and that the components of $\mathcal{L}$ with respect to W_1 are $\mathcal{L}' = \mathcal{L}\backslash E_2$ and $\mathcal{L}'' = \mathcal{L}\backslash(E_1 \cup \{e_1\})$. By Theorem 2.2, we have that the inequality $a^T x \geq a_0$ defines either a facet of $Q(\mathcal{L}')$ or a facet of $Q(\mathcal{L}'')$. In particular, if $a_{e_1} > 0$, we have that the inequality $a^T x \geq a_0$ defines a facet of $Q(\mathcal{L}')$, and hence $S(a) \subseteq E_1 \cup W$. A symmetric argument, relative to the weak cutset W_2, shows that, if $a_{e_2} > 0$, then $S(a) \subseteq E_2 \cup W$.

(iii) If $a_{e_1} > 0$ and $a_{e_2} > 0$, then we have, by (ii), that $S(a) \subseteq W$. It follows, by Proposition 1.2, that $a^T x \geq a_0$ defines a facet of the polytope $Q(\mathcal{L}^W)$.

Let $\mathcal{C}_W$ be the family of covers of $\mathcal{L}^W$ whose incidence vectors satisfy the inequality $a^T x \geq a_0$ as an equality and let C be a member of $\mathcal{C}_W$. Since R is a 2-clique, we have that $r - 1 \leq |C \cap R| \leq r$.

Suppose that $|C \cap R| = r$; since $a_{e_1}, a_{e_2} > 0$, we have that neither the set $C_1 = C - \{e_1\}$ nor $C_2 = C - \{e_2\}$ is a cover of $\mathcal{L}^W$. It follows that there exist two members of $\mathcal{L}^W$, say A_1 and A_2, such that $A_1 \cap C_1 = \emptyset$ and $A_2 \cap C_2 = \emptyset$. Now, since C is a cover of $\mathcal{L}^W$ and $R \subseteq C$, we have that $A_1 \cap C = \{e_1\} = A_1 \cap R$ and $A_2 \cap C = \{e_2\} = A_2 \cap R$. It follows, by (iii) of Definition 3.1, that there exists $A \in \mathcal{L}^W$ such that $A \subseteq (A_1 \cup A_2) - \{e_1, e_2\}$ and hence that $C \cap A = \emptyset$, contradicting the assumption that C is a cover of $\mathcal{L}^W$.

It follows that $|C \cap R| = r - 1$ and hence that the incidence vector of each cover $C \in \mathcal{C}_W$ lies on the hyperplane $\{x \in \Re^W : \sum_{f \in R} x_f = r - 1\}$. Consequently, the inequalities $a^T x \geq a_0$ and $\sum_{f \in R} x_f \geq r - 1$ define the same facet of $Q(\mathcal{L}^W)$ and, by the full-dimensionality of $Q(\mathcal{L}^W)$, (iii) follows. □

The following proposition is an easy consequence of Theorem 2.2 and Lemma 3.2.

PROPOSITION 3.3. *Let W be a weak cutset of a clutter $\mathcal{L}_{12}$ and let E_1 and E_2 be the shores of W. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be the components of $\mathcal{L}_{12}$ with respect to W. Finally, let $\{e_1, e_2\}$ be a W-pair of $\mathcal{L}_{12}$ and let $R = \{e_1, e_2, \ldots, e_r\}$ be the maximal 2-clique containing $\{e_1, e_2\}$. Then the defining linear systems for the polytopes $Q(\mathcal{L}_k)$, $(k = 1, 2)$ have the following structure*:

$$
\begin{aligned}
&\sum_{e \in E_k \cup W'} a^i_e x_e &&\geq a^i_0 &&(i \in N_k),\\
&\sum_{e \in E_k \cup W'} b^i_e x_e + x_{e_k} &&\geq b^i_0 &&(i \in I_k),\\
&\sum_{e \in W'} c^i_e x_e + x_{e_h} &&\geq c^i_0 &&(i \in J_k),\\
&\sum_{e \in R'} x_e + x_{e_1} + x_{e_2} &&\geq r - 1,
\end{aligned}
\tag{3.1}
$$

$$0 \leq x_e \leq 1 \qquad (e \in E(\mathcal{L}_k)),$$

where $W' = W - \{e_1, e_2\}$, $R' = R - \{e_1, e_2\}$ and $h = 1$ (respectively, 2) if $k = 2$ (respectively, 1). Moreover, a linear description of the polytope $Q(\mathcal{L}_{12})$ is the following:

$$
\begin{aligned}
&\sum_{e \in E_k \cup W'} a^i_e x_e &&\geq a^i_0 &&(i \in N_k;\ k = 1, 2),\\
&\sum_{e \in E_k \cup W'} b^i_e x_e + x_{e_k} &&\geq b^i_0 &&(i \in I_k;\ k = 1, 2),\\
&\sum_{e \in R'} x_e + x_{e_1} + x_{e_2} &&\geq r - 1,
\end{aligned}
\tag{3.2}
$$

$$0 \leq x_e \leq 1 \qquad (e \in E(\mathcal{L}_{12})).$$

We are now ready to introduce the concept of clutter amalgam.

DEFINITION 3.4. Let W be a weak cutset of a clutter $\mathcal{L}_{12}$ and let E_1 and E_2 be the shores of W. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be the components of $\mathcal{L}_{12}$ with respect to W, let $\{e_1, e_2\}$

be a W-pair of $\mathcal{L}_{12}$, and let $R = \{e_1, e_2, \ldots, e_r\}$ be the maximal 2-clique of $\mathcal{L}_{12}$ containing $\{e_1, e_2\}$. The *amalgam* of $\mathcal{L}_1$ and $\mathcal{L}_2$ with respect to $\{e_1, e_2\}$ is the clutter $\mathcal{L}$ of the minimal members of the family $\mathcal{F}$ of subsets of $E(\mathcal{L}) = E(\mathcal{L}_{12}) - \{e_1, e_2\}$ defined as follows:

$$\mathcal{F} = \mathcal{F}_1 \cup \{A_1 \cup A_2 - \{e_1, e_2\} : A_i \in \mathcal{L}_{12}, A_i \cap R = \{e_i\}, i = 1, 2\},$$

where $\mathcal{F}_1 = \mathcal{L}_{12} \backslash \{e_1, e_2\}$.

In the special case of edge-clutters, we can state the following.

PROPOSITION 3.5. *Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two edge-clutters. Then their amalgam $\mathcal{L}$ is an edge-clutter.*

Proof. Let $\mathcal{L}_{12}$ be the weak cutset identification of $\mathcal{L}_1$ and $\mathcal{L}_2$ and let W be the corresponding weak cutset. Moreover, let $\{e_1, e_2\}$ be a W-pair and let R be the unique maximal clique containing $\{e_1, e_2\}$. Since $\mathcal{L}_{12} = \mathcal{L}_1 \cup \mathcal{L}_2$, we have that $\mathcal{L}_{12}$ is an edge clutter. Furthermore, we have that the clutter $\mathcal{L}_{12} \backslash \{e_1, e_2\}$ is a subset of $\mathcal{L}_{12}$ and hence is an edge-clutter. Finally, for each pair $A_1, A_2 \in \mathcal{L}_{12}$ satisfying $A_1 \cap R = \{e_1\}$ and $A_2 \cap R = \{e_2\}$, we have that the set $A = A_1 \cup A_2 - \{e_1, e_2\}$ satisfies $|A| = 2$. It follows, by Definition 3.4, that, for each $A \in \mathcal{L}$, we have that $|A| = 2$ and hence that $\mathcal{L}$ is an edge-clutter. □

A crucial relation between the clutter $\mathcal{L}_{12}$, obtained as weak cutset identification of $\mathcal{L}_1$ and $\mathcal{L}_2$, and the amalgam $\mathcal{L}$ is given by the following theorem.

THEOREM 3.6. *A subset $C \subseteq E(\mathcal{L})$ is a cover of $\mathcal{L}$ if and only if $C = C' - \{e_1, e_2\}$ and C' is a cover of $\mathcal{L}_{12}$ with $|C' \cap R| = r - 1$.*

Proof. Sufficiency. Let C' be a cover of $\mathcal{L}_{12}$ with $|C' \cap R| = r - 1$ and suppose that the set $C = C' - \{e_1, e_2\}$ is not a cover of $\mathcal{L}$. It follows that there exists $A \in \mathcal{L}$ such that $A \cap C = \emptyset$. Since C' is a cover of $\mathcal{L}_{12}$, we have that C is a cover of $\mathcal{L}_{12} \backslash \{e_1, e_2\}$ and hence, by Definition 3.4, that there exist A_1 and A_2 in $\mathcal{L}_{12}$ such that $A_i \cap R = \{e_i\}$ for $i = 1, 2$ and $A = A_1 \cup A_2 - \{e_1, e_2\}$. It follows that $C' \cap (A_i - \{e_i\}) = \emptyset$ for $i = 1, 2$ and hence that $\{e_1, e_2\} \subseteq C'$. Now, since $|C' \cap R| = r - 1$, we can assume that $e_k \notin C'$ for some $e_k \in R - \{e_1, e_2\}$. By (ii) of Definition 3.1, there exists $A' \in \mathcal{L}_{12}$ such that $A' - \{e_k\} \subseteq A_1 - \{e_1\}$, and hence it follows that

$$C' \cap A' = \emptyset,$$

contradicting the assumption that C' is a cover of $\mathcal{L}_{12}$.

Necessity. Let $C \subseteq E(\mathcal{L})$ be a cover of $\mathcal{L}$. Then, by Definition 3.4, we have that C is a cover of $\mathcal{L}_{12} \backslash \{e_1, e_2\}$ and hence that $C' = C \cup \{e_1, e_2\}$ is a cover of $\mathcal{L}_{12}$. Since R is a 2-clique of $\mathcal{L}_{12}$, we have that $r - 1 \leq |C' \cap R| \leq r$.

If $|C' \cap R| = r - 1$, then we are done; hence consider the case where $|C' \cap R| = r$. We claim that either $C'_1 = C' - \{e_1\}$ or $C'_2 = C' - \{e_2\}$ is a cover of $\mathcal{L}_{12}$. Suppose that neither C'_1 nor C'_2 is a cover of $\mathcal{L}_{12}$. It follows that there exist $A_1, A_2 \in \mathcal{L}_{12}$ such that $A_i \cap C'_i = \emptyset$ $(i = 1, 2)$. Since C' is a cover of $\mathcal{L}_{12}$, we have that $A_1 \cap R = \{e_1\}$ and $A_2 \cap R = \{e_2\}$. It follows, by Definition 3.4, that there exists $A \in \mathcal{L}$ such that $A \subseteq A_1 \cup A_2 - \{e_1, e_2\}$ and hence that $A \cap C = \emptyset$, contradicting the assumption that C is a cover of $\mathcal{L}$.

Consequently, we can assume, without loss of generality, that C'_1 is a cover of $\mathcal{L}_{12}$, and, since $C = C'_1 - \{e_1, e_2\}$ and $|C'_1 \cap R| = r - 1$, the theorem follows. □

An immediate polyhedral consequence of Theorem 3.6 is given by the following corollary, whose straightforward proof is omitted.

COROLLARY 3.7. *The polytope $Q(\mathcal{L})$ is the projection onto the space $\Re^{E(\mathcal{L})}$ of the facet of the polytope $Q(\mathcal{L}_{12})$ defined by the inequality $\sum_{e \in R} x_e \geq r - 1$.*

Now, given a clutter $\mathcal{L}$ obtained as the amalgam of the clutters $\mathcal{L}_1$ and $\mathcal{L}_2$, we are able to provide the linear description of the polytope $Q(\mathcal{L})$ in terms of the linear descriptions of the polytopes $Q(\mathcal{L}_1)$ and $Q(\mathcal{L}_2)$.

THEOREM 3.8. *Let W be a weak cutset of a clutter $\mathcal{L}_{12}$ and let E_1 and E_2 be the shores of W. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be the components of $\mathcal{L}_{12}$, with respect to W, let $\{e_1, e_2\}$ be a W-pair of $\mathcal{L}_{12}$, and let R be the maximal 2-clique containing $\{e_1, e_2\}$. Let $\mathcal{L}$ be the amalgam of $\mathcal{L}_1$ and $\mathcal{L}_2$ with respect to $\{e_1, e_2\}$ and let (3.1) be the defining linear systems for the polytopes $Q(\mathcal{L}_1)$ and $Q(\mathcal{L}_2)$. Then the following is a defining linear system for the polytope $Q(\mathcal{L})$:*

$$\text{(3.3a)} \qquad \sum_{e\in E_k\cup W'} a^i_e x_e \geq a^i_0 \qquad (i \in N_k;\ k = 1,2),$$

$$\text{(3.3b)} \qquad \sum_{e\in E_1\cup W'} b^i_e x_e + \sum_{e\in E_2\cup W'} b^j_e x_e - \sum_{e\in R'} x_e \geq b^i_0 + b^j_0 - r + 1 \qquad (i \in I_1;\ j \in I_2),$$

$$\text{(3.3c)} \qquad \sum_{e\in E_k\cup W'} b^i_e x_e \geq b^i_0 - 1 \qquad (i \in I_k;\ k = 1,2),$$

$$\text{(3.3d)} \qquad \sum_{e\in R'} x_e \geq r - 3,$$

$$0 \leq x_e \leq 1 (e \in E(\mathcal{L})),$$

where $W' = W - \{e_1, e_2\}$ and $R' = R - \{e_1, e_2\}$.

Proof. By Corollary 3.7, we have that the polytope $Q(\mathcal{L})$ is the projection onto the space $\Re^{E(\mathcal{L})}$ of the facet

$$F_R = Q(\mathcal{L}_{12}) \cap \left\{ x \in \Re^{E(\mathcal{L}_{12})} \ :\ \sum_{e\in R} x_e = r - 1 \right\}$$

of the polytope $Q(\mathcal{L}_{12})$. Consequently, by Proposition 3.3, we have that the polytope F_R is described by the following system:

$$\text{(3.4)} \qquad \begin{array}{rcll} \displaystyle\sum_{e\in E_k\cup W'} a^i_e x_e & \geq & a^i_0 & (i \in N_k;\ k = 1,2), \\ \displaystyle\sum_{e\in E_1\cup W'} b^i_e x_e + x_{e_1} & \geq & b^i_0 & (i \in I_1), \\ \displaystyle\sum_{e\in E_2\cup W'} b^i_e x_e + x_{e_2} & \geq & b^i_0 & (i \in I_2), \\ \displaystyle\sum_{e\in R'} x_e + x_{e_1} + x_{e_2} & = & r - 1, & \end{array}$$

$$-x_e \geq -1, \quad x_e \geq 0 \qquad (e \in E(\mathcal{L}_{12})).$$

Now we can obtain a linear description of $Q(\mathcal{L})$ by Fourier–Motzkin elimination of the variables x_{e_1} and x_{e_2} from system (3.4) [1]. In particular, observe that the variable x_{e_1} has a nonzero coefficient in the inequalities with index in I_1, in the trivial inequalities, and in the equation $\sum_{f\in R} x_f = r-1$. Consequently, the elimination of x_{e_1} from system (3.4) produces the following system:

$$\begin{aligned}
\sum_{e\in E_k\cup W'} a^i_e x_e &\geq a^i_0 && (i\in N_k;\ k=1,2),\\
\sum_{e\in E_1\cup W'} b^i_e x_e - \sum_{e\in R'} x_e - x_{e_2} &\geq b^i_0 - r + 1 && (i\in I_1),\\
\sum_{e\in E_1\cup W'} b^i_e x_e &\geq b^i_0 - 1 && (i\in I_1),\\
\sum_{e\in E_2\cup W'} b^i_e x_e + x_{e_2} &\geq b^i_0 && (i\in I_2),\\
\sum_{e\in R'} x_e + x_{e_2} &\geq r-2,
\end{aligned}\tag{3.5}$$

$$-x_e \geq -1, \quad x_e \geq 0 \qquad (e\in E(\mathcal{L}_{12}) - \{e_1\}).$$

Now the Fourier–Motzkin elimination of the variable x_{e_2} from system (3.5) produces the following linear description of the polytope $Q(\mathcal{L})$:

$$\sum_{e\in E_k\cup W'} a^i_e x_e \geq a^i_0 \qquad (i\in N_k;\ k=1,2), \tag{3.6a}$$

$$\sum_{e\in E_1\cup W'} b^i_e x_e + \sum_{e\in E_2\cup W'} b^j_e x_e - \sum_{f\in R'} x_e \geq b^i_0 + b^j_0 - r + 1 \qquad (i\in I_1;\ j\in I_2), \tag{3.6b}$$

$$\sum_{e\in E_k\cup W'} b^i_e x_e \geq b^i_0 - 1 \qquad (i\in I_k;\ k=1,2), \tag{3.6c}$$

$$\sum_{e\in R'} x_e \geq r-3, \tag{3.6d}$$

$$\sum_{e\in E_1\cup W'} b^i_e x_e - \sum_{e\in R'} x_e \geq b^i_0 - r + 1 \qquad (i\in I_1), \tag{3.6e}$$

$$0 \leq x_e \leq 1 (e\in E(\mathcal{L}_{12}) - \{e_1, e_2\}).$$

Each inequality of system (3.6) has nonnegative coefficients since, by (i) of Lemma 3.2, we have that each inequality of system (3.4) with index $i \in I_k$ $(k = 1,2)$ has the property that $b^i_f \geq b^i_{e_k} = 1$ for each $f \in R'$.

Finally, observe that inequalities (3.6e) are redundant. In fact, they are obtained as sum of inequalities (3.6c) and inequalities $-x_e \geq -1$ $(e \in R')$. □

Remark 3.9. Observe that to describe the polytope $Q(\mathcal{L})$, we need only the facet-defining inequalities $a^T x \geq a_0$ of $Q(\mathcal{L}_1)$ with $a_{e_2} = 0$ and the facet-defining inequalities $b^T x \geq b_0$ of $Q(\mathcal{L}_2)$ with $b_{e_1} = 0$. It follows that the knowledge of the linear structure of $Q(\mathcal{L}_1 \backslash \{e_2\})$ and $Q(\mathcal{L}_2 \backslash \{e_1\})$ is sufficient to describe the linear structure of $Q(\mathcal{L})$.

Theorem 3.8 has the following interesting consequences.

COROLLARY 3.10. *If the clutters $\mathcal{L}_1$ and $\mathcal{L}_2$ are ideal, then their amalgam $\mathcal{L}$ is ideal.*

Proof. To prove the corollary, observe that, if $\mathcal{L}_i$ is ideal, then each nontrivial facet-defining inequality has the form $\sum_{e\in A} x_e \geq 1$ for some $A \in \mathcal{L}_i$ $(i = 1, 2)$. It follows that the W-pair $\{e_1, e_2\}$ is a maximal 2-clique in $\mathcal{L}_{12}$ $(R = \{e_1, e_2\})$. Consequently, we have that, in system (3.3), the inequalities of type (3.3c) and (3.3d) become redundant and that all the other inequalities have right-hand side 1; hence the clutter $\mathcal{L}$ is ideal. □

In the special case of edge-clutters, we have the following corollary.

COROLLARY 3.11. *Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two edge-clutters and let $\mathcal{L}$ be their amalgam. If the graphs $G_{\mathcal{L}_1}$ and $G_{\mathcal{L}_2}$ are perfect, then the graph $G_{\mathcal{L}}$ is perfect.*

Proof. First, observe that the graph $G_{\mathcal{L}}$ is well defined, since, by Proposition 3.5, $\mathcal{L}$ is an edge-clutter.

If the graph $G_{\mathcal{L}_h}$ is perfect for $h = 1, 2$, then each nontrivial facet-defining inequality of the polytope $Q(\mathcal{L}_h)$ has the form $\sum_{e\in R_i} x_e \geq |R_i| - 1$ for some maximal 2-clique R_i of $\mathcal{L}_h$ (2-clique facet).

It follows that the inequalities of type (3.3a) have the same structure. Consider any inequality of type (3.3b), say

$$\sum_{e\in E_1\cup W'} b^i_e x_e + \sum_{e\in E_2\cup W'} b^j_e x_e - \sum_{e\in R'} x_e \geq b^i_0 + b^j_0 - r + 1 \tag{3.7}$$

for some $i \in I_1$ and $j \in I_2$. Since each facet of $Q(\mathcal{L}_h)$ $(h = 1, 2)$ is defined by a 2-clique inequality, we have that there exists a maximal 2-clique R_1 of $\mathcal{L}_1$ containing e_1 and a maximal 2-clique R_2 of $\mathcal{L}_2$ containing e_2 such that inequality (3.7) can be rewritten as

$$\sum_{e\in R_1-\{e_1\}} x_e + \sum_{e\in R_2-\{e_2\}} x_e - \sum_{e\in R'} x_e \geq |R_1| + |R_2| - r - 1.$$

Now, by (i) of Lemma 3.2, we have that $R - \{e_2\} \subseteq R_1$ and $R - \{e_1\} \subseteq R_2$, and, since R is maximal in $\mathcal{L}_{12}$, we have that $R_1 \cap R_2 = R'$. It follows that the above inequality can be rewritten as

$$\sum_{e\in(R_1\cup R_2)-\{e_1,e_2\}} x_e \geq |(R_1 \cup R_2) - \{e_1, e_2\}| - 1.$$

Moreover, the set $\bar{R} = (R_1 \cup R_2) - \{e_1, e_2\}$ is a 2-clique of $\mathcal{L}$. In fact, each pair $\{e_h, e_k\} \subseteq \bar{R}\cap E_1$ belongs to $\mathcal{L}_1 \backslash \{e_1\}$ and hence to $\mathcal{L}$. Similarly, each pair $\{e_h, e_k\} \subseteq \bar{R}\cap E_2$ belongs to $\mathcal{L}_2 \backslash \{e_2\}$. Finally, each pair $\{e_h, e_k\}$ with $e_h \in \bar{R} \cap E_1$ and $e_k \in \bar{R} \cap E_2$ belongs to $\mathcal{L}$ by Definition 3.4.

Consequently, $\bar{R}$ is a 2-clique, and, by the maximality of R_1 and R_2 in $\mathcal{L}_1$ and $\mathcal{L}_2$, it is also maximal in $\mathcal{L}$. It follows that each facet of type (3.3b) is a 2-clique facet.

Finally, the inequalities of type (3.3c) and (3.3d) are clearly 2-clique inequalities; hence the graph $G_{\mathcal{L}}$ is perfect. □

Remark 3.12. It can be easily shown that the graph amalgam operation described by Burlet and Fonlupt [2] (also from a polyhedral point of view [3]) is a special case of clutter amalgam of edge-clutters. Moreover, there are examples of graphs obtained by clutter amalgam (Example 3.13) that cannot be obtained as a simple graph amalgam. It follows that the restriction to graphs of the clutter amalgam (and the weak cutset identification) enlarges the family of perfection-preserving operations.

Example 3.13. Consider the graph $G = (V, E)$ of Fig. 3.1 and recall that each cover of the associated edge-clutter $\mathcal{L}_G$ corresponds to a vertex cover of G. Let $\mathcal{V}(G)$ be the family of all the minimal vertex covers of the graph G (blocker of the associated edge-clutter). Let $W = \{1,2,3,4\}$ and observe that each minimal vertex cover of G that contains the node 3 (4) must contain the node 1 (2). It follows that

$$\mathcal{F}(W) = \{C \cap W : C \in \mathcal{V}(G)\} = \{\{1,3\},\{2,4\},\{1,2,3,4\},\{1,2,4\},\{1,2,3\}\}.$$

Consequently, we have that $|\mathcal{F}(W)| \leq |W|+1$. Moreover, it is easy to see that $\mathcal{F}(W') \leq |W'| + 1$ for each $W' \subset W$ and hence that W is a weak cutset of G. It follows that G is the weak cutset identification of the graphs G_1 and G_2 displayed in Fig. 3.2. Finally, since $\{3,4\}$ is a W-pair of G, we have that the graph $\bar{G}$ in Fig. 3.3 is the clutter amalgam of the graphs G_1 and G_2.

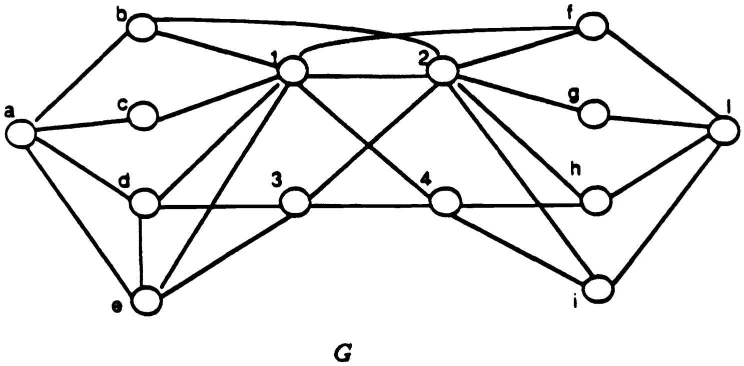


FIG. 3.1

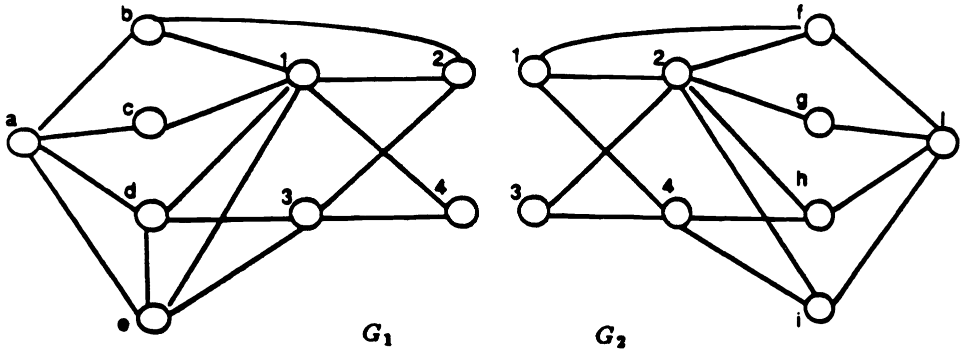
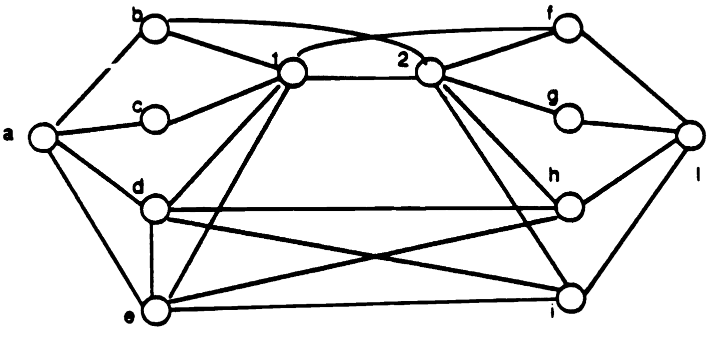


FIG. 3.2



$\tilde{G}$

FIG. 3.3

REFERENCES

[1] A. BACHEM AND M. GRÖTSCHEL, *New aspects of polyhedral theory*, in Modern Applied Mathematics, Optimization and Operations Research, B. Korte, ed., North-Holland, Amsterdam, 1982, pp. 51–106.

[2] E. BURLET AND J. FONLUPT, *Polynomial algorithm to recognize a Meyniel graph*, in Topics in Perfect Graphs, Annals of Discrete Mathematics, 21, C. Berge and V. Chvàtal, eds., North-Holland, Amsterdam, 1984, pp. 225–252.

[3] ———, *Polyhedral Consequences of the Amalgam Operation*, I.M.A.G. Laboratoire ARTEMIS (C.N.R.S.), BP 53X, Grenoble, France 1988.

[4] V. CHVÀTAL, *On certain Polytopes associated with graphs*, J. Combin. Theory Ser. B, 18 (1975), pp. 138–154.

[5] W. H. CUNNINGHAM, *Polyhedra for composed independence systems*, Ann. Discrete Math., 16 (1982), pp. 57–67.

[6] P. NOBILI AND A. SASSANO, *Facets and lifting procedures for the set covering polytope*, Math. Programming Ser. B, 43 (1989), pp. 111–137.

SIAM J. DISC. MATH.
Vol. 6, No. 1, pp. 152–166, February 1993

HAMILTON CYCLES THAT EXTEND TRANSPOSITION MATCHINGS IN CAYLEY GRAPHS OF S_n*

FRANK RUSKEY† AND CARLA SAVAGE‡

Abstract. Let B be a basis of transpositions for S_n and let Cay$(B : S_n)$ be the Cayley graph of S_n with respect to B. It was shown by Kompel'makher and Liskovets [*Kibernetica*, 3 (1975), pp. 17–21] that Cay$(B : S_n)$ is Hamiltonian. This result is extended as follows. Note that every transposition b in B induces a perfect matching M_b in Cay$(B : S_n)$. It is shown here when $n > 4$ that, for any $b \in B$, there is a Hamilton cycle in Cay$(B : S_n)$ that includes every edge of M_b. That is, for $n > 4$, for any basis B of transpositions of S_n, and, for any $b \in B$, it is possible to generate all permutations of $1, 2, \ldots, n$ by transpositions in B so that every other transposition is b.

Key words. Cayley graph, perfect matching, Hamiltonian graph, transposition

AMS(MOS) subject classifications. 05C25, 05C45

1. Introduction. For a finite group G with generating set X, the *Cayley graph of G with respect to the generating set X* is the graph Cay$(X : G)$ with vertex set G, in which g and gx are joined by an undirected edge for every $g \in G$ and $x \in X$. We will consider the edge $\{g, gx\}$ as being labeled x. A compelling question in graph theory is whether every undirected Cayley graph is Hamiltonian. Although there are results such as [CW] and [KW], which show that the answer is yes for certain subclasses of Cayley graphs, the general question remains open. If we require only a Hamilton path, the question is still open and is, in fact, a special case of the more general conjecture of Lovász that every connected, undirected, vertex transitive graph has a Hamilton path [L].

If we restrict our attention to the case when $G = S_n$, the symmetric group of all permutations of $[n] = \{1, 2, \ldots, n\}$, it is still an open problem whether every Cayley graph of S_n is Hamiltonian. The question remains open even when we require that every generator $x \in X$ satisfy $x^2 = \text{id}$. What is known is that for every generating set X of *transpositions*, the Cayley graph of S_n is Hamiltonian. This was first shown by Kompel'makher and Liskovets [KL]. Slater showed in [S] that we could always find a Hamilton path in Cay$(X : S_n)$ that starts at $12 \ldots n$ and ends at a permutation with a j in position k for any $j, k \in [n]$. Tchuente generalized both of these results by showing that any two permutations of different parity are joined by a Hamilton path in Cay$(X : S_n)$ [T]. As an example, the well-known algorithm of Steinhaus [St], Johnson [J], and Trotter [Tr], for generating permutations by adjacent transpositions, gives a Hamilton cycle through the Cayley graph of S_n with generating set $\{(12), (23), (34), \ldots, (n-1\ n)\}$.

However, an element of S_n of order 2 need not be a transposition, so it remains open whether the Cayley graph of S_n on a set of generators, each of order two, is Hamiltonian. Recently it has been shown that the Cayley graph of a Coxeter group (generated by order 2 elements that are *geometric reflections*) is Hamiltonian when the generating set X is the *standard basis* of reflections [CSW]. A related result is that, for A_n generated by the set of 3-cycles $\{(12n), (13n), \ldots, (1\ n-1\ n)\}$, the Cayley graph is Hamiltonian [GR].

In this paper, we consider S_n with any generating set of transpositions, X. Note that each $x \in X$ defines a *perfect matching* in Cay$(X : S_n)$; that is, a set M_x of edges of the

* Received by the editors May 18, 1990; accepted for publication (in revised form) February 26, 1992.

† Department of Computer Science, University of Victoria, P.O. Box 1700, Victoria, British Columbia, V8W-2Y2, Canada. This author's research was supported by Natural Sciences and Engineering Research Council of Canada grant A3379.

‡ Department of Computer Science, North Carolina State University, Box 8206, Raleigh, North Carolina 27612-8206. This author's research was supported by National Science Foundation grant CCR8906500.

graph with the property that each vertex of $\mathrm{Cay}(X : S_n)$ is the end of exactly one edge in M_x

$$M_x = \{\{g, gx\} \mid g \in S_n\}.$$

Knowing that $\mathrm{Cay}(X : S_n)$ is Hamiltonian by [KL], we can ask if M_x extends to a Hamilton cycle. Such a cycle corresponds to a listing of all permutations of $[n]$, in which successive permutations differ by a transposition in X, so that alternate transpositions correspond to the element x.

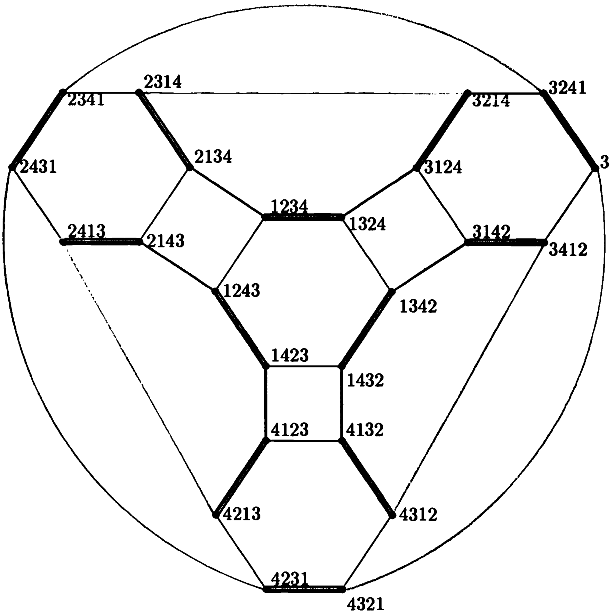


FIG. 1. *The graph* $\mathrm{Cay}(\{(12), (23), (34)\} : S_4)$ *with* $M_{(23)}$.

The graph $C = \mathrm{Cay}(\{(12), (23), (34)\} : S_4)$ is shown in Fig. 1. The tripled lines of the figure indicate edges in the matching $M_{(23)}$, and the list of permutations of Fig. 2 is a Hamilton cycle in C that contains every edge of $M_{(23)}$.

A specific instance of this problem arose initially in the work of Pruesse and Ruskey on listing the linear extensions of certain posets by transpositions [PR]. Let $\mathcal{R}$ be the class of ranked posets in which every nonmaximal element is covered by at least two distinct elements. Examples of posets in $\mathcal{R}$ include the odd fences, crowns, the Boolean algebra lattices, the lattices of subspaces of a finite-dimensional vector spaces over $GF(q)$, and partition lattices. In [PR] it is proved that the linear extensions of any poset in $\mathcal{R}$ can be listed so that every extension differs by a transposition from its predecessor in the list.

Their proof required a cyclic listing of all permutations of $[n]$ by transpositions so that every other transposition was an exchange of the elements in positions 1 and 2. Although they were able to show such a listing was always possible, in some cases the transpositions were not of elements in adjacent positions; these transpositions were the

1234	1324	3124	3214	2314	2134	2143	2413
2431	2341	3241	3421	3412	3142	1342	1432
4132	4312	4321	4231	4213	4123	1423	1243

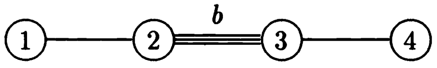


FIG. 2. $B = \{(12), (23), (34)\}$ *and* $b = (23)$(*read across*).

only ones in the proof that were In [RS] we showed that it is possible to list permutations of $[n]$ by *adjacent* transpositions so that every other transposition exchanges the elements in positions 1 and 2. See Fig. 3 for an example when $n = 5$. This result is equivalent to showing that, in the Cayley graph $\text{Cay}(X : S_n)$, where $X = \{(12), (23), \ldots, (n-1\ n)\}$, the perfect matching $M_{(12)}$ extends to a Hamilton cycle. For $n = 4$, there is a Hamilton path including every edgeof $M_{(12)}$, but no Hamilton cycle. A consequence of this result, which is a special case of our main theorem below, is that the linear extensions of the posets in $\mathcal{R}$ can, in fact, be listed by *adjacent* transpositions.

Our major result in this paper is the following theorem.

MAIN THEOREM. *Let X be a generating set of transpositions for S_n, where $n > 4$. Then, for any $x \in X$, M_x extends to a Hamilton cycle in* $\text{Cay}(X : S_n)$.

A *basis* for S_n is a minimal set of generators for S_n. Without loss of generality, we may assume that our generating set of transpositions for S_n is a basis B, so that the transpositions can be described as a tree T_B: the vertices of T_B are the positions $1, 2, \ldots, n$, where i and j are joined by an edge if and only if (ij) is a transposition in B. For $b \in B$, we refer to the ordered pair $\langle T_B, b\rangle$ as a *combination*. A combination $\langle T_B, b\rangle$ is said to be *ordinary* if there are two edges e_1, e_2 in T_B such that (a) $e_1 \neq b$, $e_2 \neq b$, and (b) the edges e_1 and e_2 are not adjacent. A combination that is not ordinary is *exceptional*.

The reason for distinguishing between ordinary and exceptional combinations is that our basic proof technique is to splice together Hamilton cycles in certain induced subgraphs. This splicing is based on small cycles that do not contain any edges labeled b. If $\langle T_B, b\rangle$ is ordinary, then every vertex of $\text{Cay}(B : S_n)$ is on a 4-cycle with no edge labeled b. Specifically, if $c, d \neq b$ are nonadjacent edges of T_B, then $(cd)^2 = \text{id}$; so, for any vertex π of $\text{Cay}(B : S_n)$, the sequence

$$\pi, \pi c, \pi cd, \pi cdc, \pi cdcd = \pi$$

is a 4-cycle. However, if $\langle T_B, b\rangle$ is exceptional, any edges $c, d \neq b$ of T_B are adjacent, so generators c and d do not commute. In this case, there will be no 4-cycles in $\text{Cay}(B : S_n)$ not containing b. Instead, $(cd)^3 = \text{id}$, which gives rise to 6-cycles not containing b.

A *star* is a tree of n vertices in which one vertex has degree $n - 1$ and a *flare* is a tree of n vertices in which one vertex has degree $n - 2$ and one vertex has degree 2. We refer to a vertex of degree 1 in a tree as a *leaf* of the tree. See Fig. 4. Exceptional combinations are characterized in the following lemma, which we state without proof.

LEMMA 1. *An exceptional combination $\langle T, (ij)\rangle$ for $n > 4$ must either be a star or be a flare in which i is a leaf and j is a vertex of degree 2 (or vice versa).*

12345	21345	23145	32145	31245	13245	13425	31425
34125	43125	43152	34152	31452	13452	13542	31542
31524	13524	13254	31254	32154	23154	23514	32514
32541	23541	25341	52341	53241	35241	35421	53421
54321	45321	45231	54231	52431	25431	25413	52413
54213	45213	42513	24513	24531	42531	42351	24351
23451	32451	32415	23415	24315	42315	43215	34215
34251	43251	43521	34521	34512	43512	45312	54312
53412	35412	35142	53142	53124	35124	35214	53214
52314	25314	25134	52134	51234	15234	15324	51324
51342	15342	15432	51432	54132	45132	41532	14532
14352	41352	41325	14325	14235	41235	42135	24135
21435	12435	12453	21453	24153	42153	41253	14253
14523	41523	45123	54123	51423	15423	15243	51243
52143	25143	21543	12543	12534	21534	21354	12354

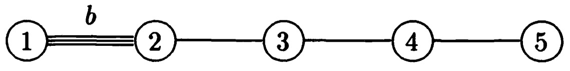


FIG. 3. $B = \{(12), (23), (34), (45)\}$ *and* $b = (12)$(*read across*).

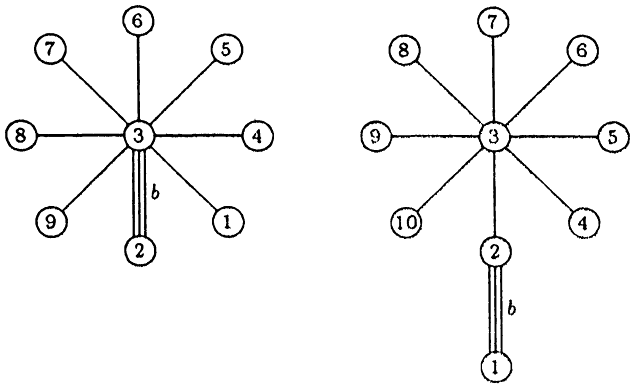


FIG. 4. *Exceptional combinations: star* (*left*) *and flare* (*right*) *with b as indicated.*

The proof of the Main Theorem uses a different construction for each of the following three families of combinations: (i) ordinary combinations, (ii) exceptional combinations in which $n > 4$ and T is a flare, and (iii) exceptional combinations in which $n > 4$ and T is a star.

Within each family, the construction relies inductively only on members of the same family, so the three cases can be handled independently. Section 2 concerns ordinary combinations. Exceptional combinations are handled in §3. Section 4 contains extensions and open problems.

2. Ordinary combinations. An ordinary combination $\langle T, (ij)\rangle$ is *minimal* if, for every leaf $k \neq i, j$, the combination $\langle T - k, (ij)\rangle$ is exceptional. The following lemma is easily proved.

LEMMA 2. *There are three nonisomorphic minimal ordinary combinations. They are shown in Figs.* 2, 3, *and* 5.

For $b \in B$, define a *b-alternating path* (*cycle*) to be a path (cycle) in $\mathrm{Cay}(B\!:\!S_n)$ in which alternate edges are labeled b. Furthermore, in the case of a b-alternating path, the first and last edge of the path must be labeled b. For example, the cycle in Fig. 3 is a (12)-alternating Hamilton cycle in $\mathrm{Cay}(B\!:\!S_n)$ where $B = \{(12), (23), \dots, (n{-}1\,n)\}$.

In this section, we show that, when B is a basis of transpositions for S_n, with $b \in B$ and $\langle T_B, b\rangle$ is an ordinary combination, then $\mathrm{Cay}(B\!:\!S_n)$ has a b-alternating Hamilton cycle. The proof is by an inductive construction and will require a somewhat stronger hypothesis.

If Q is any b-alternating cycle in $\mathrm{Cay}(B\!:\!S_n)$, an *(i, j)-insertion pair* for Q is a pair of consecutive vertices, α, β on Q satisfying (1) $\alpha(i) = \beta(i) = j$, and (2) the edge joining α and β is not labeled b (i.e., $\{\alpha, \beta\} \notin M_b$).

THEOREM 1. *Let B be a basis of transpositions for S_n and let $b \in B$ be such that $\langle T_B, b\rangle$ is an ordinary combination. Then* $\mathrm{Cay}(B\!:\!S_n)$ *has a b-alternating Hamilton cycle Q. Furthermore, Q can be chosen so that, for every $i, j \in [n]$, Q has an (i, j)-insertion pair, and, for each $i \in [n]$, there is some $j \in [n]$ for which Q has two distinct (i, j)-insertion pairs.*

Proof. If the ordinary combination $\langle T_B, b\rangle$ is minimal, then by Lemma 2 it must be isomorphic to one of the combinations in Fig. 2, Fig. 3, or Fig. 5, each shown with a cycle Q satisfying the conditions of the theorem.

Otherwise, assume inductively that the theorem is true for all ordinary combinations with fewer vertices than T_B. Since $\langle T_B, b\rangle$ is not minimal, T_B contains a leaf v, not incident with the edge labeled b, such that $\langle T_B - v, b\rangle$ is an ordinary combination. Let z be the unique vertex of T_B adjacent to v.

The Cayley graph of S_n on the set $B \setminus \{(zv)\}$ has n components $G_1, G_2, \cdots, G_n$, where G_k is the subgraph of $\mathrm{Cay}(B\!:\!S_n)$ induced by all permutations π with $\pi(v) = k$. Let G' denote the Cayley graph of permutations of $[n] \setminus \{v\}$, generated by the set $B \setminus \{(zv)\}$. Then the induction hypothesis holds for G'. Each G_k is isomorphic to G'; so, by induction, G_v, in particular, has a b-alternating Hamilton cycle Q_v. Furthermore, for each i, j satisfying $i \neq v$, $j \neq v$, Q_v has an (i, j)-insertion pair and, for each $i \neq v$, there is some $j \neq v$ for which Q_v has two (i, j)-insertion pairs.

For $k \neq v$, interchanging k and v in every permutation on Q_v gives a b-alternating Hamilton cycle Q_k in G_k.

Now, to obtain the desired cycle Q for $\mathrm{Cay}(B\!:\!S_n)$, each of the cycles Q_k, where $k \neq v$, is spliced into the cycle Q_v at a (z, k)-insertion pair of Q_v (z is the unique vertex of T_B adjacent to v.) This is done as follows (see Fig. 6). Let α, β be the (z, k) insertion

12345	21345	21354	12354	13254	31254	31245	13245
14235	41235	41253	14253	12453	21453	21435	12435
13425	31425	31452	13452	14352	41352	43152	34152
34125	43125	42135	24135	23145	32145	32154	23154
25134	52134	52143	25143	24153	42153	45123	54123
54132	45132	41532	14532	14523	41523	42513	24513
24531	42531	43521	34521	34512	43512	45312	54312
51342	15342	13542	31542	35142	53142	53124	35124
31524	13524	15324	51324	52314	25314	25341	52341
53241	35241	32541	23541	23514	32514	35214	53214
51234	15234	12534	21534	21543	12543	15243	51243
54213	45213	45231	54231	52431	25431	25413	52413
51423	15423	15432	51432	53412	35412	35421	53421
54321	45321	42351	24351	23451	32451	34251	43251
43215	34215	32415	23415	24315	42315	41325	14325

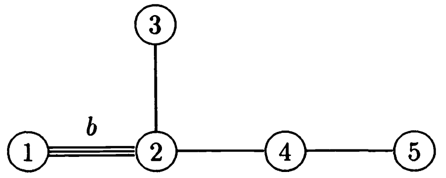


FIG. 5. $B = \{(12), (23), (24), (45)\}$ *and* $b = (12)$(*read across*).

pair of Q_v. Let α', β' be the corresponding pair on the cycle Q_k; that is, α' is obtained from α by interchanging v and k, and similarly for β and β'. Then simply delete edges $\alpha\beta$ and $\alpha'\beta'$ and add edges $\alpha\alpha'$ and $\beta\beta'$ corresponding to the generator (zv) in B.

It remains to show that, after all Q_k are spliced into Q_v to form Q, there is still an (i, j)-insertion pair for every $i, j \in [n]$ and that, for each i, there is some j for which Q has two (i, j)-insertion pairs.

First, consider $i \neq z, v$ and $j \neq v$. The cycle Q_v has an (i, j)-insertion pair and, for some $t \neq v$, there are two (i, t)-insertion pairs. These pairs are still in the final cycle Q, unless some Q_k was spliced into Q_v at a (z, k)-insertion pair α, β, which was also an (i, j)-insertion pair. Then, however, for any $l \neq j, k$, consider the consecutive pair α^*, β^* on Q_l obtained by swapping elements v and l in each of α, β (see Fig. 7). Then α^*, β^* is an (i, j)-insertion pair on Q_l. Note that $\alpha^*(z) = \beta^*(z) = k$. In splicing Q_l into Q_v, however, Q_l is split only at a pair with element v in position z, so α^*, β^* is still an (i, j)-insertion pair in Q. Thus, after splicing, there is no net loss in insertion pairs for $i \neq z, v$ and $j \neq v$.

For $i \neq z, v$ and $j = v$, choose $k \neq v$. In Q_v there was an (i, k)-insertion pair α, β. Interchanging elements v and k in each of α, β gives an (i, v)-insertion pair on Q_k. Since $i \neq z$, this is not the pair in Q_k that was split when Q_z was spliced into Q_v. Thus each Q_k, $k \neq v$ contributes an (i, v)-insertion pair to Q.

If $i = v$, the number of (v, k)-insertion pairs on Q_k is $(n-1)!/2$. During the splicing, only one pair is split for $k \neq v$ and only $n - 1$ pairs for $k = v$. So Q contains a (v, k)-

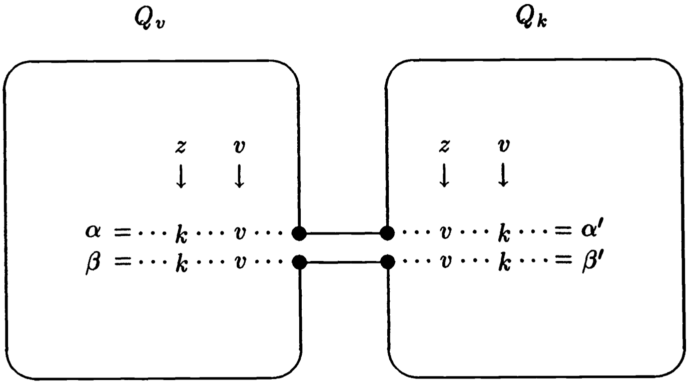

Q_v
Q_k
z v
$\alpha = \cdots k \cdots v \cdots$
$\beta = \cdots k \cdots v \cdots$
$\cdots v \cdots k \cdots = \alpha'$
$\cdots v \cdots k \cdots = \beta'$

FIG. 6. *Splicing cycle Q_k into cycle Q_v at a (z, k) insertion pair in proof of Theorem* 1.

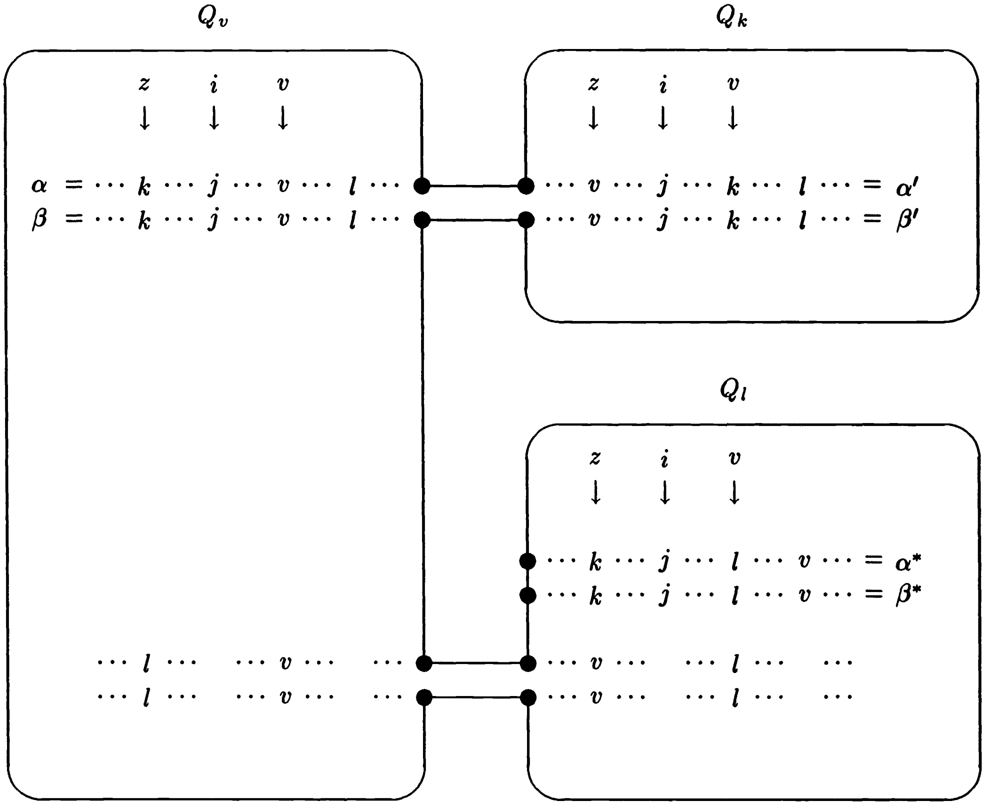

Q_v
Q_k
Q_l
z i v
$\alpha = \cdots k \cdots j \cdots v \cdots l \cdots$
$\beta = \cdots k \cdots j \cdots v \cdots l \cdots$
$\cdots v \cdots j \cdots k \cdots l \cdots = \alpha'$
$\cdots v \cdots j \cdots k \cdots l \cdots = \beta'$
$\cdots k \cdots j \cdots l \cdots v \cdots = \alpha^*$
$\cdots k \cdots j \cdots l \cdots v \cdots = \beta^*$
$\cdots l \cdots \quad \cdots v \cdots \quad \cdots$
$\cdots v \cdots \quad \cdots l \cdots \quad \cdots$

FIG. 7. *Conservation of (i, j)-insertion pairs when $i \neq z, v$ and $j \neq v$.*

insertion pair for every k, as well as two (v, v)-insertion pairs.

In the case where $i = z$, splicing Q_k into Q_v for $k \neq v$ can only split Q_k at a (z, j)-insertion pair for $j = v$. So, even after splicing, Q_k will contain (z, j)-insertion pairs for every $j \neq v, k$. Choose any l, m distinct from k and v. Then each of Q_l and Q_m contains a (z, k)-insertion pair, even after splicing.

Finally, we must check for a (z, v)-insertion pair. The cycle Q_v has none, and each Q_k, $k \neq v$ gets split at a (z, v)-insertion pair during splicing. However, by induction, there is some $j \neq v$ for which Q_v contains two distinct (z, j)-insertion pairs. Corresponding to these, Q_j contains two (z, v)-insertion pairs. Thus, even after splicing, Q_j contains a (z, v)-insertion pair. □

3. Stars and flares: Exceptional combinations. Let B be a basis of transpositions for S_n and $b \in B$. We consider here the cases where T_B is a star or a flare in which b joins the vertex of degree 2 with a leaf (see Fig. 4.) In both cases, any two edges in $B \setminus \{b\}$ are adjacent, so the technique used for ordinary combinations will not work. We focus attention on flares and then show that stars can be handled similarly.

If T_B is a flare, we can assume that $B = F_n = \{(12), (23), (34), (35), \ldots, (3n)\}$. For $n \geq 5$, flares are isomorphic to ordinary combinations, unless $b = (12)$. We show now that even in this case, $\mathrm{Cay}(F_n : S_n)$ has a (12)-alternating Hamilton cycle.

12345	21345	21435	12435	12534	21534	21354	12354
12453	21453	21543	12543	15243	51243	51423	15423
15324	51324	51234	15234	15432	51432	51342	15342
13542	31542	31452	13452	13254	31254	31524	13524
13425	31425	34125	43125	43215	**34215**	**34512**	43512
43152	34152	34251	43251	43521	34521	35421	53421
53124	35124	35214	53214	53412	35412	35142	53142
53241	35241	32541	23541	25341	52341	52143	25143
25413	**52413**	**52314**	25314	25134	52134	52431	25431
24531	42531	45231	54231	54321	45321	45123	54123
54213	45213	45312	54312	54132	45132	41532	14532
14352	41352	41253	14253	14523	41523	41325	14325
14235	41235	42135	24135	24315	42315	42513	24513
24153	42153	42351	24351	23451	32451	32154	23154
23514	32514	32415	23415	23145	32145	31245	13245

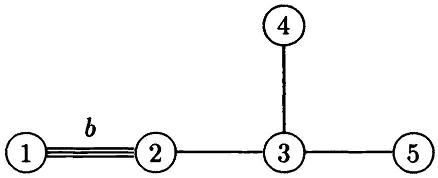


FIG. 8. *Basis case for flares (read across).*

THEOREM 2. *For $n \geq 5$, $\mathrm{Cay}(F_n : S_n)$ has a (12)-alternating Hamilton cycle H satisfying the following conditions:*

(1) *For n odd, there are consecutive permutations σ_n, τ_n on H satisfying*

$$\sigma_n(3) = 2, \quad \sigma_n(n-1) = 1, \quad \sigma_n(n) = n,$$

$$\tau_n(3) = n, \quad \tau_n(n-1) = 1, \quad \tau_n(n) = 2;$$

(2) *For $0 \leq k < (n-1)/2$ when n is even and for $1 \leq k < (n-1)/2$ when n is odd, there are consecutive permutations $\alpha_n^{(k)}$ and $\beta_n^{(k)}$ on H satisfying*

$$\alpha_n^{(k)}(3) = 2k+1, \quad \alpha_n^{(k)}(n) = 2k+2,$$

$$\beta_n^{(k)}(3) = 2k+2, \quad \beta_n^{(k)}(n) = 2k+1.$$

Proof. The theorem is true when $n = 5$, as demonstrated in Fig. 8. Note that, on the cycle of Fig. 8, the required consecutive permutations σ_5 and τ_5 are 34215 and 34512. The required consecutive permutations $\alpha_5^{(1)}$ and $\beta_5^{(1)}$ are 52314 and 52413. (Note that the order does not matter as long as the permutations appear consecutively.)

Assume that, for some $n \geq 5$, Cay$(F_n : S_n)$ has a (12)- alternating Hamilton cycle H satisfying conditions (1) and (2) of the theorem. If we append $n+1$ to every permutation on H, we have a b-alternating cycle in Cay$(F_{n+1} : S_{n+1})$, call it H_{n+1}, still satisfying (1) and (2).

For $1 \leq i \leq n$, the subgraph of Cay$(F_{n+1} : S_{n+1})$, induced by the elements of S_{n+1} with i in position $n+1$, is isomorphic to Cay$(F_n : S_n)$, and therefore it contains a (12)-alternating Hamilton cycle H_i. Note that, given any permutation π with $\pi(n+1) = i$ and any transposition of the form $(3\ k) \in F_n$, we may assume that π is followed by an edge labeled $(3\ k)$ on H_i. (Some edge labeled $(3\ k)$ must appear on H_i since F_n is a basis for S_n. Simply arrange the cyclic list of generators corresponding to the edges along H_i to begin with $(3\ k)$ and apply them, starting with permutation π. This yields a new H_i with the required property.)

The idea of the construction is to splice $H_1, \ldots, H_n$ into H_{n+1} in such a way to obtain a (12)-alternating Hamilton cycle H^* in Cay$(F_{n+1} : S_{n+1})$ and preserve properties (1) and (2) of the theorem.

For n odd, we first splice H_1, H_2, and H_n into H_{n+1} at the pair σ'_n, τ'_n on H_{n+1} corresponding to σ_n, τ_n on H (see Fig. 9.) To do this, we use the fact that the following composition of transpositions is the identity:

$$(3\ n)(3\ n{+}1)(3\ n{-}1)(3\ n{+}1)(3\ n)(3\ n{+}1)(3\ n{-}1)(3\ n{+}1) = \text{id}.$$

We know that σ'_n and τ'_n appear consecutively on H_{n+1} and, as discussed above, we may assume without loss of generality that

(i) $[\tau'_n(3\ n{+}1)]$ and $[\tau'_n(3\ n{+}1)](3\ n{-}1)$ appear consecutively on H_n;

(ii) $[\tau'_n(3\ n{+}1)(3\ n{-}1)(3\ n{+}1)]$ and $[\tau'_n(3\ n{+}1)(3\ n{-}1)(3\ n{+}1)](3\ n)$ appear consecutively on H_1; and

(iii)

$$[\tau'_n(3\ n{+}1)(3\ n{-}1)(3\ n{+}1)(3\ n)(3\ n{+}1)]$$

and

$$[\tau'_n(3\ n{+}1)(3\ n{-}1)(3\ n{+}1)(3\ n)(3\ n{+}1)](3\ n{-}1)$$

appear consecutively on H_2.

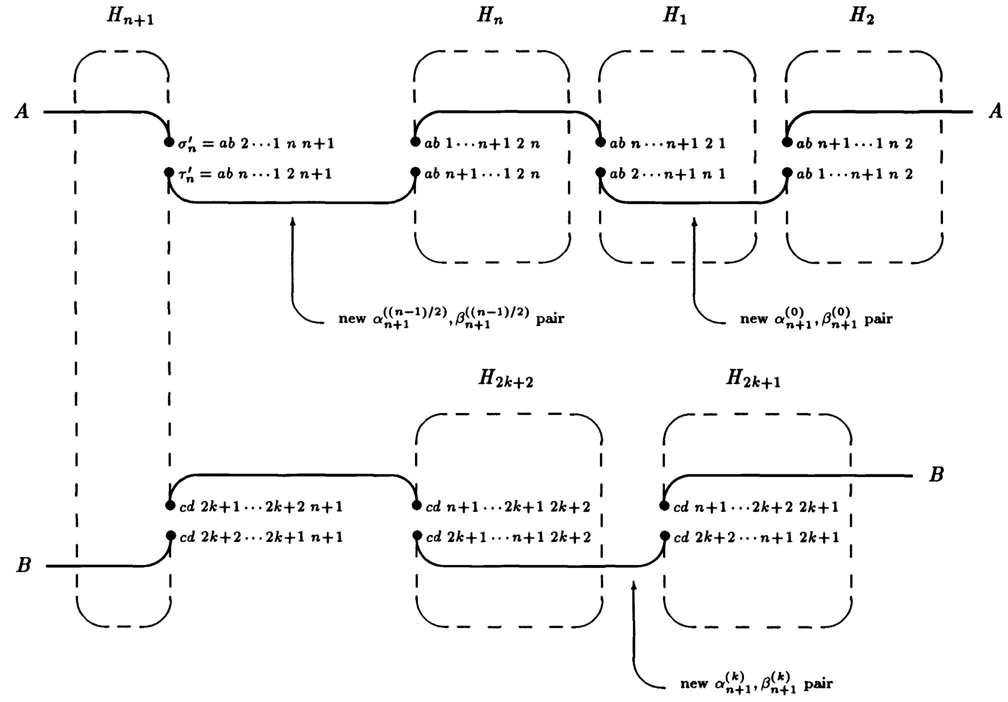

H_{n+1}
H_n
H_1
H_2
A
A
$\sigma'_n = ab\ 2 \cdots 1\ n\ n{+}1$
$\tau'_n = ab\ n \cdots 1\ 2\ n{+}1$
$ab\ 1 \cdots n{+}1\ 2\ n$
$ab\ n{+}1 \cdots 1\ 2\ n$
$ab\ n \cdots n{+}1\ 2\ 1$
$ab\ 2 \cdots n{+}1\ n\ 1$
$ab\ n{+}1 \cdots 1\ n\ 2$
$ab\ 1 \cdots n{+}1\ n\ 2$
new $\alpha_{n+1}^{((n-1)/2)}, \beta_{n+1}^{((n-1)/2)}$ pair
new $\alpha_{n+1}^{(0)}, \beta_{n+1}^{(0)}$ pair
H_{2k+2}
H_{2k+1}
B
B
$cd\ 2k{+}1 \cdots 2k{+}2\ n{+}1$
$cd\ 2k{+}2 \cdots 2k{+}1\ n{+}1$
$cd\ n{+}1 \cdots 2k{+}1\ 2k{+}2$
$cd\ 2k{+}1 \cdots n{+}1\ 2k{+}2$
$cd\ n{+}1 \cdots 2k{+}2\ 2k{+}1$
$cd\ 2k{+}2 \cdots n{+}1\ 2k{+}1$
new $\alpha_{n+1}^{(k)}, \beta_{n+1}^{(k)}$ pair

FIG. 9. *For n odd, splicing the cycles H_i into H_{n+1}.*

In each pair above, as well as for the pair σ'_n, τ'_n, delete the edges joining the two elements of the pair in their respective cycles. Then use edges corresponding to the generator $(3\,n{+}1)$ to join together the cycles as shown in Fig. 9. Note from Fig. 9 that this construction provides us with the required pairs $\alpha_{n+1}^{(0)}, \beta_{n+1}^{(0)}$ and $\alpha_{n+1}^{((n-1)/2)}, \beta_{n+1}^{((n-1)/2)}$ for the (12)-alternating cycle H^* being constructed in Cay$(F_{n+1} : S_{n+1})$.

For $0 \le k < (n-1)/2$ when n is even and $1 \le k < (n-1)/2$ when n is odd, we splice H_{2k+1} and H_{2k+2} into H_{n+1} at the consecutive pair on H_{n+1} corresponding to $\alpha_n^{(k)}, \beta_n^{(k)}$ on H, similar to the method above, but using the identity

$$(3\,n)(3\,n{+}1)(3\,n)(3\,n{+}1)(3\,n)(3\,n{+}1) = \text{id}$$

(see Figs. 9 and 10.) Note from Fig. 9 that this provides for the cycle H^* the pairs $\alpha_{n+1}^{(k)}$, $\beta_{n+1}^{(k)}$ for $1 \le k < (n-1)/2$ and, from Fig. 10, when n is even, gives σ_{n+1}, τ_{n+1}. □

The case of stars can be handled similarly. For a basis B of transpositions of S_n, if T_B is a star, we may assume that $B = R_n = \{(31), (32), (34), (35), \cdots, (3n)\}$, and that

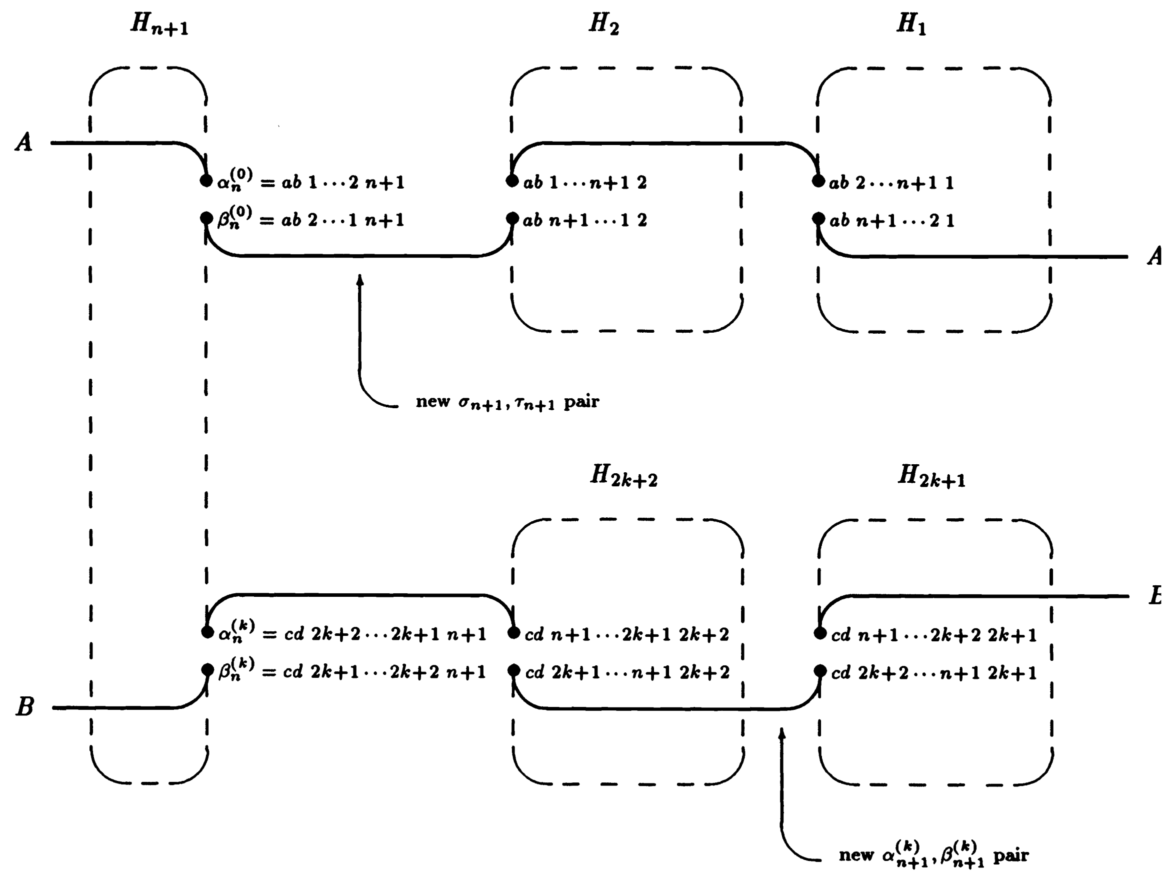

H_{n+1}
H_2
H_1
A
$\alpha_n^{(0)} = ab\ 1 \cdots 2\ n{+}1$
$\beta_n^{(0)} = ab\ 2 \cdots 1\ n{+}1$
$ab\ 1 \cdots n{+}1\ 2$
$ab\ n{+}1 \cdots 1\ 2$
$ab\ 2 \cdots n{+}1\ 1$
$ab\ n{+}1 \cdots 2\ 1$
A
new σ_{n+1}, τ_{n+1} pair
H_{2k+2}
H_{2k+1}
E
$\alpha_n^{(k)} = cd\ 2k{+}2 \cdots 2k{+}1\ n{+}1$
$\beta_n^{(k)} = cd\ 2k{+}1 \cdots 2k{+}2\ n{+}1$
$cd\ n{+}1 \cdots 2k{+}1\ 2k{+}2$
$cd\ 2k{+}1 \cdots n{+}1\ 2k{+}2$
$cd\ n{+}1 \cdots 2k{+}2\ 2k{+}1$
$cd\ 2k{+}2 \cdots n{+}1\ 2k{+}1$
B
new $\alpha_{n+1}^{(k)}, \beta_{n+1}^{(k)}$ pair

FIG. 10. *For n even, splicing the H_i into H_{n+1}.*

the distinguished edge b of B is (32) (see Fig. 4.) In this case, we have the following theorem.

THEOREM 3. *For $n \geq 5$, $\mathrm{Cay}(R_n : S_n)$ has a (32)-alternating Hamilton cycle H satisfying the following conditions*:

(i) *For n odd, there are consecutive permutations σ_n, τ_n on H satisfying*

$$\sigma_n(3) = 2, \quad \sigma_n(n-1) = 1, \quad \sigma_n(n) = n,$$

$$\tau_n(3) = n, \quad \tau_n(n-1) = 1, \quad \tau_n(n) = 2;$$

(ii) *For $0 \leq k < (n-1)/2$ when n is even and for $1 \leq k < (n-1)/2$ when n is odd,* *there are consecutive permutations $\alpha_n^{(k)}$ and $\beta_n^{(k)}$ on H satisfying*

$$\alpha_n^{(k)}(3) = 2k+1, \quad \alpha_n^{(k)}(n) = 2k+2,$$
$$\beta_n^{(k)}(3) = 2k+2, \quad \beta_n^{(k)}(n) = 2k+1.$$

Proof. The theorem is true when $n = 5$, as demonstrated in Fig. 11. Note that, on the cycle of Fig. 11, the required consecutive permutations σ_5 and τ_5 are 34215 and 34512. The required consecutive permutations $\alpha_5^{(1)}$ and $\beta_5^{(1)}$ are 25314 and 25413. The remainder of the proof is identical to the proof of Theorem 2. □

4. Final remarks. There have been some other papers written about finding Hamilton cycles through specified matchings in graphs, but not in connection with Cayley graphs [H], [W]. For example, Häggkvist [H] has shown that, if $d(u)+d(v) \geq |V(G)|+1$ for all nonadjacent vertices u and v of G, then G has a Hamilton path through any given perfect matching.

By deleting all odd permutations from our lists, we obtain listings of the alternating group A_n. In the case of a star, where $B = \{(1\ n), (2\ n), \cdots, (n-1\ n)\}$ and $b = (1\ n)$, note that, since $(1\ n)(j\ n) = (1\ j\ n)$, our results provide another proof of the result of Gould and Roth [GR] that the *digraph* $\mathrm{Cay}(X : A_n)$ is Hamiltonian for $n \geq 5$, where $X = \{(1\ j\ n) \mid 1 < j < n\}$.

Tchuente [T] showed that there is a Hamilton path between any two permutations of opposite parity in $\mathrm{Cay}(B : S_n)$ for any basis of transpositions B. The next lemma shows that it is not, in general, the case that there is a b-alternating path containing M_b between any two permutations of opposite parity.

Let $\langle T_B, b \rangle$ be a combination. If the edge b is removed from T_B then two trees remain; these trees induce a partition of $[n]$ into two sets, say X and Y.

LEMMA 3. *Let X, Y be the partition of $[n]$ induced by $\langle T_B, b \rangle$. If there is a b-alternating Hamilton path in* $\mathrm{Cay}(B : S_n)$ *that starts at the permutation π and ends at the permutation π', then π and π' must satisfy the following condition*:

$$\bigcup_{i \in X} \pi(i) = \bigcup_{i \in X} \pi'(i).$$

Proof. Consider the multigraph $\mathcal{M}$ formed from $\mathrm{Cay}(B : S_n)$ by condensing into a single vertex, for each k-subset S of $[n]$, those permutations π for which $\{\pi(i) \mid i \in X\} = S$. Thus $\mathcal{M}$ has $\binom{n}{k}$ vertices, and each vertex has degree $k!(n-k)!$. Every edge of $\mathcal{M}$ is labeled b, since every transposition other than b either swaps two elements with positions in X or swaps two elements with positions in Y. A b-alternating path in $\mathrm{Cay}(B : S_n)$ that contains every edge of M_b becomes an Euler tour in $\mathcal{M}$. Clearly, this tour must start and end at the same condensed vertex. □

If $n = 4$, then there are two nonisomorphic exceptional combinations $\langle T_B, b \rangle$, namely, the star $B = \{(12), (13), (14)\}$ with $b = (12)$ and the path $B = \{(12), (23), (34)\}$, again with $b = (12)$. In these cases, it is not too difficult to show that there is no b-alternating Hamilton cycle. However, there are b-alternating Hamilton paths containing M_b, as shown in Fig. 12.

Below, we list some questions for further investigation.

(1) Is there an efficient algorithm to generate the permutations on a b-alternating Hamilton cycle? We would like an algorithm whose total storage requirement is

12345	13245	13542	15342	15432	14532	54132	51432
51234	52134	52314	53214	23514	**25314**	**25413**	24513
24153	21453	41253	42153	42351	43251	43521	45321
35421	34521	34125	31425	31245	32145	32415	**34215**
34512	35412	35142	31542	51342	53142	53241	52341
52431	54231	54321	53421	53124	51324	31524	35124
35214	32514	32154	31254	31452	34152	34251	32451
32541	35241	25341	23541	23145	21345	21543	25143
15243	12543	12453	14253	14352	13452	43152	41352
41532	45132	45312	43512	53412	54312	54213	52413
52143	51243	51423	54123	14523	15423	15324	13524
13254	12354	12534	15234	25134	21534	21354	23154
23451	24351	24531	25431	45231	42531	42135	41235
21435	24135	24315	23415	43215	42315	42513	45213
45123	41523	41325	43125	13425	14325	14235	12435

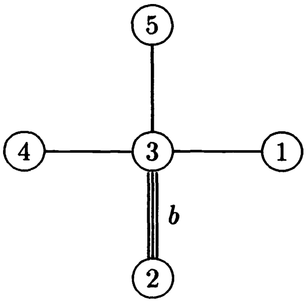


FIG. 11. *Basis case for stars (read across).*

$O(n)$ and whose total running time is $O(n!)$. A straightforward implementation of our proofs leads to algorithms that require $\Theta(n \cdot n!)$ time and $\Theta(n \cdot n!)$ space.

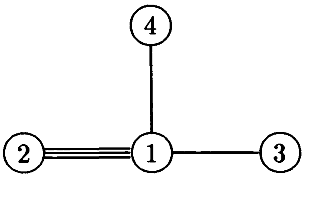
(2) Is the necessary condition of Lemma 3, together with the condition that π and π' have opposite parity, also a sufficient condition for the existence of a Hamilton path from π to π'? We conjecture that the condition is sufficient.

(3) Given a matching M in the n-cube Q_n, is there a Hamilton cycle in Q_n that includes every edge of M?

(4) If X is a set of generators for a group G, and $x \in X$ is an involution (i.e., $x^2 =$ id), then x induces a perfect matching, M_x, in Cay$(X:G)$. A natural question is whether there is a x-alternating path in Cay$(X:G)$.

In general, there is no x-alternating Hamilton path in Cay$(X:G)$. For example, if $X = \{(1\,2), (1\,2\,\ldots\,n)\}$, where $n \geq 3$ is odd, then the following argument, similar to the proof of Lemma 3, shows that Cay$(X:S_n)$ has no $(1\,2)$-alternating path. Condense into single supervertices all those permutations equivalent under the rotation $(1\;2\;\ldots\;n)$. The resulting multigraph has $(n-1)!$ vertices, each of degree n, and any Hamilton path

1234	2134	3124	1324	2314	3214
4213	2413	3412	4312	1342	3142
4132	1432	2431	4231	3241	2341
4321	3421	1423	4123	2143	1243



1234	2134	2314	3214	3124	1324
1342	3142	3412	4312	4321	3421
3241	2341	2431	4231	4213	2413
2143	1243	1423	4123	4132	1432

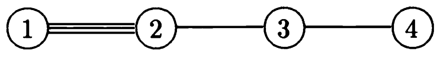


FIG. 12. (12)-*alternating Hamilton paths.*

in Cay($X : S_n$) becomes an Euler tour in the multigraph. Clearly, there is no Euler tour if n is odd.

On the other hand, if G is a Coxeter group and X is a standard basis of reflections, it is likely, for every $x \in X$, that G has an x-alternating Hamilton path. This has been verified already for the groups S_n (this paper) and B_n [Sm], and for several other G, x pairs.

REFERENCES

[CSW] J. H. CONWAY, N. J. A. SLOANE, AND A. R. WILKS, *Gray codes for reflection groups*, Graphs Combin., 5 (1989), pp. 315–325.

[CW] S. J. CURRAN AND D. WITTE, *Hamilton paths in cartesian products of directed cycles*, in Cycles in Graphs, Annals of Discrete Mathematics 27, B. R. Alspach and C. D. Godsil, eds., North-Holland, Amsterdam, 1985.

[GR] R. J. GOULD AND R. ROTH, *Cayley digraphs and* $(1, j, n)$ *sequencings of the alternating group* A_n, Discrete Math., 66 (1987), pp. 91–102.

[H] R. HÄGGKVIST, *On F-Hamiltonian graphs*, in Graph Theory and Related Topics, Academic Press, New York, London, 1979, pp. 219–231.

[J] S. M. JOHNSON, *Generation of permutations by adjacent transpositions*, Math. Comp., 17 (1963), pp. 282–285.

[KL] V. L. KOMPEL'MAKHER AND V. A. LISKOVETS, *Sequential generation of arrangements by means of a basis of transpositions*, Kibernetica, 3 (1975), pp. 17–21.

[KW] K. KEATING AND D. WITTE, *On Hamilton cycles in Cayley graphs of groups with cyclic commutator subgroup*, in Cycles in Graphs, B. R. Alspach and C. D. Godsil, eds., Annals of Discrete Mathematics 27, North–Holland, Amsterdam, 1985.

[L] L. LOVÁSZ, *Combinatorial Structures and Their Applications, Problem* II, Gordon and Breach, London, 1970.

[PR] G. PRUESSE AND F. RUSKEY, *Generating the linear extensions of certain posets by transpositions*, SIAM J. Discrete Math., 4 (1991), pp. 413–422.

[RS] F. RUSKEY AND C. SAVAGE, *Generating permutations by restricted adjacent transpositions*, 1989, unpublished manuscript.

[S] P. J. SLATER, *Generating all permutations by graphical transpositions*, Ars Combin., 5 (1978), pp. 219–225.

[Sm] M. SMITH, *Hamilton cycles alternating the same edge label in Cayley graphs of the hyper-octahedral groups generated by reflections*, Master's project, Dept. of Mathematics, North Carolina State University, Raleigh, NC, 1991.

[St] H. STEINHAUS, *One Hundred Problems in Elementary Mathematics*, Basic Books, New York, 1964.

[T] M. TCHUENTE, *Generation of permutations by graphical exchanges*, Ars Combin., 14 (1982), pp. 115–122.

[Tr] H. F. TROTTER, *PERM* (*Algorithm* 115), Comm. of the ACM, 8 (1962), pp. 434–435.

[W] A. P. WOJDA, *Hamiltonian cycles through matchings*, Demonstratio Math., 22 (1988), pp. 547–553.

SIAM J. DISC. MATH.
Vol. 6, No. 1, pp. 167–180, February 1993

013

A NEW TRIANGULATION FOR SIMPLICIAL ALGORITHMS*

MICHAEL J. TODD[†][‡] AND LEVENT TUNÇEL[†][§]

Abstract. Triangulations are used in simplicial algorithms to find the fixed points of continuous functions or upper semicontinuous mappings; applications arise from economics and optimization. The performance of simplicial algorithms is very sensitive to the triangulation used. Using a facetal description, Dang's D_1 triangulation is modified to obtain a more efficient triangulation of the unit hypercube in R^n, and then, by means of translations and reflections, we derive a new triangulation, D_1', of R^n. It is shown that D_1' uses fewer simplices (asymptotically 30 percent fewer) than D_1 while achieving comparable scores for other performance measures such as the diameter and the surface density. The results of Haiman's recursive method for getting asymptotically better triangulations from D_1, D_1' and other triangulations are also compared.

Key words. subdivisions, simplicial algorithms, triangulations

AMS(MOS) subject classifications. 65H10, 57Q15, 65D05

1. Introduction. Scarf [Sc] was the first to provide a constructive proof of Brouwer's and Kakutani's fixed-point theorems, which have important applications in proving the existence of competitive price equilibria in certain economic models. Scarf used the notion of primitive sets, but most subsequent work used triangulations to discretize the continuous problem. The resulting methods to compute the approximate fixed points, known as simplicial algorithms, are described, for instance in Allgower and Georg [AG1], [AG2], Eaves [E], and Todd [T1]. The performance of such methods depends critically on the triangulation used, and this led to much work on devising efficient triangulations of R^n. Among those used in simplicial algorithms are those of Freudenthal [F], Tucker (Lefschetz [Le, p. 140]), Todd [T2], and Dang [D], known as K_1, J_1, J_1', and D_1, respectively. These triangulations have relatively simple descriptions of their simplices and their pivoting rules, i.e., rules indicating the adjacent simplex found when a specified vertex of a given simplex of the triangulation is dropped. Other triangulations, with attractive properties but with much more complicated descriptions and pivoting rules, are independently devised by Sallee [S1] and by Lee [L], and Sallee's middle cut triangulation [S2]. In this paper, we modify Dang's triangulation to get a more efficient triangulation, which we denote D_1'.

A triangulation of an n-dimensional convex subset of R^n is a locally finite collection of n-dimensional simplices that cover the subset, any two of which intersect in a common face (possibly empty). All of the triangulations above (except J_1') also triangulate the unit cube $I^n := [0,1]^n$ in that their simplices in I^n form a triangulation of R^n. The triangulations of R^n are then obtained by replicating this triangulation using reflections and/or translations. One basic measure of such a triangulation is the number of simplices used to triangulate I^n. This is $n!$ for K_1 and J_1, about $(\mathrm{e}-2)n!$ for D_1, and about $(\mathrm{e}-2)^2 n!$ for D_1'. The triangulation of Lee [L] and Sallee [S1] is slightly better, and that of Sallee [S2] is considerably better, but at a price of increased complexity.

* Received by the editors December 3, 1990; accepted for publication (in revised form) March 12, 1992.

[†] School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York 14853-3801.

[‡] This author's research was supported in part by National Science Foundation grant DMS-88904406 and by the National Science Foundation, the Air Force Office of Scientific Research, and the Office of Naval Research through National Science Foundation grant DMS-8920550.

[§] This author was partially supported by the Ministry of Education of the Republic of Turkey through scholarship program 1416.

An n-simplex can be described as the convex hull of $n+1$ affinely independent vertices or, alternatively, as the solution set of $n+1$ linear inequalities, provided that it is bounded with nonempty interior. The latter description, called a facetal description, often provides a simpler proof that a given collection of simplices forms a triangulation (see, e.g., Todd [T1] and [T2]). We use this description to derive D_1'. A typical simplex of K_1 or J_1 in I^n has the form

$$\{x \in R^n : 1 \geq x_1 \geq x_2 \geq \cdots \geq x_n \geq 0\};$$

all possible orderings of the components give the $n!$ simplices in I^n (the triangulations differ in how this triangulation of I^n is replicated to cover R^n). A typical simplex of D_1 in I^n can easily be shown to be of the form

$$\left\{x \in R^n : x_1, x_2, \ldots, x_p \geq \frac{\sum_{i=1}^p x_i - 1}{p-1} \geq x_{p+1} \geq \cdots \geq x_n \geq 0\right\},$$

where $1 < p \leq n$. As we shall see, the typical simplices of D_1' have a more symmetrical facetal description, which also distinguishes the last $n-q+1$ components of x, where $1 < p < q < n$.

Section 2 defines D_1' and proves that it is indeed a triangulation. In §3 we provide the pivot rules of D_1'. Finally, §4 compares all the triangulations mentioned above according to the number of simplices in the unit cube, their diameters, and their average directional or surface densities. We conclude by comparing the results of applying Haiman's recursive method [H] for obtaining yet better triangulations from these.

2. The triangulation D_1'. We first describe how we triangulate the unit cube $I^n := [0,1]^n$. Copies of the triangulation are then constructed by standard methods (using reflections and translations) to give a triangulation of R^n.

Let $e^1, e^2, \ldots, e^n$ denote the standard basis of R^n and let $e := \sum_j e^j$. We divide the unit cube into a shell S and a core C, which is a neighborhood of the diagonal from zero to e. We triangulate S and C separately; the collection of all the resulting simplices triangulates the unit cube.

C is the convex hull of 0, e, e^i for $i \in N := \{1, 2, \ldots, n\}$, and $e - e^k$ for $k \in N$. We triangulate it into $2^n + 2$ simplices as follows: First, the hyperplane $\{x : e^T x = 1\}$ cuts off the simplex

$$\sigma_- := \operatorname{conv}\{0, e^1, e^2, \ldots, e^n\}, \tag{2.1}$$

and the hyperplane $\{x : e^T x = n-1\}$ cuts off the simplex

$$\sigma_+ := \operatorname{conv}\{e, e - e^1, e - e^2, \ldots, e - e^n\}. \tag{2.2}$$

What remains is $\operatorname{conv}\{e^1, e^2, \ldots, e^n, e - e^1, e - e^2, \ldots, e - e^n\}$, which is an affine transformation of the standard octahedron $\operatorname{conv}\{\pm e^1, \pm e^2, \ldots, \pm e^n\}$. We triangulate this into 2^n simplices, corresponding to the 2^n partitions of N into $I \cup K$; a typical simplex is

$$\sigma_{I,K} = \operatorname{conv}\{\tfrac{1}{2}e, e^i, i \in I, e - e^k, k \in K\}, \tag{2.3}$$

(which corresponds to the simplex $\operatorname{conv}\{0, -e^i, i \in I, e^k, k \in K\}$ of the canonical triangulation of the standard octahedron). It is clear that this provides a triangulation of C (note that it is possible to use just $2^{n-1} + 2$ simplices by joining the center simplices in pairs; if $1 \in I$, replace $\frac{1}{2}e$ by $e - e^1$, while if $1 \in K$, replace $\frac{1}{2}e$ by e^1. Then all simplices include e^1 and $e - e^1$. For symmetry, we have retained the central vertex $\frac{1}{2}e$.).

For future reference, we need a facetal (by linear inequalities) description of the octahedron $\mathrm{conv}\{e^i, i \in N, e - e^k, k \in N\}$, as well as the simplices described above. We write $x(I)$ for $\sum_{i \in I} x_i$, and so forth.

LEMMA 2.1. *With the notation above,* $\mathrm{conv}\{e^i, i \in N, e - e^k, k \in N\} = \{x | (|K| - 1)x(I) - (|I| - 1)x(K) \leq |K| - 1$ *for all partitions* $I \cup K$ *of* $N\}$.

Proof. It suffices to check that the inequality given is satisfied at equality by e^i, $i \in I$, and $e - e^k, k \in K$, and strictly by the other vertices, so that it describes the facet $\mathrm{conv}\{e^i, i \in I, e - e^k, k \in K\}$ of the octahedron. □

For the next result, we call 0, $\frac{1}{2}e$, or e the zeroth vertex of any simplex in which it appears, while e^j or $e - e^j$ is the jth vertex of a simplex in which it appears.

LEMMA 2.2. (a) $\sigma_- = \{x \in R^n | x(N) \leq 1, x_j \geq 0, j \in N\}$. *Moreover, if* $x \in \sigma_-$, *the* j*th barycentric coordinate of* x *is positive if and only if the inequality indexed by* j *is satisfied strictly.* (*Here* $x(N) \leq 1$ *is indexed zero.*)

(b) $\sigma_+ = \{x \in R^n | x(N) \geq n - 1, x_j \leq 1, j \in N\}$. *Moreover, if* $x \in \sigma_+$, *the* j*th barycentric coordinate of* x *is positive if and only if the inequality indexed by* j *is satisfied strictly.* (*Again, the first-listed inequality is indexed by zero.*)

(c) $\sigma_{I,K} = \{x \in R^n | (|K| - 1)x(I) - (|I| - 1)x(K) \leq |K| - 1, x_i \geq (x(N) - 1)/(n - 2), i \in I, x_k \leq (x(N) - 1)/(n - 2), k \in K\}$. *Moreover, if* $x \in \sigma_{I,K}$, *the* j*th barycentric coordinate of* x *is positive if and only if the inequality indexed by* j *is satisfied strictly.* (*Again, the first-listed inequality is indexed by zero.*)

Proof. In each case, we merely check that all the vertices satisfy all the inequalities, strictly if and only if the indices correspond. □

Now we define the shell S and its subdivision. Let $1 < p < q < n$ and let π be a permutation of N. Then let

$$\sigma_{p,q,\pi} := \{x | x_{\pi(1)}, \ldots, x_{\pi(p)} \geq \frac{x(\{\pi(1), \ldots, \pi(p)\}) - 1}{p - 1} \geq x_{\pi(p+1)}$$
$$\geq \cdots \geq x_{\pi(q-1)} \geq \frac{x(\{\pi(q), \ldots, \pi(n)\})}{n - q} \geq x_{\pi(q)}, \ldots, x_{\pi(n)}\}$$

and let S be the union of all such $\sigma_{p,q,\pi}$.(Note that the order of $\{\pi(1), \ldots, \pi(p)\}$ and of $\{\pi(q), \ldots, \pi(n)\}$ is immaterial.) By summing the first p inequalities above, except that indexed by $\pi(i)$, we can deduce that $x_{\pi(i)} \leq 1$ for each i less than or equal to p. Proceeding similarly with the last $n - q + 1$ inequalities yields $x_{\pi(k)} \geq 0$ for each k greater than or equal to q. Hence $\sigma_{p,q,\pi}$ is in the unit cube. It is easy to find an x satisfying all inequalities strictly, whence we can see that $\sigma_{p,q,\pi}$ contains an open ball. Since it is defined by $n + 1$ inequalities, it is an n-simplex. We label the inequalities $\pi(1), \ldots, \pi(p); \pi(p + \frac{1}{2}), \ldots, \pi(q - \frac{1}{2}); \pi(q), \ldots, \pi(n)$, as they appear above. Of course, $\pi(p + \frac{1}{2})$ is a purely formal notation, connoting that it is "between $\pi(p)$ and $\pi(p + 1)$" in some sense.

LEMMA 2.3. *The vertices of* $\sigma_{p,q,\pi}$ *are*

$$e^{\pi(i)}, i = 1, 2, \ldots, p, \tag{2.4}$$

$$\sum_{i=1}^{j} e^{\pi(i)}, j = p, p + 1, \ldots, q - 1, \tag{2.5}$$

and

$$e - e^{\pi(k)}, k = q, q + 1, \ldots, n. \tag{2.6}$$

If indexed by $\pi(1), \ldots, \pi(p), \pi(p+\frac{1}{2}), \ldots, \pi(q-\frac{1}{2})$, *and* $\pi(q), \ldots, \pi(n)$, *then they correspond to the facets with the same index. That is, each vertex is off just the facet with the same index.*

Proof. Again, merely check the inequalities. □

Given $x \in [0,1]^n$, let us suppose the components of x are ordered as follows:

$$1 \geq x_1 \geq \cdots \geq x_n \geq 0.$$

For $p > 1$ and $q < n$, let us write

$$x_{1p} := x(\{1, 2, \ldots, p\}),$$

$$x_{qn} := x(\{q, \ldots, n\}),$$

$$f(p) := \frac{x_{1p} - 1}{p - 1},$$

$$g(q) := \frac{x_{qn}}{n - q}.$$

(We suppress the dependence of f and g on x.) We can think of $f(p)$ as approximately the average of p largest components of x, and $g(q)$ as approximately that of the $n-q+1$ smallest. In fact, if $1 = x_1$ and $x_n = 0$, $f(p)$ is the average of the p largest components of x without the largest, and similarly for $g(q)$.

Clearly, $f(p)$ and $g(q)$ are important in the description of $\sigma_{p,q,\iota}$, where ι is the identity permutation. The following result is very useful.

LEMMA 2.4. *Let* $p > 1$ *and* $q < n$. *Then*

(a) $f(p+1) = ((p-1)/p)f(p) + (1/p)x_{p+1}$;

(b) $g(q-1) = ((n-q)/(n-q+1))g(q) + (1/(n-q+1))x_{q-1}$;

(c) $x_{p+1}\{<, =, >\}f(p)$ *according as* $x_{p+1}\{<, =, >\}f(p+1)$;

(d) $x_{q-1}\{<, =, >\}g(q)$ *according as* $x_{q-1}\{<, =, >\}g(q-1)$;

(e) *If* $2 < p < n-1$, *then* $f(p-1) \leq g(p)$ *and* $g(p) \geq x_p$ *imply that* $f(p) \leq g(p+1)$, *and the third inequality is strict if either of the first two is*;

(f) *If* $2 < p < n-1$, *then* $f(p) \leq g(p+1)$ *and* $x_p \geq f(p)$ *imply that* $f(p-1) \leq g(p)$, *and the third inequality is strict if either of the first two is.*

Proof. Parts (a) and (b) of Lemma 2.4 follow directly from the definition. Since $f(p+1)$ is a strict convex combination of $f(p)$ and x_{p+1}, part (c) follows; similarly, part (d) follows from (b). For part (e), the hypotheses imply that $f(p)$, as a strict convex combination of $f(p-1)$ and x_p, is at most $g(p)$. However, $g(p)$ is a convex combination of $g(p+1)$ and x_p, so $g(p) \geq x_p$ implies that $g(p+1) \geq g(p)$. This gives the weak inequality, and the claim on when it is strict follows also. Part (f) is similar. □

We can now show that our simplices cover the unit cube.

PROPOSITION 2.1. *The simplices* σ_-, σ_+, $\sigma_{I,K}$ *and* $\sigma_{p,q,\pi}$, *where* I, K, p, q, *and* π *range over all appropriate values, cover the unit cube.*

Proof. Choose $x \in [0,1]^n$ and, without loss of generality, assume that

$$1 \geq x_1 \geq x_2 \geq \cdots \geq x_{n-1} \geq x_n \geq 0.$$

Since $x_1, x_2 \leq 1$, we find that $x_1, x_2 \geq f(2) = (x_1 + x_2 - 1)/(2-1)$, and, since $x_{n-1}, x_n \geq 0$, we see that $x_{n-1}, x_n \leq g(n-1) = (x_{n-1} + x_n)/(n-(n-1))$.

Now we proceed as follows. We have $x_1, x_2, \ldots, x_p \geq f(p)$ for $p = 2$, and $g(q) \geq x_q, \ldots, x_{n-1}, x_n$ for $q = n-1$. If $x_{p+1} > f(p)$ and $p < q-1$, we replace p by $p+1$. Then, if $x_{q-1} < g(q)$ and $p < q-1$, we replace q by $q-1$. By (c) and (d) in Lemma 2.4, we see that $x_1, \ldots, x_p \geq f(p)$ and $g(q) \geq x_q, \ldots, x_n$ are preserved.

Suppose that the procedure ends with $p < q$ and

$$x_1 \geq \cdots \geq x_p \geq \frac{x_{1p} - 1}{p-1} \geq x_{p+1} \geq \cdots \geq x_{q-1} \geq \frac{x_{qn}}{n-q} \geq x_q \geq \cdots \geq x_n.$$

Then $x \in \sigma_{p,q,\iota}$, where ι is again the identity permutation.

Otherwise, we want to increase p or decrease q, but we cannot since $p = q-1$. Hence

$$x_1 \geq \cdots \geq x_p \geq f(p), \qquad g(p+1) \geq x_{p+1} \geq \cdots \geq x_n;$$

$x_p < g(p+1)$ or $x_{p+1} > f(p)$.

In either case, $g(p+1) > f(p)$. Now (e) in Lemma 2.4 implies that $g(j+1) > f(j)$ for $p \leq j < n-1$, and (f) implies that $g(j+1) > f(j)$ for $2 \leq j \leq p$. We can now show that $x \in C$.

If $x(N) \leq 1$ or $x(N) \geq n-1$, then $x \in \sigma_-$ or $x \in \sigma_+$, respectively. If $1 \leq x(N) \leq n-1$, then the inequality

$$(|K|-1)x(I) - (|I|-1)x(K) \leq |K| - 1$$

is satisfied for $I = \emptyset$ and $I = N$. Since $x \geq 0$ and $x \leq e$, this inequality is satisfied for I or K a singleton. So assume that I has j elements, $1 < j < n-1$; then the inequality is certainly satisfied if

$$(|K|-1)x_{1j} - (|I|-1)x_{j+1,n} \leq |K| - 1,$$

since the left-hand side only increases by taking the indices of the j largest components of x as I and those of the $n-j+1$ smallest as K. This inequality, however, is exactly equivalent to $f(j) - g(j+1) \leq 0$, which holds as shown above. Hence, if x lies in no $\sigma_{p,q,\pi}$, nor in σ_- or σ_+, it lies in the octahedron $\operatorname{conv}\{e^i, i \in N, e - e^k, k \in N\}$ and hence in some $\sigma_{I,K}$. □

Since there are clearly only a finite number of simplices in our description, to show that we have a triangulation, it only remains to show that any point in the unit cube lies in the relative interior of just one face of a simplex of our collection. First, we need the following lemma.

LEMMA 2.5. *Suppose that $x \in \sigma := \sigma_{p,q,\pi}$ and $x \in \sigma' := \sigma_{p',q',\pi'}$. Then*

$$\frac{x(\{\pi(1), \ldots, \pi(p)\}) - 1}{p-1} = \frac{x(\{\pi'(1), \ldots, \pi'(p')\}) - 1}{p'-1}$$

and

$$\frac{x(\{\pi(q), \ldots, \pi(n)\})}{n-q} = \frac{x(\{\pi'(q'), \ldots, \pi'(n)\}) - 1}{n-q'}.$$

Proof. Without loss of generality, we assume that π is the identity, so that

$$x_1 \geq x_2 \geq \cdots \geq x_n.$$

Since also $x_{\pi'(1)} \geq \cdots \geq x_{\pi'(n)}$, it follows that $x(\{\pi'(1), \ldots, \pi'(p')\})$ is the sum of the p' largest components of x. We must therefore show that $f(p) = f(p')$ and similarly that $g(q) = g(q')$. We prove just the first equation. Assume that $p' > p$. By Lemma 2.4(a), $f(j) \geq x_{j+1}$ implies that $f(j+1) \leq f(j)$ and $f(j+1) \geq x_{j+1} \geq x_{j+2}$. Hence $f(p) \geq f(p+1) \geq \cdots \geq f(p')$. Now either $x_{p+1} = f(p)$ or $x_{p+1} < f(p)$. In the first case, $f(p+1) = f(p)$, while, in the second, Lemma 2.4(c) shows that $x_{p+2} \leq x_{p+1} < f(p+1)$. Thus, as we proceed from p to p', either $f(p) = f(p+1) = \cdots = f(p')$, as desired, or at some stage $x_{j+1} < f(j)$, in which case $x_{j+2} < f(j+1), \ldots, x_{p'} < f(p'-1)$, which implies that $x_{p'} < f(p')$. However, $x \in \sigma'$ shows that the p' largest components of x are at least $f(p')$, a contradiction. Hence $f(p) = f(p')$.

A similar argument yields $g(q) = g(q')$. □

PROPOSITION 2.2. *Each $x \in [0,1]^n$ lies in the relative interior of just one face of a simplex of our collection.*

Proof. If $x \in \sigma$, then the face of σ containing x in its relative interior is called the carrier of x in σ; its vertices are just those corresponding to the positive barycentric coordinates of x in σ.

If x lies in no $\sigma_{p,q,\pi}$, then, by Proposition 2.1, x lies in the core C, and the result is clear. Suppose therefore that $x \in \sigma := \sigma_{p,q,\pi}$ and assume without loss of generality that π is the identity.

We show first that any vertex of the carrier of x in σ is a vertex of the carrier of x in any other simplex of our collection in which x lies, and then the converse follows easily. We distinguish several cases.

First, let e^i be a vertex of the carrier of x in σ. Then the ith barycentric coordinate of x in σ is positive, so, by Lemma 2.3,

$$x_i > \frac{x_{1p} - 1}{p - 1}.$$

If $x \in \sigma' := \sigma_{p',q',\pi'}$, then Lemma 2.5 shows that the ith barycentric coordinate of x in σ' is also positive, so e^i is also a vertex of the carrier of x in σ'. If $x \in \sigma'' := \sigma_{I,K}$, then the argument in the proof of Lemma 2.5 shows that $f(p) \geq f(n)$, so that $x_i > (x_{1n} - 1)/(n-1)$; hence e^i is also a vertex of the carrier of x in σ'' by Lemma 2.2. If $x \in \sigma_-$, then $x_i > x_n \geq 0$ shows that e^i is a vertex of the carrier of x in σ_-. Finally, we show that x cannot belong to σ_+ as follows: For $j = 1$ to $q-1$, $x_j \geq x_{qn}/(n-q)$, with at least one strict inequality. Hence $(n-q)x_{1,q-1} > (q-1)x_{qn}$. Adding $(q-1)x_{1,q-1}$ to both sides gives

$$(q-1)x_{1n} < (n-1)x_{1,q-1} \leq (n-1)(q-1),$$

so $x_{1n} < n - 1$ and $x \notin \sigma_+$.

Next, let $v := e^1 + \cdots + e^p + \cdots + e^j$ be a vertex of the carrier of x in σ, so that the inequality indexed $j + \frac{1}{2}$ of σ is strict, as follows:

$$x_j > x_{j+1},$$

where x_j is replaced by $f(p)$ if $j = p$, and x_{j+1} is replaced by $g(q)$ if $j = q - 1$. Suppose that $x \in \sigma' := \sigma_{p',q',\pi'}$. There is a gap between the jth largest component of x (or $f(p)$) and the $(j+1)$th (or $g(q)$), and, since $f(p) = f(p')$ and $g(q) = g(q')$, this also holds true when x is regarded as a member of σ'. The vertex v is just the sum of the coordinate vectors corresponding to the j largest components of x; this is also a vertex of the carrier of x in σ'. Also, $f(p) > g(q)$ and $x_{p+1} \geq g(q)$ if $p < q - 1$, so, in this case, $f(p+1) > g(q)$. Continuing, $f(q-1) > g(q)$, which implies that x violates one of the inequalities defining C, so it lies in none of its simplices.

Now, let $e - e^k$ be a vertex of the carrier of x in σ. Then we have

$$x_k < \frac{x_{qn}}{n-q}.$$

The argument follows exactly the lines of that for the first case. (Alternatively, we may replace x by $e - x$, the permutation $\pi = \iota$ by its reverse, p by $n + 1 - q$, and q by $n + 1 - p$; the argument is then identical.)

Hence every vertex of the carrier of x in σ is also a vertex of the carrier of x in every other simplex containing it. To show the reverse, we simply observe that, if x lies in a simplex, then the barycentric coordinates of x in that simplex is unique. This completes the proof. □

We have proved the following result.

THEOREM 2.1. *The simplices σ_-, σ_+, $\{\sigma_{I,K}\}$, and $\{\sigma_{p,q,\pi}\}$ triangulate the unit cube $[0,1]^n$.*

To triangulate R^n, we first reflect our triangulation in each of the coordinate hyperplanes $x_j = 0$, to get a triangulation of $[-1,1]^n$. Then we translate this triangulation by each vector in $(2Z)^n$ (with even integer components) to triangulate R^n. Each unit cube corresponds to a vector $v \in (2Z)^n$ and a sign vector $s \in \{-1,+1\}^n$, and is the set $\{x | x_j$ between v_j and $v_j + s_j$, $j \in N\}$. This is the image of the unit cube $[0,1]^n$ under the nonsingular affine transformation $x \to (v + \Sigma x)$, where Σ is the nonsingular diagonal matrix whose diagonal entries are the components of s. Then an explicit description of the vertices of the resulting simplex is obtained by applying the same transformation to the vertices of σ_-, σ_+, $\{\sigma_{I,K}\}$, and $\{\sigma_{p,q,\pi}\}$ given in equations (2.1)–(2.6). We call the resulting triangulation of R^n D_1'; it is a modification of Dang's D_1 triangulation [D].

3. Pivot rules. Here we describe the rules for obtaining the adjacent simplex $\sigma' \in D_1'$, which contains all vertices of $\sigma \in D_1'$ except a specified one v. We confine ourselves to the case where $\sigma \subseteq [0,1]^n$.

Case 1: $\sigma = \sigma_-$. If $v = 0$, it is replaced by $v' = \frac{1}{2}e$, and $\sigma' = \sigma_{I,K}$, where $I = N$, $K = \emptyset$. If $v = e^i$, then it is replaced by $v' = -e^i$, and σ' is the reflection of σ in $x_i = 0$.

Case 2: $\sigma = \sigma_+$. If $v = e$, it is replaced by $v' = \frac{1}{2}e$, and $\sigma' = \sigma_{I,K}$, where $I = \emptyset$, $K = N$. If $v = e - e^k$, then it is replaced by $v' = e + e^k$, and σ' is the reflection of σ in $x_k = 1$.

Case 3: $\sigma = \sigma_{I,K}$. If $v = e^i$, then it is replaced by $v' = e - e^i$, and $\sigma' = \sigma_{I',K'}$, where $I' = I \backslash \{i\}$, $K' = K \cup \{i\}$. If $v = e - e^k$, then it is replaced by $v' = e^k$, and $\sigma' = \sigma_{I',K'}$ with $I' = I \cup \{k\}$, $K' = K \backslash \{k\}$. Finally, if $v = \frac{1}{2}e$, then if $I = N$ $v' = 0$ and $\sigma' = \sigma_-$; if $I = \emptyset$ $v' = e$ and $\sigma' = \sigma_+$; else $v' = \sum_{i \in I} e^i$ and $\sigma' = \sigma_{p,q,\pi}$, where $p = |I| = q - 1$ and π is any permutation placing all $i \in I$ before all $k \in K$.

Case 4: $\sigma = \sigma_{p,q,\pi}$. Suppose that $v = e^j$. Then $v' = \sum_{i=1}^{p} e^{\pi(i)} - e^j$ and $\sigma' = \sigma_{p-1,q,\pi'}$, where π' moves j to position p, i.e., $\pi'(p) = j$ (if it was not already there), as long as $p > 2$. If $p = 2$, then $\{\pi(1), \pi(2)\} = \{j, j'\}$, $v' = e^j + 2e^{j'}$, and σ' is the reflection of σ in $x_{j'} = 1$.

Suppose that $v = \sum_{i=1}^{p} e^{\pi(i)}$. Then $v' = e^{\pi(p+1)}$, and $\sigma' = \sigma_{p+1,q,\pi}$, as long as $p < q - 1$. If $p = q - 1$, then $v' = \frac{1}{2}e$ and $\sigma' = \sigma_{I,K}$, where $I = \{\pi(1), \ldots, \pi(p)\}$, $K = \{\pi(q), \ldots, \pi(n)\}$.

Suppose that $v = \sum_{i=1}^{j} e^{\pi(i)}$, $p < j < q - 1$. Then $v' = \sum_{i=1}^{j-1} e^{\pi(i)} + e^{\pi(j+1)}$ and $\sigma' = \sigma_{p,q,\pi'}$, where $\pi' = (\pi(1), \ldots, \pi(j-1), \pi(j+1), \pi(j), \pi(j+2), \ldots, \pi(n))$.

Suppose that $v = \sum_{i=1}^{q-1} e^{\pi(i)}$. Then $v' = e - e^{\pi(q-1)}$ and $\sigma' = \sigma_{p,q-1,\pi}$, as long as $p < q - 1$. (The case $p = q - 1$ was considered earlier.)

Finally, suppose that $v = e - e^j$. Then $v' = \sum_{i=1}^{q-1} e^{\pi(i)} + e^j$ and $\sigma' = \sigma_{p,q+1,\pi'}$, where π' moves j to position q (if it was not already there), as long as $q < n - 1$. If $q = n - 1$, then $\{\pi(n-1), \pi(n)\} = \{j, j'\}$, $v' = e - e^j - 2e^{j'}$, and σ' is the reflection of σ in $x_{j'} = 0$.

4. Efficiency measures. The performance of simplicial algorithms is very sensitive to the triangulation used. To evaluate the triangulations, several measures of efficiency have been proposed in the literature; see Todd [T1]. In this section, we calculate the values of the efficiency measures for the new triangulation D_1' and compare them with those of D_1 and other previously developed triangulations. Here, we consider D_1' with "paired" simplices in the core, i.e., without the interior vertex $\frac{1}{2}e$.

4.1. The number of simplices in the unit cube. Let $P_n(D_1')$ be the number of simplices used by D_1' to triangulate I^n. The number of simplices in the core is $2 + 2^{n-1}$, and we count the number of simplices in the shell as follows: We know that $2 \le p < q \le n-1$ and the order of the indices $\pi(j)$ for $j \in \{1, \dots, p\}$ (and similarly for $j \in \{q, \dots, n\}$) is irrelevant. Therefore, given p and q, we choose p indices out of n indices, then we choose $(n - q + 1)$ indices out of $(n - p)$ indices, and finally we have $(q - p - 1)!$ different ways of ordering indices $\pi(j)$ for $j \in \{p + 1, \dots, q - 1\}$.

So, for any given p and q, we have

$$\begin{aligned}\binom{n}{p}\binom{n-p}{n-q+1}(q-p-1)! &= \frac{n!}{(n-p)!p!}\frac{(n-p)!}{(q-p-1)!(n-q+1)!}(q-p-1)! \\ &= \frac{n!}{p!(n-q+1)!}\end{aligned}$$

simplices. Hence

$$\begin{aligned}P_n(D_1') &= 2 + 2^{n-1} + \sum_{q=3}^{n-1}\sum_{p=2}^{q-1}\frac{n!}{p!(n-q+1)!} \\ &\le 2 + 2^{n-1} + \sum_{q=3}^{n-1}\frac{n!}{(n-q+1)!}\sum_{p=2}^{\infty}\frac{1}{p!} \\ &= 2 + 2^{n-1} + (\mathbf{e}-2)n!\sum_{q=3}^{n-1}\frac{1}{(n-q+1)!} \\ &\le 2 + 2^{n-1} + (\mathbf{e}-2)n!\sum_{k=2}^{\infty}\frac{1}{k!} \\ &= 2 + 2^{n-1} + (\mathbf{e}-2)^2 n!.\end{aligned}$$

(We use **e** for the base of the natural logarithm since e is reserved for the vector of ones.) Moreover, it is easy to see that the ratio of the left-hand side and the right-hand side approaches 1 as $n \to \infty$. Hence we have the following theorem.

THEOREM 4.1. *$P_n(D_1') \le (\mathbf{e}-2)^2 n! + 2^{n-1} + 2$ and*

$$\lim_{n\to\infty}\frac{P_n(D_1')}{n!} = (\mathbf{e}-2)^2.$$

4.2. The diameter of D_1'. Let τ and τ' be the two facets of a triangulation. The distance between τ and τ' is defined as the minimum number of adjacent simplices that must be visited to get from τ to τ'; i.e., if $\sigma_0, \sigma_1, \ldots, \sigma_m$ is a sequence of simplices in the triangulation such that $\tau \subset \sigma_0$, $\tau' \subset \sigma_m$, and σ_i and σ_{i+1} are adjacent for all $i \in \{0, 1, \ldots, m-1\}$, then this sequence of simplices define a path of length $(m+1)$. So, the distance between τ and τ' is defined as the minimum length of such a path. The diameter of a triangulation is the distance between the farthest two facets or, in other words, the maximum of all such distances.

For our analysis, it is easier to work with full-dimensional simplices. We find the maximum distance between two simplices in D_1'; the diameter is then one more.

If $I \cup K$ is a partition of $N' := \{2, \ldots, n\}$, we denote by $\sigma'_{I,K}$ the simplex conv$\{e^1, e - e^1, e^i, i \in I, e - e^k, k \in K\}$. Let $\sigma'_- := \sigma'_{N',\emptyset}$ and $\sigma'_+ := \sigma'_{\emptyset,N'}$. Then σ_- is adjacent to σ'_-; σ_+ is adjacent to σ'_+, and, clearly, any $\sigma'_{I,K}$ is a distance of at most $n/2$ from either σ'_- or σ'_+.

Now, let $\sigma := \sigma_{p,q,\pi}$ be in the shell and assume that π is the identity permutation. Let $I = \{1, 2, \ldots, p\}$, $J = \{p+1, \ldots, q-1\}$, and $K = \{q, \ldots, n\}$. From σ, we can reach σ'_- in at most $n-1$ steps as follows. First cross the facet defined by $f(p) = x_{p+1}$, so that index $p+1$ moves from J to I. Then successively move $p+2, \ldots, q-1$ from J to I; $|J|$ steps are necessary. Now p has become $q-1$; move across the facet defined by $f(p) = g(q)$. The vertex $e^1 + e^2 + \cdots + e^p$ is replaced by $e - e^1$, and we have entered the core. Finally, move the elements of K one by one into I, in $|K|$ steps. The total is $|J| + 1 + |K| = n - |I| + 1$. Since $|I| \geq 2$, at most $n-1$ steps are necessary. Similarly, at most $n-1$ steps are necessary to move from σ to σ'_+ (actually, only $n-2$, since $1 \in I$ does not have to be moved).

Since $n-1 \geq n/2$, it follows that we can move from any simplex to any other simplex in at most $2n-2$ steps, via either σ'_- or σ'_+.

We now show that $2n-2$ steps are necessary to go from $\sigma' := \sigma_{p,q,\pi'}$, where $p = 2$, $q = n-1$, and $\pi' = (2, 3, 1, 4, 5, \ldots, n)$ (here $n \geq 5$), to $\sigma'' := \sigma_{p,q,\pi''}$, where $\pi'' = (n, n-1, \ldots, 5, 4, 1, 3, 2)$. Let $I' = \{2, 3\}$, $J' = \{1, 4, 5, \ldots, n-2\}$, $K' = \{n-1, n\}$, and $I'' = K'$, $J'' = J'$, $K'' = I'$. We let I, J, K denote the index sets during a typical simplex on the path from σ' to σ''. First, consider an index $j \in J'$. If it leaves J at some step, it must return at a later step, so we charge this index two steps. If it remains in J at all steps, then each index in I' and K' must cross this index, so we charge this index four steps. This accounts for at least $2|J'| = 2n-8$ steps.

Next, if we never reach the core, then each index in $I' \cup K'$ must enter J and leave at the other end, for two steps each or eight in total. This gives $2n$ steps in all. Hence we must reach the core and leave it again; this costs two steps.

Finally, each index in $I' \cup K'$ must cross from one end to the other. (Note that none of the indices is the special index 1, which is "at both ends" in the core.) This takes at least one step for each such index, for a total of 4. Hence $2n-2$ steps in all are necessary.

When we add the extra one to account for the diameter for the facets, we have the following theorem.

THEOREM 4.2. diam$(D_1') = 2n - 1$.

Note that, even though the diameter of D_1' is $2n-1$, when we take a line that goes through the unit cube it might intersect as many as $\frac{1}{2}(n-4)(n-5)$ simplices. In diameter calculations, we free ourselves in taking the shortest distance between two facets; as a result, the shortest path does not necessarily follow a line.

4.3. The surface density of D_1'. The average directional density of a triangulation, a measure introduced by Todd [T1], was shown to be equivalent to the surface density of the same triangulation by Eaves and Yorke [EY], as long as it satisfies certain regularity conditions, which hold for D_1'. In fact, they showed the equivalence for a larger class. The equivalence holds for tilings that do not necessarily have convex cells. They concluded that, given a subdivision of R^n, the average directional density does not depend on how the cells are assembled, but it does depend on the cells used, and they give the following relationship:

average directional density = (surface density)$\cdot g_n$, where

$$g_n = \frac{\Gamma(n/2)}{(n-1)\Gamma(1/2)\Gamma((n-1)/2)}.$$

Here, we calculate the volumes and the surface areas of the simplices in D_1'. Then we can compute the surface density of D_1', $SD(D_1')$, by one of two means, shown below:

$$SD(D_1') = \frac{\sum_{\sigma\in D_1',\sigma\subset I^n} SA(\sigma)}{\sum_{\sigma\in D_1',\sigma\subset I^n} Vol(\sigma)} = \sum_{\sigma\in D_1',\sigma\subset I^n} SA(\sigma)$$

or

$$SD(D_1') = \frac{\sum_{\sigma\in D_1',\sigma\subset I^n} SD(\sigma)Vol(\sigma)}{\sum_{\sigma\in D_1',\sigma\subset I^n} Vol(\sigma)} = \sum_{\sigma\in D_1',\sigma\subset I^n} SD(\sigma)Vol(\sigma).$$

Here, $SA(\sigma)$, $SD(\sigma)$, and $Vol(\sigma)$ denote the surface area, the surface density, and the volume of simplex σ, respectively. Note that the second equation implies that the worst surface density over all individual simplices cannot be better than the surface density of the triangulation.

To calculate the volume of a simplex, we construct an $(n+1)$ by $(n+1)$ matrix M_σ whose columns are the vertices of that particular simplex σ augmented with a $+1$ in the $(n+1)$st position. Then the absolute value of the determinant of the constructed matrix divided by $n!$ is the volume of the simplex.

To calculate the area of a particular facet, we take the vertices of the facet, find the normal of the hyperplane defined by the facet, and create a new point by taking a unit step (in Euclidean norm) from a vertex of the facet in the direction of the normal. Then the convex hull of the vertices of the facet and the new point define an n-simplex, and n times the volume of this simplex is the same as the surface area of the facet.

4.3.1. The simplices in the core. We have two different types of simplices in the core. Simplices $\sigma_- = \text{conv}\{0, e^1, e^2, \dots, e^n\}$ and $\sigma_+ = \text{conv}\{e-e^1, e-e^2, \dots, e-e^n, e\}$ are of type 1, and the remainder of type 2.

For type 1 simplices, we have

$$\text{Vol}(\sigma_-) = \frac{1}{n!}.$$

One of the facets of σ_- is $\text{conv}\{e^1, \dots, e^n\}$, and all other n facets are congruent to $\text{conv}\{0, e^1, \dots, e^{n-1}\}$. Hence

$$SA(\sigma_-) = SA(\text{conv}\{e^1, \dots, e^n\}) + nSA(\text{conv}\{0, e^1, \dots, e^{n-1}\}) = \frac{n+\sqrt{n}}{(n-1)!}.$$

So, we obtain the surface density of type 1 simplices as follows:

$$SD(\sigma_-) = \frac{SA(\sigma_-)}{Vol(\sigma_-)} = (n+\sqrt{n})n.$$

Let $\sigma^{'}$ be a type 2 simplex. Then we have

$$Vol(\sigma^{'}) = \frac{(n-2)}{n!}.$$

Note that any type 2 simplex has e^1 and $e - e^1$ as its vertices. Let τ_1 and τ_2 be the facets that we get from $\sigma^{'}$ by throwing away e^1 and $e - e^1$, respectively. All other facets of $\sigma^{'}$ have the same surface area; let τ_3 denote such a facet. Let p be the number of e^i's that are vertices of $\sigma^{'}$; then the surface areas of the facets of $\sigma^{'}$ are as follows:

$$SA(\tau_1) = \frac{\sqrt{(n-p+1)(p-2)^2 + (p-1)(n-p)^2}}{(n-1)!},$$

$$SA(\tau_2) = \frac{\sqrt{p(n-p-1)^2 + (n-p)(p-1)^2}}{(n-1)!},$$

$$SA(\tau_3) = \frac{\sqrt{(n-2)(n-3)+2}}{(n-1)!}.$$

So if $\sigma_p^{'}$ is a type 2 simplex with parameter p, we obtain

$$SA(\sigma_p^{'}) = SA(\tau_1) + SA(\tau_2) + (n-1)SA(\tau_3).$$

From this formula, we can easily get an upper bound on the surface densities of the type 2 simplices independent of p:

$$SA(\sigma_p^{'}) \le \frac{n(n-2) + n\sqrt{n}}{(n-1)!}, \qquad SD(\sigma_p^{'}) \le n^2 + \frac{n^2\sqrt{n}}{n-2}.$$

4.3.2. The simplices in the shell. For a generic simplex $\sigma_{p,q,\iota}$ in the core, we construct the corresponding matrix $M_{p,q,n}$ as described at the beginning of this section. We then have

$$M_{p,q,n} := \begin{array}{|c|c|c|}
I_{p\times p} & E_{(p-1)\times(q-p)} & E_{(q-1)\times(n-q+1)} \\ \cline{1-2}
0 & triu(E_{(q-p)\times(q-p)}) & \\ \cline{2-3}
 & 0 & (E-I)_{(n-q+1)\times(n-q+1)} \\ \hline
\multicolumn{3}{|c|}{e^T}
\end{array},$$

where $E_{r\times t}$ is the $r \times t$ matrix of ones, $I_{r\times r}$ is the $r \times r$ identity matrix, and $\mathrm{triu}(A)$ is an upper triangular matrix that is the upper triangular portion of A. Hence

$$\mathrm{Vol}(\sigma_{p,q,\iota}) = \frac{1}{n!}|\det(M_{p,q,n})| = \frac{(p-1)(n-q)}{n!}.$$

Let $\tau_{p-1,q-1,n-1}$ be a facet of $\sigma_{p,q,\iota}$ that does not have one of the first p vertices of $\sigma_{p,q,\iota}$ (all such facets are congruent). We find that

$$SA(\tau_{p-1,q-1,n-1}) = \frac{(n-q)\sqrt{p^2-3p+3}}{(n-1)!}.$$

Similarly, we get $\tau_{p,q,n-1}$ as a facet of $\sigma_{p,q,\iota}$ when we throw away one of the last $(n-q+1)$ vertices of $\sigma_{p,q,\iota}$ (again all such facets are congruent). We find that

$$SA(\tau_{p,q,n-1}) = \frac{(p-1)\sqrt{(n-q+1)^2-3(n-q+1)+3}}{(n-1)!}.$$

Finally, we define $\tau^j_{p,q-1,n-1}$ as the facet obtained when the jth vertex, $j \in \{p+1,\ldots,q\}$ of $\sigma_{p,q,\iota}$ is thrown away. We find that

$$SA(\tau^j_{p,q-1,n-1}) = \begin{cases} \dfrac{\sqrt{2}(n-q)(p-1)}{(n-1)!} & j \neq q, j \neq p+1; \\ \dfrac{(n-q)\sqrt{p^2-p+1}}{(n-1)!} & j = q \neq p+1; \\ \dfrac{(p-1)\sqrt{(n-q+1)^2-(n-q+1)+1}}{(n-1)!} & j = p+1 \neq q; \\ \dfrac{\sqrt{(n-q+1)(p-1)^2+p(n-q)^2}}{(n-1)!} & j = q = p+1. \end{cases}$$

So, we have p facets like $\tau_{p-1,q-1,n-1}$, $(n-q+1)$ facets like $\tau_{p,q,n-1}$, and $(q-p)$ facets like $\tau^j_{p,q-1,n-1}$. Thus the total surface area for the simplex $\sigma_{p,q,\iota}$ is

$$SA(\sigma_{p,q,\iota}) = pSA(\tau_{p-1,q-1,n-1}) + (n-q+1)SA(\tau_{p,q,n-1}) + \sum_{p+1}^{q} SA(\tau^j_{p,q-1,n-1}).$$

As $n \to \infty$, the worst surface density is given by the simplices that have *small* p and *large* q as parameters. In particular, the worst simplices are those with $p = 2$ and $q = n-1$, giving

$$SD(\sigma_{2,n-1,\iota}) = \sqrt{2}n^2 + o(n^2).$$

Note that the surface density of the triangulation cannot be worse than the worst simplex in the triangulation; therefore

$$SD(D_1') \leq \sqrt{2}n^2 + o(n^2).$$

(In fact, there are $n!/4$ simplices with $p = 2$ and $q = n - 1$, with total volume $\frac{1}{4}$. If we next consider the simplices with $p = 3$ and $q = n - 1$, or $p = 2$ and $q = n - 2$, which have almost as bad a surface density, the volume increases to $\frac{7}{12}$. Continuing, we find that $SD(D_1') = \sqrt{2}n^2 + o(n^2)$.)

4.4. Comparison of the triangulations in terms of the efficiency measures. We define P_∞ of a triangulation as $\lim_{n\to\infty} P_n/n!$, where P_n is the number of simplices of the triangulation in I^n. Then we have Table 1.

TABLE 1

Triangulation	$P\infty$	Diameter
Freudenthal [F] (K_1)	1	$O(n^2)$
Tucker [Le] (J_1)	1	$O(n^2)$
Sallee [S1] and Lee [L]	0.4762	not known
Sallee [S2]	0	$O(n^2)$
Dang [D] (D_1)	0.7183	$2n - 3$
D_1'	0.5159	$2n - 1$

In terms of P_∞, D_1' is superior to J_1, K_1, and D_1. In terms of their diameters, D_1 and D_1' are the only ones that are known to have $O(n)$ bounds. In terms of the surface densities, D_1' is slightly better than J_1, K_1, and D_1, yet asymptotically they all have the same surface density $\sqrt{2}n^2 + o(n^2)$. (Published version of [D] corrects error.)

4.5. Asymptotically better triangulations. We first mention an elegant result by Haiman [H].

THEOREM 5.1. *If I^n can be triangulated into P_n simplices, then I^{kn} can be triangulated into $[(kn)!/(n!)^k]P_n^k = \rho^{kn}(kn)!$ simplices, where $\rho = (P_n/n!)^{1/n}$.*

Note that, according to the measure $R_n := (P_n/n!)^{1/n}$, $R_\infty = \lim_{n\to\infty} R_n$ we have $R_\infty = 1$ for all triangulations in the previous table. Haiman's result implies that, if a triangulation achieves some $R_n = \rho$ for some n, then the same number ρ is asymptotically achievable, i.e., $R_\infty = \rho$. In other words, this result enables us to get triangulations with $P_\infty = 0$ from those that have $P_\infty < 1$.(Note that this is weaker than saying that $R_\infty = \rho < 1$, which is also true.)

Using this result, we can define new triangulations recursively by using those in the previous table, and choosing the best possible ρ for each triangulation.

We observe that, for each triangulation, R_n converges to 1 very fast. As a result, the best value for ρ is achieved for $n < 10$ for all these triangulations (as expected, smaller ρ values are achieved by those triangulations which have smaller P_∞ values).

Finally, we note that all triangulations in Table 2 except D_1' achieve the minimum value of P_3, all except D_1 achieve the minimum for P_4, and all except D_1 achieve (or are within 1 of) the minimum for P_5. See Mara [M], Cottle [C], Böhm [B], and Hughes [Hu1]. Hughes also shows that any triangulation that slices alternate corners off the unit cube in R^6 cannot achieve fewer than 324 simplices, which is achieved by Sallee's middle-cut triangulation; however, Hughes [Hu2] recently showed that a 6-cube can be triangulated into 312 simplices.

TABLE 2

	Sallee [S1] and Lee [L]		Sallee [S2]		D_1		D_1'	
n	P_n	R_n	P_n	R_n	P_n	R_n	P_n	R_n
3	5	.9410	5	.9410	5	.9410	6	1
4	16	.9036	16	.9036	18	$\boxed{.9306}$	16	.9036
5	67	$\boxed{.8900}$	67	.8900	87	.9377	68	$\boxed{.8926}$
6	364	.8925	324	.8754	518	.9466	384	.9005
7	2445	.9018	1962	.8739	3621	.9539	2628	.9112
8	19296	.9120	13248	$\boxed{.8701}$	28962	.9595	20864	.9201
9	173015	.9210	106181	.8724	260651	.9639	187356	.9292
10	1720924	.9281	931300	.8728	2606502	.9675	1872496	.9360

REFERENCES

[AG1] E. L. ALLGOWER AND K. GEORG, *Simplicial and Continuation Methods for Approximating Fixed Points*, SIAM Rev., 22 (1980), pp. 28–85.

[AG2] ———, *Numerical Continuation Methods: An Introduction*, Springer Series in Mathematics 13, Springer-Verlag, New York, 1990.

[B] J. B. BÖHM, *Complete Enumeration of All Optimal Vertex Preserving Triangulations of the 5-Cube*, Sektion Mathematik der Friedrich-Schiller-Universtät, Jena, DDR, 1988, manuscript.

[C] R. W. COTTLE, *Minimal triangulation of the 4-cube*, Discrete Math., 40 (1982), pp. 25–29.

[D] C. DANG, *The D_1 triangulation of R^n for simplicial algorithms for computing the solutions of nonlinear equations*, Math. Oper. Res., 16 (1991), pp. 148–161.

[E] B. C. EAVES, *A Course in Triangulations for Solving Equations with Deformations*, Lecture Notes in Economics and Mathematical Systems 234, Springer-Verlag, Berlin, 1984.

[EY] B. C. EAVES AND J. A. YORKE, *Equivalence of surface density and average directional density*, Math. Oper. Res., 9 (1984), pp. 363–375.

[F] H. FREUDENTHAL, *Simplizialzerlegungen von Beschränkter Flachheit*, Ann. of Math., 43 (1942), pp. 580–582.

[H] M. HAIMAN, *A simple and relatively efficient triangulation of the n-cube*, Discrete Comput. Geom., 6 (1991), pp. 287-289.

[Hu1] R .B. HUGHES, *Minimum cardinality triangulations of the d-cube for d =5 and d =6*, Dept. of Mathematics, Boise State University, Boise, ID, 1990; Discrete Math., to appear.

[Hu2] ———, *A Triangulation of the 6-Cube with 312 Simplices*, Dept. of Mathematics, Boise State University, Boise, ID, 1992, manuscript.

[L] C. LEE, *Triangulating the d-cube*, in Discrete Geometry and Convexity, J. E. Goodman, E. Lutwak, J. Malkevitch, and R. Pollack, eds., New York Academy of Sciences, New York, 1985, pp. 205–211.

[Le] S. LEFSCHETZ, *Introduction to Topology*, Princeton University Press, Princeton, 1949.

[M] P. S. MARA, *Triangulations for the Cube*, J. Combin. Theory Ser. A, 20 (1976), pp. 170-177.

[Me] O. H. MERRILL, *Applications and Extensions of an Algorithm that Computes Fixed Points of Certain Upper Semi-Continuous Point to Set Mappings*, Ph.D. thesis, Department of Industrial Engineering, University of Michigan, Ann Arbor, MI, 1972.

[S1] J. F. SALLEE, *A Triangulation of the n-cube*, Discrete Math., 40 (1982), pp. 81–86.

[S2] ———, *Middle cut triangulations of the n-cube*, SIAM J. Alg. Discrete Meth., 5 (1984), pp. 407–418.

[Sc] H. SCARF, *The approximation of fixed points of a continuous mapping*, SIAM J. Appl. Math., 15 (1967), pp. 1328–1343.

[T1] M. J. TODD, *The Computation of Fixed Points and Applications*, Lecture Notes in Economics and Mathematical Systems 124, Springer-Verlag, Berlin, 1976.

[T2] ———, *J': A new triangulation of R^n*, SIAM J. Alg. Discrete Meth., 5 (1984), pp. 244–254.

SIAM J. DISC. MATH.
Vol. 6, No. 2, pp. 181-188, May 1993

001

THE PATHWIDTH AND TREEWIDTH OF COGRAPHS*

HANS L. BODLAENDER† AND ROLF H. MÖHRING‡

Abstract. It is shown that the pathwidth of a cograph equals its treewidth, and a linear time algorithm to determine the pathwidth of a cograph and build a corresponding path-decomposition is given.

Key words. graph algorithms, cographs, treewidth, pathwidth

AMS(MOS) subject classifications. 05C05, 05C85, 68R10

1. Introduction.

The pathwidth and treewidth of a graph are two notions with a large number of different applications in many areas, like algorithmic graph theory, very large scale integration (VLSI) design, and others (see, e.g., [1], [3], [9], [15]). The notions also play a major role in the theory of graph minors (see, e.g., [20] and, for applications, [10]). Unfortunately, determining the pathwidth or treewidth of a given graph is NP-complete [2]. In this paper, we show that there are efficient algorithms for determining the pathwidth or treewidth of a cograph. We also derive some technical lemmas, which are not only necessary to prove correctness of the algorithms, but are also interesting in their own right. For instance, we show that the pathwidth of a cograph equals its treewidth.

The complexity of the problems to determine the pathwidth and treewidth of graphs has also been studied for other interesting classes of graphs. Gustedt [12] showed that the pathwidth problem stays NP-complete when restricted to chordal graphs. Sundaram, Singh, and Rangan have obtained a polynomial algorithm to determine the treewidth (but not the pathwidth) of a circular arc graph [22]. For fixed k, the problem of determining whether the pathwidth or treewidth of a given graph is at most k, and if so, building the corresponding path- or tree-decomposition can be solved in $O(n \log n)$ time (by combining the results in [17] and [6]).

The notions of pathwidth and treewidth have several equivalent characterizations (see, e.g., [1], [15], [21]). For instance, a graph is a partial k-tree if and only if its treewidth is at most k.

This paper is organized as follows. In § 2 we give most necessary definitions and some preliminary results. In § 3 we prove a number of interesting graph-theoretic lemmas and theorems. In § 4 we show how these can be used to obtain linear time algorithms for pathwidth and related notions on cographs. Some final remarks are made in § 5.

2. Definitions and preliminary results.

In this section, we give most necessary definitions and some preliminary results. We start with introducing the notion of *cographs*.

Notation. Let $G = (V, E)$, $H = (W, F)$ be undirected graphs.

(i) We denote the disjoint union of G and H by $G \dot{\cup} H = (V \dot{\cup} W, E \dot{\cup} F)$ (where $\dot{\cup}$ is the disjoint union on graphs and sets, respectively);

(ii) With $G \times H$, we denote the following type of "product" of G and H: $G \times H = (V \dot{\cup} W, E \dot{\cup} F \cup \{(v, w) | v \in V, w \in W\})$;

* Received by the editors August 20, 1990; accepted for publication (in revised form) April 1, 1992.

† Department of Computer Science, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, the Netherlands. The research of this author was partially supported by the ESPRIT II Basic Research Actions Program of the European Community contract 3075 (project ALCOM).

‡ Department of Mathematics, Technical University of Berlin, Straße des 17. Juni 136, W-1000 Berlin 12, Germany. The research of this author was partially supported by the Deutsche Forschungsgemeinschaft grant Mo446/1-1.

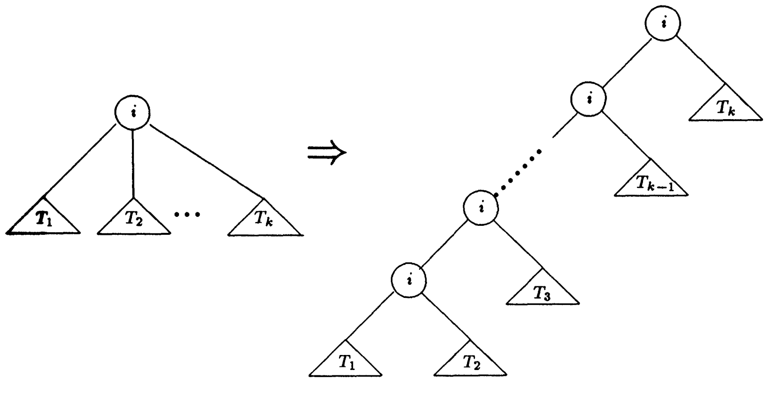


FIG. 1. *Transformation to a binary co-tree* $i \in \{0, 1\}$.

(iii) The complement of G is denoted by $G^c = \{V, E^c\}$, with $E^c = \{(v, w) \mid v, w \in V, v \neq w, (v, w) \notin E\}$.

PROPOSITION 2.1. $\dot{\cup}$, $\times$ *are commutative and transitive.* $G \times H = (G^c \,\dot{\cup}\, H^c)^c$.

DEFINITION 2.1. A graph $G = (V, E)$ is a cograph if and only if one of the following conditions holds:

1) $|V| = 1$;

2) There are cographs $G_1, \ldots, G_k$ and $G = G_1 \,\dot{\cup}\, G_2 \,\dot{\cup} \cdots \dot{\cup}\, G_k$;

3) There are cographs $G_1, \ldots, G_k$ and $G = G_1 \times G_2 \times \cdots \times G_k$.

There are other equivalent characterizations of the class of cographs. Rule 3) can be replaced by the following:

3)′ There is a cograph H and $G = H^c$.

Also, we can restrict k in rules 2) and 3) to be 2. Alternatively, we can define the class of cographs as the graphs that do not contain P_4, a path with four vertices, as an induced subgraph (see, e.g., [7]).

With each cograph $G = (V, E)$, we can associate a labeled tree, called the *cotree* T_G of G. Each vertex of G corresponds to a unique leaf in T_G. Internal vertices of T_G have a label $\in \{0, 1\}$. To each vertex in T_G, we can associate a cotree in the following manner: a leaf corresponds to a cotree, consisting of one vertex. The cograph corresponding to a 0-labeled vertex v in T_G is the disjoint union of the cographs, corresponding to the sons of v in T_G. The cograph corresponding to a 1-labeled vertex v in T_G is the product ("×") of the cographs, corresponding to the sons of v in T_G. Note that $(v, w) \in E$ if and only if the lowest common ancestor of v and w in T_G is labeled with a 1. (There are other very similar notions of "cotree.")

Corneil, Perl, and Stewart [8] gave an $O(n + e)$ algorithm for determining whether a given graph $G = (V, E)$ is a cograph and, if so, for building the corresponding cotree.

A cotree T_G can easily be transformed to an equivalent cotree T'_G such that every internal vertex in T'_G has exactly two sons. (Note that $G_1 \,\dot{\cup} \cdots \dot{\cup}\, G_k = (G_1 \,\dot{\cup} \cdots \dot{\cup}\, G_{k-1}) \,\dot{\cup}\, G_k$, and $G_1 \times \cdots \times G_k = (G_1 \times \cdots \times G_{k-1}) \times G_k$. The resulting operation on trees is illustrated in Fig. 1.)

So, in the remainder of this paper, we assume that cographs G are given together with a binary cotree T_G. Next, we give the definitions of pathwidth and treewidth, introduced by Robertson and Seymour [18], [19].

DEFINITION 2.2. A tree-decomposition of a graph $G = (V, E)$ 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, and T a tree, such that

(i) $\bigcup_{i \in I} X_i = V$;

(ii) For all $(v, w) \in E$, there exists $i \in I$, $v \in X_i \wedge w \in X_i$;

(iii) For all $v \in V$, $\{i \in I \mid v \in X_i\}$ forms a subtree of T.

The treewidth of a tree-decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ is $\max_{i \in I} |X_i| - 1$. The treewidth of G is the minimum treewidth over all possible tree-decompositions of G.

Note that the third condition can be replaced by

$$\forall i, j, k \in I\text{: if } j \text{ is on the path from } i \text{ to } k \text{ in } T\text{, then } X_i \cap X_k \subseteq X_j.$$

There are several other notions that are equivalent to the notion of treewidth, e.g., a graph G is a partial k-tree if and only if *treewidth*$(G) \leqq k$ (see [1], [21]).

The notion of pathwidth is obtained from the notion of treewidth by requiring that the tree T in the tree-decompositions is a path.

DEFINITION 2.3. A path-decomposition of a graph $G = (V, E)$ is a pair $(\{X_i \mid i \in I\}, I)$, with $\{X_i \mid i \in I\}$ a family of subsets, and there exists $r \in \mathbb{N}$: $I = \{1, 2 \cdots, r\}$ such that

(i) $\bigcup_{i \in I} X_i = V$;

(ii) For all $(v, w) \in E$, there exists $i \in I$, $v \in X_i \wedge w \in X_i$;

(iii) For all $v \in V$, there exists $b_v, e_v \in I$, such that for all $i \in I$, $v \in X_i \Leftrightarrow b_v \leqq i \leqq e_v$.

The pathwidth of $(\{X_i \mid i \in I\}, I)$ is $\max_{i \in I} |X_i|$. The pathwidth of G is the minimum pathwidth over all possible path-decompositions of G.

The third condition states that, for all $v \in V$, $\{i \in I \mid v \in X_i\}$ forms an interval in I and is equivalent to "$\forall i, j, k$: $i < j < k \Rightarrow X_i \cap X_k \subseteq X_j$."

The notion of pathwidth is closely related to several other notions, including node search number and interval thickness. (See, e.g., [13]–[15].)

DEFINITION 2.4. The node search number of a graph $G = (V, E)$ is the minimum number of searchers needed to clear all edges of G under the following rules:

(i) Initially, all edges are contaminated;

(ii) A move can consist of

1) putting a searcher on a vertex,

2) removing a searcher from a vertex,

3) moving a searcher over an edge from a vertex to an adjacent vertex;

(iii) A contaminated edge becomes cleared when there is a searcher on both ends of the edge;

(iv) A cleared edge becomes recontaminated when there is a path from the edge to a contaminated edge that does not pass through a vertex with a searcher on it.

DEFINITION 2.5. A graph $G = (V, E)$ is an interval graph if, to each $v \in V$, an interval $[b_v, e_v] \subseteq \mathbb{R}$ can be associated such that for all $v, w \in V$; $(v, w) \in E \Leftrightarrow [b_v, e_v] \cap [b_w, e_w] \neq \emptyset$.

LEMMA 2.1 (see [5], [11]). *Let $G = (V, E)$ be an interval graph. Let the chromatic number of G be $\chi(G)$ and let the maximum size of a clique in G be $\omega(G)$. Then $\chi(G) = \omega(G) =$ treewidth$(G) + 1 =$ pathwidth$(G) + 1$.*

DEFINITION 2.6. The interval thickness of a graph $G = (V, E)$ is the minimum chromatic number of an interval graph H that contains G as a subgraph.

THEOREM 2.1 (see [13], [15]). *For every graph $G = (V, E)$, the pathwidth of G + 1, the node search number of G, and the interval thickness of G are equal.*

3. Graph-theoretic results. In this section, we derive some new and interesting graph theoretic results, which are needed to derive the algorithm but also have interest on their own. We start with a very short proof of a known result.

DEFINITION 3.1. A family $\{T_i \mid i \in I\}$ of subsets of a set T is said to satisfy the Helly property if, for all $J \subseteq I$ with, for all $i, j \in J$, $T_i \cap T_j \neq \emptyset$, it holds that $\bigcap_{j \in J} T_j \neq \emptyset$.

THEOREM 3.1 (see [11, p. 92]). *A family of induced subtrees of a tree satisfies the Helly property.*

LEMMA 3.1 ("clique containment lemma"). *Let $(\{X_i \mid i \in I\}, T = (I, F))$ be a tree-decomposition of $G = (V, E)$ and let $W \subseteq V$ be a clique in G. Then there exists $i \in I$ with $W \subseteq X_i$.*

Proof. Let $T_v = \{i \in I \mid v \in X_i\}$. $\{T_v \mid v \in W\}$ is a family of subtrees of T. By Theorem 3.1, there exists $i \in I$ with, for all $u \in W$, $i \in T_u$. Hence, there exists $i \in I$ with $W \subseteq X_i$. □

Older and longer proofs of Lemma 3.1 can be found in [5], [21]. With the Helly property for trees, we can also prove a variant of the clique containment lemma for bipartite subgraphs.

LEMMA 3.2 ("complete bipartite subgraph containment lemma"). *Let $(\{X_i \mid i \in I\}, T = (I, F))$ be a tree-decomposition of $G = (V, E)$. Let $A, B \subseteq V$ and suppose that $\{(v, w) \mid v \in A, w \in B\} \subseteq E$, $A \cap B = \emptyset$. Then there exists $i \in I$ with $A \subseteq X_i$ or $B \subseteq X_i$.*

Proof. Let $(\{X_i \mid i \in I\}, T = (I, F))$, $G = (V, E)$, and A and B be given. Suppose that for all $i \in I$, $B \not\subseteq X_i$. Let $T_v = \{i \in I \mid v \in X_i\}$. Consider the family $\{T_v \mid v \in B\}$ of subtrees of T. As $\bigcap_{v \in B} T_v = \emptyset$, it follows from Theorem 3.1 that there are $b_1, b_2 \in B$ such that T_{b_1} and T_{b_2} are vertex disjoint. Consider the unique path of T connecting T_{b_1} and T_{b_2}, and let k and l be the border vertices of this path (see Fig. 2).

Each $a \in A$ must be contained in a set X_i with $i \in T_{b_1}$ and in a set X_j with $j \in T_{b_2}$. Hence $a \in X_k$. (Use the definition of tree-decomposition.) So $A \subseteq X_k$. □

LEMMA 3.3. *Let $(\{X_i \mid i \in I\}, T = (I, F))$ be a tree decomposition of $G = (V, E)$, let $A, B \subseteq V$, and suppose that $\{(v, w) \mid v \in A, w \in B\} \subseteq E$, $A \cap B = \emptyset$. Suppose that there exists $i \in I$ with $A \subseteq X_i$. Then there exists an induced subtree $T' = (I', F')$ of T such that*

(i) *For all $i \in I'$, $A \subseteq X_i$;*

(ii) *$B \subseteq \bigcup_{i \in I'} X_i$;*

(iii) *$(\{X'_i \mid i \in I'\}, T' = (I', F'))$ with $X'_i = X_i \cap (A \cup B)$ is a tree-decomposition of the subgraph of G induced by $A \cup B$.*

Proof. Let $I' = \{i \in I \mid A \subseteq X_i\}$. Take $T' = (I', F')$ to be the subgraph of T induced by I'. By definition of tree-decomposition, T' is again a tree. Clearly, condition (i) is fulfilled.

Because there exists an i, with $A \subseteq X_i$, $(\{X_i \mid i \in I\}, T)$ is a tree-decomposition of $G' = (V, E')$ with $E' = E \cup \{(v, w) \mid v, w \in A, v \neq w\}$. For all $b \in B$, $A \cup \{b\}$ forms a clique in G', and hence, by the clique containment lemma, there exists an $i \in I$ with

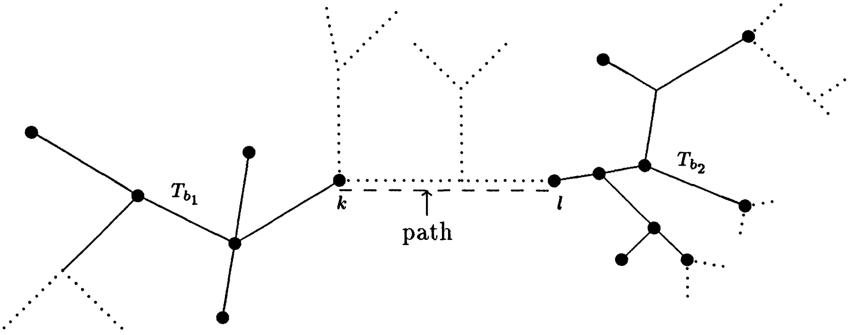


FIG. 2. *An illustration of the proof of Lemma* 3.2.

$A \cup \{b\} \subseteq X_i \Rightarrow$ there exists $i \in I'$, $b \in X_i$. So condition (ii) is fulfilled. Consider an edge $(b, c) \in E$, $b, c \in B$. $A \cup \{b, c\}$ forms a clique in G', and hence, by the clique containment lemma, there exists $i \in I$ with $A \cup \{b, c\} \subseteq X_i$; hence there exists $i \in I'$, $\{b, c\} \subseteq X_i$. It now easily follows that condition (iii) is fulfilled. □

LEMMA 3.4. *Let $G = (V, E)$, $H = (W, F)$ be graphs. Then the following conditions hold*:

(i) *treewidth* $(G \dot{\cup} H) = \max$ (*treewidth* (G), *treewidth* (H));

(ii) *pathwidth* $(G \dot{\cup} H) = \max$ (*pathwidth* (G), *pathwidth* (H));

(iii) *treewidth* $(G \times H) = \min$ (*treewidth* $(G) + |W|$, *treewidth* $(H) + |V|$);

(iv) *pathwidth* $(G \times H) = \min$ (*pathwidth* $(G) + |W|$, *pathwidth* $(H) + |V|$).

Proof. The proofs of conditions (i) and (ii) are trivial. To prove condition (iii), we first show that *treewidth* $(G \times H) \leqq$ *treewidth* $(G) + |W|$. Take a tree-decomposition $(\{X_i | i \in I\}, T = (I, F))$ of G with *treewidth treewidth* (G). Then $(\{X_i \cup W | i \in I\}, T = (I, F))$ is a tree-decomposition of $G \times H$ with *treewidth treewidth* $(G) + |W|$. So *treewidth* $(G \times H) \leqq$ *treewidth* $(G) + |W|$. Similarly, we can show that *treewidth* $(G \times H) \leqq$ *treewidth* $(H) + |V|$.

Next, we show that *treewidth* $(G \times H) \geqq \min$ (*treewidth* $(G) + |W|$, *treewidth* $(H) + |V|$). Consider a tree-decomposition $(\{X_i | i \in T\}, T = (I, F))$ of $G \times H$. From the complete bipartite subgraph containment lemma, it follows that there exists an i with $V \subseteq X_i$, or there exists an i with $W \subseteq X_i$.

Suppose that there exists an i with $V \subseteq X_i$. Let $T = (I', F')$ be a subtree of T such that for all $i \in I'$, $V \subseteq X_i$, $W \subseteq \bigcup_{i \in I'} X_i$ and $(\{X_i | i \in I'\}$. $T' = (I', F'))$ is a tree-decomposition of $G \times H$. T' exists by Lemma 3.3. Note that $(\{X_i \cap W | i \in I'\}, T' = (I', F))$ is a tree-decomposition of H, so there exists an $i \in I'$ with $|X_i \cap W| \geqq$ *treewidth* $(H) + 1 \Rightarrow$ there exists an $i \in I'$ with $|X_i| \geqq |V| +$ *treewidth* $(H) + 1$. So the *treewidth* of the tree-decomposition $(\{X_i | i \in I\}, T = (I, F))$ is at least $|V| +$ *treewidth* (H).

Similarly, if there exists an i with $W \subseteq X_i$, we can show that the *treewidth* of $(\{X_i | i \in I\}, T = (I, F))$ is at least $|W| +$ *treewidth* (G). Hence *treewidth* $(G \times H) \geqq \min (|V| +$ *treewidth* (H), $|W| +$ *treewidth* $(G))$.

The proof of (iv) is similar to that of (iii). □

THEOREM 3.2. *For every cograph $G = (V, E)$, treewidth (G) = pathwidth (G).*

Proof. To prove the theorem, use induction on $|V|$. If G consists of a single vertex, then *treewidth* $(G) = 0 =$ *pathwidth* (G). If $G = G_1 \dot{\cup} G_2$, then *treewidth* $(G) = \max$ (*treewidth* (G_1), *treewidth* (G_2)) = (*i.h.*) max (*pathwidth* (G_1), *pathwidth* (G_2)) = *pathwidth* (G). If $G = G_1 \times G_2$, with $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$, then *treewidth* $(G) = \min$ (*treewidth* $(G_1) + |V_2|$, *treewidth* $(G_2) + |V_1|$) = (*i.h.*) min (*pathwidth* $(G_1) + |V_2|$, *pathwidth* $(G_2) + |V_1|$) = *pathwidth* (G). □

4. Algorithms for pathwidth and related notions on cographs. In this section, we give linear algorithms for determining treewidth, pathwidth, path-decompositions, optimal node search strategies, and interval graph augmentations with minimum clique size of cographs.

In Fig. 3 we give two recursive procedures. COMPUTE-SIZE computes for every vertex in a binary cotree the number of vertices of the corresponding cograph. COMPUTE-PATHWIDTH computes for every vertex in a binary cotree the pathwidth of the cograph corresponding to that vertex. To compute the pathwidth of a cograph G, let r be the root of the binary cotree corresponding to G. Now first call COMPUTE-SIZE(r) and then COMPUTE-PATHWIDTH(r). As per vertex in the cotree, a constant number of operations are performed, this costs $O(n)$ time in total. Correctness follows from Lemma 3.4.

```
procedure COMPUTE-SIZE (v: vertex);
  begin if v is a leaf of $T_G$
    then size (v) := 1
    else begin COMPUTE-SIZE (left son of v)
               COMPUTE-SIZE (right son of v);
               size (v) := size (left son of v) + size (right son of v)
         end
  end

procedure COMPUTE-PATHWIDTH (v: vertex);
  begin if v is a leaf of $T_G$
    then pathwidth(v) := 0
    else begin COMPUTE-PATHWIDTH (left son of v);
               COMPUTE-PATHWIDTH (right son of v);
               if label(v)=0
                 then pathwidth(v) := max(pathwidth (left son of v),
                        pathwidth (right son of v))
                 else pathwidth(v) := min(size (left son of v) +
                        pathwidth (right son of v),
                        pathwidth (left son of v) + size (right son of v))
         end
  end
```

FIG. 3

THEOREM 4.1. *The pathwidth and treewidth of a cograph given with a corresponding cotree can be computed in $O(n)$ time.*

It is easy to construct corresponding path-decompositions in time, linear in the output, i.e., linear in $\Sigma_{i \in I} |X_i|$. However, in some cases, this may be quadratic in n. (Consider a cograph $G = G_1 \times G_2$, where G_1 is a clique with $n/2$ vertices, and G_2 consists of $n/2$ isolated vertices. The optimal tree decomposition of G will consist of $n/2$ sets X_i, each containing each vertex of G_1 and one vertex of G_2.)

Thus we are seeking a more compact representation of path-decompositions. We solve this in the following way: For each $v \in V$, we compute numbers first $(v) = \min \{i \in I | v \in X_i\}$ and last $(v) = \max \{i \in I | v \in X_i\}$. These numbers fix the path-decomposition, because, for all $v \in V$, $i \in I$, $v \in X_i \Leftrightarrow$ first $(v) \leqq i \leqq$ last (v).

Note that this representation corresponds to assigning to each $v \in V$ an interval such that the corresponding interval graph contains G: the chromatic number equals the maximum clique size of this interval graph equals the pathwidth of G plus 1. Thus we also find a representation of G corresponding to its interval thickness.

The numbers first (v) and last (v) for all $v \in V$ are computed in the procedure MAKE-INTERVALS of Fig. 4, which is called with MAKE-INTERVALS $(r, 1, m)$, where r is the root of the binary cotree of G, and where m is an integer variable. In the procedure, "start" always denotes the smallest value that can be used for first (w) with w a leaf in the subtree of the cotree rooted at v, and "finish" will yield the largest value used for last (w), with w again leaf in the subtree rooted at v. Correctness of the procedure easily follows. Clearly, the procedure is linear in the size of the cotree $= O(n)$.

THEOREM 4.2. *A representation of a path decomposition with optimal pathwidth of a cograph, given with a corresponding cotree, can be computed in $O(n)$ time.*

THEOREM 4.3. *The pathwidth and treewidth of cographs and corresponding path-decompositions or tree-decompositions can be computed in $O(n+e)$ time.*

Proof. Recall that the cotree of a cograph can be found in $O(n+e)$ time (see § 2). We now use the fact that optimal path-decompositions of cotrees fulfill $\Sigma_{i \in I} |X_i| = O(n+e)$. □

```
procedure MAKE-INTERVALS (v: vertex, start: integer, finish: var integer);
  var help: integer:
  begin if v is a leaf of $T_G$
    then begin first(v) := start; last(v) := start; finish := start end
    else if label(v) = 0
      then begin MAKE-INTERVALS (left son of v, start, help);
                 MAKE-INTERVALS (right son of v, help + 1, finish)
        end
      else (*label(v) = 1)
      if size (left son of v) + pathwidth(right son of v) > size (right son of v) + pathwidth(left son of v)
        then begin MAKE-INTERVALS (left son of v, start, finish);
                   for each w ∈ V that is a leaf descendant of the right son of v
                   do begin first(w) := start; last(w) := finish end
             end
        else begin MAKE-INTERVALS (right son of v, start, finish)
                   for each w ∈ V that is a leaf descendant of the left son of v
                   do begin first(w) := start; last(w) := finish end
             end
  end
```

FIG. 4

THEOREM 4.4. *There exists an algorithm that, given a cograph G and a corresponding cotree of G, determines in $O(n)$ time an interval graph H that contains G as a subgraph and has chromatic number equal to the interval thickness of G.*

THEOREM 4.5. *There exists an algorithm that, given a cograph G and a corresponding cotree of G, determines the node search number of G and corresponding search strategy in $O(n)$* time.

Proof. Compute first (v) and last (v) for all $v \in V$, as described above. Now use the following search strategy:

```
put a searcher on each vertex $v$ with first$(v) = 1$
for $i := 1$ to max$\{$last$(v) \mid v \in V\} - 1$
do begin for all $v \in V$ with last$(v) = i$: remove searcher from $v$;
         for all $v \in V$ with first$(v) = i + 1$: put a searcher on $v$
  end
```

With this search strategy, all edges will be cleared, no recontamination can take place, and the optimal number of searchers (*pathwidth* $(G) + 1$) is used. Determining the sets $\{v \mid \text{first } (v) = i\}$, and $\{v \mid \text{last } (v) = i\}$ can be done with bucket sort in $O(n)$ time in total. □

5. Final remarks. In this paper, we gave a linear time algorithm to determine the treewidth and pathwidth of cographs. Currently, we are investigating how to extend the results of this paper to larger classes of graphs, e.g., graphs that are built with modular composition with small neighborhood modules (see [16]). Another interesting problem is whether these results can be extended to distance-hereditary graphs.

REFERENCES

[1] S. ARNBORG, *Efficient algorithms for combinatorial problems on graphs with bounded decomposability—A survey*, BIT, 25 (1985), pp. 2–23.

[2] S. ARNBORG, D. G. CORNEIL, AND A. PROSKUROWSKI, *Complexity of finding embeddings in a k-tree*, SIAM J. Alg. Disc. Meth., 8 (1987), pp. 277–284.

[3] S. ARNBORG AND A. PROSKUROWSKI, *Linear time algorithms for* NP-*hard problems restricted to partial k-trees*, Discrete Appl. Math., 23 (1989), pp. 11–24.

[4] H. L. BODLAENDER, *Classes of graphs with bounded treewidth*, Tech. Report RUU-CS-86-22, Dept. of Computer Science, Univ. of Utrecht, Utrecht, the Netherlands, 1986.

[5] ———, *Dynamic programming algorithms on graphs with bounded treewidth*, in Proc. 15th Internat. Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science, Vol. 317, Springer-Verlag, Berlin, New York, 1988, pp. 105–119.

[6] H. L. BODLAENDER AND T. KLOKS, *Better algorithms for the pathwidth and treewidth of graphs*, in Proc. 18th Internat. Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science, Vol. 510, Springer-Verlag, Berlin, New York, 1991, pp. 544–555.

[7] D. G. CORNEIL, H. LERCHS, AND L. STEWART BURLINGHAM, *Complement reducible graphs*, Discrete Appl. Math., 3 (1981), pp. 163–174.

[8] D. G. CORNEIL, Y. PERL, AND L. STEWART, *A linear recognition algorithm for cographs*, SIAM J. Comput., 4 (1985), pp. 926–934.

[9] M. R. FELLOWS AND M. A. LANGSTON, *Layout permutation problems and well-partial ordered sets*, in Proc. Fifth M.I.T. Conference on Advanced Research in VLSI, published as Advanced Research in VLSI, J. Allen and F. T. Leighton, eds., MIT Press, Cambridge, MA, 1988, pp. 315–327.

[10] ———, *Nonconstructive tools for proving polynomial-time decidability*, J. Assoc. Comput. Mach., 35 (1988), pp. 727–739.

[11] M. C. GOLUMBIC, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[12] J. GUSTEDT, *Path width for chordal graphs is* NP-*complete*, Tech. Report 221/1989, Technical Univ. Berlin, Berlin, Germany, 1989; Discrete Appl. Math., to appear.

[13] L. M. KIROUSIS AND C. H. PAPADIMITRIOU, *Interval graphs and searching*, Discrete Math., 55 (1985), pp. 181–184.

[14] ———, *Searching and pebbling*, Theoret. Comput. Sci., 47 (1986), pp. 205–218.

[15] R. H. MÖHRING, *Graph problems related to gate matrix layout and* PLA *folding*, in Computational Graph Theory, G. Tinhofer et al., eds., Computing Supplementum 7, Springer-Verlag, Wien, 1990, pp. 17–52.

[16] J. H. MULLER AND J. SPINRAD, *Incremental modular decomposition*, J. Assoc. Comput. Mach., 36 (1989), pp. 1–19.

[17] B. REED, *Finding approximate separators and computing treewidth quickly*, in Proc. 24th Annual Sympos. on Theory of Computing (STOC'92), 1992, pp. 221–228.

[18] N. ROBERTSON AND P. D. SEYMOUR, *Graph minors* I. *Excluding a forest*, J. Combin. Theory Ser. B, 35 (1983), pp. 39–61.

[19] ———, *Graph minors* II. *Algorithmic aspects of treewidth*, J. Algorithms, 7 (1986), pp. 309–322.

[20] ———, *Graph minors—A survey*, in Surveys in Combinatorics, I. Anderson, ed., Cambridge Univ. Press, Cambridge, UK, 1985, pp. 153–171.

[21] P. SCHEFFLER, *Die Baumweite von Graphen als ein Maß für die Kompliziertheit algorithmischer Probleme*, Ph.D. thesis, Akademie der Wissenschaften der DDR, Berlin, 1989.

[22] R. SUNDARAM, K. S. SINGH, AND C. P. RANGAN, *Treewidth of circular-arc graphs*, manuscript, 1991.

SIAM J. Disc. Math.
Vol. 6, No. 2, pp. 189-196, May 1993

HAMILTONIAN PROPERTIES OF BIPARTITE GRAPHS AND DIGRAPHS WITH BIPARTITE INDEPENDENCE 2*

ODILE FAVARON†, PEDRO MAGO‡, CONSUELO MAULINO§, and OSCAR ORDAZ¶

Abstract. This paper studies the bipartite graphs G in which $\alpha_{\mathrm{BIP}}(G)$, the maximum order of an induced balanced bipartite subgraph without edges, is equal to 2. When its order is at least 10, it is shown that G contains a Hamiltonian path, provided that it is connected, and that, if its minimum degree is at least 2, then it is bipancyclic.

Similar results concerning the bipartite digraphs D in which $\alpha^2_{\mathrm{BIP}}(D)$ are given, and the maximum order of an induced balanced bipartite subdigraph without 2-cycles, is equal to 2.

Key words. bipartite graph, independent set, Hamiltonian

AMS(MOS) subject classifications. 68R10, 68Q05, 68E10

1. Introduction. The graphs (respectively, digraphs) considered here are without loops or multiple edges (respectively, arcs). We denote a bipartite graph by $G = (A, B, E)$, where A and B are the bipartition sets, E is the edge set, and where there is a bipartite digraph by $D = (A, B, E \cup F)$, where F is the set of the antisymmetric arcs of D, and E the set of the symmetric arcs (i.e., 2-cycles), which are also called the *edges* of D. Given this notation, the graph G is said to be associated to the digraph D. If H is a subgraph of a bipartite graph (A, B, E), H_A (respectively, H_B) is the set of vertices of H in A (respectively, in B). An edge (respectively, arc) between a and b (respectively, from a to b) is written ab (respectively, (a, b)). In a digraph, the words, cycles, and paths are used in their directed sense.

An obvious necessary condition for a bipartite graph to be Hamiltonian (respectively, to have a Hamiltonian path) is to be *balanced*, that is, $|A| = |B|$ (respectively, to be balanced or *almost balanced*; that is, $||A| - |B|| = 1$). A balanced graph or digraph is *bipancyclic* if it contains cycles of all even lengths. The minimum degree of a graph (respectively, minimum indegree or outdegree of a digraph) is denoted by δ (respectively, δ^- and δ^+). We denote by $d(x)$ the degree of x in G. For $S \subseteq A \cup B$, we define $N(S) = \{y\colon zy \text{ edge in } G \text{ and } z \in S\}$ and $N^+(S) = \{y\colon (z, y) \text{ arc in } D \text{ and } z \in S\}$.

The notion of an independent set (set of pairwise nonadjacent vertices) of a graph has been generalized to the directed case in three ways, two of which are used as follows: A set S of vertices of a digraph D is said to be *α^0-independent* (respectively, *α^2-independent*) if $D[S]$, the subdigraph induced by S, contains no arcs (respectively, no edges). The α^2-independent sets of D are thus the independent sets of the graph G associated to D. The

* Received by the editors January 22, 1991; accepted for publication (in revised form) September 20, 1991.

† L.R.I., U.A. 410, Centre National de la Recherche Scientifique, Bat. 490, Université de Paris-Sud, 91405 Orsay Cedex, France. This work was done in part while this author was visiting the Mathematics Department, Faculty of Science, Universidad Central de Venezuela, Caracas, Venezuela. This author's research was partially supported by Programme de Recherche Coordonne Mathematics Information.

‡ Mathematics Department, Faculty of Science, Universidad de Oriente, Cumaná, Venezuela. Thus author's research was partially supported by the French Postgraduate Cooperation Program in Informatics between Venezuela and France.

§ School of Computer Science, Faculty of Science, Universidad Central de Venezuela, Apartado 47002, Caracas 1041-A, Venezuela.

¶ Mathematics Department, Faculty of Science, Universidad Central de Venezuela, Apartado 47567, Caracas 1041-A, Venezuela. This work was begun while this author was visiting the L.R.I., Université de Paris-Sud, Paris, France.

maximum cardinality of an α^i-independent set is the α^i-independence number and $\alpha^0 \leqq \alpha^2$.

In a bipartite graph, the *bipartite independence number* α_{BIP} is the maximum cardinality of a balanced independent set (note that α_{BIP} is always even). In [1] and [7], α_{BIP} was equal to half this number, but following [8], we find the second definition more convenient. In a bipartite digraph, α^0_{BIP} and α^2_{BIP} are defined in the same manner, so that $\alpha^0_{\mathrm{BIP}} \leqq \alpha^2_{\mathrm{BIP}}$ and $\alpha^2_{\mathrm{BIP}}(D) = \alpha_{\mathrm{BIP}}(G)$, where G is the graph associated to D. It is clear that $\alpha_{\mathrm{BIP}}(G) = 0$ if and only if G is a complete bipartite and, in this case, G is bipancyclic. In the proofs of the theorems of §§ 2 and 3, we will thus only consider the case where $\alpha_{\mathrm{BIP}}(G) = 2$ or where $\alpha^2_{\mathrm{BIP}}(D) = 2$.

Few results have been obtained that extend to digraphs the Hamiltonian properties of undirected graphs involving the independence number α and the connectivity k, the most famous of them being that $\alpha \leqq k$ implies Hamiltonicity (see the theorem of Chvâtal and Erdös). These results concern essentially digraphs of a small independence number (see the survey of Jackson and Ordaz [8]). According to the type of independence number, α^0 or α^2, that is considered, we will refer to the two following theorems.

THEOREM 1.1 (Chen and Manalastas [4]). *Every strongly connected digraph with $\alpha^0 \leqq 2$ has a Hamiltonian path.*

THEOREM 1.2 (Chakroun and Sotteau [3]). *For $k = 2$ or $k = 3$, every k-connected digraph with $\alpha^2 \leqq k$ is pancyclic, with a few specified exceptions.*

On the other hand, general conditions of this kind are not interesting for bipartite graphs since the independence number is always at least equal to the connectivity. For these graphs, a specific independence parameter α_{BIP} was introduced by Ash [1], and Fraisse [7] proved the following important theorem (the statement of which is given here with the new definition of α_{BIP}).

THEOREM 1.3 (Fraisse [7]). *Every 2-connected balanced bipartite graph with minimum degree δ and $\alpha_{\mathrm{BIP}} \leqq \delta - 1$ is Hamiltonian.*

This is, at first, not a Chvâtal–Erdös condition. It is known [6], however, that the hypothesis $\alpha_{\mathrm{BIP}} \leqq \delta - 1$ of Theorem 1.3 is actually equivalent to $\alpha_{\mathrm{BIP}} \leqq k - 1$ and that the condition of 2-connectedness may be removed.

Finally, Bondy's metaconjecture incited us to study the bipancyclicity. For this, we use Theorem 1.4, below, which is a bipartite version of Bondy's classical result and which allows us, when the existence of a Hamiltonian cycle is already known, to weaken the required condition on the number of edges to guarantee the existence of cycles of all even lengths (see also the short and clear survey [9]).

THEOREM 1.4 (Entringer and Schmeichel [5]). *Let G be a Hamiltonian bipartite graph on $2n \geqq 8$ vertices. If the number m of edges of G is greater than $n^2/2$, then G is bipancyclic.*

In § 2 we prove that every balanced bipartite graph of order at least 10, with minimum degree δ and bipartite independence number equal to 2, has a Hamiltonian path if $\delta \geqq 1$ and is bipancyclic if $\delta \geqq 2$. In § 3 we show that, by replacing δ by min $\{\delta^+, \delta^-\}$, the same results hold for the digraphs D satisfying $\alpha^2_{\mathrm{BIP}}(D) = 2$.

2. Bipartite graphs with $\alpha_{\mathrm{BIP}} \leqq 2$.

When $\alpha_{\mathrm{BIP}} = 2$, Theorem 1.3 establishes the Hamiltonicity of G if $\delta \geqq 3$ but is not sufficient for $\delta = 2$. So we will prove it directly, following the same idea as in Fraisse's demonstration and using Veldman's notation and Theorems A, B, and C.

Two disjoint induced subgraphs H_1 and H_2 of G are *remote* if no edge exists between H_1 and H_2. The *α_λ-independence number* $\alpha_\lambda(G)$ is the maximum number of mutually remote connected subgraphs of order λ of G. The *degree $d(H)$ of a subgraph H* of G is

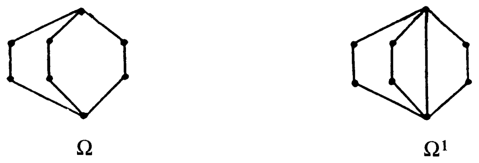


FIG. 1

the number of vertices in $V(G) - V(H)$ adjacent to at least one vortex of H. A cycle C of a graph G is a D_λ-*cycle* if all the connected components of $G - V(C)$ have order less than λ. Thus a D_1-cycle is a Hamiltonian cycle and a D_2-cycle, also called a *dominating cycle* by some authors is such that $G - V(C)$ is empty or consists of isolated vertices.

THEOREM A (Veldman [10]). *Let k and λ be positive integers such that $k \geqq 2$. If G is a k-connected graph with $\alpha_\lambda \leqq k$, then G admits a D_λ*-cycle.

THEOREM B (Veldman [10]). *Let G be a 2-connected graph of order p such that the degree-sum of every three mutually remote connected subgraphs of order $\lambda \geqq 2$ is at least $p - 3\lambda + 5$. Then G admits a D-cycle.*

THEOREM C (see [6]). *A balanced bipartite graph of order at least 4 with $\alpha_{\mathrm{BIP}} \leqq \delta$ is 2-connected.*

THEOREM 2.1. *Let G be a balanced bipartite graph of order $2n \geqq 4$, with minimum degree $\delta \geqq 2$ and $\alpha_{\mathrm{BIP}}(G) \leqq 2$. If G is not isomorphic to one of the graphs Ω and Ω^1* (Fig. 1), *then G is Hamiltonian.*

Proof. The result is clearly true for graphs of order 4 or 6, so we suppose that $2n \geqq 8$.

If G does not contain 3 remote edges and k is the connectivity of G, then $\alpha_2 \leqq 2 \leqq k$ by Theorem C, and, by Theorem A, G has a D_2-cycle.

If G contains three remote edges $e_w = a_wb_w$, $w = 1, 2, 3$, then, since $\alpha_{\mathrm{BIP}}(G) \leqq 2$, each vertex of A-$\{a_1, a_2, a_3\}$ (respectively, of B-$\{b_1, b_2, b_3\}$) has at least two neighbours in $\{b_1, b_2, b_3\}$ (respectively, $\{a_1, a_2, a_3\}$). Therefore $\sum_{w=1}^{3} d(e_w) \geqq 4(n - 3)$ and for $2n \geqq 12$, G has a $D2$-cycle by Theorem B.

Case A. G contains no D_2-cycle.

From above, $2n = 8$ or 10, and G contains 3 remote edges $e_w = a_wb_w$, $w = 1, 2, 3$. If $2n = 8$, the graph G is isomorphic to Ω or Ω^1. If $2n = 10$, let a_4, a_5 in A and b_4, b_5 in B be the four other vertices of G. Since $\delta \geqq 2$, G contains a spanning subgraph isomorphic to H_1 (Fig. 2) or H_2 (Fig. 3, where H_2 is drawn in two different ways).

Since G admits no D_2-cycle, it does not contain H_1, which is Hamiltonian, and thus contain H_2. Since the graph G is not Hamiltonian, a_4b_4, a_4b_5, and a_5b_4 are not in E, and, since $\alpha_{\mathrm{BIP}}(G) \leqq 2$, a_5b_5 is in E. Similarly, a_5b_2 and a_5b_4 are not in E, since G is not Hamiltonian. a_3b_2 is also not in E, since e_2 and e_3 are remote. Because the set $\{a_3, a_5, b_2, b_4\}$ is not independent, a_3b_4 is in E. Then, however, the cycle $a_3b_4a_1b_1a_4b_2a_2b_5a_5b_3a_3$

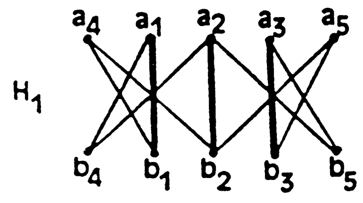


FIG. 2

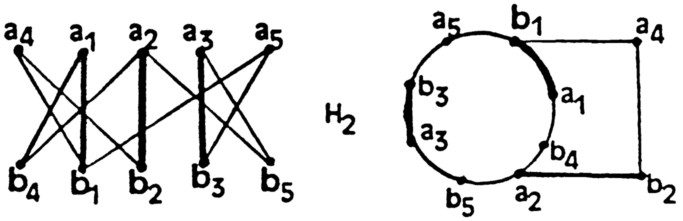


FIG. 3

is Hamiltonian, a contradiction. Therefore G contains a D_2-cycle, unless it is isomorphic to Ω or Ω^1.

Case B. G contains a D_2-cycle.

Suppose that G is a non-Hamiltonian graph and let C be a maximum D_2-cycle. Let a in A and b in B be two nonadjacent vertices not on C. We will show that in all cases there is a D_2-cycle containing $C \cup \{a, b\}$. This is a contradiction. Since $\delta \geqq 2$, each vertex a and b has at least two neighbours on C. We discuss according to the relative position of these neighbours. An arbitrary orientation of C being chosen, we use classical notation. If x is a vertex of C, x^+ (respectively, x^-) is the successor (respectively, predecessor) of x on C. If x and y are vertices of C, then xC^+y (respectively, xC^-y) is the path of C joining x to y following the directed (respectively, inverse) orientation.

At each step of the proof (labelled as subcases, below), we suppose that the conditions of the previous cases, and of the analogous ones, obtained by changing a into b or the orientation of C, are not satisfied.

Subcase B1. The vertex a has two neighbours b_1 and b_2 on C such that b is adjacent to b_1^+ and b_2^+. The cycle $ab_2C^-b_1^+bb_2^+C^+b_1a$ is a D_2-cycle.

Subcase B2. The vertex a has two neighbours b_1 and b_2 on C such that b is adjacent to b_1^+ and to another vertex a_1 on $b_1C^+b_2$. Since the conditions of Subcase B1 are not satisfied, aa_1^- and $bb_2^+ \notin E$. The balanced set $\{a, b_2^+, b, a_1^-\}$ is not independent. Thus $a_1^-b_2^+ \in E$ and $ab_2C^-a_1bb_1^+C^+a_1^-b_2^+C^+b_1a$ is a D_2-cycle.

Subcase B3. The vertex a has two neighbours b_1 and b_2 on C such that b is adjacent to b_1^+ and to another vertex a_1 on $b_2C^+b_1$.

Since the conditions of Subcase B2 are not satisfied, aa_1^+ and $bb_2^- \notin E$. The balanced set $\{a, b_2^-, b, a_1^+\}$ is not independent. Thus $a_1^+b_2^- \in E$ and $ab_2C^+a_1bb_1^+C^+b_2^-a_1^+C^+b_1a$ is a D_2-cycle.

Subcase B4. The neighbours on C of a and b are never consecutive. Let b_1 and b_2 (respectively, a_1 and a_2) be two neighbours of a (respectively, b) on C. We may suppose that $a_1 \in b_1C^+b_2$. Since aa_w^+ and bb_w^+ are not in E and the balanced set $\{a, b_w^+, b, a_w^+\}$ is not independent, $a_w^+b_w^+$ is in E for $w = 1, 2$. If $a_2 \in b_2C^+b_1$, then $ab_2C^-a_1^+b_1^+C^+a_1ba_2C^-b_2^+a_2^+C^+b_1a$ is a D_2-cycle. If $a_2 \in b_1C^+b_2$ (in this case, we can choose a_1 and a_2 such that $a_2 \in b_1C^+a_1$), then $ab_2C^-a_1^+b_1^+C^+a_2ba_1C^-a_2^+b_2^+C^+b_1a$ is a D_2-cycle.

Since Subcases B1–B4 lead to a contradiction, the graph G is Hamiltonian. □

THEOREM 2.2. *Any balanced bipartite graph G of order $2n \geqq 8$ with $\delta \geqq 2$ and $\alpha_{\text{BIP}} \leqq 2$ is bipancyclic, except if G is isomorphic to Ω, Ω^1, or C_g.*

Proof. It is sufficient, by Theorems 2.1 and 1.4, to check that m number of arcs of G verify $m > n^2/2$.

Let $a_1b_1a_2b_2 \cdots a_nb_n$ be a Hamiltonian cycle of G with a_w in A and b_w in B and let d_w be the degree of the vertex a_w for $w = 1, 2, \cdots, n$. Since $\alpha_{\text{BIP}}(G) = 2$, we get $d_w + d_{w+1} \geqq n$ for any w (with $a_{n+1} = a_1$).

By adding these n inequalities, we obtain $2m \geqq n^2$. Suppose that $m = n^2/2$. This implies that n is even, $d_1 = d_3 = \cdots = d_{n-1} = p$, and $d_2 = d_4 = \cdots = d_n = n - p$. Again, by $\alpha_{\mathrm{BIP}}(G) = 2$, we have $d_1 + d_3 \geqq n - 1$ and $d_2 + d_4 \geqq n - 1$. Therefore, by parity, $p = n/2$. Among the n subsets $B_w = N(a_w)$ of order $n/2$ of the set B of even order $n \geqq 4$, at least two, say B_w and B_j, have two or more common vertices. Indeed if $|B_1 \cap B_2| < 2$ and $|B_1 \cap B_3| < 2$, then $|B_2 \cap B_3| \geqq n/2 - 2$. Thus $|B_2 \cap B_3| \geqq 2$ for $n \geqq 8$. For $n = 6$, the property is easy to verify directly. For $n = 4$, the only case where the four subsets B_w have pairwise at most one common vertex corresponds to C_8, which is excluded. Therefore $n - |B_w \cup B_j| \geqq 2$ and $\alpha_{\mathrm{BIP}} \geqq 4$, a contradiction. Hence, $m > n^2/2$, and G is bipancyclic. □

Note that the cycle C_8 is an example of a Hamiltonian but not bipancyclic balanced bipartite graph of order 8 with $\alpha_{\mathrm{BIP}} = \delta = 2$.

In the following corollary, we denote by Ω_1 (respectively, Ω_1^1) the graph obtained from Ω (respectively, Ω^1) by deleting an edge whose endvertices have degree 2.

COROLLARY 2.3. *Let G be a connected balanced bipartite graph of order $2n \geqq 8$ with $\alpha_{\mathrm{BIP}} \leqq 2$. If G is not isomorphic to Ω_1 or Ω_1^1, then G contains a Hamiltonian path.*

Proof. If $\delta \geqq 2$, then G is Hamiltonian or isomorphic to Ω or Ω^1, and the result is obvious. Henceforth, assume that $\delta = 1$. Since $\alpha_{\mathrm{BIP}} = 2$ and $n \geqq 4$, there cannot exist two vertices of degree 1 in either A or B. Therefore there exists at most one vertex of degree 1 in A and in B, and these two vertices are not adjacent because of the connectivity of G. Suppose that $d(a) = 1$ for a vertex a in A. Let b be the vertex of B of degree 1 if such a vertex exists; if not, let b be a vertex of B not adjacent to a. The graph $G^1 = G + ab$, which satisfies the hypothesis of Theorem 3.1, is Hamiltonian or isomorphic to Ω or Ω^1, and thus G, which is not isomorphic to Ω^1 or Ω_1^1, has a Hamiltonian path. □

Note that the condition $n \geqq 4$ is necessary, as shown by the graph consisting of a path P_5 with a pendant edge incident to the third vertex.

COROLLARY 2.4. *Let $G = (A, B, E)$ be a bipartite graph with $|A| = |B| + 1 > 4$, $\alpha_{\mathrm{BIP}}(G) \leqq 2$, and $d(y) \geqq 2$ for every vertex y of B. Then G contains a Hamiltonian path.*

Proof. The graph G^1 obtained from G by adding a new vertex v to B and joining it to every vertex of A is Hamiltonian by Theorem 2.1. Thus $G = G^1 - \{v\}$ contains a Hamiltonian path. □

Note that the condition $d(y) \geqq 2$ for every vertex y in B is necessary, as shown by the graphs obtained by adding a pendent edge to a vertex of the $(r + 2)$ stable set of a $K_{r,r+2}$. Moreover, if $|A| = 4$, the graph obtained from Ω or Ω^1 by deleting a vertex of degree 4 has no Hamiltonian path.

We end this section by the study of the Hamiltonian connectedness of G. Let us recall that the *p-biclosure* of a bipartite-graph G is the graph G_p obtained from G by recursively joining pairs of nonadjacent vertices in A and B whose degree sum is at least p until no such pair remains. Also, if the $(n + 2)$-biclosure G_{n+2} of a balanced bipartite graph G of order $2n$ is a complete bipartite, then G is *Hamiltonian biconnected*; that is, G admits a Hamiltonian path between every pair of vertices in A and B (cf., for instance, [2]).

THEOREM 2.5. *Any balanced bipartite graph of order $2n$ and minimum degree δ that satisfies $\alpha_{\mathrm{BIP}}(G) = 2$ and $\delta \geqq 3$ is Hamiltonian biconnected.*

Proof. We will show that the $(n + 2)$-biclosure G_{n+2} of G is complete bipartite. Let a be a vertex of degree δ in A. Since $\alpha_{\mathrm{BIP}}(G) = 2$, every vertex x in $A - \{a\}$ satisfies $d(x) \geqq n - \delta - 1$.

If $d(y) \geqq \delta + 3$ for every vertex y of B, then $d(x) + d(y) \geqq n + 2$ for all x in $A - \{a\}$ and y in B. After adding all the missing edges between $A - \{a\}$ and B, the sums $d(a) + d(y)$ become at least $n - 1 + \delta \geqq n + 2$ for all y in B. So G_{n+2} is a complete bipartite. If there exists a vertex b in B with $d(b) \leqq \delta + 2$, then, since $\alpha_{\mathrm{BIP}}(G) = 2$, every vertex y of $B - \{b\}$ satisfies $d(y) \geqq n - \delta - 3$. Thus $d(x) + d(y) \geqq 2n + 2\delta - 4 \geqq n + 2$ for all x in $A - \{a\}$ and y in $B - \{b\}$; we add all the missing edges between $A - \{a\}$ and $B - \{b\}$. So G_{n+2} is a complete bipartite. □

3. Bipartite digraphs with $\alpha^2_{\mathrm{BIP}} \leqq 2$. In the case of a balanced bipartite digraph, the proofs of the existence of a Hamiltonian path and of the bipancyclicity use the same method based on the study of the structure of the associated graph G. These two proofs will thus be given together. First, however, we describe this structure when G is not 2-connected.

The following lemmas are easy to check.

LEMMA 3.1. *Any nonconnected balanced (respectively, almost balanced) bipartite graph of order at least* 8 (*respectively*, 7) *with* $\alpha_{\mathrm{BIP}}(G) = 2$ *has necessarily one of the following three structures*:

(1) *G consists of a connected balanced (respectively, almost balanced) bipartite graph H with* $\alpha_{\mathrm{BIP}}(H) = 0$ *or* 2 *and a graph* $K_{1,1}$ *with vertices a in A and b in B*;

(2) *G consists of a connected almost balanced (respectively, balanced) bipartite graph H with* $\alpha_{\mathrm{BIP}}(H) = 0$ *or* 2 *and a trivial graph* $\{b\}$ *with b in B*;

(3) *G consists of a complete balanced (respectively, almost balanced) bipartite graphs H and two isolated vertices a* in A and b in B.

Furthermore, in the above cases (1) *and* (2), *we have*

$$d_H(y) \geqq |H_A| - 1 \quad \textit{for any } y \textit{ in } H_B, \tag{3.1}$$

and, in case (1),

$$d_H(y) \geqq |H_B| - 1 \quad \textit{for any } y \textit{ in } H_A. \tag{3.2}$$

If the connectivity of the balanced bipartite graph G is equal to 1, *let x be a cutvertex, the graph* $G - \{x\}$ *is a nonconnected, almost balanced bipartite with* $\alpha_{\mathrm{BIP}}(G - \{x\}) = 2$ *and is described in the previous lemma.*

Noting that the third case of Lemma 3.1 cannot correspond to a connected graph G, we get the second structure lemma.

LEMMA 3.2. *Any balanced bipartite graph G of order at least* 8, $\alpha_{\mathrm{BIP}} = 2$, *connectivity* 1 *with cutvertex x has one of the following two structures*:

(1) *G consists of a connected almost balanced bipartite graph H with* $|H_A| = |H_B| + 1$ *and* $\alpha_{\mathrm{BIP}}(H) = 0$ *or* 2, *an edge ab with a in A and b in B, with a vertex x in B that is adjacent to a and to some vertices in* H_A, *or*

(2) *G consists of a connected balanced bipartite graph H with* $\alpha_{\mathrm{BIP}}(H) = 0$ *or* 2, *a vertex b in B and a vertex x in A that is adjacent to b and to some vertices in* H_B.

Furthermore, we have

$$d_H(y) \geqq |H_A| - 1 \quad \textit{for any } y \textit{ in } H_B, \tag{3.3}$$

and, in case (1),

$$d_H(y) \geqq |H_B| - 1 \quad \textit{for any } y \textit{ in } H_A. \tag{3.4}$$

As a consequence of these two lemmas, we find again that any balanced bipartite graph G with $\alpha_{\mathrm{BIP}}(G) \leqq 2 \leqq \delta$ is 2-connected (Theorem C).

THEOREM 3.3. *Any balanced bipartite digraph D of order* $2n \geqq 10$ *with* $\delta^+ \geqq 2$, $\delta^- \geqq 2$ *and* $\alpha^2_{\mathrm{BIP}} \leqq 2$ *is bipancyclic.*

THEOREM 3.4. *Any connected balanced bipartite digraph D of order* $2n \geqq 10$ *with* $\delta^+ \geqq 1$, $\delta^- \geqq 1$ *and* $\alpha^2_{\mathrm{BIP}}(D) \leqq 2$ *contains a Hamiltonian path.*

Proof of Theorems 3.3. *and* 3.4. The graph $G = (A, B, E)$ of order $2n \geqq 10$ associated to $D = (A, B, E \cup F)$, as said in the Introduction, satisfies $\alpha_{\mathrm{BIP}}(G) = 2$. When it is not 2-connected, its structure is given by Lemmas 3.1 and 3.2. Below, we refer to the notation of these lemmas:

(i) $k(G) \geqq 2$. In this case, by Theorem 2.2, G, and thus D, is bipancyclic.

(ii) $k(G) = 1$. The graph G satisfies the hypothesis of Corollary 2.3. Therefore D contains, as G, a Hamiltonian path.

For the bipancyclicity (with the additional hypothesis $\delta^+ \geqq 2$, $\delta^- \geqq 2$), we discuss the two possibilities given for G in Lemma 3.2.

Case 1. Since $d^+(b) \geqq 2$ and $d^-(b) \geqq 2$, there exists a vertex a^2 in H_A such that (a^2, b) or (b, a^2) (not both) belongs to F, say (a^2, b) is in F. The graph $G^1 = G + a^2b$ that satisfies $\alpha_{\mathrm{BIP}}(G^1) = 2$ and $\delta(G^1) = 2$ by (3.3) and (3.4) is bipancyclic by Theorem 2.2. Hence, replacing the edge a^2b by the arc (a^2, b), every cycle of G^1 yields a cycle in D that is also bipancyclic.

Case 2(a). If the degree of every vertex of H_A is at least 2, the graph $G^1 = G + ba^1$, where (b, a^1) is an arc of D, satisfies $\alpha_{\mathrm{BIP}}(G^1) = 2$ and $\delta(G^1) = 2$ and is bipancylic. Replacing the edge ba^1 by the arc (b, a^1), we thus see that D is also bipancyclic.

Case 2(b). Suppose now that a vertex a of H_A has degree 1 in H and let b^1 be its neighbour in H_B. By (2.3), such a vertex is unique, the graph $H^1 = H - \{a, b^1\}$ is complete bipartite, and b^1 is adjacent to every vertex of $H_A - \{a\}$, except perhaps a vertex a_1. Furthermore, because of $\alpha_{\mathrm{BIP}} = 2$, x is adjacent to every vertex of $H_B - \{b^1\}$, except perhaps a vertex b_1.

If one of the two arcs (a, b) or (b, a) belongs to F, the graph $G^1 = G + ab$ is bipancyclic of Theorem 2.2. D is also bipancyclic. If neither (a, b) nor (b, a) belong to F, there exists at least one arc (a, v) or (v, a), say, for example, (a, v), with v in $H_B - \{b_1, b^1\}$. By Theorem 2.2, the graph $H + av$ is bipancyclic and D contains cycles of all even lengths at most $2n - 2$. It remains to show that D is Hamiltonian. Because of the degree of b, there exists at least one arc (u, b) with u in $H_A - \{a\}$, and, since $n \geqq 5$, x has a neighbour b_2 in $H_B^1 - \{v, b_1\}$ and b^1 has a neighbour a_2 in $H_A^1 - \{u, a_1\}$. The complete balanced bipartite graph $H^1 - \{u, v\}$ admits a Hamiltonian path with end-vertices b_2 and a_2 that forms with the path $a_2b^1avubxb_2$, a Hamiltonian cycle of D.

(iii) $k(G) = 0$.

We discuss the following three possibilities as given in Lemma 3.1 for balanced bipartite graphs.

Case 1. By (3.1), (3.2), and Theorem 2.2, the graph H is bipancyclic. Since D is connected, there exists at least one arc, say (b^1, a), between H and $\{a, b\}$, that yields a Hamiltonian path in D. Furthermore, if $\delta^+ \geqq 2$ and $\delta^- \geqq 2$, there also exists an arc (b, a^1). The graph $G^1 = G + ba^1 + ab^1$ satisfies $\alpha_{\mathrm{BIP}}(G^1) = 2$ and $\delta(G^1) = 2$. It admits a Hamiltonian cycle that produces a Hamiltonian cycle in D.

Case 2(a). If all the vertices of H_A have degree at least 2, let (b, a^1) and (a^2, b) be arcs of D. The graph $G^1 = G + a^1b + a^2b$ that satisfies $\alpha_{\mathrm{BIP}}(G^1) = 2$ and $\delta(G^1) = 2$ is bipancyclic by Theorem 2.2. Therefore D is also bipancyclic.

Case 2(b). Suppose now that a vertex a of H_A has degree 1 in H and let b^1 be its neighbour in H_B. By (3.1), such a vertex is unique, and the balanced bipartite graph $H^1 = H - \{a\}$ is complete, except perhaps for one missing edge b^1a_1. Therefore H^1 is bipancyclic, and D admits cycles of all even lengths at most $2n - 2$. Moreover, given an arbitrary vertex a^1 in $H_A - \{a\}$, H^1 admits a Hamiltonian path P whose endpoints are b^1 and a^1.

If there exists an arc between a and b, say, for example, (a, b), let (b, a^1) with a^1 in H_A^1 be an arc of D. The graph $G^1 = G + ab + ba^1$ is bipancyclic by Theorem 2.2. D is also bipancyclic.

If neither (a, b) nor (b, a) belong to F, let (b, a^1) and (a^2, b) with a^1 and a^2 in H^1 be two arcs of D. The Hamiltonian path P of endpoints b^1 and a^1 in H^1 joined with the arc (b, a^1) and the edge b^1a forms a Hamiltonian path in D. When, furthermore, $\delta^+ \geqq 2$ and $\delta^- \geqq 2$, one may choose a^1 different from a_1 so that b^1a^1 belongs to E, and there exists (a, b^2) with b^2 in $H_B - \{b^1\}$. The complete balanced bipartite graph $G^2 = G + \{b^1, a^1, a, b\}$ admits a Hamiltonian path between b^2 and a^2 that yields with the path $a^2ba^1b^1ab^2$, a Hamiltonian cycle of D.

Case 3. The complete balanced bipartite graph H is bipancyclic. Furthermore, there exist arcs (b^1, a), (a, b^2), (a^1, b), (b, a^2) that yield with H a Hamiltonian cycle, and thus a Hamiltonian path, in D.

This last case completes the proof. □

As a corollary, we obtain the existence of a Hamiltonian path in the case of almost balanced bipartite digraphs.

COROLLARY 3.5. *Let $D = (A, B, E \cup F)$ be an almost balanced bipartite digraph with $|A| = |B| + 1 \geqq 5$; $\delta^+ \geqq 1$ and $\delta^- \geqq 1$; and $d^+(y) \geqq 2$ and $d^-(y) \geqq 2$ for y in B and $\alpha^2_{\mathrm{BIP}}(D) \leqq 2$. Then D admits a Hamiltonian path.*

The proof, the same as in Corollary 2.4, is obtained by adding a new vertex v joined by an edge to each vertex of A.

In conclusion, let us note that we cannot hope to obtain a bipartite version of the Chen–Manalastas conditions involving $\alpha^0_{\mathrm{BIP}}(D)$ as in Theorem 1.1. Indeed, for every positive integer k, the following construction gives a balanced bipartite digraph of arbitrarily large order $2n$, with minimum indegree and outdegree k, connectivity k, $\alpha^0_{\mathrm{BIP}}(D) = 0$, and no Hamiltonian path.

Let $n > 3k$, $A = A_1 \cup A_2$, $B = B_1 \cup B_2$ with $|A_1| = |B_2| = n - k$, and $|A_2| = |B_1| = k$. D is the bipartite digraph $(A, B, E \cup F)$, where E is the set of all the edges between A_i and B_i $(i = 1, 2)$, and F is the set of all the arcs from A_1 to B_2 and from B_1 to A_2, together with $(k - 1)$ independent arcs directed from B_2 to A_1 and $(k - 1)$ independent arcs directed from A_2 to B_1.

The digraph D has no Hamiltonian path, since $|N^+(B_2)| = 2k - 1 < |B_2| - 1$.

REFERENCES

[1] P. ASH, *Two sufficient conditions for the existence of Hamiltonian cycles in bipartite graphs*, Ars Combin., 16 (1983), pp. 33–37.

[2] K. S. BAGGA AND B. N. VARMA, *Bipartite graphs and the degree conditions*, private communication.

[3] N. CHAKROUN AND D. SOTTEAU, *Chvátal-Erdos condition for pancyclability in digraphs with stability at most* 3, in Proceedings of the Workshop of Cycles and Rays, Montréal, Hahm, Sabidussi, Woodrow, eds., 1990, pp. 75–86.

[4] C. CHEN AND P. MANALASTAS, *Every finite strongly connected digraph of stability* 2 *has a Hamiltonian path*, Discrete Math., 44 (1983), pp. 243–250.

[5] R. ENTRINGER AND E. SCHMEICHEL, *Edge conditions and cycle structure in bipartite graphs*, Ars Combin., 26 (1988), pp. 229–232.

[6] O. FAVARON, P. MAGO, AND O. ORDAZ, *On the bipartite independence number of a balanced bipartite graph*, Discrete Math., to appear.

[7] P. FRAISSE, *D_λ-cycles and their applications for Hamiltonian graphs,* Ph.D. thesis, Université Paris-Sud, Paris, France, 1986.

[8] B. JACKSON AND O. ORDAZ, *Chvátal-Erdos conditions for paths and cycles in graphs and digraphs. A survey*, Discrete Math., 84 (1990), pp. 241–254.

[9] J. MITCHEM AND E. SCHMEICHEL, *Pancyclic and bipancyclic graphs. A survey*, in Graphs and Applications, Proceedings of the First Colorado Symposium on Graph Theory, F. Harary and J. S. Maybees, eds. John Wiley, New York, 1985, pp. 271–278.

[10] H. J. VELDMAN, *Existence of D_λ-cycles and D_λ-paths*, Discrete Math., 44 (1983), pp. 309–316.

SIAM J. DISC. MATH.
Vol. 6, No. 2, pp. 197-213, May 1993

LIMITING DISTRIBUTION FOR THE DEPTH IN PATRICIA TRIES*

BONITA RAIS†§, PHILIPPE JACQUET‡, AND WOJCIECH SZPANKOWSKI†¶

Abstract. Digital tries occur in a variety of computer and communication algorithms, including symbolic manipulations, compiling, comparison-based searching and sorting, digital retrieval techniques, algorithms on strings, file systems, codes, and communication protocols. The depth of the PATRICIA trie in a probabilistic framework is studied. The PATRICIA trie is a digital tree in which nodes that would otherwise have only one branch have been collapsed into nodes having more than one branch. Because of this characteristic, the depth of the PATRICIA trie provides a measure on the compression of the keys stored in the trie. Here, n independent keys that are random strings of symbols from a V-ary alphabet are considered. This model is known as the Bernoulli model. This paper shows that the depth in the asymmetric case (i.e., symbols from the alphabet do *not* occur with the same probability) is asymptotically *normally* distributed. In the symmetric case, which surprisingly proved to be more difficult, the limiting generating function and the limiting distribution are presented. In either case, the results point to the conclusion that the PATRICIA trie is with high probability a well-balanced tree.

Key words. compact tries, analysis of algorithms, complex analysis, Mellin transform, limiting distribution for depth

AMS(MOS) subject classifications. 68Q25, 68P05

1. Introduction. This paper establishes the limiting distribution for the depth of keys in a PATRICIA trie. A PATRICIA trie is a variation of the trie, a well-known tree structure, which is a frequently used data structure in many applications of computer science and telecommunications. These applications include dynamic hashing [5], [7], data compression [1], [2], pattern matching [1], and conflict resolution algorithms for broadcast communications [3], [13].

The *depth* of a leaf in a trie, also known as *depth of insertion* or *successful search time*, is the number of internal nodes on the path from the root of the trie to the leaf. It is of particular interest since it provides useful information in many applications. For example, when keys are stored in the leaves of the trie, the depth of a key gives an estimate of the search time for that key in searching and sorting algorithms [20]. Depth also gives the length of a conflict resolution session for tree-based communication protocols or, in compression algorithms, provides the length of a substring that may be copied or compressed [1].

The primary purpose of a trie is to store a set $\mathcal{S}$ of keys. Each key $X = x_1x_2x_3\cdots$ is a finite or infinite string of symbols taken from a finite alphabet $\mathcal{A} = \{\omega_1, \ldots, \omega_V\}$. The trie over $\mathcal{S}$ is built recursively as follows. For $|\mathcal{S}| = 0$, the trie is, of course, empty. For $|\mathcal{S}| = 1$, trie $(\mathcal{S})$ is a single node. If $|\mathcal{S}| > 1$, $\mathcal{S}$ is split into V subsets $\mathcal{S}_1$, $\mathcal{S}_2$, $\ldots$, $\mathcal{S}_V$, so that a key is in $\mathcal{S}_j$ if its first symbol is ω_j. The tries: trie $(\mathcal{S}_1)$, trie $(\mathcal{S}_2)$, $\ldots$, trie $(\mathcal{S}_V)$ are constructed in the same way, except that, at the kth step, the splitting of sets is based on the kth symbol. They are then connected from their respective roots

* Received by the editors April 8, 1991; accepted for publication (in revised form) January 28, 1992.

† Department of Computer Science, Purdue University, West Lafayette, Indiana 47907.

‡ Institut National de Recherche en Informatique et en Automatique (INRIA), Rocquencourt, 78153 Le Chesnay Cedex, France. This author's research was primarily supported by North Atlantic Treaty Organization (NATO) collaborative grant 00571/89.

§ This author's research was partially supported by Air Force Office of Scientific Research grant 90-0107 and National Science Foundation grant CCR-8900305.

¶ This author's research was partially supported by NATO collaborative grant 00571/89, by Air Force Office of Scientific Research grant 90-0107, by National Science Foundation grants CCR-8900305 and INT-8912631, and by grant R01 LM05118 from the National Library of Medicine.

to a single node to create trie ($\mathscr{S}$). A trie may have nodes with only one branch leading from it, and it is this waste of space that the PATRICIA trie eliminates by collapsing one-way branches into a single node. Thus the depth of a key in a PATRICIA trie may be less than that of the same key in a regular trie.

Consider the following example. Let $\mathscr{A} = \{0, 1, 2\}$ and $\mathscr{S} = \{A, B, C, D, E, F\}$, as defined in Fig. 1. The PATRICIA trie built over the set $\mathscr{S}$ is shown in Fig. 1. We can also vary both the trie and PATRICIA trie to a more general structure by allowing a leaf to hold at most b keys [7], [12]. This is the case in algorithms for extendible hashing in which the capacity of a page or other storage unit is b.

Tries have been analyzed by many authors under various probabilistic models, most having independent keys [7], [12], [25], [16], [18], [22]. Frequently, the symbols of $\mathscr{A}$ are also independent with Pr $\{x_j = \omega_i\} = p_i$ for any $j = 1, 2, \dots,$ where $\sum_{i=1}^{V} p_i = 1$, and we adopt these assumptions in this paper. Such a model is known as the *Bernoulli model*, provided that the number of keys n is fixed. If $p_1 = p_2 = \cdots = p_V = 1/V$, then the distribution of symbols is *symmetric*, otherwise it is *asymmetric*. Studies of the binary symmetric model have been carried out by Knuth [20], Flajolet [7], and Kirschenhofer and Prodinger [16]. The variance of the depth was also obtained in [16] (see also [18]). The limiting distribution for the depth of a regular trie was obtained independently by Jacquet and Régnier [12] (limiting distribution), Pittel [22] (limiting distribution), and Szpankowski [25] (all moments for the asymmetric independent model). The limiting distribution of depth in tries using a Markovian dependency among symbols is presented by Jacquet and Szpankowski in [14]. Pittel [21] has proved convergence *in probability* for the depth for a more general dependency among symbols (i.e., mixing stationary sequences).

PATRICIA tries have not been studied as extensively, but the moments of the successful search for the asymmetric model (see also [16], [20] for the binary symmetric case) and moments of the unsuccessful search for binary symmetric model have been obtained in Szpankowski [26], and the variance of the external path length by Kirschenhofer, Prodinger, and Szpankowski [18]. Pittel [22] provided the leading term in the *almost sure* convergence for the depth and the height. Also, Devroye [4] obtained results for depth and height of PATRICIA tries under a model in which the keys are random variables with a continuous density f on [0, 1]. In this paper, we obtain the convergence *in distribution* of the depth in the Bernoulli model. From the probabilistic viewpoint, this is the best and the strongest possible result regarding typical behavior of the depth in the PATRICIA.

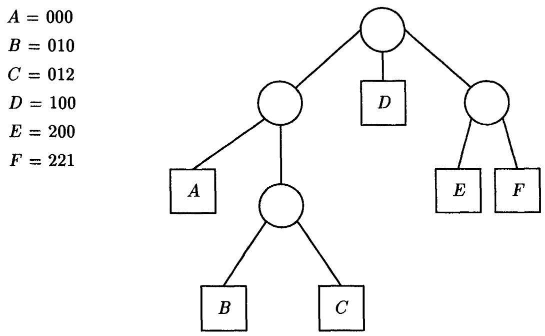


FIG. 1. *Example of a 3-ary digital trie with $n = 6$.*

Assuming independence among keys as well as symbols, our aim is to analyze the limiting distribution of the depth for a PATRICIA trie. To accomplish this, we use the Poisson transform and study the *Poisson model*, in which the number of keys in the trie follows a Poisson distribution with parameter n. After deriving results for this model, we use the inverse Poisson transform to obtain the results for our Bernoulli model. In either model, the distribution of the depth in the asymmetric case is asymptotically a Gaussian-like distribution. In our analysis of this, we use properties of the Mellin transform and follow the method suggested by Jacquet and Régnier in [12]. However, in the symmetric case where we obtain very different results, another approach is necessary.

The paper is organized as follows. The next section will give all necessary definitions and tools not yet defined. It will also give a statement of the main results and consequences of our findings. In the last section, we will prove all the results given in § 2.

2. Main results. Before making precise statements of our results, it is necessary to give some definitions and notation. We let the random variable D_n be the depth of a randomly chosen key in a PATRICIA trie holding n keys. Then $\Pr\{D_n = k\}$ is the probability that the depth of a key is k when the PATRICIA trie holds n keys; that is, k is the length of the path from the root to a randomly selected key. Then $D_n(u)$ is the *ordinary generating function*, and $D(z, u)$ is the *Poisson generating function*, where

$$D_n(u) = \sum_{k=0}^{\infty} \Pr\{D_n = k\} u^k, \qquad D(z, u) = \sum_{n=0}^{\infty} D_n(u) \frac{z^n}{n!} e^{-z}.$$

The first function $D_n(u)$ is used in the Bernoulli model where the number of keys is fixed at n and the probability of generating the ith symbol from the alphabet $\mathcal{A}$ is equal to p_i for $1 \leqq i \leqq V$. When z is real, $D(z, u)$ is the generating function for the Poisson model in which the number of keys in the trie follows a Poisson distribution with parameter z. These functions are well defined for any complex numbers z and u such that $|u| < 1$. However, in our analysis, we must analytically extend the functions to $|u| < 1 + \delta$ for some $\delta > 0$. When we replace u by e^t, where t is a complex number, we obtain the *characteristic function* of the respective distribution.

We summarize the main results of our study in the following theorems. The first theorem gives a complete characterization of the asymptotic behavior of the depth in a PATRICIA trie under the Bernoulli model with an asymmetric alphabet.

THEOREM 1. *Consider the* asymmetric *model of* PATRICIA *tries described above. Then*

(i) *For large n, the average depth ED_n of a* PATRICIA *trie is*

$$ED_n = \frac{1}{H} \log n + O(1),$$

and the variance var D_n *of the depth is*

$$\operatorname{var} D_n = \frac{H_2 - H^2}{H^3} \log n + O(1),$$

where $H = -\sum_{i=1}^{V} p_i \log p_i$ is the entropy of the alphabet, and $H_2 = \sum_{i=1}^{V} p_i \log^2 p_i$;

(ii) *The random variable $((D_n - ED_n)/\sqrt{\operatorname{var} D_n})$ is asymptotically normal with mean zero and variance* 1; *that is,*

$$\lim_{n \to \infty} \Pr\{D_n \leqq ED_n + x\sqrt{\operatorname{var} D_n}\} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2}\, dt.$$

In addition, for all integer m, the following convergence in moments holds:

$$E\left(\frac{D_n - ED_n}{\sqrt{\operatorname{var} D_n}}\right)^m \to \begin{cases} \dfrac{m!}{2^{m/2}(m/2)!} & m \text{ even}, \\ 0 & m \text{ odd}, \end{cases}$$

where the right-hand side of the above presents moments of the normal distribution.

The second theorem provides the limiting generating function, as well as the limiting distribution for the symmetric case. Proofs for both theorems are presented in the next section.

THEOREM 2. *Consider the* symmetric *model of* PATRICIA *tries described above.*

(i) Limiting generating function. *The limiting generating function $D_n(u)$ for the depth in a* PATRICIA *trie for large n is*

$$(1) \qquad D_n(u) = u^{\log_V n} \exp\left\{\frac{\alpha(u)}{\log V} - \frac{\log u}{2} + \frac{1}{\log V}\sum_{k\neq 0} g^*(s_k, u)n^{-s_k}\right\} + O\left(\frac{1}{\sqrt{n}}\right),$$

where $s_k = 2\pi ik/\log V$, $k \neq 0$,

$$(2) \qquad \alpha(u) = (1-u)\int_0^\infty \frac{e^{-z}\log z}{e^{-z} + u(1-e^{-z})}\,dz,$$

and

$$(3) \qquad g^*(s, u) = \int_0^\infty \log\,[e^{-z} + u(1-e^{-z})]z^{s-1}\,dz,$$

where $\alpha(u)$ is defined for $u \notin (-\infty, 0]$.

(ii) Limiting distribution. *Let $k_0 = \lfloor \log_V n \rfloor$ and $q = e^{-n/V^{k_0}}$. Toss a k_0-long sequence of biased coins with the probability of heads (success) for the jth coin equal to q^{V^j}. Let X_1 be the number of successes. Toss an infinite sequence of biased coins with probability of heads (success) for the jth coin equal to $q^{V^{-j}}$. Let X_2 be the number of failures. Then, asymptotically for such x that $x + \log_V n$ is integer,*

$$(4) \qquad \Pr\{D_n - \log_V n \leqq x\} = \Pr\{X_2 - X_1 \leqq x\} + O(1/\sqrt{n}),$$

and X_1 and X_2 are independent. Moreover, this characterization leads to

$$\lim_{n\to\infty} \Pr\{D_n - \log_V n \leqq x\} = f^P(V^{-x}),$$

where

$$(5) \qquad f^P(y) = \sum_{m=0}^{\infty} B_m(yV^{m-1})e^{-yV^{m-1}}$$

and

$$B_m(z) = e^{-z}\sum_{\substack{J\subset\{1,2,\cdots\}\\ |J|=m}} \prod_{j\in J}(e^{V^{-j}(V-1)z} - 1).$$

In particular, $B_0(z) = e^{-z}$ and $B_1(z) = e^{-z}\sum_{j=1}^{\infty}(e^{V^{-j}(V-1)z} - 1)$.

Remarks. (i) *Symmetric* PATRICIA. Although the limiting distribution is computed here for the first time, it was shown previously [20] that, for large n, the average ED_n depth of a PATRICIA trie is

$$ED_n = \log_V n + \frac{\gamma}{\log V} + \log_V(1 - 1/V) + \frac{1}{2} + P_1(\log n),$$

and the variance var D_n of the depth is constant; more precisely [16], [26],

$$\operatorname{var} D_n = \frac{\pi^2}{6 \log^2 V} + \frac{1}{12} - \frac{2}{\log V} \log \prod_{j=1}^{\infty} \left(1 + \frac{1}{V^j}\right) + P_2(\log_V n),$$

where $\gamma = 0.577$ is the Euler constant and $P_1(\log_V n)$ and $P_2(\log_V n)$ are fluctuating functions of very small amplitude. We also obtain these results from the limiting generating function of (1). These formulas follow from our limiting distribution (4). Finally, we also note that

$$\lim_{n \to \infty} \Pr \{D_n - \log_V n \leqq x\} = F_1(x) \cdot F_2(x),$$

where

$$F_1(x) = \exp(-V^{-x}), \qquad F_2(x) = \sum_{m=0}^{\infty} B_m(V^{-x+m-1}) \exp(-V^{-x}(V^{m-1} - 1)).$$

The function $F_1(x)$ is a distribution function. In fact, it is known as the standard extreme distribution (i.e., the so-called *double exponential* distribution $\exp(-e^{-x})$). Let Z be a random variable distributed according to $F_1(x)$. Then $EZ = \gamma/\log V - \frac{1}{2}$ and $\operatorname{var} Z = \pi^2/(6 \log^2 V) + \frac{1}{12}$, where the terms $\frac{1}{2}$ and $\frac{1}{12}$ are Sheppard's corrections for continuity (cf. [15]) of the average value and the variance, respectively. In fact, as discussed in remark (iii), below, $\log_V n + Z$ is distributed as the depth in regular tries.

(ii) *Aldous's representation for the symmetric case*. For completeness, we include an observation of Aldous, who noted that the problem of depth in the symmetric PATRICIA trie can be alternatively described in the following way. Consider an infinite row of boxes, and box i receives a Poisson number of balls with mean $V^{-i}(1 - 1/V)$. Let D_n^T be the number of the rightmost box containing a ball and let C be the number of empty boxes to the left of box D_n^T. In terms of tries, D_n^T gives the depth of a key in a regular symmetric trie, and C gives the number of nonbranching nodes on the path from the root to the key (in the Poisson model). Thus the depth in a PATRICIA trie D_n^P is given by $D_n^T - C$. Unfortunately, the random variables D_n^T and C are *dependent*, and therefore this representation is rather useless for the limiting distribution analysis. However, it is interesting to note that in the symmetric binary case, $\Pr\{C = d\} = 2^{-d}$ as is shown by Knuth[1]. This distribution of C was independently discovered by Pittel and Rubin [23].

(iii) *Comparison with regular tries*. When considering either the case of a symmetric or asymmetric alphabet, we can make the following observations. Although the expected depth of either the regular or PATRICIA trie is $\log n/H + O(1)$, the constant is not the same. Examination of this constant shows that the expected depth of a regular trie is greater than that of a PATRICIA trie. More importantly, the variance of the depth for a PATRICIA is smaller than for regular tries. This leads us to conclude that the PATRICIA trie is a better balanced trie than the regular trie.

We can offer further support of this claim in the symmetric case. In particular, as shown in [26], the difference in the variance for small alphabet is significant. For example, for binary regular tries, we have $\operatorname{var} D_n^T = 3.507 \ldots$, while, for binary PATRICIA, $\operatorname{var} D_n^P = 1.000 \ldots$. In fact, as proved in [17], the variance is $\operatorname{var} D_n^P = 1.000000000000 \ldots$ (twelve zeros). We also note that the difference becomes smaller for larger values of V, as expected. We can also compare the limiting distribution for the

[1] Amer. Math. Monthly, 94 (1987), p. 189.

depth in a PATRICIA trie with that for a regular trie. From [12], [22], we know that the limiting distribution for a regular trie under the symmetric alphabet is given by

$$\lim_{n\to\infty} \Pr\{D_n - \log_V n \leqq x\} = e^{-V^{-x}}. \tag{6}$$

A simple proof of this is given in Pittel [22], which proceeds as follows. First, observe that $\Pr\{D_n \leqq k\} = (1 - V^{-k})^{n-1}$. By letting $k = x + \log_V n$ and taking the limit as $n \to \infty$, Pittel obtains the limiting distribution $f^T(V^{-x})$ as given in (6), where $f^T(x) = e^{-x}$.

As is easy to see from Fig. 2, which compares $f^T(2^{-x})$ and $f^P(2^{-x})$, the probability of the depth of a randomly chosen key being at most $\log_V n + x$ is greater in a PATRICIA trie than in a regular trie. Since the mean depth is $\log_V n + O(1)$ for both structures, this supports the conclusion that the PATRICIA trie is better balanced than the regular trie.

(iv) *How well is the* PATRICIA *balanced*? A tree built over a V-ary alphabet is well balanced if (a) the average depth of a key is $\log_V n + O(1)$, (b) the variance of the depth is significantly smaller than the average depth, and (c) large derivations from the average value are very unlikely. Many algorithms using trees need balanced trees (e.g., the extendible hashing algorithm [5]) to run efficiently, so oftentimes these algorithms include a costly rebalancing step. This rebalancing operation is customarily justified by the worst-case analysis. Our average case analysis, however, shows that this costly operation seems to be unnecessary, since, with high probability, the tree is already balanced; that is, a random shape of a PATRICIA trie resembles the shape of the well-balanced structure of a complete tree [2]. In the symmetric case, we know that the expected depth of a PATRICIA trie is $\log_V n + O(1)$ and that its variance is $O(1)$; thus we can expect that the PATRICIA trie is well balanced. In the asymmetric case, we show that the limiting distribution for the depth is normal with mean $\log n/H + O(1)$ and variance $((H_2 - H^2)/H^3)\log n + O(1)$. The coefficient $1/H$ in the mean shows that the more asymmetric the distribution of the symbols is, the more skew the PATRICIA trie is. However, the standard deviation is $O(\sqrt{\log n})$, so the PATRICIA trie is still, on average, balanced. Efficiently preprocessing the asymmetric alphabet to obtain a more symmetric alphabet will improve the balance of the PATRICIA trie.

(v) *Poisson model*. In the proofs of our theorems, we will also establish similar convergence results for the Poisson model in which the number of keys is not fixed but rather a random variable distributed according to Poisson law. That is, in the asymmetric

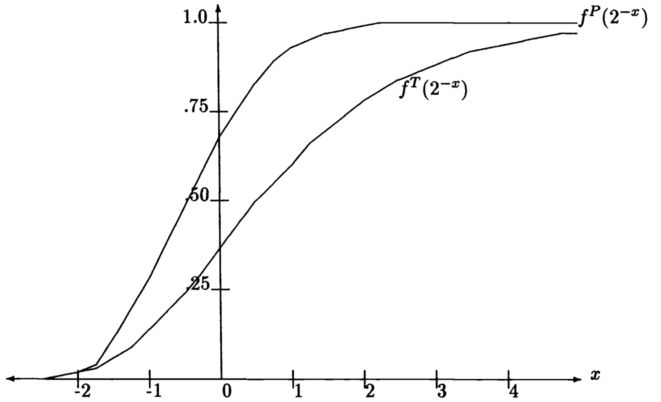


FIG. 2. *Comparison of distributions of tries and* PATRICIA *tries for* $V = 2$.

case, the depth of a key in a PATRICIA trie with Poisson number of records, once normalized and centered, is asymptotically normal.

3. Analysis. The primary focus of this section is the proof of our results. As mentioned earlier, different approaches are necessary to compute the limiting distributions for the depth of a PATRICIA trie in the symmetric and asymmetric cases. Before giving details of our analysis, we briefly identify tools that are useful in manipulating the generating functions defined in the previous section in both the symmetric and asymmetric cases. Then we will prove Theorems 1 and 2 in the following sections.

An important tool that will enable us to obtain asymptotic results is the *Mellin transform*, an integral transform from complex analysis, which is defined as follows [8] (cf. also [10]). Let $F(x)$ be a piecewise continuous function on the interval $[0, \infty)$. If $F(x) = O(x^{\alpha})$ for $x \to 0$ and $F(x) = O(x^{\beta})$ for $x \to \infty$, then the Mellin transform of $F(x)$, denoted $F^*(s)$, is defined for any complex number s in the strip $-\alpha < \Re(s) < -\beta$ and

$$F^*(s) = \int_0^{\infty} F(x)x^{s-1}\,dx.$$

The importance of the Mellin transform is that it provides information concerning the asymptotic behavior of a function $F(x)$ around zero and infinity through the poles of $F^*(s)$. In fact, the asymptotic expansion of $F(x)$ is obtained directly from the poles of its transform [8] as follows:

$$F(x) \sim \pm \sum_{\lambda \in \mathscr{H}} \operatorname{Res}\{F^*(s)x^{-s}, s = \lambda\},$$

where $\operatorname{Res}\{f(s), s = \lambda\}$ is the residue of $f(s)$ at $s = \lambda$ and $\mathscr{H}$ is the set of poles of $F^*(s)$ in the left (right) half plane giving the asymptotic expansion as $x \to 0$ $(x \to \infty)$. This is obtained using the inverse Mellin transform [10]

$$F(x) = \frac{1}{2i\pi} \int_{c-i\infty}^{c+i\infty} F^*(s)x^{-s}\,ds,$$

where $c \in (-\alpha, -\beta)$, and residue theory holds. Provided that $F^*(s)$ is small at $\pm i\infty$ and has only isolated singularities, for the case in which $x \to \infty$, we can close the contour to the right and derive (cf. [8], [10])

$$F(x) = -\sum_{\lambda} \operatorname{Res}\{F^*(s)x^{-s}, s = \lambda\} + O(x^{-M}),$$

where the sum is taken over all poles λ such that $-\beta \leqq \Re(\lambda) \leqq M$ for any M.

A second tool of great importance that will allow us to extract results for $D_n(u)$ from the results for $D(z, u)$ is the Poisson generating function. It is derived from Cauchy's integral formula, which says that [10]

$$D_n(u) = \frac{n!}{2\pi i} \oint D(z, u)e^z \frac{dz}{z^{n+1}}, \tag{7}$$

where the integration is taken over a circle of arbitrary radius centered at the origin. The following important result is derived from this formula when the radius of the circle is chosen to be n.

DEPOISSONIZATION LEMMA. *Let S_θ be a cone $S_\theta = \{z : |\arg z| < \theta, 0 < \theta < \pi/2\}$. If, for $z \in S_\theta$ and $z \to \infty$, the following holds for all u in a compact set $\mathscr{U}$:*

$$|D(z, u)| < \beta_1 |z|^{\varepsilon} \tag{8}$$

for some $\beta_1, \varepsilon > 0$, *and, for* $z \notin S_\theta$ *and all* $u \in \mathcal{U}$

$$|D(z, u)e^z| \leqq \beta_2 |z|^\varepsilon e^{\alpha|z|} \tag{9}$$

for some $0 < \alpha < 1$ *and a constant* $\beta_2 > 0$, *then, for large n uniformly in* $u \in \mathcal{U}$, *the generating function* $D_n(u)$ *satisfies*

$$D_n(u) = D(n, u) + O(n^{\varepsilon - 1/2}) \tag{10}$$

with $\varepsilon < \frac{1}{2}$.

Proof. We proceed along the lines of [13] with some necessary modifications. We apply Cauchy's formula (7). By the change of variable $z = ne^{it}$, we have

(11)

$$D_n(u) = \frac{n!}{n^n e^{-n}\sqrt{2\pi n}} \sqrt{\frac{n}{2\pi}} \int_{-\theta}^{\theta} D(ne^{it}, u) \exp(n(e^{it} - it - 1))\, dt + O(n^\varepsilon e^{-(1-\alpha)n}),$$

where the last term of the above is a simple consequence of our condition (9). We note that, by Stirling's formula, $n!/(n^n e^{-n}\sqrt{2\pi n}) = 1 + O(1/n)$. Hence, (11) can be reduced to $D_n(u) = (1 + O(1/n))I_n(u) + O(n^\varepsilon e^{-(1-\alpha)n})$, where

$$I_n(u) = \sqrt{\frac{n}{2\pi}} \int_{-\theta}^{\theta} D(ne^{it}, u) \exp(n(e^{it} - it - 1))\, dt.$$

Now we evaluate the integral $I_n(u)$. After introducing a new change of variable $x = t\sqrt{n}$, we obtain

$$I_n(u) = \frac{1}{\sqrt{2\pi}} \int_{-\theta}^{\theta} D(ne^{ix/\sqrt{n}}, u) \exp(n(e^{ix/\sqrt{n}} - ix/\sqrt{n} - 1))\, dx.$$

The following expansions are easy to derive:

$$\exp(n(e^{ix/\sqrt{n}} - ix/\sqrt{n} - 1)) = e^{-x^2/2}\left(1 - \frac{ix^3}{6\sqrt{n}} + O(1/n)\right)$$

and

$$D(ne^{ix/\sqrt{n}}, u) = D(n, u) + n(e^{ix/\sqrt{n}} - 1)D'(n, u) + O(1/n),$$

where $D'(n, u)$ is the first derivative of $D(z, u)$ with respect to z at $z = n$. However, again using the Cauchy formula in the form (cf. [10])

$$D'(n, u) = \frac{1}{\pi i} \oint \frac{D(z, u)}{(z - n)^2}\, dz$$

and our first condition (8), we finally prove that $D'(n, u) = O(n^{\varepsilon - 1})$ uniformly in $u \in \mathcal{U}$. This leads to (10) by noting that $(1/\sqrt{2\pi}) \int_{-\infty}^{\infty} e^{-x^2/2}\, dx = 1$ (cf. [6]). □

This lemma gives us the conditions necessary to transform our Poisson model results into those for the Bernoulli model, so we call it the Depoissonization Lemma (in fact, (10) can be called the inverse Poisson transform; see also [9] and [11]).

3.1. Asymmetric case. In this section, we adopt the approach of Jacquet and Régnier [12], making necessary changes required by the PATRICIA trie. At first, we give a rough plan of our analysis, which leads to the proof of our main results. To get the limiting distribution for depth in a PATRICIA trie under our Bernoulli model, we begin by deriving its probability generating function $D_n(u)$. Unfortunately, it is not easy to derive the limiting distribution directly. However, we use the Poisson transform to compute the generating function $D(z, u)$ for the Poisson model in which the number of keys follows a Poisson distribution with parameter z. This model is easier to analyze, since the generating function for the Poisson model has a closed form. This is not so in the Bernoulli model.

Since we are interested in the asymptotic behavior of the Poisson probability generating function $D(z, u)$, we use the Mellin transform. We also replace u by e^t, where t is complex. This will guarantee that each generating function in our sequence is analytic. The limit function of a sequence of analytic function is again analytic, so all of its derivatives are well defined. In this way, we also get convergence in moments.

We then show the following limit, where $\mu(z)$ is the mean and $\sigma(z)$ is the standard deviation for the Poisson model with parameter z, and $\tau = i\nu$ for any real number ν:

$$(12) \qquad \lim_{z \to \infty} e^{-\tau\mu(z)/\sigma(z)} D(z, e^{\tau/\sigma(z)}) = e^{\tau^2/2},$$

which is a modification of Goncharov's theorem (cf. [19, Chap. 1.2.10, Ex. 13]). By proving (12), we show that the depth in the Poisson model is asymptotically normal with mean $\mu(z)$ and variance $\sigma^2(z)$. The next step is to extract from (12) information about the Bernoulli model, that is, to *depoissonizate* the above formula. We do this by applying the Depoissonization Lemma to (12), and we obtain

$$\lim_{n \to \infty} e^{-\tau\mu_n/\sigma_n} D_n(e^{\tau/\sigma_n}) = e^{\tau^2/2}$$

for all $\tau = i\nu$ and $-\infty < \nu < \infty$, where $D_n(u)$ is the generating function of the depth D_n. However, the above is exactly Goncharov's theorem (cf. [19, Chap. 1.2.10, Ex. 13]), which states that a sequence of random variables D_n with mean μ_n and standard deviation σ_n approaches a normal distribution if the above holds. In this way, we prove Theorem 1.

Noting this plan, we present more details below. To simplify our notation, we will hereafter assume a binary alphabet, noting that our derivation easily extends to any finite alphabet. We denote the probabilities of the symbols ω_1 and ω_2 as p and q, where $p + q = 1$.

Let $\mathcal{I}_n$ be the set of all possible PATRICIA tries of n keys from the alphabet $\mathcal{A}$ and $\mathcal{T}$ be a particular trie from $\mathcal{I}_n$. If $S^k_{\mathcal{T}}$ denotes the number of keys at depth k in $\mathcal{T}$, then the generating function associated with $\mathcal{T}$ is given by

$$S_{\mathcal{T}}(u) = \sum_{k=0}^{\infty} S^k_{\mathcal{T}} u^k.$$

Note that the sum is actually finite, since the maximum depth in a PATRICIA trie with n keys is $n - 1$. Clearly, the following statements are true when the left subtree $\mathcal{T}_\alpha$ and right subtree $\mathcal{T}_\beta$ hold k and $n - k$ keys, respectively, and $\delta_{j,k}$ is the Kronecker delta (i.e., $\delta_{j,k} = 1$, if $j = k$, $\delta_{j,k} = 0$, otherwise)

$$S_{\mathcal{T}}(u) = n, \qquad n \leqq b,$$

$$S_{\mathcal{T}}(u) = u\{S_{\mathcal{T}_\alpha}(u) + S_{\mathcal{T}_\beta}(u)\} + (1 - u)(\delta_{k,n} + \delta_{n-k,n})S_{\mathcal{T}}(u), \qquad n > b.$$

We note that the above recurrence holds for a particular tree $\mathcal{T}$ in $\mathcal{I}_n$. The leading factor u must be present, since the depth of a key in either $\mathcal{T}_\alpha$ or in $\mathcal{T}_\beta$ is one less than its depth in the trie $\mathcal{T}$. The second term avoids one-way branching. For example, if $k = 0$, then the right branch is a one-way branch and $S_{\mathcal{T}}(u) = uS_{\mathcal{T}}(u) + (1 - u)S_{\mathcal{T}}(u)$, which means that the subtree begins at the root of $\mathcal{T}$.

Averaging $S_{\mathcal{T}}(u)$ over all tries $\mathcal{T}$ in $\mathcal{I}_n$, we derive a new generating function $S_n(u) = E(S_{\mathcal{T}}(u))$. Of course, for $n \leqq b$, we have $S_n(u) = n$. Otherwise,

$$(13) \qquad S_n(u) = u \sum_{k=0}^{n} \binom{n}{k} p^k q^{n-k}[S_k(u) + S_{n-k}(u)] - (u-1)(p^n + q^n)S_n(u).$$

Here the sum in the first term ranges over k from zero to n. We know that, in a PATRICIA trie, there are no one-way branches; thus we must subtract those terms from the sum. However, it is possible that all n keys begin with the same symbol from $\mathcal{A}$. This occurs with probability $p^n + q^n$, and so we add this term. The last term in (13) makes the analysis of the depth of the PATRICIA trie different from that of the regular trie.

Define $S(z, u) = \sum_{n=0}^{\infty} S_n(u)(z^n/n!)e^{-z}$ and note that $S(z, u)$ is the generating function of the depth in the Poisson model. Since $S_n(1) = n$, we see that, for any z, we have $S(z, 1) = z$. Using the relation in (13), we obtain the following functional equation:

$$\begin{aligned} S(z, u) = {} & u[S(pz, u) + S(qz, u)] + (1-u)e^{-z}ze_{b-1}(z) \\ & + (1-u)[S(pz, u)e^{-qz} + S(qz, u)e^{-pz}] \\ & + (u-1)e^{-z}[pze_{b-1}(pz) + qze_{b-1}(qz)], \end{aligned}$$

where $e_m(x) = 1 + x + \cdots + x^m/m!$.

The generating function $D_n(u) = S_n(u)/n$ is the *probability* generating function [22] for the depth of a leaf in a PATRICIA, since the coefficient of u^k is the probability that a randomly chosen key in a randomly chosen trie $\mathcal{T}$ in $\mathcal{I}_n$ is at depth k. Thus the Poisson generating function for depth of a leaf is $D(z, u) = S(z, u)/z$ and $D(\alpha z, 1) = 1$ for all α and z. (Note, in fact, that $D(z, u) = e^{-z}\sum_{n \geqq 0} D_{n+1}(u)(z^n/n!)$.) Consequently, we have

$$(14) \qquad \begin{aligned} D(z, u) = {} & upD(pz, u) + uqD(qz, u) + (1-u)e^{-z}e_{b-1}(z) \\ & + (1-u)[D(pz, u)pe^{-qz} + D(qz, u)qe^{-pz}] \\ & + (u-1)e^{-z}[pe_{b-1}(pz) + qe_{b-1}(qz)]. \end{aligned}$$

We have in (14) the functional equation corresponding to the Poisson generating function for the depth in a PATRICIA trie. The first three terms give the functional equation for the regular trie [12]. Since we are interested in its asymptotic behavior, we would like to solve it. This, however, is too difficult, so we use the Mellin transform to calculate the asymptotics of $D(z, u)$. Since the strip on which the Mellin transform of $D(z, u)$ is defined is empty, let $D^*(s, u)$ be the Mellin transform of $D(z, u) - 1$. Note that it is defined for all s in the strip where $-1 < \Re(s) < 0$. (In fact, it is defined on the larger strip $-b < \Re(s) < 0$, since, as $z \to 0$, $D(z, u) - 1 = O(z^b)$, and $z \to \infty$, $D(z, u) - 1 = O(z^\varepsilon)$ for some $\varepsilon > 0$; see the Appendix. Subsequent integral computations require the smaller strip.) Computing $D^*(s, u)$ requires the evaluation of many integrals, but ultimately we arrive at the following:

$$(15) \qquad D^*(s, u) = \frac{(1-u)G^*(s, u)}{1 - u(p^{1-s} + q^{1-s})},$$

where

$$(16) \qquad \begin{aligned} G^*(s, u) = {} & \int_0^\infty (pe^{-qz}[D(pz, u) - 1] + qe^{-pz}[D(qz, u) - 1])z^{s-1}\,dz \\ & + \frac{\Gamma(s+b)}{s(b-1)!} - \sum_{j=0}^{b-1} \Gamma(s+j)\frac{p^{j+1} + q^{j+1}}{j!} - \Gamma(s)(pq^{-s} + qp^{-s}). \end{aligned}$$

Although (15) looks very much like that for regular tries in [12], it is, in fact, quite different. For regular tries, $G^*(s, u)$ is exactly the second term of (16), but in (15) we see that $D^*(s, u)$ is only implicitly given, since $G^*(s, u)$ contains an integral depending on $D(pz, u)$ and $D(qz, u)$. The analysis of $D(z, u)$ is clearly more difficult in the case of PATRICIA tries than in the case of regular tries.

Now, to prove (12), we let $u = e^t$, where t is complex. We want to evaluate the asymptotics of $D(z, e^t)$ as t goes to zero. We can recover $D(z, e^t)$ from $D^*(s, u)$ by evaluating the integral

$$D(z, u) = \frac{1}{2i\pi} \int_{-1/2 - i\infty}^{-1/2 + i\infty} z^{-s} D^*(s, u)\, ds.$$

This is the inverse Mellin transform [10]. We use Cauchy's residue theorem to evaluate this integral, but we must first find the poles of the integrand. These correspond to the roots of

$$e^t(p^{1-s} + q^{1-s}) = 1. \tag{17}$$

Now, following [12], we analyze the roots of (17) lying in the strip $\Re(s) \leqq 1$. We denote them as $s_k(t)$. Let also $R_k(t)$ be the residues of $1/(1 - e^t(p^{1-s} + q^{1-s}))$ at these points for $k = 0, \pm 1, \pm 2, \ldots$. Then we can write $D(z, e^t)$ as follows:

$$\begin{aligned} D(z, e^t) = {} & R_0(t) G^*(s_0(t), e^t)(1 - e^t) z^{-s_0(t)} \\ & + (1 - e^t) \sum_{k \neq 0} R_k(t) G^*(s_k(t), e^t) z^{-s_k(t)} + O(z^{-M}), \end{aligned} \tag{18}$$

for arbitrary $M > 0$. Now we compute the components of (18), and we begin with $s_0(t)$. Since (17) and the equation

$$e^{-t} = p^{1-s} + q^{1-s} \tag{19}$$

are equivalent, we solve above for $s_0(t)$. First, expand both sides using a Taylor's series up to terms of degree two. We then have

$$1 - t + t^2/2 + O(t^3) = 1 + Hs_0(t) + H_2 s_0(t)^2/2 + O(s_0(t)^3), \tag{20}$$

where $H = -(p \log p + q \log q)$ and $H_2 = p \log^2 p + q \log^2 q$. Since $s_0(0) = 0$, we can write

$$s_0(t) = at + bt^2 + O(t^3). \tag{21}$$

Substitute (21) into (19). Equating coefficients and solving for a and b, we see that $s_0(t) = -t/H + \frac{1}{2}(1/H - H_2/H^3)t^2 + O(t^3)$. We also note that its residue is $R_0(t) = -1/H + O(t)$.

Now we are ready to show that (18) can be written as

$$D(z, e^t) = z^{-s_0(t)}(1 + O(t|z|^{-At^2})) \tag{22}$$

for some constant A. We begin with the first term of (18). The behavior of $R_0(t)$ and $(1 - e^t)$ when $t \rightarrow 0$ has already been determined, so we continue by examining $G^*(s_0(t), e^t)$, which is given below in an alternate form than (16):

$$\begin{aligned} G^*(s, e^t) = {} & \int_0^\infty (pe^{-qz}[D(pz, e^t) - 1] + qe^{-pz}[D(qz, e^t) - 1]) z^{s-1}\, dz \\ & + \sum_{j=1}^{b-1} \Gamma(s + j) \frac{1 - (p^{j+1} + q^{j+1})}{j!} - \Gamma(s)(pq^{-s} + qp^{-s}). \end{aligned} \tag{23}$$

Near zero, $\Gamma(z) = z^{-1} - \gamma + O(z)$ and $n^{-z} = 1 - z \log n + O(z^2)$. Thus the last term of (23) behaves as $-H/t + O(1)$, and the middle term behaves as a constant as $t \to 0$. Finally, for the integral in (23), we note that, since $D(\alpha z, e^t)$ is continuous and $D(\alpha z, 1) = 1$, as $t \to 0$, we have $D(\alpha z, e^t) - 1 \to 0$ as $t \to 0$. Therefore, the integral will converge to zero, provided that it converges uniformly. This can be shown to be true [10]. Thus, as $t \to 0$, $G^*(s_0(t), e^t) \to H/t$ and $R_0(t)G^*(s_0(t), e^t)(1 - e^t) \to 1$.

Finally, for the Poisson model, it remains only to show that the sum in (18) is small when t is small. The proof of this is similar to the one that appears in [12] and [14]. It relies on showing that $\sum_{k\neq 0} |R_k(t)G^*(s_k(t), e^t)| = O(1)$ and that $\Re(s_k(t)) \geqq s_0(\Re(t))$. This implies that, for some $A > 0$,

$$|D(z, e^t)| \leqq z^{-\Re(s_0(t))}(1 + O(t|z|^{-At^2}),$$

which behaves as $z^{-\Re(s_0(t))}$ as $t \to 0$. Therefore, the sum in (18) contributes $z^{-s_0(t)}o(1)$, giving (22). Writing (22) in a more convenient form, we have

$$(24)\quad D(z, e^t) = \exp\left\{\frac{t}{H}\log z - \frac{1}{2}\left[\frac{1}{H} - \frac{H_2}{H^3}\right]t^2 \log z + O(t^3)\log z\right\}(1 + O(t|z|^{At^2})).$$

This directly leads to (12); that is,

$$e^{-t\mu(z)/\sigma(z)}D(z, e^{t/\sigma(z)}) = e^{t^2/2}(1 + o(1)).$$

Hence, we see that the mean of the distribution of the Poisson model with parameter z is $\mu(z) = \log z/H + O(1)$, and its variance is $\sigma^2(z) = -[1/H - H_2/H^3]\log z + O(1)$.

Finally, to prove our main result for the Bernoulli model, we must use the Depoissonization Lemma. However, this requires us to verify hypotheses (8) and (9). (This is rather technical and appears in the Appendix.) Then we can compute $D_n(e^t)$ from (24) and (10), and, by Goncharov's theorem, we prove that the limiting distribution of the Bernoulli model is normal with mean $\mu_n = \log n/H + O(1)$ and variance $\sigma_n^2 = -[1/H - H_2/H^3]\log n + O(1)$. This completes the proof of Theorem 1.

3.2. Symmetric case. Note that, in the preceding analysis, when $p = q = \frac{1}{2}$, the variance var D_n becomes $O(1)$ because, in this case, $H_2 = H^2 = \log^2 V$. Hence, from (24) we conclude that Goncharov's theorem cannot hold, and we need a somewhat different analysis. More precisely, the Mellin transform (15) in this case becomes

$$D^*(s, u) = \frac{G^*(s, u)}{1 - ue^{s\log V}}.$$

The poles are all on the axis defined by $\Re(s\log V + \log u) = 0$. Therefore, by the Mellin inverse formula, we obtain

$$(25)\quad \begin{aligned} D(z, e^t) = {} & \frac{z^{t/\log V}}{\log V}\left[G^*\left(-\frac{t}{\log V}, e^t\right) + \sum_{k\neq 0} G^*\left(-\frac{t - 2ik\pi}{\log V}, e^t\right)z^{2ik\pi/\log V}\right] \\ & + O(z^{-M}) \end{aligned}$$

with M as large as we want. Then, from the Depoissonization Lemma, we have

$$D_n(e^t) = D(n, e^t) + O(n^{\varepsilon - 1/2}).$$

This form for the limiting distribution in the symmetric case is unsatisfying, since it gives little information except that the distribution is periodic with period log V. Thus we must search for an alternative representation. We henceforth consider the case where $b = 1$.

We begin another approach by again considering $S_n(u)$. The recurrence relation for $S_n(u)$ is obtained from (22) by setting $p = q = \frac{1}{2}$, as follows:

$$(26) \qquad S_n(u) = u \sum_{k=0}^{n} \binom{n}{k} 2^{-n}\{S_k(u) + S_{n-k}(u)\} + (1 - u)2^{-n}2S_n(u).$$

We cannot solve the recurrence in (26) directly, so we define $S(z, u) = \sum_{n=0}^{\infty} S_n(u)(z^n/n!)$ as we previously did. This gives another recurrence

$$S(z, u) = 2S(z/2, u)\{ue^{z/2} - u + 1\}$$

similar to that derived in the asymmetric model. Finally, since $D_n(u) = S_n(u)/n$ (we set $D_0(u) = 0$), by defining $D(z, u) = S(z, u)e^{-z}/z$, we obtain the following Poisson generating function for the depth in a PATRICIA trie:

$$(27) \qquad D(z, u) = D(z/2, u)\{e^{-z/2} + u(1 - e^{-z/2})\}.$$

Iterating it and knowing that $D(0, u) = 1$, we are able to express $D(z, u)$ as an infinite product (cf. [13])

$$(28) \qquad D(z, u) = \prod_{k=1}^{\infty} \{e^{-z/2^k} + u(1 - e^{-z/2^k})\}.$$

We need this form of the generating function to prove our theorem. It should be noted here that the function given in (28) is the generating function of $\sum_{k\geqq 1} Y_k$, where the Y_k are independent random variables that take on the values zero or 1 with $\Pr\{Y_k = 1\} = 1 - e^{-z/2^k}$.

We start with proving Theorem 2(i) concerning the limiting generating function of the depth. Although (28) provides the generating function, it is difficult to extract information concerning the distribution. However, since we are primarily interested in deriving a limiting law, we can use the Mellin transform. We let $l(z, u) = \log(D(z, u))$ and compute its Mellin transform. Note that $l(z, u)$ can be written as

$$l(z, u) = \sum_{k=1}^{\infty} \log[e^{-z/2^k} + u(1 - e^{-z/2^k})].$$

Let $g(z, u) = \log[e^{-z} + u(1 - e^{-z})]$. Using a special property of the Mellin transform concerning harmonic sums (cf. [8]), $l^*(s, u)$ can be computed as the product of $g^*(s, u)$ and $\sum_{k=1}^{\infty}(2^{-k})^{-s}$. First, we determine the strip on which $g^*(s, u)$ is defined. Note that, as $z \to 0$, $g(z, u) = O(z)$, and, as $z \to \infty$, $g(z, u) = O(1)$. Thus the Mellin transform of $g(z, u)$, and therefore of $l(z, u)$, is defined in the strip $-1 < \Re(s) < 0$. Furthermore, using (3) to compute $g^*(s, u)$, we obtain

$$(29) \qquad g^*(s, u) = \frac{-\log u}{s} + \alpha(u) + O(s),$$

where $\alpha(u)$ is defined as in (2). Finally, we have

$$(30) \qquad l^*(s, u) = \frac{2^s}{1 - 2^s} g^*(s, u).$$

Now we can use $l^*(s, u)$ to determine the asymptotic expansion of $l(z, u)$. By definition, the inverse Mellin transform is given by

$$l(z, u) = \frac{1}{2\pi i} \int_{-1/2 - i\infty}^{-1/2 + i\infty} l^*(s, u) z^{-s}\, ds.$$

This integral can be computed using Cauchy's theorem on residues. Since we want the expansion to hold for large values of z, we close the contour to the right, with the left boundary as the line $\Re(s) = -\frac{1}{2}$. Our next step then is to identify the poles of the integrand $l^*(s, u)z^{-s}$ with respect to s and to determine their residues.

Clearly, the only poles are those of $l^*(s, u)$. There are poles of multiplicity 1 at $s_k = 2\pi ik/\log 2$ for all integers k, since $1 - 2^{s_k} = 0$. However, $s_0 = 0$ is actually a double pole, since it also is a pole of $g^*(s, u)$. The residues at the single poles s_k, $k \neq 0$ are easily calculated and are equal to $(-g^*(s_k, u)/\log 2)z^{-s_k}$. The residue associated with s_0 requires more work.

We begin this computation by expanding all factors of $l^*(s, u)z^{-s}$. The expansion of $g^*(s, u)$ is shown in (29). The other factors are then written as

$$(31) \qquad \frac{2^s}{1 - 2^s} = \frac{-1}{s \log 2} - \frac{1}{2} + O(s)$$

and

$$(32) \qquad z^{-s} = 1 - s \log z + O(s^2).$$

Multiplying (29), (31), and (32) and taking the coefficient of $1/s$ gives us the residue at s_0, namely, $-\log z(\log u/\log 2) - [\alpha(u)/\log 2 - (\log u)/2]$. We can therefore write

$$l(z, u) = \log z \frac{\log u}{\log 2} + \frac{\alpha(u)}{\log 2} - \frac{\log u}{2} + \frac{1}{\log 2} \sum_{k \neq 0} g^*(s_k, u)z^{-s_k} + O(z^{-M}).$$

Finally, $D(z, u) = e^{l(z,u)}$. This gives us the Poisson generating function; so we have a result for the Poisson model. From this, we obtain the results for the Bernoulli model, the generating function $D_n(u)$ of (1), by applying the Depoissonization Lemma from the previous section. Of course, the hypotheses of the lemma must first be verified, but this is a simple variation of the proof that appears in the Appendix. Thus the proof of part (i) of Theorem 2 is complete.

We have not yet obtained the limiting distribution that is our ultimate goal. Ordinarily, this can be found from $D_n(u)/(1 - u)$ using Cauchy's integral formula [10]. However, we cannot use this technique here, since the expression we obtained for $D_n(u)$ does not appear to be analytic within any circle about the origin, a necessary condition of Cauchy's formula, due to the presence of $\log u$. Therefore, another approach is necessary to obtain the limiting distribution for the depth in a PATRICIA trie. We can, however, compute all moments from this limiting generating function since all of its derivatives exist at $u = 1$.

Now we turn our attention to the limiting distribution and the proof of part (ii) of Theorem 2. To show that $D_n - \log_2 n$ can be written as the difference of the random variables X_1 and X_2 as defined in Theorem 2, let $G_{X_1}(u)$ and $G_{X_2}(u)$ be their respective generating functions. Recall from probability theory [6] that $G_{X_2 - X_1}(u) = G_{X_2}(u)G_{X_1}(1/u)$. Clearly, then,

$$(33) \qquad G_{X_2 - X_1}(u) = \prod_{j=1}^{\infty} [q^{2^{-j}} + u(1 - q^{2^{-j}})] \prod_{j=0}^{k_0} [u^{-1}q^{2^j} + (1 - q^{2^j})].$$

By applying the Depoissonization Lemma, we have $D_n(u) = D(n, u) + O(1/\sqrt{n})$. Using this in (28), dividing both sides of the result by u^{k_0}, and replacing $e^{-n/2^{k_0}}$ by q, we have exactly (33). This proves that asymptotically $D_n - \log_2 n = X_2 - X_1$.

To show the second part of Theorem 2, consider the functional equation of (27) and define a new function $F(z, u) = D(z, u)/(1 - u)$. The generating function $F(z, u)$ is then

$$F(z, u) = uF(z, u) + e^{-z/2}D(z/2, u).$$

For now, we assume that $u < 1$. Iterating repeatedly and again using the fact that $D(0, u) = 1$, we obtain

$$F(z, u) = \sum_{k=0}^{\infty} u^k e^{-z/2^{k+1}} D(z/2^{k+1}, u). \tag{34}$$

Define $B_m(z)$ so that $D(z, u) = \sum_{m=0}^{\infty} B_m(z) u^m$. Then, substituting this into (34), we obtain

$$F(z, u) = \sum_{k=0}^{\infty} u^k \sum_{m=0}^{k} \{B_m(z/2^{k+1-m}) e^{-z/2^{k+1-m}}\}. \tag{35}$$

So, we can obtain the limiting distribution if we can compute $B_m(z)$ for $m \geqq 0$. To do this, consider (28). The coefficient of u^m in the expansion of this product is exactly $B_m(z)$. Clearly, then, with some algebraic manipulation, $B_0(z) = e^{-z}$ and

$$B_1(z) = e^{-z} \sum_{j_1=1}^{\infty} (e^{z/2^{j_1}} - 1),$$

$$B_2(z) = e^{-z} \sum_{j_1=1}^{\infty} \left\{ (e^{z/2^{j_1}} - 1) \sum_{j_2=1, j_2 \neq j_1}^{\infty} (e^{z/2^{j_2}} - 1) \right\}.$$

Other $B_m(z)$ for $m > 2$ are similar to $B_2(z)$, having m sums with the condition that no two j_i's are equal.

Again using the Depoissonization Lemma, $F(n, u) = D_n(u)/(1 - u) + O(n^{\varepsilon - 1/2})$, making

$$\Pr\{D_n \leqq k\} = \sum_{m=0}^{k} B_m(n/2^{k+1-m}) e^{-n/2^{k+1-m}}.$$

Let $k = \log_2 n + x$ be an integer. Then

$$\Pr\{D_n - \log_2 n \leqq x\} = \sum_{m=0}^{\log_2 n + x} B_m(2^{-(x+1-m)}) e^{-2^{-(x+1-m)}}. \tag{36}$$

The proof of Theorem 2 is now complete.

Appendix. In this appendix, we prove that conditions of the Depoissonization Lemma hold for our problem, and we can use the inverse Poisson transform to prove the limiting distribution of PATRICIA for the Bernoulli model. Although the proof is written for the asymmetric case, it also holds in the symmetric case. We start with the following proposition.

PROPOSITION 1. *For each* $\varepsilon > 0$, *there exists a neighborhood of* 1, $\mathcal{U}(1)$, *such that, for all* u *in* $\mathcal{U}(1)$, z *in* S_θ, *and* $|z| > 1$, *the following holds*: $|D(z, u)| < |z|^\varepsilon$.

Proof. Let us define ρ such that $\rho > 1$ and $\rho(p^{1+\varepsilon} + q^{1+\varepsilon}) < 1 - \varepsilon'$, for some $\varepsilon' > 0$. Suppose also that $p > q$. Let us choose A such that $A > 1/q$ and such that, for $z \in S_\theta$ and $|z| \geqq A$, the following holds:

$$(1 + \rho)\{|pz|^\varepsilon |e^{-pz}| + |qz|^\varepsilon |e^{-qz}| + |e^{-z}\{e_{b-1}(z) - pe_{b-1}(pz) - qe_{b-1}(qz)\}|\} < \varepsilon' |z|^\varepsilon.$$

Let us define a sequence of domains $R_0 = \{z : z \in S_\theta, B \leqq |z| \leqq A\}$ with $1 < B < A$ and, for m natural, $R_m = \{z : z \in S_\theta, 1 < |z| < A/\rho^m\}$. An interesting fact is that $z \in R_m - R_{m-1}$ implies that $qz \in R_{m-1}$ and $pz \in R_{m-1}$. We prove our proposition by recursion on domains $R_m \cap S_\theta$. Since $R_0 \cap S_\theta$ is compact, $D(z, 1) = 1$, and $|z|^\varepsilon > 1$, there is a neighborhood $\mathcal{U}(1)$ of 1 such that, for all $u \in \mathcal{U}(1)$ and $z \in R_0 \cap S_\theta$, we have $|D(z, u)| < |z|^\varepsilon$. We can restrict u such that $|u| < \rho$ (by redefining $\mathcal{U}(1)$ if necessary).

Now let us suppose that, for all $z \in R_m \cap S_\theta$ and for all $u \in V(1)$, Proposition 1 holds; that is, $|D(z, u)| < |z|^\varepsilon$. We prove that the proposition is true for all $z \in R_{m+1} \cap S_\theta$. Let $z \in (R_{m+1} - R_m) \cap S_\theta$ and u in $\mathcal{U}(1)$. Then, by (14),

$$D(z, u) = upD(pz, u) + uqD(qz, u) + (1 - u)\{D(pz, u)pe^{-qz} + D(qz, u)qe^{-pz}\} \\ + (1 - u)\{e_{b-1}(z) - pe_{b-1}(pz) - qe_{b-1}(qz)\}e^{-z}.$$

Since pz and qz are in $R_m \cap S_\theta$, we can use the fact that $|D(pz, u)| < |pz|^\varepsilon$ and $|D(qz, u)| < |qz|^\varepsilon$. So

$$|D(z, u)| < \rho(p^{1+\varepsilon} + q^{1+\varepsilon})|z|^\varepsilon + (1 + \rho)\{|pz|^\varepsilon p|e^{-qz}| + |qz|^\varepsilon q|e^{-pz}|\} \\ + (1 + \rho)|\{e_{b-1}(z) - pe_{b-1}(pz) - qe_{b-1}(qz)\}e^{-z}|.$$

According to the fact that $|z| > A$ and $z \in S_\theta$, we can use the following hypothesis about A:

$$|D(z, u)| < (1 - \varepsilon')|z|^\varepsilon + \varepsilon'|z|^\varepsilon = |z|^\varepsilon,$$

and this completes the induction step. □

To verify the second condition of the Depoissonization Lemma, we must now only check that $D(z, u)$ outside cone S_θ does not grow faster than exponential. We prove this below.

PROPOSITION 2. *There exists* $\alpha < 1$ *and a neighborhood* $\mathcal{U}(1)$ *of* 1 *such that, for all* $u \in \mathcal{U}(1)$, $z \notin S_\theta$, *and* $|z| > 1$ *implies that* $|D(z, u)e^z| \leqq |z|^\varepsilon e^{\alpha|z|}$.

Proof. Essentially, we have $\cos\theta < \alpha < 1$ because $|e^z| = e^{\Re(z)} \leqq e^{\cos\theta|z|}$ for z not in S_θ. Let S_θ^c be the complementary set of S_θ in the complex plane. Let us choose $A > 0$ such that $A > 1/q$ and such that, for $z \in S_\theta^c$ and $|z| \geqq A$, the following holds:

$$(1 + \rho)\{|pz|^\varepsilon pe^{p\alpha|z|} + |qz|^\varepsilon qe^{q\alpha|z|} + |e_{b-1}(z) - pe_{b-1}(pz) - qe_{b-1}(qz)|\} \\ < \varepsilon'|z|^\varepsilon e^{\alpha|z|}.$$

Using the domains R_m as defined in the previous proof, we can establish Proposition 2 by mathematical induction on domains $R_m \cap S_\theta^c$. Since the above sets are compact, $D(z, 1)e^z = e^z$, and $|e^z| < |z|^\varepsilon e^{\alpha|z|}$, there exists a neighborhood $\mathcal{U}(1)$ such that, for all u in $\mathcal{U}(1)$ and $z \in R_0 \cap S_\theta^c$, the following holds: $|D(z, u)e^z| \leqq |z|^\varepsilon e^{\alpha|z|}$.

Now let us suppose that the property is true on $R_m \cap S_\theta^c$, and we will prove that it also holds for $m + 1$. Let $z \in (R_{m+1} - R_m) \cap S_\theta^c$ with u in $\mathcal{U}(1)$. Then by (14) we have

$$D(z, u)e^z = upD(pz, u)e^z + uqD(qz, u)e^z + (1 - u)\{D(pz, u)pe^{pz} + D(qz, u)qe^{qz}\} \\ + (1 - u)\{e_{b-1}(z) - pe_{b-1}(pz) - qe_{b-1}(qz)\}.$$

Therefore, taking into account the mathematical induction hypotheses, that is, $|D(az, u)e^{-az}| < |az|^\varepsilon e^{\alpha a|z|}$ with a either p or q, we finally obtain

$$|D(z, u)e^z| \leqq \{\rho p^{1+\varepsilon}e^{\alpha p|z|}|e^{qz}||z|^\varepsilon + \rho q^{1+\varepsilon}e^{\alpha q|z|}|e^{pz}||z|^\varepsilon\} + \varepsilon'|z|^\varepsilon e^{\alpha|z|} \\ \leqq (\rho p^{1+\varepsilon} + \rho q^{1+\varepsilon})|z|^\varepsilon e^{\alpha|z|} + \varepsilon'|z|^\varepsilon e^{\alpha|z|} \leqq |z|^\varepsilon e^{\alpha|z|},$$

and this completes the proof of Proposition 2 and also verification of hypotheses (8) and (9) in the Depoissonization Lemma. □

Acknowledgments. We are very grateful to Dr. Philippe Flajolet for a suggestion that led to part (ii) of our Theorem 2. We also would like to thank an anonymous referee for suggestions that improved the presentation of this paper.

REFERENCES

[1] A. Apostolico, *The myriad virtue of suffix trees*, NATO Adv. Sci. Inst. Ser. F: Comput. Systems Sci. 12 (1985), pp. 85–96.

[2] A. Aho, J. Hopcroft, and J. Ullman, *Data Structures and Algorithms*, Addison-Wesley, Reading, MA, 1983.

[3] J. Capetanakis, *Tree algorithms for packet broadcast channels*, IEEE Trans. Inform. Theory, 25 (1980), pp. 605–615.

[4] L. Devroye, *A study of trie-like structures under the density model*, Ann. Appl. Probab. 2 (1992), pp. 402–434.

[5] R. Fagin, J. Nievergelt, N. Pippenger, and H. Strong, *Extendible hashing: A fast access method for dynamic files*, ACM Trans. Database Systems, 4 (1979), pp. 315–344.

[6] W. Feller, *An Introduction to Probability Theory and Its Applications*, John Wiley, New York, 1988.

[7] P. Flajolet, *On the performance evaluation of extendible hashing and trie searching*, Acta Inform., 20 (1983), pp. 345–369.

[8] P. Flajolet, M. Régnier, and R. Sedgewick, *Some uses of the Mellin transform techniques on the analysis of algorithms*, in Combinatorial Algorithms on Words, NATO Adv. Sci. Inst. Ser. F: Comput. Systems Sci., 12 (1985), pp. 241–254.

[9] G. Gonnet and J. Munro, *The analysis of linear probing sort by the use of a new mathematical transform*, J. Algorithms, 5 (1984), pp. 451–470.

[10] P. Henrici, *Applied and Computational Complex Analysis*, John Wiley, New York, 1977.

[11] M. Hofri, *Probabilistic Analysis of Algorithms*, Springer-Verlag, Berlin, New York, 1987.

[12] P. Jacquet and M. Régnier, *Limiting distributions for trie parameters*, Lecture Notes in Comput. Sci., 214 (1986), pp. 196–210.

[13] P. Jacquet and W. Szpankowski, *Ultimate characterization of the burst response of an interval searching algorithm: A study of a functional equation*, SIAM J. Comput., 18 (1989), pp. 777–791.

[14] ———, *Analysis of digital tries with Markovian dependency*, IEEE Trans. Inform. Theory, 37 (1991), pp. 1470–1475.

[15] M. Kendall and A. Stuart, *The Advanced Theory of Statistics*, Vol. 1, 4th ed., Charles Griffin and Co. Ltd., London, 1977.

[16] P. Kirschenhofer and H. Prodinger, *Further results on digital search trees*, Theoret. Comput. Sci., 58 (1988), pp. 143–154.

[17] P. Kirschenhofer, H. Prodinger, and J. Schoissengeier, *Zur Auswertung gewisser Reihen mit Hilfe modularer Functionen*, in Zahlentheoretische Analysis 2, F. Hlawka, ed., Lecture Notes in Mathematics 1262, Springer, Berlin, 1987, pp. 108–110.

[18] P. Kirschenhofer, H. Prodinger, and W. Szpankowski, *On the balance property of PATRICIA tries: External path length viewpoint*, Theoret. Comput. Sci., 68 (1989), pp. 1–17.

[19] D. Knuth, *The Art of Computer Programming*, Vol. I, Addison-Wesley, Reading, MA, 1973.

[20] ———, *The Art of Computer Programming. Sorting and Searching*, Vol. III, Addison-Wesley, Reading, MA, 1973.

[21] B. Pittel, *Asymptotic growth of a class of random trees*, Ann. Probab., 18 (1985), pp. 414–427.

[22] ———, *Paths in a random digital tree: Limiting distributions*, Adv. Appl. Probab., 18 (1986), pp. 139–155.

[23] B. Pittel and H. Rubin, *How many random questions are necessary to identify n distinct objects?*, J. Combin. Theory Ser. A, 55 (1990), pp. 292–312.

[24] A. Renyi, *Probability Theory*, North-Holland, Amsterdam, 1970.

[25] W. Szpankowski, *Some results on V-ary asymmetric tries*, J. Algorithms, 9 (1988), pp. 224–244.

[26] ———, *Patricia tries again revisited*, J. Assoc. Comput. Mach., 37 (1991), pp. 691–711.

SIAM J. DISC. MATH.
Vol. 6, No. 2, pp. 214-229, May 1993

REPRESENTATIONS OF PLANAR GRAPHS*

GRAHAM R. BRIGHTWELL† AND EDWARD R. SCHEINERMAN‡

Abstract. This paper shows that every 3-connected planar graph G can be represented as a collection of circles, one circle representing each vertex and each face, so that, for each edge of G, the four circles representing the two endpoints and the two neighboring faces meet at a point, and furthermore the vertex-circles cross the face-circles at right angles. This extends a result of W. Thurston [*The Geometry and Topology of Three Manifolds*, unpublished] and, independently, Andreev. From this we deduce two corollaries: (1) The partial order formed by taking the vertices, edges, and bounded faces of G, ordered by inclusion, is a circle order; (2) One can represent G and its dual simultaneously in the plane with straight-line edges so that the edges of G cross the dual edges at right angles. This answers a question first asked by W. Tutte [*Proc. LMS*, 13 (3) (1963), pp. 743–768].

Key words. planar graphs, partially ordered sets, coin graphs

AMS(MOS) subject classifications. 05C10, 06A10

1. Introduction. Our aim in this paper is to prove various closely related results concerning representations of 3-connected planar graphs. The central result is that every such graph G can be represented in a natural way as a collection of circles in the plane. The existence of such a representation extends a result of Thurston. We use this representation of G to derive others: In particular, we solve a long-standing problem of Tutte by showing the existence of a simultaneous straight-line drawing of G and its dual, so that edges of G cross dual edges at right angles.

Our main result can also be thought of as a result about convex polytopes, and in this form it has been found independently by Pulleyblank and Rote [12].

Throughout this paper, G will be a 3-connected planar graph. In this case, the *faces* of G are combinatorially well-defined, up to the free choice of an outside face. We shall frequently blur the distinction between a planar graph and a plane map representing it.

Our starting point is the following result of Thurston [19], which has been independently discovered by Andreev.

THEOREM 1 (Thurston's coin-graph theorem). *Every planar graph G can be represented by a set of nonoverlapping circles in the plane, one circle for each vertex, so that two vertices are adjacent in G if and only if the corresponding circles are tangent.*

A graph that can be represented in this way is sometimes known as a *coin graph* (see Fig. 1). The converse of Theorem 1 is obvious, so we have that a graph is a coin graph if and only if it is planar.

We remark that the proof of Theorem given in [19] uses some deep results from the theory of orbifolds. In the course of this paper, we shall give a somewhat more elementary proof, which is based on an interpretation of Thurston's proof due to Lovász and communicated to us by Pulleyblank.

Note. Since preparing this paper, we have received a manuscript from Sachs [13], in which he sets out more of the history of the subject. It appears that the coin-graph theorem, Theorem 1, was first proved by Koebe [9] in 1935. It seems that Theorem 6

* Received by the editors February 25, 1991; accepted for publication (in revised form) June 5, 1992.

† Department of Statistical and Mathematical Sciences, London School of Economics and Political Science, Houghton Street, London WC2A 2AE, United Kingdom.

‡ Department of Mathematical Sciences, The Johns Hopkins University, Baltimore, Maryland 21218-2689.
This research was supported in part by Office of Naval Research contract N00014-85-K-0622.

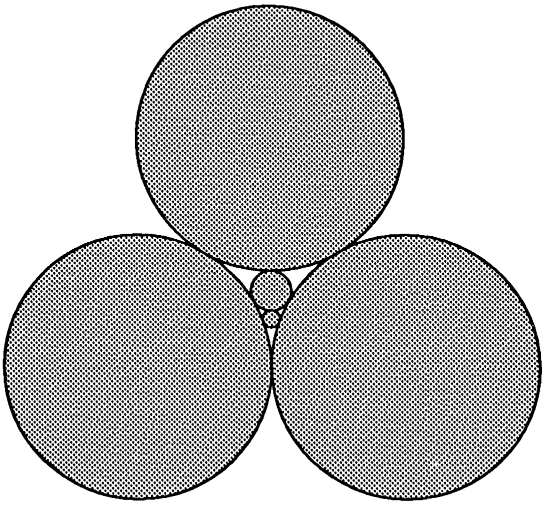


FIG. 1. *A representation of $K_5 - e$ as a coin graph.*

(or an equivalent version) was proved several times independently in 1990 and 1991. Schramm [17] has proved a more general version. See also [6].

Scheinerman [14] used Theorem 1 to prove a result (Theorem 4) about circle orders and planar graphs. A *circle order* is a partial order that can be represented as a set of disks in the plane, ordered by inclusion. Circle orders have been the subject of much study in the last few years—the major aim in the area has been to settle Conjecture 2 below. Here, as usual, the *dimension* of a partial order $(X, <)$ is the minimum number of linear orders on X whose intersection is exactly $(X, <)$.

CONJECTURE 2. *Every finite three-dimensional partial order is a circle order.*

Evidence has been produced both for and against Conjecture 2. At the present time, it seems that most researchers in the field believe that Conjecture 2 is false, and indeed that the poset $\{0, 1, 2, 3\}^3$, with the coordinatewise order, is a counterexample. However, no one has succeeded in proving that this poset is not a circle order, and so it is of interest to investigate other classes of three-dimensional orders to see if these consist of circle orders.

For further information on circle orders, the reader is referred to Hurlbert [8], Scheinerman and Wierman [15], Sidney, Sidney, and Urrutia [18], or Urrutia [21].

Given any graph G, the *incidence poset* P_G of G is defined by taking as elements the set of vertices and edges of G, ordered by inclusion, so $x < y$ in P_G if and only if y is an edge and x is one of its endpoints. Interest in P_G was stimulated by the following beautiful result of Schnyder [16].

THEOREM 3 (Schnyder). *A graph G is planar if and only if the dimension of P_G is at most* 3.

In the same spirit, Scheinerman [14] proved the following.

THEOREM 4 (Scheinerman). *A graph G is planar if and only if P_G is a circle order.*

In [14], the "only if" half of Theorem 4 is deduced from Theorem 1—since the ideas of that proof are central to what follows, we give the following brief sketch.

Let, then, G be a planar graph and take a collection of circles in the plane representing G as in Theorem 1. Now we form a representation of the dual of P_G as a circle order by taking the vertex-circles to represent the corresponding vertices and the single points where two neighboring circles touch to represent the indicated edge. (In case the reader is unhappy about taking single points rather than circles, we note that we can simply add 1 to the radius of each circle, thinking of a single point as a circle of radius 0; our new collection of circles will then have the same containment relations as the old.) Since the dual of a finite circle order is always a circle order (see [2] for a more elaborate discussion), this implies that P_G is a circle order.

A recent result of Brightwell and Trotter [3] extends Schnyder's theorem. For G a planar graph with a fixed plane drawing, let $P(G)$ be the poset formed by taking the vertices, edges, and (closed) faces of G, considered as subsets of the plane and ordered by inclusion, with the outside face removed. If G is 3-connected, then $P(G)$ does not depend on the plane drawing chosen, but only on the choice of outside face. For convenience, we state the theorem for the 3-connected case only: the result has been extended to cover general plane maps by Brightwell and Trotter [4].

THEOREM 5 (Brightwell–Trotter). *Let G be a 3-connected planar graph with a designated outside face. The dimension of $P(G)$ is exactly 3.*

A 3-connected planar graph G corresponds to a convex polytope in 3-space, and $P(G)$ to the *face lattice* of the polytope, i.e., the set of zero-, one-, and two-dimensional faces, ordered by inclusion, except that one face is omitted. It can be shown that the entire face lattice of a convex polytope always has dimension at least 4, and therefore, since adding one point to a poset increases the dimension by at most one, Theorem 5 implies the dimension of the face lattice is always exactly 4.

Hence, the class of posets $P(G)$, with G a 3-connected planar graph, forms a large collection of three-dimensional posets, and it is natural to ask whether these posets are all circle orders. We prove that this is the case, thus eliminating the posets $P(G)$ as possible counterexamples to Conjecture 2.

To apply the method of proof of Theorem 4 sketched above, we seek an extension of Thurston's coin-graph theorem to include the faces of our 3-connected planar graph. Since, in the derivation of Theorem 4, we considered the vertex-circles to have positive radius and the edge-circles to have zero radius, it seems natural to construct the face-circles to have "negative radius." To make sense of this, it helps to view circle orders rather differently, as we now discuss.

Consider the space $R^2 \times R$ and define a partial order on it by $(x,t) \leq (y,t')$ if and only if $|x-y| \leq t'-t$, where $|\cdot|$ denotes the Euclidean metric. Thus the set of points comparable to a given point (x,t) forms a cone in R^3. Another way of thinking of this is to consider the space as Minkowski spacetime, with x representing position and t time. Then, in units where the speed of light is 1, the order we give is the *causality order*, with $(x,t) \leq (y,t')$ if (y,t') is in the future light-cone of (x,t).

We claim that the finite suborders of this causality order are precisely the circle orders. To see this, consider any finite subset X of $R^2 \times R$. Take a reference plane H defined by $t = t_0$, where t_0 is less than the "time component" of every point of X. Now represent each point $(x,t) \in X$ by the set of points in H, which are below (x,t) in the order. Evidently, this set is a closed disk with center (x,t_0) and radius $t-t_0$, and we see that $(x,t) \leq (y,t')$ if and only if the disk representing (x,t) is contained in that representing (y,t'). This procedure can clearly be inverted, so a finite partial order is a suborder of the causality order precisely when it is a circle order. (This is not true for infinite suborders—see [2].)

For more information on how circle orders and their higher-dimensional analogues, sphere orders, relate to the study of spacetime, see Brightwell and Gregory [1], Brightwell and Winkler [5], and Meyer [10], [11].

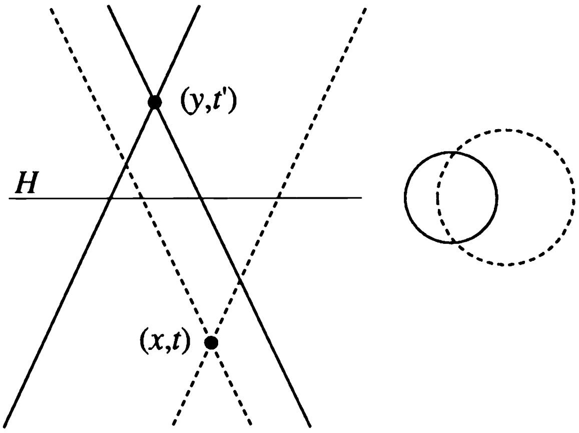


FIG. 2. *Points above and below H related in the causality order.*

Returning to our problem of how to represent the faces of G as "circles of negative radius," the previous discussion suggests that we should represent vertices by points above H, edges by points in H, and faces by points below H. Converting back to circles, a point (x, t) to the past of H can be identified with the set of points in H, which are *above* (x, t). Now it is immediate that a point (x, t) below H is less than a point (y, t') above H in the causality order if and only if the two corresponding circles intersect (see Fig. 2).

Our proposed extension of Theorem 1 would then assert, for every 3-connected planar graph G, the existence of a set of circles in the plane, representing G in the following manner. The set of circles is in 1–1 correspondence with the set of vertices and faces of G. The circles corresponding to vertices are termed *vertex-circles*, and those corresponding to faces are the *face-circles*. The circles satisfy the following properties.

(P1) No two vertex-circles cross, and no two face-circles cross.

(P2) Corresponding to every edge e of G, there is a point in the plane where four circles meet, namely, those corresponding to the two endpoints of e and the two faces bounded by e. This point will be called an *edge-point* and is to be thought of as representing e.

(P3) A face-circle and a vertex-circle intersect only when the corresponding vertex is on the boundary of the corresponding face.

(P4) The region bounded by the circle corresponding to the outside face contains all other face-circles. With this exception, none of the disks bounded by one of the circles contains another of the circles.

Note that our representation includes a circle for the outside face of G. This is particularly important when we think of the circles as raised to the sphere, but it also frees us from any problems associated with the choice of the outside face in the plane representation of G. We shall discuss this further a little later.

It is remarkable that, for every 3-connected planar graph G, there is such a representation. Even more remarkably, we can insist on one more property.

(P5) At each edge-point, the two vertex-circles cross the two face-circles at right angles.

Let us now state our main theorem.

THEOREM 6. *Let G be a 3-connected planar graph, with a designated outside face. There is a collection of circles in the plane, one circle representing each vertex and each face*

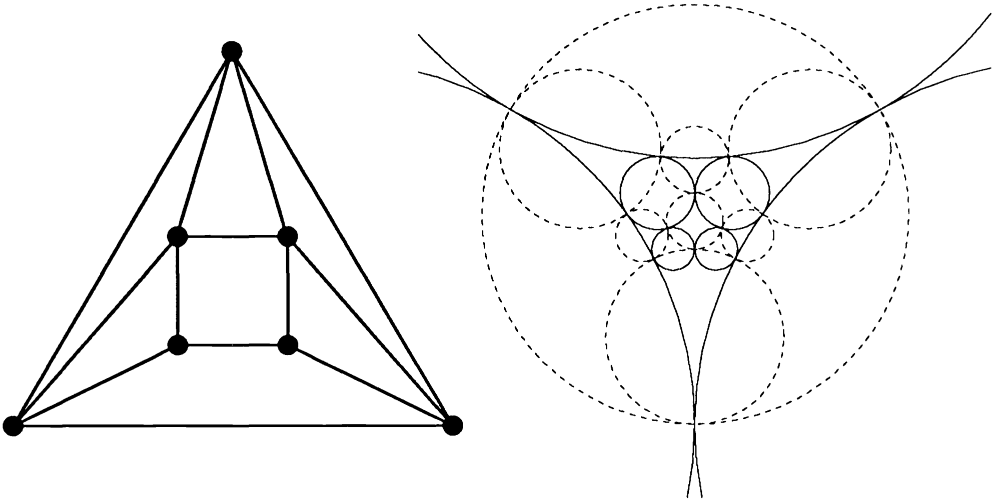


FIG. 3. *A planar graph G, and a representation by circles satisfying* (P1)–(P5).

of G, satisfying properties (P1)–(P5). *Furthermore, this collection is unique up to linear fractional transformations and reflections of the plane.*

We call such a representation of G a *circle representation*.

See Fig. 3 for an example showing a circle representation of the simple planar graph G. The next section will be devoted to a proof of Theorem 6. Note that Theorem 6 implies Theorem 1, since the vertex-circles by themselves satisfy the requirements for a coin-graph representation.

Given Theorem 6 and the idea of a face-circle having negative radius, it is a simple matter to deduce that $P(G)$ is a circle order.

THEOREM 7. *If G is a* 3*-connected planar graph with a designated outside face, then $P(G)$ is a circle order.*

Proof. Given a circle representation of G, we identify the dual of $P(G)$ with a suborder of the causality order as in our previous discussion. Namely, for a vertex-circle with center x and radius t, we take the point (x, t); for a face-circle with center y and radius t', we take the point $(y, -t')$, and for an edge-point at z, we take the point $(z, 0)$.

Thus the dual of $P(G)$ is a suborder of the causality order. Hence it is a circle order, and therefore, so is $P(G)$ itself. □

Instead of the above proof, we can proceed more directly by a suitable interpretation of the rule: increase all radii by the same large constant.

The fact that we can also insist on property (P5) points the way to our next theorem. This result was first conjectured by Tutte [20], but has also occurred independently to others and has recently been popularized by Sachs [13].

THEOREM 8. *Let G be a* 3*-connected planar graph, and G^* its planar dual. It is possible to draw G and G^* simultaneously in the plane with straight-line edges so that the edges of G cross the edges of G^* at right angles.*

Proof. Take a circle representation of G. Place each vertex of G at the center of the corresponding vertex-circle and each vertex of G^* at the center of the corresponding face-circle. Now put in straight-line edges between adjacent vertices of G and of G^*. If any two edges of G cross, then either some pair of vertex-circles cross, or one is contained in another, contradicting either (P1) or (P4). So we have straight-line representations

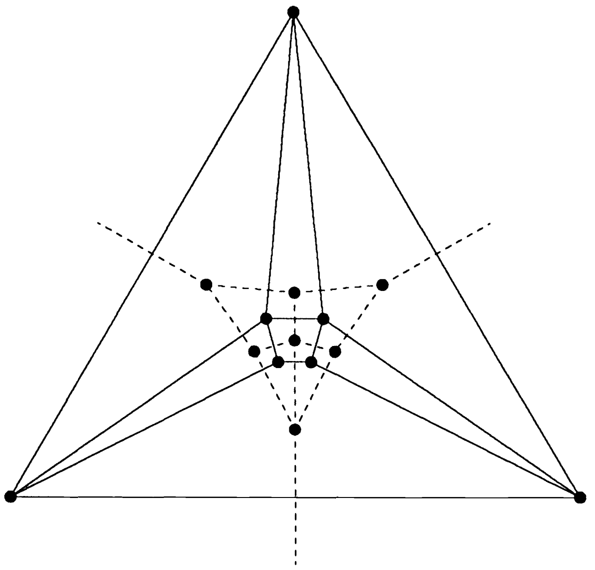


FIG. 4. *The graph G of* Fig. 3 *and its dual, simultaneously drawn with straight-line edges, so that edges cross dual edges at right angles.*

of G and G^*. Also, each edge in the straight-line representation of G goes through the corresponding edge-point in the circle representation, at right angles to the vertex-circles, and similarly for G^*. Thus, by (P5), the edges of G cross the dual edges at right angles, as claimed. □

This proof avoids mention of the outside face F, but it is clear that we can incorporate it by running edges to infinity from all neighboring vertices of F in G^*. These edges can cross their dual edges at right angles: they will then all project to a common point, namely the center of the circle representing F. It is easy to see that when G is a triangulation, we cannot have these dual edges run to a common finite point.

Figure 4 shows the straight-line representation of the graph G and its dual derived from the circle representation shown in Fig. 3.

Further motivation for this treatment of the outside face is provided by considering our circles as living on the sphere. Indeed, Theorem 6 translates immediately to the analogous result on the sphere, as stereographic projection from the plane to the sphere maps circles to circles. Viewed this way, the circle representing the outside face should be thought of as bounding its *exterior* on the plane. Mapped up to the sphere, this region is simply a cap containing the point at infinity, requiring no exceptional treatment.

Let us then define $P^+(G)$ to be the containment order of vertices, edges, and *all* faces of a planar 3-connected graph G. It follows that $P^+(G)$ is a *cap order*, i.e., each element of $P^+(G)$ can be assigned a "cap" on the surface of the sphere in an order-preserving fashion. We might hope that $P^+(G)$ is actually a circle order (eliminating special treatment of the exterior face), but we note that $P^+(K_4)$ is exactly the middle three levels of the Boolean algebra $2^{[4]}$ and is therefore, as shown by Jamison, not a circle order. For a proof of this, see Brightwell and Winkler [5].

We state our theorem for the plane case partly because our proof is a little cleaner in that case and partly because that form of the result is closer to its applications in Theorems 7 and 8.

2. Proof of Theorem 6. We will produce the required circles rather indirectly. To motivate the first few steps, imagine that we have circles in the plane as desired. If we draw lines between the centers of tangent vertex-circles and also between the centers of tangent face-circles (the exterior-face-circle being handled as mentioned at the end of §1), the polygon defined by the outside vertices is divided into kite-shaped regions: in fact, these kites can be seen in Fig. 4, as the lines we draw are those we put in for the straight-line representation of G and G^*. Our approach is to construct the kites, from which the circles can be recovered immediately.

Each of the kites is defined by its two side-lengths corresponding to the radii of the two circles it intersects. Our goal is to specify radii for all the circles so that the kites defined do tile the plane in the desired manner. Circles α and β give rise to a kite $K_{\alpha\beta}$ only if one of α and β is a face circle and the other is a vertex-circle corresponding to a vertex on the face. Thus it is natural to consider the graph of this incidence relation.

So, let G be a 3-connected plane map and form the *vertex-face incidence graph* $\hat{G}$ by taking as vertices the vertices and faces of G (including the outside face) and putting in an edge when a vertex of G is incident with a face of G. Thus $\hat{G}$ is a bipartite planar graph, and, in fact, every face is a quadrilateral corresponding in an obvious way to an edge of G.

LEMMA 9. *Let G be a 3-connected plane map, and let $\hat{G}$ be its vertex-face incidence graph.*

(i) *If $\hat{G}$ has v vertices, then it has $2v - 4$ edges, and so has a vertex x of degree 3.*

(ii) *If S is any subset of $V(\hat{G})$, which* (a) *is nonempty and* (b) *does not contain x or any of its neighbors, then S spans at most $2|S| - 2$ edges of $\hat{G}$ and is incident with at least $2|S| + 1$ edges of $\hat{G}$.*

Proof. Part (i) follows immediately from Euler's formula.

For part (ii), we take a fixed planar embedding of $\hat{G}$ and let $\hat{G}(S)$ be the map defined by taking just the vertices of S and the edges spanned by S. Note that $\hat{G}(S)$ is also a bipartite plane map. If $\hat{G}(S)$ has at least two edges, then, since the graph is bipartite, every face is incident with at least four edges (counting twice if "both sides" of the edge are on the face), and so an application of Euler's formula shows that the number of edges in $\hat{G}(S)$ is at most $2|S| - 4$. The cases where $\hat{G}(S)$ has one edge or no edges are trivial.

For the second assertion of part (ii), set $\overline{S} = V(\hat{G}) \setminus S$. If there are as few as $2|S|$ edges incident with S, then $\hat{G}(\overline{S})$ has $2|\overline{S}| - 4$ edges. By hypothesis, $\hat{G}(\overline{S})$ certainly contains at least two edges, so this means that every face of $\hat{G}(\overline{S})$ is a quadrilateral. One such face contains some element of S, and the vertices of $\hat{G}$ on that face separate $\hat{G}$. (Note that $\overline{S}$ contains some vertex not on this face.) These vertices correspond to two vertices and two faces whose removal separates G, contradicting the assumption that G is 3-connected. (When $|S| > 2$, the bound in (ii) can be improved to $2|S| - 4$.) □

It turns out to be convenient to assume that the outside face of G is a triangle. By Lemma 9(i), we have that the map G contains either a triangular face or a vertex of degree 3. In the former case, we draw G so that some triangle is the exterior face: in the latter, we draw G^* with a triangle as the exterior face. We will construct our circles for the map thus chosen and convert back to the desired circles by means of a linear fractional transformation of the plane at the end. (Alternatively, we can think of the circles as being on the sphere.)

Let us return to our hypothetical kites tiling the plane. What conditions must the radii of the defining circles satisfy so that the kites fit together in the appropriate manner? One obvious condition is that, if α is any vertex or face of G, the kites constructed using

α must fit together around the center of the corresponding circle. It turns out that this necessary local condition is also sufficient for the construction to go through globally.

Let us proceed a little more formally. We will take v to denote the number of vertices of $\hat{G}$, counting the outside face, and use subscripts $\alpha, \beta, \ldots$ to denote these vertices. To each vertex α of $\hat{G}$, except that corresponding to the outside face F_0 of G, we associate a variable r_α, which is to be thought of as the radius of the circle corresponding to α. For the three outside vertices a, b, c, the radii r_a, r_b, r_c are fixed at 1 throughout. All other r_α are to be regarded as variables. Let r denote the vector $(r_\alpha)_{\alpha \in \hat{G}-F_0}$.

We next define a function $\theta : (R^+)^{v-1} \to (R^+)^{v-4}$ by

$$\theta(r) = (\theta_\alpha(r))_{\alpha \in \hat{G}-\{F_0,a,b,c\}},$$

where

$$\theta_\alpha(r) = \sum_{\beta \perp \alpha} \tan^{-1}\left(\frac{r_\beta}{r_\alpha}\right),$$

and this sum is over all neighbors of α in $\hat{G}$. For the kite $K_{\alpha\beta}$ defined by α and a neighbor β, the angle between the two sides of length r_α is given by $2\tan^{-1}(r_\beta/r_\alpha)$, so $\theta_\alpha(r)$ represents half the total angle spanned by all the kites meeting at the center of the circle corresponding to α. If the kites are to fit at this point, we must have $\theta_\alpha(r) = \pi$. Thus we must find a radius vector r to satisfy the equation $\theta(r) = \underline{\pi}$, where $\underline{\pi}$ is the vector with every entry equal to π.

The proof now splits into two parts: first, finding a solution to the above equation; and second, showing that a solution will suffice to enable us to construct the kites and hence the circles so as to satisfy all the conditions (P1)–(P5). For the moment, we concentrate on the first of these tasks.

We know of no way to produce an explicit solution to the equation $\theta(r) = \underline{\pi}$. The best we can do is describe an iterative process that converges to a solution. As set out, this process requires two stages, although, as we shall see at the end of the proof, the second stage is always superfluous. We should also remark that there are several other fairly obvious iterative schemes, both discrete and continuous, which we expect to converge to a solution as well. Indeed, our experiments suggest that the scheme we set out below converges rather more slowly than some of the alternatives.

To begin with, note that the function θ behaves in a uniform manner when just one of the coordinates of r is changed. Indeed, increasing r_α while keeping all the other radii fixed decreases $\theta_\alpha(r)$, but increases $\theta_\beta(r)$ for β adjacent to α in $\hat{G}$.

The process is loosely as follows. We produce a sequence of vectors $r^i = (r^i_\alpha, r^i_\beta, \ldots)$, starting with any strictly positive vector r^0. If any coordinate $\theta_\alpha(r^i)$ is less than π, we find r^{i+1}_α by decreasing the variable r_α until $\theta_\alpha(r) = \pi$. We shall show that iterating this process and taking the limit results in a set of values for r_α such that all the r_α are still strictly positive and all the $\theta_\alpha(r)$ are at least π. In the second stage, if any $\theta_\alpha(r)$ is greater than π, we increase r_α to the appropriate level. We shall prove that all the r_α remain bounded during this process and that the process converges to a solution of the equation.

Let us consider the first phase of this process. We define a sequence of vectors $r^i = (r^i_\alpha, r^i_\beta, \ldots)$ inductively as follows. We initialize by setting $r^0 = \mathbf{1}$. (As we have already indicated, the particular values chosen are not important.) In general, some of the coordinates $\theta_\alpha(r^0)$ will be greater than π, and others less than π. If we ever come to a vector r^i satisfying $\theta(r^i) \geq \underline{\pi}$, we move on to the second stage. Suppose that, for our current vector r^i, some of the coordinates $\theta_\alpha(r^i)$ are less than π. We then define r^{i+1} according to the following rules. If $\theta_\alpha(r^i) \geq \pi$, then set $r^{i+1}_\alpha = r^i_\alpha$. If $\theta_\alpha(r^i) < \pi$, then set r^{i+1}_α equal to the unique solution of the equation

$$\sum_{\beta \perp \alpha} \tan^{-1}\left(\frac{r^i_\beta}{r^{i+1}_\alpha}\right) = \pi.$$

Note that no coordinate r^i_α is ever reduced to 0 by this process. Thus either the process terminates with a vector r^k with all coordinates positive, satisfying $\theta(r^k) \geq \underline{\pi}$, or the process continues indefinitely. In the latter case, each sequence $(r^j_\alpha)^\infty_{j=0}$ is a nonincreasing sequence of positive real numbers, so tends to some nonnegative limit r^∞_α. We shall show that in fact each limit r^∞_α is nonzero. It follows immediately from the continuity of θ that $\theta(r^\infty) \geq \underline{\pi}$, and we move on to the second stage of the process, taking r^∞ as the initial vector.

Suppose then that some of the sequences (r^j_α) tend to 0 as $j \to \infty$, and let $S = \{\alpha : \lim_{j\to\infty} r^j_\alpha = 0\}$. Note that, for every $\alpha \in S$, $\theta_\alpha(r^j)$ is less than π for some j, since otherwise, r_α would never be decreased below its initial value. Also, once the sequence $(\theta_\alpha(r^j))$ falls below π, it remains at most π thereafter, since r_α is only decreased far enough to allow $\theta_\alpha(r^{i+1}) = \pi$, and decreases in other r_β can never increase $\theta_\alpha(r)$. Thus we have

$$\sum_{\alpha \in S} \theta_\alpha(r^j) \leq \pi|S|$$

for all sufficiently large j. On the other hand, we have

$$\sum_{\alpha \in S} \theta_\alpha(r^j) = \sum_{\alpha \in S} \sum_{\beta \perp \alpha} \tan^{-1}\left(\frac{r^j_\beta}{r^j_\alpha}\right).$$

Every edge $\alpha\beta$ of $\hat{G}$ with both endpoints in S contributes exactly $\pi/2$ to this sum for every j, since $\tan^{-1}(r_\beta/r_\alpha) + \tan^{-1}(r_\alpha/r_\beta) \equiv \pi/2$. Furthermore, if S contains α but not β, then $r^j_\beta/r^j_\alpha \to \infty$ as $j \to \infty$, so $\lim_{j\to\infty} \tan^{-1}(r_\beta/r_\alpha) = \pi/2$. Thus the contribution to this sum of every edge with at least one endpoint in S tends to $\pi/2$. By Lemma 9(ii), however, there are at least $2|S| + 1$ such edges, so the limit of the sum as $j \to \infty$ is at least $\pi|S| + \pi/2$, a contradiction.

Thus, either taking some r^k or taking the limit r^∞, we arrive at a strictly positive vector s^0 satisfying $\theta(s^0) \geq \underline{\pi}$.

In the second phase, we define a sequence of vectors $s^i = (s^i_\alpha, s^i_\beta, \ldots)$ inductively, starting from s^0 as follows.

If $\theta_\alpha(s^i) = \pi$, set $s^{i+1}_\alpha = s^i_\alpha$. If $\theta_\alpha(s^i) > \pi$, set s^{i+1}_α equal to the unique solution of

$$\sum_{\beta \perp \alpha} \tan^{-1}\left(\frac{s^i_\beta}{s^{i+1}_\alpha}\right) = \pi.$$

Note that no $\theta_\alpha(s^j)$ ever drops under π as we iterate this process and that the sequences $(s^j_\alpha)^\infty_{j=1}$ are nondecreasing.

If all these sequences stay bounded as $j \to \infty$, then they all tend to limits s^∞_α. Again by the continuity of θ, we will then have $\theta(s^\infty) = \underline{\pi}$ as required.

Suppose then that some sequence (s^j_α) escapes to infinity, and let S be the set of α such that $s^j_\alpha \to \infty$ as $j \to \infty$. Consider the sum $\sum_{\alpha\in S} \theta_\alpha(s^j)$. Each edge of $\hat{G}$

with both endpoints α and β in S contributes $\pi/2$ to this sum for each j, again since $\tan^{-1}(s_\alpha/s_\beta) + \tan^{-1}(s_\beta/s_\alpha) \equiv \pi/2$. However, the sum of the contributions from other edges tends to 0 as $j \to \infty$, since $\tan^{-1}(s^j_\beta/s^j_\alpha) \to 0$ as $s^j_\alpha \to \infty$ while s^j_β remains bounded (i.e., whenever α is in S, but β is not).

Let $\hat{G}(S)$ be the subgraph of $\hat{G}$ spanned by the vertices of S and suppose that $\hat{G}(S)$ has n vertices and e edges. From the above, we have that

$$\sum_{\alpha \in S} \theta_\alpha(s^j) \to e\pi/2$$

as $j \to \infty$ while, for every j, each individual $\theta_\alpha(s^j)$ is at least π, so the above sum is at least $n\pi$. Thus we have $e \geq 2n$, but this contradicts Lemma 9(ii). Hence S is empty, and the process indeed converges to give a vector s^∞ satisfying $\theta(s^\infty) = \underline{\pi}$.

We turn for a moment to the question of uniqueness. Suppose that there are two solutions r and r' to the equation $\theta(r) = \underline{\pi}$, with $r_a = r'_a$ for each outside vertex a. Let $S = \{\alpha : r_\alpha > r'_\alpha\}$, and suppose that S is nonempty. Then we have

$$\begin{aligned}
\sum_{\alpha \in S} \theta_\alpha(r) &= \sum_{\alpha \in S} \sum_{\beta \perp \alpha} \tan^{-1}(r_\beta/r_\alpha) \\
&= \sum_{\alpha, \beta \in S} \pi/2 + \sum_{\alpha \in S, \beta \notin S} \tan^{-1}(r_\beta/r_\alpha) \\
&< \sum_{\alpha, \beta \in S} \pi/2 + \sum_{\alpha \in S, \beta \notin S} \tan^{-1}(r'_\beta/r'_\alpha) \\
&= \sum_{\alpha \in S} \theta_\alpha(r'),
\end{aligned}$$

but both the initial and the final sums are equal to $|S|\,\pi$, a contradiction. Thus S is empty, and by symmetry there is no α for which $r_\alpha < r'_\alpha$. Therefore, the two solutions are identical.

From now on, let $r = (r_\alpha, r_\beta, \ldots)$ be the unique solution of $\theta(r) = \underline{\pi}$ with $r_a = 1$ for each outside vertex a. The positive real number r_α will then be taken as the radius of the circle representing the vertex α of $\hat{G}$.

In some sense, given these radii, there is very little left to do: We form our kites, lay them down in the plane, draw in the circles, and note that they have the desired properties (P1)–(P5). Indeed, the remainder of the proof is basically checking, but there are still a lot of details to look after.

The next step is to give a more formal description of the kites and the manner of laying them down in the plane. For each edge $\alpha\beta$ of $\hat{G}$, we specify a kite-shaped quadrilateral $K_{\alpha\beta}$ as follows. The kite has two sides, the *α-sides*, of length r_α, and the other two sides, the *β-sides*, of length r_β. The α-sides meet the β-sides at right angles. We note once more that the angle where the two α-sides meet is $2\tan^{-1}(r_\beta/r_\alpha)$ and similarly for the β-sides (see Fig. 5).

Consider the plane map G. We find a collection $(C_i)_{i=1}^k$ of simple cycles in G with the property that C_1 bounds a single face F_1, each subsequent cycle C_{i+1} bounds the faces $F_1, \ldots, F_i$ together with just one more face F_{i+1}, and the final cycle is the exterior triangle. Our order of placing the kites onto the plane will be so as to form the faces F_i in turn.

The plane representation of the original graph G contains the information of which kites should abut. For instance, if α is a face, and β and γ are consecutive vertices on its

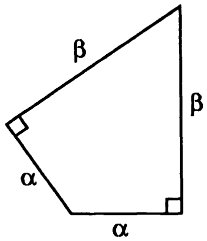


FIG. 5. *A kite $K_{\alpha\beta}$.*

boundary, reading clockwise, then the kites $K_{\alpha\beta}$ and $K_{\alpha\gamma}$ are to meet, with the common side being an α-side, and $K_{\alpha\beta}$ being to the left of $K_{\alpha\gamma}$ as viewed from the center of the α-circle.

The various kites $K_{F_1\beta}$ involving the face F_1 fit together in this manner to tile a convex polygon, as guaranteed by the condition $\theta_{F_1}(r) = \pi$. In fact, the polygon admits an inscribed circle of radius r_{F_1}, which will of course be the circle representing F_1. We place this polygon on the plane arbitrarily. If v is a vertex on F_1, the point where the v-sides of $K_{F_1 v}$ meet will be the center of the circle representing v: for the moment, we shall think of this point as itself representing v. The other two corners of the kite $K_{F_1 v}$ will be the two edge-points corresponding to the edges bounding F_1 incident with v.

Once we have placed the kites involving faces $F_1, \ldots, F_i$, we form the polygon corresponding to the face F_{i+1}, just as we did for F_1. We shall place this in the plane according to the map G, so that all the kites meet in the prescribed manner. We must check that the requirements do not conflict.

Consider the boundary between F_{i+1} and those faces already dealt with. Due to the manner of choosing F_{i+1}, this boundary consists of just one connected section of the exterior of each region. Let $x_1, x_2, \ldots$ be the vertices on this boundary, reading clockwise around F_{i+1}. Some points of the plane have already been chosen to represent the x_i, and we have to check that these choices do not prevent us from placing the polygon corresponding to the new face.

The distance between the points representing x_i and x_{i+1} is $r_{x_i} + r_{x_{i+1}}$, both in the plane and in the new polygon. Also, if x_i, x_{i+1} and x_{i+2} are all in the boundary, then all the kites involving x_{i+1} are either already on the plane or appearing in the polygon, so the angle around x_{i+1} in the plane is exactly 2π minus the angle in the new polygon. Hence, the two boundaries fit snugly together.

Also, the region occupied by the tiles has corners only at the points representing vertices on the bounding cycle C_i. Hence, when all the tiles are in place, there are only three corners to the region tiled corresponding to the external vertices a, b, c, and each of the three sides has length $r_a + r_b = 2$.

Thus, the kites tile an equilateral triangle, as required. Once we have the kites drawn in the plane, we construct the vertex- and face-circles by inscribing them into the region consisting of all the kites involved with the appropriate vertex/face. The outside face-circle is inscribed in the bounding equilateral triangle, so it passes through the relevant edge-points at the midpoint of each side of the triangle.

It remains to check that this system of circles satisfies properties (P1)–(P5).

(P1) The vertex-circles are inscribed into mutually disjoint polygons, so no two can cross. The same is true for face-circles, except that we have to be a little careful with the outside face-circle: we defer consideration of this for a while.

(P2) If α is a vertex or face incident with an edge e of G, then by construction the edge-point representing e is at distance r_α from the center of the circle representing α.

(P5) Let e be an edge of G with endpoints x and y, separating faces F_1 and F_2. The straight line between the center of the circle representing x and the point representing e follows the common boundary between the kites K_{xF_1} and K_{xF_2}. The circle representing x passes through the point representing e at right-angles to this line, and similarly for the other circles through the edge-point. Property (P5) now follows from the fact that the angle between the α-sides and β-sides of each kite is a right-angle.

The outside face-circle again needs special treatment, but here all we need to note is that it passes through the three edge-points on the bounding equilateral triangle, and is tangent to the triangle at those points.

(P3) If a vertex and a face of G are not incident, they do not share a kite and therefore cannot intersect. Conversely, an incident vertex-face pair shares a kite, and clearly the two representing circles intersect at the two edge-points at the right-angled corners of that kite.

(P4) It is obvious that, with the exception of the outside face-circle, no representing circle can bound another. We must now show that the outside face-circle bounds all other face-circles—note that this includes the missing assertion from (P1).

The outside face-circle lies in the union of those kites involving one of the outside vertices, so any face-circle that is not bounded by the outside face-circle certainly crosses one of the vertex-circles corresponding to an outside vertex. So any offending face-circle crosses an outside vertex-circle at right angles. Also, both crossings occur inside the bounding equilateral triangle. This implies that such a face-circle does after all lie inside the circle inscribed into the triangle, and the two circles can only touch if they do so at one of the three external edge-points.

Incidentally, note that no circle can have radius larger than 1, which shows that the second stage of our convergence process is indeed unnecessary.

We have now proved the result in the case where the outside face of G is a triangle. If this is not the case, then we have been working with a different planar representation of G or G^*. Our approach now is to take the point of the plane corresponding to the center of the face (or vertex) that we wish to be the outside face and apply any linear fractional transformation of the plane that maps this point to infinity. Our circles are mapped to circles, and the right-angle crossings between circles are preserved, so we have a circle representation corresponding to the required plane map.

Finally, we return to the question of uniqueness. We have already seen that, if the radii of the three exterior vertex-circles are given, then all the radii are determined. It is evident that, given the radii, once the first two kites are placed in the plane, the positions of all others are determined. Furthermore, the only choice for the position of the second kite is that of placing it to the right or left of the first. So, given the radii, the circles are determined up to an isometry of the plane.

Now, if we have a circle representation and we take a linear fractional transformation of the plane, we obtain another circle representation, possibly with a different "outside face-circle." Also, given three pairwise tangent circles of radius 1 and three positive reals r_a, r_b, r_c, there is a linear fractional transformation that maps the three circles to circles of radius r_a, r_b, and r_c. (To see this, consider the three points of tangency of the original circles and three points of tangency for suitable "target" circles. Take a linear fractional transformation taking the first three points to the second three. The image circles must now have the correct radii.)

Therefore, all circle representations can be obtained from the one we find by a combination of linear fractional transformation and plane isometry. Since a sense-preserving isometry is itself a linear fractional transformation, we have the desired result, namely that the circle representation is unique up to linear fractional transformations and reflections.

This completes the proof of the result. □

As a final remark in this section, let us note that the conditions of planarity and 3-connectedness are necessary for the existence of a circle representation of a graph G. Planarity is obvious, since we obtain from the circle representation a planar representation of G. It remains to be shown that, if G is not 3-connected, then there is no circle representation.

Suppose then that G is a planar, non-3-connected graph with a circle representation. The cases where G is disconnected, or has a cutvertex, are easy to rule out. In the remaining cases, there is a subset S of $\hat{G}$, not including the outside face, which is incident with just $2|S|$ edges of $\hat{G}$. For $\alpha \in \hat{G}$, let r_α be the radius of the circle representing α, and define the functions θ_α as in the proof of Theorem 6 above. Then, as usual, we have

$$\pi\,|S| = \sum_{\alpha\in S} \theta_\alpha(r) = \sum_{\alpha,\beta\in S} \pi/2 + \sum_{\alpha\in S,\beta\notin S} \tan^{-1}\left(\frac{r_\beta}{r_\alpha}\right).$$

The right-hand side is at most $(\pi/2)\cdot(2|S|) = \pi\,|S|$, with strict inequality unless $r_\alpha = 0$ for every $\alpha \in S$. This is not possible, so we have a contradiction.

3. Integral representations. The uniqueness of the circle representation up to linear fractional transformations of the plane suggests that one could investigate the set of radii of the representing circles. In particular, is it always possible to arrange for the circles to have integer radii? This would imply that every planar graph possesses a straight-line embedding in the plane all of whose edge lengths are integers. Whether or not such an integer-length embedding exists is an open question [7]. Since the representation is a drawing with many circles, it is also natural to ask if such a representation is *constructible*, i.e., can be drawn with the classical construction tools: straight edge and compass.

In this section, we show that some planar graphs admit no coin-graph representation (and hence certainly no circle representation) with integer, or even constructible radii.

Each point (a, b) in the plane is naturally associated with a complex number $a+bi$. It is well known that a point in the plane can be constructed with straight edge and compass if and only if the number $a + bi$ lies in an iterated quadratic extension of the rationals; i.e., $a + bi$ can be computed from integers using only finitely many applications of the operations $+$, $-$, $\times$, $\div$, and $\sqrt{}$.

Call a planar graph G *constructible* if it admits a coin-graph representation all of whose centers and radii are constructible.

We shall give an example of a planar graph that is not constructible. Let the *bipyramid graph* B_n be the graph consisting of an n-cycle $v_1v_2\dots v_nv_1$ together with two additional vertices u and w: u and w are not adjacent, but each is adjacent to every v_i. Clearly, B_n is a planar graph: indeed, it is a *triangulation* as each of its faces (including the exterior face) is a triangle. We shall prove that B_{28} is not constructible.

Note first that a coin-graph representation of a triangulation can be extended to a circle representation by taking, for each face, the circle inscribed into the triangle defined by the centers of the three incident vertex-circles. Thus the coin-graph representation of a triangulation is unique up to linear fractional transformations and reflections of the plane.

Next, we give some lemmas concerning constructibility of graphs and circles.

LEMMA 10. *For a triangulation G, the following are equivalent*:

(i) *G is constructible*,

(ii) *G's radii are constructible*,

(iii) *G's centers are constructible.*

Proof. It is enough to show the equivalence of (ii) and (iii). First, we suppose that G's radii are constructible and show that we can choose constructible centers for the representing circles. Without loss of generality, we can put the center for one vertex, say v_1, at $z_1 = 0$ and the center of one of its neighbors, say v_2, at $z_2 = r_1 + r_2$. We can build up G by triangles, finding the center of a circle based on the its radius and the centers and radii of two of its adjacent neighbors. Assuming that v_a, v_b, and v_c form a triangular face of G and that r_a, r_b, r_c, z_a, z_b are constructible, we know that the center z_c must be at one of the intersection points of the circle with radius $r_a + r_c$ centered at z_a and the circle with radius $r_b + r_c$ centered at z_b. Both of these locations are constructible; hence z_c is constructible.

Conversely, suppose that the centers z_i are constructible. To construct the radii, note that any v_a has adjacent neighbors v_b and v_c. First, construct the point x, which is the intersection of the angle bisectors of triangle $z_a z_b z_c$; this is the center of the inscribed circle of the triangle. Next, construct a perpendicular from x to line segment $z_a z_b$, which meets the segment in a point y. Finally, put $r_a = |z_a - y|$, which is the correct (and constructible) radius for v_a. This completes the proof. □

LEMMA 11. *Let C be a circle. There exist three constructible points on C if and only if the center and radius of C are constructible.*

Proof. Suppose that the three constructible points are p_1, p_2, and p_3. Construct the perpendicular bisectors of the three line segments determined by these points. This gives the center of the circumscribing circle, C. Thus C's center and radius are constructible. Conversely, suppose that C's center z and radius r are constructible. Then $z + r$, $z - r$, and $z + ir$ are constructible points on C. □

LEMMA 12. *If z_1, z_2, z_3 are three distinct constructible points and if z_1', z_2', z_3' are also three distinct constructible points, then there exist constructible a, b, c, d so that $\mu(z) = (az + b)/(cz + d)$ satisfies $\mu(z_i) = z_i'$ for $i = 1, 2, 3$.*

(We call such a μ a *constructible* linear fractional transformation.)

Proof. Recall that the cross ratio

$$[x_1, x_2, x_3, x_4] = \frac{(x_1 - x_3)(x_2 - x_4)}{(x_1 - x_2)(x_3 - x_4)}$$

is invariant under linear fractional transformations, i.e.,

$$[x_1, x_2, x_3, x_4] = [\mu(x_1), \mu(x_2), \mu(x_3), \mu(x_4)]$$

for any complex x_1, x_2, x_3, x_4 (including ∞) and any linear fractional transformation μ. Thus, given any z, we can compute $\mu(z)$ uniquely by solving the equation

$$[z_1, z_2, z_3, z] = [z_1', z_2', z_3', z']$$

for z'; indeed, z' is a rational combination of the z_i's, the z_i''s, and z. Hence if z is constructible, then $z' = \mu(z)$ is also constructible.

We know there is a unique $\mu(x) = (ax + b)/(cx + d)$ with $ad - bc \neq 0$, which takes z_i to z_i'; we need to show that a, b, c, and d can be chosen to be constructible.

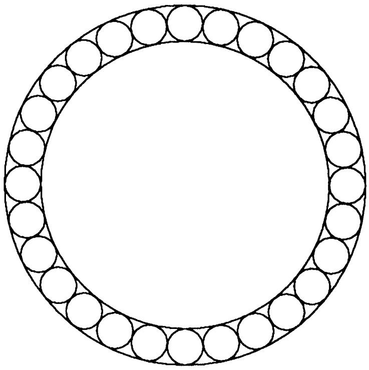


FIG. 6. *A representation of B_{28}.*

Consider $\mu(0)$. In the case where $\mu(0) = \infty$, then we must have $d = 0$ and, without loss of generality, $b = 1$. Otherwise, ($\mu(0) \neq \infty$), and we can take $d = 1$ and $b = \mu(0)$, both of which are constructible. Thus, we know we can take b and d constructible.

Next, consider $\mu(1)$. If $\mu(1) = \infty$, we have that $c = -d = -1$ (since $\mu(0) \neq \infty$). In this case, $\mu(-1) = (b-a)/2$, and, since b is constructible, so is a. Hence a and c are constructible.

Otherwise, ($\mu(1) \neq \infty$); consider $\mu(-1)$. If $\mu(-1) = \infty$, we argue as above that a and c must be constructible.

Finally, if neither $\mu(1)$ nor $\mu(-1)$ are infinite, then we can solve for a and c in the linear equations

$$a + b = (c+d)\mu(1), \qquad b - a = (d-c)\mu(-1),$$

which show that a and c are constructible as required. (Note: these equations are solvable, provided $\mu(1) + \mu(-1) \neq 0$. This must be the case, for otherwise, we find that $ad - bc = 0$.) □

THEOREM 13. *The bipyramid B_{28} is not constructible.*

Proof. Let the cycle in B_{28} be $v_0, v_1, \ldots, v_{27}$ and the other two vertices be u and w. Suppose, for the sake of contradiction, that B_{28} were constructible. Fix a constructible representation and identify the (constructible) points t_0, t_9, and t_{18}, which are the points of tangency where the circles for v_0, v_9, and v_{18} meet the circle for u. By Lemma 12, there is a constructible μ, which maps $t_0 \mapsto 1$, $t_9 \mapsto i$, and $t_{18} \mapsto -1$. Since the coin-graph representation is unique once three points have been fixed, the transformed representation must be as in Fig. 6.

Now, since the original representation is constructible, every circle in that representation contains three constructible points, by Lemma 11. Those three points are mapped onto constructible points by μ, and so by Lemma 11 the new representation is also constructible.

However, by joining the centers of every fourth circle in the cycle, we can construct a regular 7-gon, which is known to be impossible. Thus B_{28} is not constructible. □

In particular, note that B_{28} cannot be represented by circles with integer radii, since if so it would be constructible by Lemma 10.

Although there are planar graphs that cannot be represented as coin graphs with integer radii, it remains an open question as to whether an arbitrary graph can be straight-line embedded in the plane with all edges of integral length [7].

4. Open problems.

4.1. Other convergence methods. Consider the dynamical system

$$\frac{dr_\alpha}{dt} = \theta_\alpha(r) - \pi.$$

We know that, provided that the incidence relation defining θ is that of a 3-connected planar graph, this system has a unique fixed point. Is this fixed point asymptotically stable? Indeed, will

$$\sum_\alpha (\theta_\alpha(r) - \pi)^2$$

serve as a Liapunov function? (Our experiments suggest that it does.)

4.2. Stability. Is there any sense in which the circle representation is stable under small changes to the graph? Put another way, what can be said about the change in the radii of the representing circles if an edge is added to or removed from the graph "far away"?

4.3. Higher dimensions. If C is a simplicial complex that embeds in R^n, then is C an n-sphere order, i.e., can it be represented by balls in R^n ordered by inclusion? By contrast with Theorem 5, it is known that the face lattice of a convex polytope in R^4 can have arbitrary high dimension, so an affirmative answer to this question would be of great interest.

REFERENCES

[1] G. R. Brightwell and R. A. W. Gregory, *Structure of random discrete spacetime*, Phys. Rev. Lett., 66 (1991), pp. 260–263.

[2] G. R. Brightwell and E. R. Scheinerman, *The dual of a circle order is not necessarily a circle order*, submitted.

[3] G. R. Brightwell and W. T. Trotter, *The order dimension of convex polytopes*, SIAM J. Discrete Math., 6 (1993), pp. 230–245, this issue.

[4] ———, *The order dimension of planar maps*, preprint.

[5] G. R. Brightwell and P. M. Winkler, *Sphere orders*, Order, 6 (1989), pp. 235–240.

[6] A. Darmet, *Représentation convexe d'un graphe planaire et de son dual par familles orthogonales de cercles tangents dans le plan ou sur la sphère*, manuscript, 1992.

[7] H. Harborth, *Ganzzahlige planare Darstellungen der platonischen*, Körper. El. Math., 42 (1987), pp. 118-122.

[8] G. H. Hurlbert, *A short proof that* $\mathbf{N}^3$ *is not a circle containment order*, Order, 5 (1988), pp. 235–237.

[9] P. Koebe, *Kontaktprobleme der konformen Abbildung*, Berichte über die Verhandlungen der Sächsischen Akademie der Wissenschaften zu Leipzig, Math.-Phys. Klasse, 88 (1936), pp. 141–164.

[10] D. Meyer, *The dimension of causal sets* I: *Minkowski dimension*, Syracuse University preprint, 1988.

[11] ———, *The dimension of causal sets* II: *Hausdorff dimension*, Syracuse University preprint, 1988.

[12] W. Pulleyblank and G. Rote, *Disk packings, planar graph and combinatorial optimization*, in preparation.

[13] H. Sachs, *Coin graphs, polyhedra and conformal mapping*, preprint.

[14] E. R. Scheinerman, *A note on planar graphs and circle orders*, SIAM J. Discrete Math., 4 (1991), pp. 448–451.

[15] E. R. Scheinerman and J. C. Wierman, *On circle containment orders*, Order, 4 (1988), pp. 315–318.

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

[17] O. Schramm, *How to cage an egg*, preprint.

[18] J. B. Sidney, S. J. Sidney, and J. Urrutia, *Circle orders, N-gon orders and the crossing number of partial orders*, Order, 5 (1988), pp. 1–10

[19] W. Thurston, *The Geometry and Topology of Three-Manifolds*, unpublished.

[20] W. Tutte, *How to draw a graph*, Proc. LMS, 13 (3) (1963), pp. 743–768.

[21] J. Urrutia, *Partial orders and Euclidean geometry*, in Algorithms and Order, I. Rival, ed., 1989, pp. 387–434.

SIAM J. DISC. MATH.
Vol. 6, No. 2, pp. 230–245, May 1993

THE ORDER DIMENSION OF CONVEX POLYTOPES*

GRAHAM BRIGHTWELL† AND WILLIAM T. TROTTER‡

Abstract. With a convex polytope $\mathbf{M}$ in $\mathbb{R}^3$, a partially ordered set $\mathbf{P_M}$ is associated whose elements are the vertices, edges, and faces of $\mathbf{M}$ ordered by inclusion. This paper shows that the order dimension of $\mathbf{P_M}$ is exactly 4 for every convex polytope $\mathbf{M}$. In fact, the subposet of $\mathbf{P_M}$ determined by the vertices and faces is critical in the sense that deleting any element leaves a poset of dimension 3.

Key words. convex polytopes, planar graphs, dimension

AMS(MOS) subject classifications. 06A07, 05C35

1. Introduction. We consider a planar map $\mathbf{M}$ as a finite connected planar graph $\mathbf{G} = (V, E)$ together with a plane drawing D of $\mathbf{G}$, i.e., a representation of $\mathbf{G}$ by points and arcs in the plane $\mathbb{R}^2$ in which there are no edge crossings. We do not distinguish between a vertex (edge) of $\mathbf{G}$ and the corresponding point (arc) in the plane. Deleting the vertices and edges of $\mathbf{G}$ from the plane leaves several connected components whose closures are the *faces* of $\mathbf{M}$. The unique unbounded face is called the *exterior* or *outside* face.

With a planar map $\mathbf{M}$, we associate a partially ordered set (poset) $\mathbf{P_M}$ whose elements are the vertices, edges, and faces (including the exterior face) of $\mathbf{M}$ ordered by inclusion. As an example, a planar map $\mathbf{M}$ and its associated poset $\mathbf{P_M}$ are shown in Fig. 1, below.

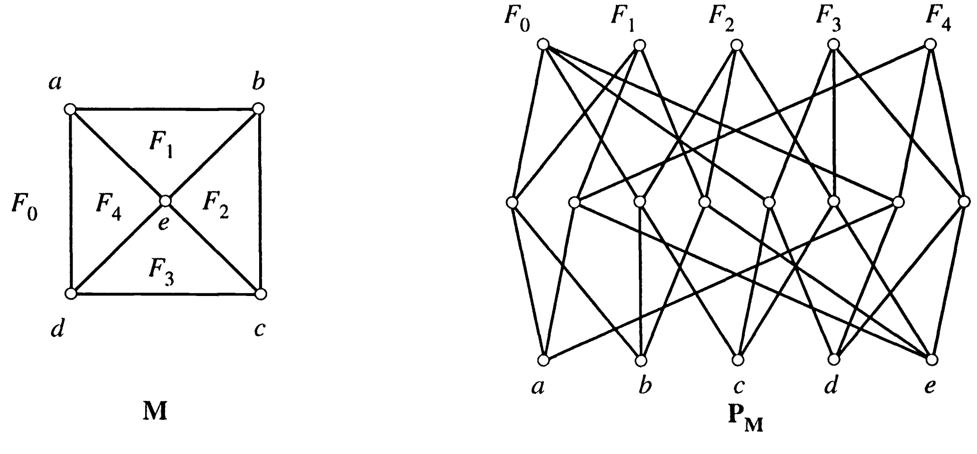


FIG. 1

With a convex polytope $\mathbf{M}$ in $\mathbb{R}^3$, there is associated a planar map, which we also denote by $\mathbf{M}$. Among all planar maps, a well-known theorem of Steinitz [13] character-

*Received by the editors January 24, 1992; accepted for publication (in revised form) May 8, 1992.

†Department of Mathematics, London School of Economics, Houghton Street, London WC2A 2AE, England.

‡Bell Communications Research, 445 South Street 2L-367, Morristown, New Jersey 07962, and Department of Mathematics, Arizona State University, Tempe, Arizona 85287. The research of this author was supported in part by National Science Foundation grants DMS87-13994 and DMS 89-02481.

izes those associated with convex polytopes in $\mathbb{R}^3$. These are exactly the three-connected planar maps. For example, the planar map in Fig. 1 is such a map.

Dushnik and Miller [2] defined the *order dimension* of a finite poset $\mathbf{P}$, denoted $\dim(\mathbf{P})$, as the least positive integer t for which $\mathbf{P}$ is the intersection of t linear orders. The principal result of this paper will be the following theorem.

THEOREM 1.1. *Let* $\mathbf{M}$ *be a planar map associated with a convex polytope in* $\mathbb{R}^3$, *and let* $\mathbf{P}_\mathbf{M}$ *be the partially ordered set of vertices, edges and faces of* $\mathbf{M}$ *ordered by inclusion. Then* $\dim(\mathbf{P}_\mathbf{M}) = 4$.

Before proceeding with the proof, we pause to make a few comments concerning the origin of this problem. Our original motivation comes from the study of convex polytopes in $\mathbb{R}^n$. The *face lattice* of a convex polytope $\mathbf{M}$ is the poset consisting of all vertices, edges, faces, hyperfaces, and so forth, partially ordered by inclusion. In Birkhoff's lattice theory book [1], the problem of determining the order dimension of the face lattice of a polytope in $\mathbb{R}^n$ is posed and is credited to Kurepa (see also Golumbic's book [3, p. 137]). In $\mathbb{R}^2$, the poset of vertices and edges of a convex polygon has the following form. The point set is $\{x_i : 1 \leq i \leq m\} \cup \{y_i : 1 \leq i \leq m\}$, and the order is given by $x_i < y_i$ and $x_i < y_{i+1}$ (cyclically) for $i = 1, 2, \ldots, m$, where $m \geq 3$ is the number of vertices. Such posets are easily seen to be three-dimensional. They belong to a well-known family of posets called *crowns* [14]. (See Fig. 2.)

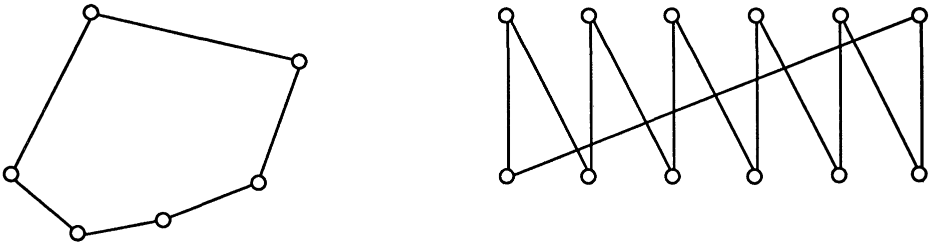


FIG. 2

If $n \geq 4$, there exist convex polytopes in $\mathbb{R}^n$ for which the face lattice has arbitrarily large dimension. This phenomenon is due to the existence of cyclical polytopes that have the property that they contain large sets of vertices each pair of which is contained in an edge. Spencer [12] showed that the order dimension $d(m)$ of the poset of all 1- and 2-element subsets of an m-element set satisfies $\log\log m \leq d(m) \leq 2\log\log m$.

Accordingly, the problem is of interest only in $\mathbb{R}^3$. Sedmak [11] reports on the existence of (nonconvex) polyhedra in $\mathbb{R}^3$ with face lattices of arbitrarily large dimension. However, our Theorem 1.1 implies that the order dimension of $\mathbf{P}_\mathbf{M}$ is 4 whenever $\mathbf{M}$ is associated with a convex polytope in $\mathbb{R}^3$, so for example, the poset shown in Fig. 1 has order dimension 4.

Also, we are motivated by the work of Schnyder [10], who proved the following elegant characterization of planar graphs.

THEOREM 1.2. *Let* $\mathbf{G} = (V, E)$ *be a graph and let* $\mathbf{Q}_\mathbf{G}$ *denote the poset consisting of the vertices and edges of* $\mathbf{G}$ *partially ordered by inclusion. Then* $\mathbf{G}$ *is planar if and only if the order dimension of* $\mathbf{Q}_\mathbf{G}$ *is at most* 3.

It is relatively easy to show that $\mathbf{G}$ is planar if $\dim(\mathbf{Q}_\mathbf{G}) \leq 3$. Schnyder's argument to show that $\dim(\mathbf{Q}_\mathbf{G}) \leq 3$ when $\mathbf{G}$ is planar is quite complex and requires the development of some entirely new concepts for planar graphs. However, Schnyder is able to capitalize on the fact that in this part of the proof, it can be assumed that $\mathbf{G}$ is a maximal planar graph. In this case, a plane drawing of $\mathbf{G}$ without edge crossings produces a planar triangulation $\mathbf{M}$, i.e., a planar map $\mathbf{M}$ in which every face (including the exterior face) is a triangle.

It is natural to ask what happens to the order dimension of the poset associated with a planar graph if we add the faces determined by a particular drawing. It is not at all clear why the order dimension should be bounded by any absolute constant, and it is conceivable that a planar graph can be drawn as two different maps for which the associated posets have different order dimension.

In the final section of this paper, Schnyder comments that it follows easily from his Theorem 1.2 that if $\mathbf{M}$ is a convex polytope in $\mathbb{R}^3$ in which every face is a triangle, then $\dim(\mathbf{P_M}) \leq 4$. By duality, the upper bound $\dim(\mathbf{P_M}) \leq 4$ also holds if every vertex has degree 3. For these reasons, the problem of finding an upper bound (if one exists) when $\mathbf{M}$ is an arbitrary convex polytope in $\mathbb{R}^3$ is a natural one.

We comment that Schnyder's theorem can be derived easily from our results. Also, we have been successful in establishing the upper bound $\dim(\mathbf{P_M}) \leq 4$ when $\mathbf{M}$ is an arbitrary planar map—allowing loops and multiple edges. As this result requires additional machinery, it will appear in a subsequent paper. For the general theorem, the results and techniques of this paper will serve as an essential first step.

In the next section of this paper, we collect some facts from dimension theory. The major part of the proof of Theorem 1.1 in §§3 and 4 involves the construction of a family of paths in a planar map. We fix three special vertices v_1, v_2, v_3 on the outside face and then, for each other vertex x, find three vertex-disjoint paths from x to the v_i. Menger's theorem tells us that, provided no pair of vertices separates any other vertex from $\{v_1, v_2, v_3\}$, we can find such a family of paths in the graph. We show that the family we construct has certain other properties related to the plane representation of the graph. This enables us to define three partial orders on the vertex set of the map, which we use in turn to define three linear extensions of $\mathbf{P_M}$. In the fourth linear extension, we require only that the outside face is below all vertices not on that face. These four linear extensions then intersect to give $\mathbf{P_M}$.

2. Necessary tools from dimension theory. In this section, we describe briefly some basic concepts of dimension theory needed in this paper. We refer the reader to the monograph [17] by Trotter, the survey article by Kelly [5] and by Kelly and Trotter [6] and the chapters in [15], [16] by Trotter for additional background material and an extensive list of references.

Let $\mathbf{P}$ be a finite poset. We write $x \| y$ to indicate that x and y are incomparable points in $\mathbf{P}$. A family $\mathbf{F} = \{L_1, L_2, \ldots, L_t\}$ of linear extensions of $\mathbf{P}$ is called a *realizer* of $\mathbf{P}$ if $\mathbf{P} = L_1 \cap L_2 \cap \cdots \cap L_t$, i.e., $x < y$ in $\mathbf{P}$ if and only if $x < y$ in L_i for $i = 1, 2, \ldots, t$. The dimension of $\mathbf{P}$ is then the minimum cardinality of a realizer.

An ordered pair (x, y) of incomparable points is called a *critical pair* if $z < x$ implies $z < y$ and $w > y$ implies $w > x$ for all $z, w \in \mathbf{P}$. In Fig. 3, we show a critical pair in a poset.

If (x, y) is a critical pair in a poset $\mathbf{P}$ and L is a linear extension of $\mathbf{P}$, we say *L reverses* (x, y) if $y < x$ in L. A family $\{L_1, L_2, \ldots, L_t\}$ of linear extensions of $\mathbf{P}$ is a realizer of $\mathbf{P}$ if and only if, for every critical pair (x, y), there is some i so that L_i reverses (x, y).

When $\mathbf{M}$ is a planar map, y a vertex, and F a face not containing y, then (y, F) is a critical pair in $\mathbf{P_M}$. So every realizer must (at least) reverse each critical pair of this type. We let $D(\mathbf{P_M})$ denote the least positive integer for which there exist t linear extensions

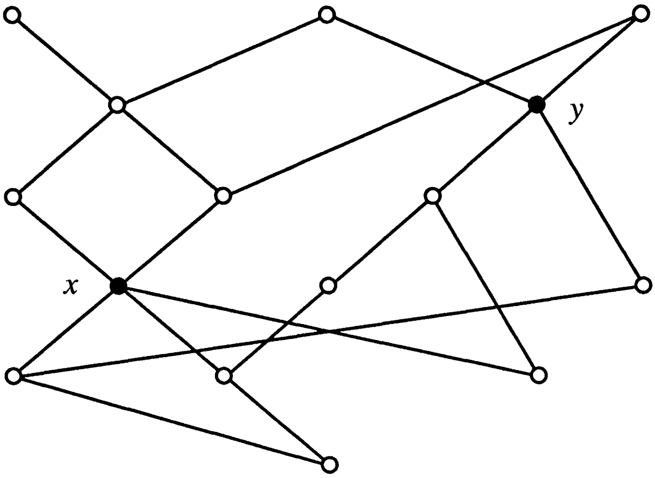


FIG. 3

$L_1, L_2, \ldots, L_t$ reversing all critical pairs of the form (y, F), where y is a vertex and F is a face not containing y. Of course, we always have $D(\mathbf{P_M}) \leq \dim(\mathbf{P_M})$.

We say that a planar map $\mathbf{M}$ is *well formed* if the critical pairs of $\mathbf{P_M}$ are exactly the pairs of the form (y, F), where y is a vertex, F is a face, and $y \notin F$. It is an easy exercise to show that if $\mathbf{M}$ is a planar map associated with a convex polytope in $\mathbb{R}^3$, then $\mathbf{M}$ is well formed so that $\dim(\mathbf{P_M}) = D(\mathbf{P_M})$.

When L is a linear order on the vertex set V of a planar map $\mathbf{M}$, y is a vertex and F is a face of $\mathbf{M}$, we write $y > F$ in L when $y > x$ in L for every vertex $x \in F$. It is easy to see that if L is any linear order on V, then there exists a linear extension L^* of $\mathbf{P_M}$ so that $y > F$ in L^* whenever $y > F$ in L. Accordingly, to show that $\dim(\mathbf{P_M}) \leq 4$ when $\mathbf{M}$ is a well-formed planar map, we must produce four linear orders, L_1, L_2, L_3, L_4 of the vertex set V so that for every critical pair (y, F), there is some i with $y > F$ in L_i.

3. Normal families of paths. When x and y are distinct vertices on the exterior face of $\mathbf{M}$, we denote by $\mathbf{M}[x, y]$ the sequence of vertices encountered in proceeding clockwise around the exterior face of $\mathbf{M}$ beginning at x and ending at y. For the sequence obtained by proceeding in a counterclockwise direction, we write $\mathbf{M}^r[x, y]$. For example, in the planar map $\mathbf{M}$ shown in Fig. 4, $\mathbf{M}[f, a] = (f, g, v_1, g, a)$ and $\mathbf{M}^r[e, a] = (e, v_2, b, a)$.

We call a triple (v_1, v_2, v_3) of distinct vertices from the exterior face of $\mathbf{M}$ a *triad* if $v_{\alpha+2} \notin M[v_\alpha, v_{\alpha+1}]$ for $\alpha = 1, 2, 3$. (Throughout this paper, subscripts are interpreted cyclically.) The triple (v_1, v_2, v_3) is a triad for the map $\mathbf{M}$, shown in Fig. 4.

When $P_1, P_2, \ldots, P_k$ are paths in $\mathbf{M}$, we denote by $S(P_1, P_2, \ldots, P_k)$ the set of all points in the plane that belong to an edge in some P_i together with those points inside any cycle formed by edges in the union of the edge sets of these paths. For example, in Fig. 4, let $P_1 = \mathbf{M}[v_1, v_2], P_2 = (c, a, g, v_1), P_3 = (c, b, d, v_2)$. Then $S(P_1, P_2, P_3)$ contains the points from the edges in these paths and points inside the triangles T_1 and T_2.

Now let $\mathbf{M}$ be a planar map and let (v_1, v_2, v_3) be a triad for $\mathbf{M}$. Let $\mathcal{F} = \{P(x, v_\alpha) : x \in V, \alpha = 1, 2, 3\}$ be a family of paths in $\mathbf{M}$. We say that $\mathcal{F}$ is a *normal family* of paths for (v_1, v_2, v_3), provided the following five properties are satisfied.

Path Property 1. For all $x \in V$ and each $\alpha = 1, 2, 3$, $P(x, v_\alpha)$ is a path from x to v_α.

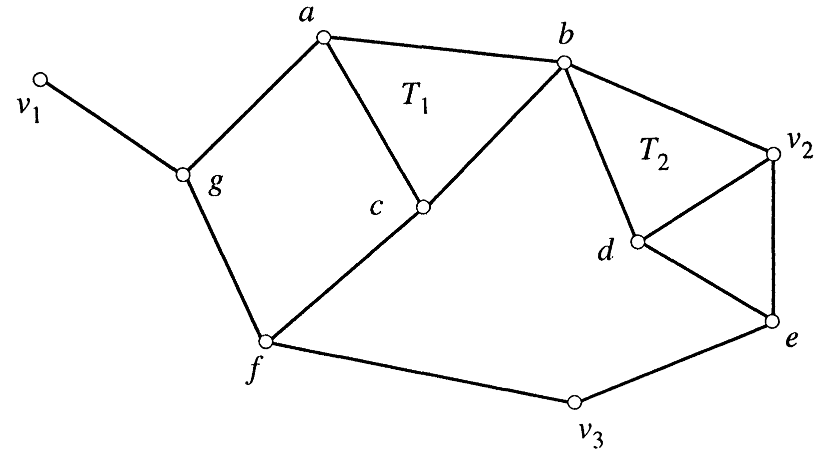


FIG. 4

Path Property 2. For all $x \in V - \{v_1, v_2, v_3\}$ and each $\alpha = 1, 2, 3$, the paths $P(x, v_\alpha)$ and $P(x, v_{\alpha+1})$ have only the vertex x in common.

Path Property 3. For each $\alpha = 1, 2, 3$, $P(v_\alpha, v_{\alpha+1}) = \mathbf{M}[v_\alpha, v_{\alpha+1}]$ and $P(v_{\alpha+1}, v_\alpha) = \mathbf{M}^r[v_{\alpha+1}, v_\alpha]$.

Path Property 4. For all $x, y \in V$ and each $\alpha = 1, 2, 3$, if $P(x, v_\alpha)$ is the path $(x = u_0, u_1, \ldots, u_t = v_\alpha)$ and $y = u_i$ for some i, then $P(y, v_\alpha)$ is the path $(y = u_i, u_{i+1}, \ldots, u_t = v_\alpha)$, i.e., $P(y, v_\alpha)$ is a terminal segment of $P(x, v_\alpha)$.

Path Property 5. For all $x \in V$ and each $\alpha = 1, 2, 3$, let $S(x, \alpha) = S(P(x, v_{\alpha+1}), P(x, v_{\alpha+2}), P(v_{\alpha+1}, v_{\alpha+2}))$. Then, for all $x, y \in V$ and each $\alpha = 1, 2, 3$, if $y \in S(x, \alpha)$, then $S(y, \alpha) \subseteq S(x, \alpha)$.

For the planar map shown in Fig. 4, it is easy to see that there are two normal families of paths for the triad (v_1, v_2, v_3). The only option is to choose $P(b, v_3)$ as either (b, c, f, v_3) or (b, d, e, v_3). We say that x and y are *α-equivalent* when $S(x, \alpha) = S(y, \alpha)$. The reader is invited to compare Schnyder's proof [10] of Theorem 1.1 and his construction of families of paths in a planar triangulation. Note that when $\mathbf{M}$ is a planar triangulation, Schnyder's argument gives an explicit construction of a normal family of paths for which there is no pair of α-equivalent vertices.

Recall that a 3-connected planar map is well formed. In the next section, we will show that a 3-connected planar map has a normal family of paths for every triad. To provide clear motivation for the concept of a normal family, we show how such a family is used to establish the upper bound $\dim(\mathbf{P_M}) \leq 4$ when $\mathbf{M}$ is 3-connected. First, we will need some additional properties of normal families of paths and binary relations defined in terms of them. In what follows, let (v_1, v_2, v_3) be a triad for a planar map $\mathbf{M}$ and let $\mathcal{F} = \{P(x, v_\alpha) : x \in V, \alpha = 1, 2, 3\}$ be a normal family of paths for (v_1, v_2, v_3).

LEMMA 3.1. *If $\alpha \in \{1, 2, 3\}$, $x \in V$, $y \in S(x, \alpha)$ and $y \notin P(x, v_{\alpha+1}) \cup P(x, v_{\alpha+2})$, then $x \notin P(y, v_{\alpha+1}) \cup P(y, v_{\alpha+2})$.*

Proof. If $x \in P(y, v_{\alpha+1}) \cup P(y, v_{\alpha+2})$, then $x \in S(y, \alpha)$, so $S(x, \alpha) \subseteq S(y, \alpha)$. However, $y \in S(x, \alpha)$ and $y \notin P(x, v_{\alpha+1}) \cup P(x, v_{\alpha+2})$ require $S(y, \alpha) \subsetneqq S(x, \alpha)$. The contradiction completes the proof. □

For each $\alpha \in \{1, 2, 3\}$, the binary relation Q_α defined on the vertex set V of M by $Q_\alpha = \{(x, y) : S(x, \alpha) \subsetneqq S(y, \alpha)\}$ is obviously a partial order. Note that when $S(x, \alpha) \not\subseteq S(y, \alpha)$ and $S(y, \alpha) \not\subseteq S(x, \alpha)$, we have $x \| y$ in Q_α. We simplify this by writing

$S(x,\alpha)\|S(y,\alpha)$. However, we also have $x\|y$ in Q_α when x and y are distinct α-equivalent points, i.e., $S(x,\alpha) = S(y,\alpha)$. Note that when $x\|y$ in Q_α, there is a unique $\beta \in \{\alpha + 1, \alpha+2\}$, so that $S(x,\beta) \subsetneq S(y,\beta)$.

The general plan is to take a linear extension L_α of the partial order Q_α for each $\alpha = 1,2,3$. However, we need for each L_α to satisfy certain other conditions. Ideally, we would like $x > F$ in L_α whenever $x \notin F$ and $F \subseteq S(x,\alpha)$. Since L_α extends Q_α, this will certainly occur unless F contains a vertex y, which is α-equivalent to x. Indeed, it may well be that x and y are α-equivalent, say with $y \in P(x,\alpha+1)$, and there are faces F and G in $S(x,\alpha)$ with F containing y but not x and G containing x but not y. In this situation, we clearly cannot have both $x > F$ and $y > G$ in L_α. Can we put $x > F$ in one of the other linear extensions? Not in $L_{\alpha+1}$, since $(x,y) \in Q_{\alpha+1}$. If F contains a vertex w with $(x,w) \in Q_{\alpha+2}$, then we cannot put $x > F$ in $L_{\alpha+2}$ either. Fortunately, if F contains such a vertex w, then G cannot contain a vertex z with $(y,z) \in Q_{\alpha+1}$, (see Lemma 3.4), so we may put $(y,x) \in L_\alpha$, and force $x > F$ in L_α. On the other hand, if F contains no such vertex w, then we want to put $(u,x) \in L_{\alpha+2}$ for every $u \in F$, to get $x > F$ in $L_{\alpha+2}$. We then must check that these relations do not conflict and that we can find linear extensions L_1, L_2, L_3 satisfying these various requirements.

More formally, we proceed by defining for each $\alpha \in \{1,2,3\}$ a suitable extension Q'_α of the order Q_α, and then taking a linear extension L_α of Q'_α. To accomplish this, we must first define some new binary relations on V. For each $\alpha \in \{1,2,3\}$, define $\mathcal{L}_\alpha = \{(x,y) \in V\times V : x\|y \text{ in } Q_\alpha \text{ and } S(y,\alpha+2) \subsetneq S(x,\alpha+2)\}$ and $\mathcal{R}_\alpha = \{(x,y) \in V\times V : x\|y \text{ in } Q_\alpha \text{ and } S(y,\alpha+1) \subsetneq S(x,\alpha+1)\}$.

Recall that the *dual* of a binary relation Q on a set V is the relation $\{(x,y) \in V\times V : (y,x) \in Q\}$. The following result is then immediate.

LEMMA 3.2. *For each $\alpha \in \{1,2,3\}$, $\mathcal{L}_\alpha$ and $\mathcal{R}_\alpha$ are partial orders on V, and $\mathcal{R}_\alpha$ is the dual of $\mathcal{L}_\alpha$.*

We think of $\mathcal{L}_\alpha$ and $\mathcal{R}_\alpha$ as denoting "left" and "right," respectively. In what follows, we will define binary relations $\mathcal{L}''_\alpha \subseteq \mathcal{L}_\alpha$ and $\mathcal{R}''_\alpha \subseteq \mathcal{R}_\alpha$; however, $\mathcal{L}''_\alpha$ and $\mathcal{R}''_\alpha$ will not be dual. First set $\mathcal{L}'_\alpha = \{(x,y) \in \mathcal{L}_\alpha$: there is a face F and a vertex $u \neq y$ such that (1) $u,x \in F$, (2) $y \in S(u,\alpha+1)$, and (3) $u \notin P(y,v_\alpha)\}$. If F and u are as above, we say that (F,u) *witnesses* $(x,y) \in \mathcal{L}'_\alpha$.

Now we set $\mathcal{L}''_\alpha = \{(x,z) \in \mathcal{L}_\alpha$: there is some y with $(x,y) \in \mathcal{L}'_\alpha$ and $(y,z) \in Q_\alpha$ or $y = z\}$. If y is as above and (F,u) witnesses $(x,y) \in \mathcal{L}'_\alpha$, we say that the triple (F,u,y) *witnesses* $(x,z) \in \mathcal{L}''_\alpha$. Thus, $\mathcal{L}'_\alpha$ is designed to capture both of the cases discussed above, where we must impose $(x,y) \in L_\alpha$, although $(x,y) \notin Q_\alpha$, at least where $(x,y) \in \mathcal{L}_\alpha$.

We define $\mathcal{R}'_\alpha$ and $\mathcal{R}''_\alpha$ in the corresponding way. Thus we set $\mathcal{R}'_\alpha = \{(x,y) \in \mathcal{R}_\alpha$: there is a face F and a vertex $u \neq y$, such that (1) $u,x \in F$, (2) $y \in S(u,\alpha+2)$, and (3) $u \notin P(y,v_\alpha)\}$. As before, in this situation we say that (F,u) witnesses $(x,y) \in \mathcal{R}'_\alpha$. Again, just as before, we set $\mathcal{R}''_\alpha = \{(x,z) \in \mathcal{R}_\alpha$: there is some y with $(x,y) \in \mathcal{R}'_\alpha$ and $(y,z) \in Q_\alpha$ or $y = z\}$. If here (F,u) witnesses $(x,y) \in \mathcal{R}'_\alpha$, then we say (F,u,y) witnesses $(x,z) \in \mathcal{R}''_\alpha$.

The next lemma provides some information about the binary relations $\mathcal{L}'_\alpha$ and $\mathcal{L}''_\alpha$. There is, of course, a symmetric version for $\mathcal{R}'_\alpha$ and $\mathcal{R}''_\alpha$.

LEMMA 3.3. *Let $\alpha \in \{1,2,3\}$ and suppose that (F,u,y) witnesses $(x,z) \in \mathcal{L}''_\alpha$.*

(1) *If x and z are α-equivalent, then $F \subseteq S(x,\alpha)$, and $y = z$ (i.e., $(x,z) \in \mathcal{L}'_\alpha$).*

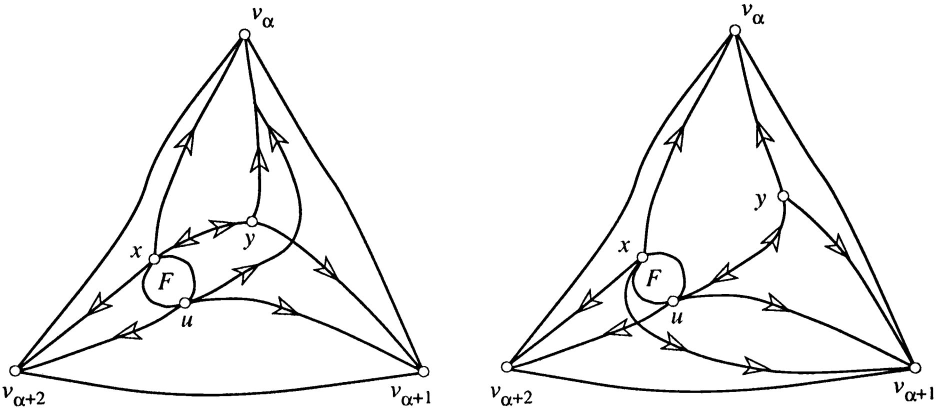
(2) *If $S(x,\alpha)\|S(z,\alpha)$, then u and y are $(\alpha+1)$-equivalent, and both y and u are on $P(z,v_{\alpha+2})$.*

Proof. We first verify statement (1). Suppose, then, that x and z are α-equivalent. If $z \neq y$, then $S(y,\alpha) \subsetneq S(z,\alpha) = S(x,\alpha)$, which is not possible. Thus, in this case,

$z = y$. If $F \not\subseteq S(x, \alpha)$, then $F \subseteq S(y, \alpha + 1)$, so in particular $u \in S(y, \alpha + 1)$. Since also $y \in S(u, \alpha + 1)$, this implies that u and y are $(\alpha + 1)$-equivalent, so we must have $u \in P(y, v_\alpha)$, a contradiction. This completes the proof of (1).

We now prove (2). Since $S(x, \alpha) \| S(y, \alpha)$ and $(x, y) \in \mathcal{L}_\alpha$, it is clear that $F \subseteq S(y, \alpha + 1)$. Thus $S(u, \alpha + 1) \subseteq S(y, \alpha + 1)$. However, we also have $S(y, \alpha + 1) \subseteq S(u, \alpha + 1)$, so u and y are $(\alpha + 1)$-equivalent. We do not have $u \in P(y, v_\alpha)$, so we must have $y \in P(u, v_\alpha)$. If $y = z$, this completes the proof, so suppose that $(y, z) \in Q_\alpha$. Then $u \in S(z, \alpha)$, but x is not in this region, so u is on $P(z, v_{\alpha+2})$. Finally, $y \in S(u, \alpha + 1) \subseteq S(z, \alpha + 1)$, and, since also $y \in S(z, \alpha)$, this implies that y is on $P(z, v_{\alpha+2})$. □

Note that when (F, u) witnesses $(x, y) \in \mathcal{L}'_\alpha$, the face F can be located in $S(x, \alpha)$ or in $S(x, \alpha + 2)$. See Figs. 5(a) and 5(b).



(a) (b)

FIG. 5

Our goal is to prove that the binary relation given by $Q'_\alpha = Q_\alpha \cup \mathcal{L}''_\alpha \cup \mathcal{R}''_\alpha$ is acyclic, and then to take L'_α to be a linear extension of the transitive closure of Q'_α.

LEMMA 3.4. *If* $\alpha \in \{1, 2, 3\}$, $(x, z) \in \mathcal{L}''_\alpha$ *and* $(z, w) \in \mathcal{R}'_\alpha$, *then* $x \neq w$ *and* $(x, w) \in Q_\alpha \cup \mathcal{L}''_\alpha$.

Proof. Take (F, u, y), witnessing $(x, z) \in \mathcal{L}''_\alpha$, and (G, v), witnessing $(z, w) \in \mathcal{R}'_\alpha$.

First, we consider the case where $S(x, \alpha) \| S(z, \alpha)$. Let R be the region bounded by $P(x, v_\alpha)$, $P(z, v_\alpha)$, $P(z, v_{\alpha+2})$, and the clockwise path from x to u round F. Note that there are two slightly different situations, depending on whether F is in $S(x, \alpha)$ or $S(x, \alpha + 2)$. (See Figs. 6(a) and 6(b).) We claim that $w \in R$.

Since z is in the interior of $S(x, \alpha+2)$ and shares a face with v, v is also in $S(x, \alpha+2)$, and hence, so is w. Also $w \in S(z, \alpha+1)$. If $F \subseteq S(x, \alpha)$, this suffices to prove our claim, so suppose that $F \subseteq S(x, \alpha + 2)$. Now if $v \in S(u, \alpha + 2)$, then so is w, and we are done. However, $z \notin P(u, v_{\alpha+1})$, so the only other possibility is that $v \in R$, in which case w is also in R, as required. Note that this also rules out the case where $w = x$ and $F \subseteq S(x, \alpha + 2)$, since that requires $v \notin R$.

Consider the path $P = P(w, v_{\alpha+2})$ and the point it leaves R. If P joins the path $P(z, v_{\alpha+2})$ and exits via u, then $u \in S(w, \alpha)$, so $y \in S(w, \alpha)$, and hence either $(y, w) \in Q_\alpha$, when $(x, w) \in \mathcal{L}''_\alpha$, or y and w are α-equivalent when $(w, z) \in Q_\alpha$, a contradiction.

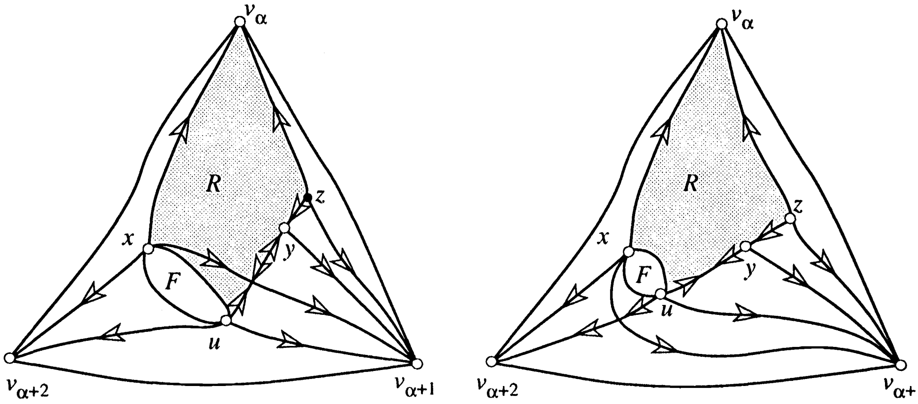


FIG. 6

The path P does not cross $P(z, v_\alpha)$, so the only remaining possibility is that it crosses $P(x, v_\alpha)$. In this case, $x \in S(w, \alpha)$, and so $(x, w) \in Q_\alpha$, unless $S(x, \alpha) = S(w, \alpha)$.

By an earlier remark, we cannot have $x = w$ and $F \subseteq S(x, \alpha + 2)$, so if $S(x, \alpha) = S(w, \alpha)$, we have $F \subseteq S(x, \alpha)$. Now v is on $P(x, v_{\alpha+1})$, but is not in $S(z, \alpha)$, since, then, $P(v, v_\alpha)$ cannot go via w. Finally, $P(v, v_{\alpha+2})$ exits R via u, but this contradicts Lemma 3.1. This completes the proof in the case where $S(x, \alpha) \| S(z, \alpha)$.

Now suppose that x and z are α-equivalent. We know that in this case $y = z$. Suppose next that z and w are also α-equivalent. If w is on $P(x, v_{\alpha+1})$, with $w \neq x$, then (F, u) witnesses $(x, w) \in \mathcal{L}'_\alpha$, so we may suppose that $w \in P(x, v_{\alpha+2})$. Then (G, v) witnesses also $(z, x) \in \mathcal{R}'_\alpha$. If $v \in S(u, \alpha + 2)$, then $x \in S(v, \alpha + 2) \subseteq S(u, \alpha + 2)$, which is clearly not possible. By symmetry, we are also done if $u \in S(v, \alpha + 1)$. So suppose $v \in S(u, \alpha + 1)$ and $u \in S(v, \alpha + 2)$. (Clearly, we cannot have, for instance, v in the interior of $S(u, \alpha)$.) Then v is in the region R bounded by $P(y, v_{\alpha+2})$, $P(u, v_\alpha)$, and F. Now consider $P(v, v_{\alpha+2})$. It cannot cross $P(u, v_\alpha)$, since that would imply $u \in S(v, \alpha + 1)$. Thus the path must join $P(y, v_{\alpha+2})$ and leave R via x. This clearly contradicts $x \in S(v, \alpha + 2)$.

Finally, suppose that x and z are α-equivalent, but that z and w are not. If $(x, w) \notin Q_\alpha$, then $(x, w) \in \mathcal{R}_\alpha$, and $x \in S(w, \alpha + 2) = S(v, \alpha + 2)$. If v is on $P(x, v_\alpha)$, then so is w, which implies $(x, w) \in Q_\alpha$. If v is not on $P(x, v_\alpha)$, then (G, v) witnesses $(z, x) \in \mathcal{R}'_\alpha$, which we have just seen is not possible. □

Now for each $\alpha = 1, 2, 3$, let $Q'_\alpha = Q_\alpha \cup \mathcal{L}''_\alpha \cup \mathcal{R}''_\alpha$. We will show that Q'_α is an acyclic binary relation on V so that the transitive closure of Q'_α is a partial order extending Q_α.

LEMMA 3.5. *For each* $\alpha = 1, 2, 3$, *the binary relation* Q'_α *is acyclic.*

Proof. Suppose to the contrary that Q'_α is not acyclic and choose a sequence $x_1, x_2, \ldots x_s$ so that $(x_i, x_{i+1}) \in Q'_\alpha$ for $i = 1, 2, \ldots, s$. Without loss of generality, we may assume that this sequence has been chosen so that s is minimum. Then the points $x_1, x_2, \ldots, x_s$ are all distinct. Furthermore, $(x_i, x_{i+2}) \notin Q'_\alpha$ for $i = 1, 2, \ldots, s$.

Since Q_α is acyclic, we know that at least one of the pairs in $\{(x_i, x_{i+1}) : 1 \leq i \leq s\}$ belongs to $\mathcal{L}''_\alpha \cup \mathcal{R}''_\alpha$. By symmetry, we will assume one (or more) of these pairs is in $\mathcal{L}''_\alpha$.

Since $\mathcal{L}''_\alpha \subseteq \mathcal{L}_\alpha$, we know that the relation $\mathcal{L}''_\alpha$ is acyclic. It follows that there is some $i \leq s$ for which $(x_i, x_{i+1}) \in \mathcal{L}''_\alpha$ and $(x_{i+1}, x_{i+2}) \in Q_\alpha \cup \mathcal{R}''_\alpha$. If $(x_{i+1}, x_{i+2}) \in Q_\alpha$, then (x_i, x_{i+2}) is clearly in $\mathcal{L}''_\alpha$; whereas if (G, v, y) witnesses $(x_{i+1}, x_{i+2}) \in \mathcal{R}''_\alpha$, then by the previous lemma we have $(x_i, y) \in Q_\alpha \cup \mathcal{L}''_\alpha$, so $(x_i, x_{i+2}) \in Q_\alpha \cup \mathcal{L}''_\alpha$. □

With the preceding lemma, we are now ready to establish the upper bound, $\dim(\mathbf{P}_\mathbf{M}) \leq 4$, when $\mathbf{M}$ is a 3-connected planar map—under the assumption that $\mathbf{M}$ has a normal family of paths.

THEOREM 3.6. *Let (v_1, v_2, v_3) be a triad for a planar map $\mathbf{M}$ and suppose that $\mathcal{F} = \{P(x, v_\alpha) : x \in V, \alpha = 1, 2, 3\}$ is a normal family of paths for (v_1, v_2, v_3); then $D(\mathbf{P}_\mathbf{M}) \leq 4$.*

Proof. As before, for each $\alpha = 1, 2, 3$, let Q'_α be the acyclic binary relation on the vertex set V defined by $Q'_\alpha = Q_\alpha \cup \mathcal{L}'_\alpha \cup \mathcal{R}'_\alpha$. Then the transitive closure of Q'_α is a partial order on V. Let L_α be a linear extension of this partial order. Then let L_4 be any linear order on V for which $x < y$ in L_4 whenever x is on the exterior face of $\mathbf{M}$ and y is not.

Now let (y, F) be a critical pair in $\mathbf{P}_\mathbf{M}$. We show that $y > F$ in some L_i. If F is the exterior face, then $y > F$ in L_4. So we assume F is an interior face. In this case, we actually prove a stronger statement. We show that there is some $\alpha \in \{1, 2, 3\}$ for which $(x, y) \in Q'_\alpha$ for every $x \in F$. For such an α, we have $y > F$ in L_α.

To see this, choose $\alpha \in \{1, 2, 3\}$ so that $F \subseteq S(y, \alpha)$. Then $S(x, \alpha) \subseteq S(y, \alpha)$ for every $x \in F$. If $S(x, \alpha) \subsetneq S(y, \alpha)$ for every $x \in F$, then $y > F$ in Q_α, and thus $y > F$ in Q'_α. So we may assume that there is a point $x_0 \in F$ for which $S(x_0, \alpha) = S(y, \alpha)$. By symmetry, we may assume that $(x_0, y) \in \mathcal{L}_\alpha$. If F contains a point u for which $S(y, \alpha + 1) \subseteq S(u, \alpha + 1)$, then (F, u) witnesses $(x, y) \in \mathcal{L}'_\alpha$ for every $x \in F$ with $x \| y$ in Q_α. For any other $x \in F$, we have $S(x, \alpha) \subsetneq S(y, \alpha)$ and $(x, y) \in Q_\alpha$. Together, these statements imply $(x, y) \in Q'_\alpha$ for every $x \in F$.

It remains only to consider the case where F contains no point u for which $S(y, \alpha + 1) \subseteq S(u, \alpha + 1)$. In this case, we claim that $y > F$ in $Q'_{\alpha+1}$. To see this, observe that for each $x \in F$, either $S(x, \alpha + 1) \subsetneq S(y, \alpha + 1)$ or $x \| y$ in $Q_{\alpha+1}$. However, when $x \| y$ in $Q_{\alpha+1}$, the face F and the vertex x_0 witness $(x, y) \in \mathcal{R}'_{\alpha+1}$. Then $(x, y) \in Q'_{\alpha+1}$ for every $x \in F$. □

Since $D(\mathbf{P}_\mathbf{M}) = \dim(\mathbf{P}_\mathbf{M})$ when $\mathbf{M}$ is 3-connected, Theorem 3.6 yields the upper bound of our principal theorem once we have established the existence of a normal family of paths.

4. Constructing normal families of paths. Let $\mathbf{M}$ be a planar map, and let X, Y, and Z be vertices or sets of vertices in $\mathbf{M}$, with $X \cap Z = \emptyset$. We say that Z *separates* X from Y if every path in $\mathbf{M}$ from X to Y includes a vertex in Z.

Let $\mathbf{M}$ be a planar map and let (v_1, v_2, v_3) be a triad for $\mathbf{M}$. We say that $\mathbf{M}$ satisfies the *star-property* for (v_1, v_2, v_3) if for every vertex $x \in V - \{v_1, v_2, v_3\}$, no pair $\{y, z\} \subseteq V - \{x\}$ separates x from $\{v_1, v_2, v_3\}$. From Menger's theorem, it follows that $\mathbf{M}$ satisfies the star-property for (v_1, v_2, v_3) if and only if there is a family $\{P(x, v_\alpha) : x \in V, \alpha = 1, 2, 3\}$ satisfying Path Properties 1 and 2.

LEMMA 4.1 (normal family lemma). *Let $\mathbf{M}$ be a planar map and let (v_1, v_2, v_3) be a triad for $\mathbf{M}$. Then $\mathbf{M}$ has a normal family of paths for (v_1, v_2, v_3) if and only if $\mathbf{M}$ satisfies the star-property for (v_1, v_2, v_3).*

Proof. As noted previously, necessity follows from consideration of Path Properties 1 and 2 alone. We now prove sufficiency. We proceed by induction on the sum $\mathbf{S}(\mathbf{M})$ of the number of edges and the number of faces of $\mathbf{M}$. The lemma is true for the two maps (K_3 and $K_{1,3}$) where $\mathbf{S}(\mathbf{M})$ is at most 5.

So we consider a planar map $\mathbf{M}$, having $\mathbf{S}(\mathbf{M}) > 5$, with a triad (v_1, v_2, v_3) and we assume that the lemma holds for all planar maps $\mathbf{M}'$ with $\mathbf{S}(\mathbf{M}') < \mathbf{S}(\mathbf{M})$.

The remainder of the argument is organized into a series of cases. In treating these cases, we will consider maps $\mathbf{M}_0$, $\mathbf{M}_1$, $\mathbf{M}_2$, and so forth. These maps are either submaps of $\mathbf{M}$ or are formed by making minor changes in submaps of $\mathbf{M}$. When working with such a map, say $\mathbf{M}_i$, we will use the notation $\mathcal{F}_i$ for a normal family in $\mathbf{M}_i$, and a path from x to y in $\mathbf{M}_i$ will be denoted $P_i(x, y)$. The vertex set of $\mathbf{M}_i$ will be denoted V_i, and so forth. If $P(x, y)$ and $P(y, z)$ are paths having only the vertex y in common, we denote by $P(x, y) \oplus P(y, z)$ the path from x to z formed by their concatenation. We also use the notation $P(x, y) \oplus P(z, w)$ for the path formed by the union of two vertex disjoint paths for which yz is an edge.

Case 1. $\mathbf{M}$ has a cut-vertex.

Suppose $\mathbf{M} - \{x\}$ is the union of r components $C_1, C_2, \ldots, C_r$ with $r \geq 2$. If C_i is one of these components and $C_i \cap \{v_1, v_2, v_3\} = \emptyset$, then any vertex in C_i is separated from $\{v_1, v_2, v_3\}$ by x. So each C_i contains at least one element from $\{v_1, v_2, v_3\}$. Since $r \geq 2$, we may assume without loss of generality that C_1 contains exactly one element from $\{v_1, v_2, v_3\}$, say v_α. If v_α is not the only element of C_1, choose a point $y \in C_1 - \{v_\alpha\}$. Then y is separated from $\{v_1, v_2, v_3\}$ by x and v_α. So it follows that v_α is the only element of C_1 and that the edge $e = v_\alpha x$ is a bridge.

Clearly, $\mathbf{M}_0 = \mathbf{M} - \{v_\alpha\}$ satisfies the star-property for the triad $(x, v_{\alpha+1}, v_{\alpha+2})$. Now let $\mathcal{F}_0$ be a normal family of paths in $\mathbf{M}_0$. Then define $\mathcal{F}$ by $P(y, v_\alpha) = P_0(y, x) \oplus (x, v_\alpha)$ for every $y \in V - \{v_\alpha\}$, while $P(v_\alpha, v_\alpha)$ is trivial.

It is straightforward to verify that $\mathcal{F}$ is a normal family for $\mathbf{M}$. The only difficulty is to make sure that $P(x, v_{\alpha+1})$ and $P(x, v_{\alpha+2})$ have no vertex in common other than x. However, if z is common to these paths, then x is separated from $\{v_1, v_2, v_3\}$ by v_α and z. So in the remainder of the proof, we will assume $\mathbf{M}$ has no cut-vertices.

Case 2. For some $\alpha \in \{1, 2, 3\}, v_\alpha v_{\alpha+1}$ is an edge in $\mathbf{M}$.

Consider the planar map $\mathbf{M}_0$ obtained by deleting the edge $v_\alpha v_{\alpha+1}$ from $\mathbf{M}$. It is easy to show that (v_1, v_2, v_3) is a triad for $\mathbf{M}_0$ and $\mathbf{M}_0$ satisfies the star-property for (v_1, v_2, v_3). Let $\mathcal{F}_0$ be a normal family of paths for (v_1, v_2, v_3) in $\mathbf{M}_0$. Construct $\mathcal{F}$ from $\mathcal{F}_0$ by setting $P(v_\alpha, v_{\alpha+1}) = (v_\alpha, v_{\alpha+1})$ and $P(v_{\alpha+1}, v_\alpha) = (v_{\alpha+1}, v_\alpha)$ as required by Path Property 3. All other paths are the same in $\mathcal{F}$ as in $\mathcal{F}_0$. Clearly, $\mathcal{F}$ is a normal family of paths for (v_1, v_2, v_3), so in what follows, we assume that $\{v_1, v_2, v_3\}$ is an independent set.

Now we pause to make an important observation about the faces of $\mathbf{M}$. If F is an interior face, then the boundary of F is a simple cycle. If we label the vertices of F as $x_1, x_2, \ldots, x_t$ in clockwise order, then $x_i x_{i+1}$ is an edge for each i, but these are the only edges among the vertices of F. For if $x_i x_j$ is an edge, and these vertices are not consecutive, then one of x_{i+1} and x_{j+1} is an interior vertex separated from $\{v_1, v_2, v_3\}$ by x_i and x_j.

Also, a similar argument shows that if F and G are interior faces having one or more common vertices, then their common vertices occur consecutively on their boundaries.

Case 3. For some $\alpha \in \{1, 2, 3\}$, there exists an interior face F that contains v_α and a point from $\mathbf{M}[v_{\alpha+1}, v_{\alpha+2}]$.

Label the points on the boundary of F in clockwise order $x_1, x_2, \ldots, x_t$ so that x_1 belongs to $\mathbf{M}[v_{\alpha+1}, v_{\alpha+2}]$ but x_t does not. Let i be the largest integer for which $x_i \in \mathbf{M}[v_{\alpha+1}, v_{\alpha+2}]$. Then either $i = 1$ or $i = 2$, for if $i \geq 2$, then x_2 is separated from $\{v_1, v_2, v_3\}$ by x_1 and x_3.

Suppose next that $x_1 = v_{\alpha+1}$. Choose a vertex $x \in \mathbf{M}[v_\alpha, v_{\alpha+1}]$ with $x \notin \{v_\alpha, v_{\alpha+1}\}$. Then x is separated from $\{v_1, v_2, v_3\}$ by v_α and $v_{\alpha+1}$. The contradiction shows $x_1 \neq v_{\alpha+1}$. Similarly, $x_i \neq v_{\alpha+2}$.

The removal of x_i and v_α from $\mathbf{M}$ disconnects the map and leaves $v_{\alpha+1}$ in a component C_1. We let $\mathbf{M}_1$ be the submap generated by the vertices in C_1 together with x_i and v_α. Then $(v_\alpha, v_{\alpha+1}, x_i)$ is a triad for $\mathbf{M}_1$, and $\mathbf{M}_1$ satisfies the star-property for $(v_\alpha, v_{\alpha+1}, x_i)$.

The map $\mathbf{M}_2$ is formed in an analogous fashion considering the component C_2 containing $v_{\alpha+2}$ when x_1 and v_α are removed. Then $\mathbf{M}_2$ satisfies the star-property for the triad $(v_\alpha, x_1, v_{\alpha+2})$.

Now let $\mathcal{F}_1$ be a normal family in $\mathbf{M}_1$ for $(v_\alpha, v_{\alpha+1}, x_i)$, and let $\mathcal{F}_2$ be a normal family in $\mathbf{M}_2$ for $(v_\alpha, x_1, v_{\alpha+2})$. Define the normal family $\mathcal{F}$ in $\mathbf{M}$ as follows. For a vertex $x \in C_1$ with $x \neq v_\alpha$, set $P(x, v_\alpha) = P_1(x, v_\alpha)$ and $P(x, v_{\alpha+1}) = P_1(x, v_{\alpha+1})$ while $P(x, v_{\alpha+2}) = P_1(x, x_i) \oplus \mathbf{M}[x_i, v_{\alpha+2}]$. For a vertex $y \in C_2$ with $x \neq v_\alpha$, $P(y, v_\alpha) = P_2(y, v_\alpha)$ and $P(y, v_{\alpha+2}) = P_2(y, v_{\alpha+2})$ while $P(y, v_{\alpha+1}) = P_2(y, x_1) \oplus \mathbf{M}^r(x_1, v_{\alpha+1})$. If $i = 1$, we may choose $P(x_1, v_\alpha)$ as either $\mathbf{M}_1[x_1, v_\alpha]$ or $\mathbf{M}_2^r[x_1, v_\alpha]$.

It is straightforward to verify that $\mathcal{F}$ is a normal family for (v_1, v_2, v_3), so in the remainder of the proof we will assume that there is no interior face containing some v_α and a vertex from $\mathbf{M}[v_{\alpha+1}, v_{\alpha+2}]$.

A set $\{F_1, F_2, F_3\}$ of three distinct faces is called a *ring* if there exists a simple cycle C with the following three properties:

1. Every edge of C belongs to exactly one of the faces F_1, F_2, F_3.
2. No point in the interior of C belongs to the interior of any of the three faces F_1, F_2, F_3.
3. If $\alpha \in \{1, 2, 3\}$ and v_α is a vertex on C, then there is some $i \in \{1, 2, 3\}$ for which $v_\alpha \in F_i \cap F_{i+1}$.

Note that in the definition of a ring, we allow one of the three faces to be the exterior face. Also note that the cycle C is uniquely determined.

Case 4. $\mathbf{M}$ has a ring $\{F_1, F_2, F_3\}$.

Let C be the uniquely determined cycle that demonstrates that $\{F_1, F_2, F_3\}$ is a ring. Then there exist unique vertices u_1, u_2, u_3 on C so that u_i belongs to F_i and F_{i+1} for $i = 1, 2, 3$.

For each $i = 1, 2, 3$, let $u_i' = u_i$ if u_i has two or more neighbors outside C, i.e., u_i is the unique point shared by F_i and F_{i+1} in $\mathbf{M}$. Otherwise, let u_i' be the unique neighbor of u_i outside C. In this situation, u_i' also belongs to F_i and F_{i+1}.

We illustrate these definitions in Fig. 7. For the map shown, $\{F_1, F_2, F_3\}$ is a ring and the cycle $C = \{u_1, a, u_2, u_3, c\}$.

Let $\mathbf{M}_0$ be the submap of $\mathbf{M}$ induced by the vertices inside and on the cycle C. We may assume that the faces F_1, F_2, and F_3 have been labeled so that (u_1, u_2, u_3) is a triad for $\mathbf{M}_0$, i.e., $u_{\alpha+2} \notin \mathbf{M}_0[u_\alpha, u_{\alpha+1}]$ for each $\alpha = 1, 2, 3$. We now observe that $\mathbf{M}_0$ satisfies the star-property for (u_1, u_2, u_3). To see that this statement is valid, let $x \in V_0 - \{u_1, u_2, u_3\}$. In the map $\mathbf{M}$, there are three paths P_1, P_2, P_3 so that P_α is a path from x to v_α and $P_\alpha \cap P_{\alpha+1} = \{x\}$ for each $\alpha = 1, 2, 3$. It is clear that there is some β for which $u_\alpha \in P_{\alpha+\beta}$ for each $\alpha = 1, 2, 3$. Thus the initial segments of P_1, P_2, and P_3 show that $\mathbf{M}_0$ satisfies the star-property for the triad (u_1, u_2, u_3). By the inductive hypothesis, there is a normal family of paths $\mathcal{F}_0$ in $\mathbf{M}_0$ for (u_1, u_2, u_3).

Next, let $\mathbf{M}_1$ be the submap of $\mathbf{M}$ induced by the vertices outside C together with those elements of $\{u_1', u_2', u_3'\}$ that are on C. Then form $\mathbf{M}_2$ from $\mathbf{M}_1$, by adding a new vertex u_0 in the area formerly occupied by the interior of C and making u_0 adjacent to

u_1', u_2' and u_3'. The modified faces adjacent to u_0 in M_2 are denoted by F_1', F_2', and F_3' with $u_\alpha' \in F_\alpha' \cap F_{\alpha+1}'$ for each $\alpha = 1, 2, 3$. We illustrate this definition for the map shown in Fig. 8.

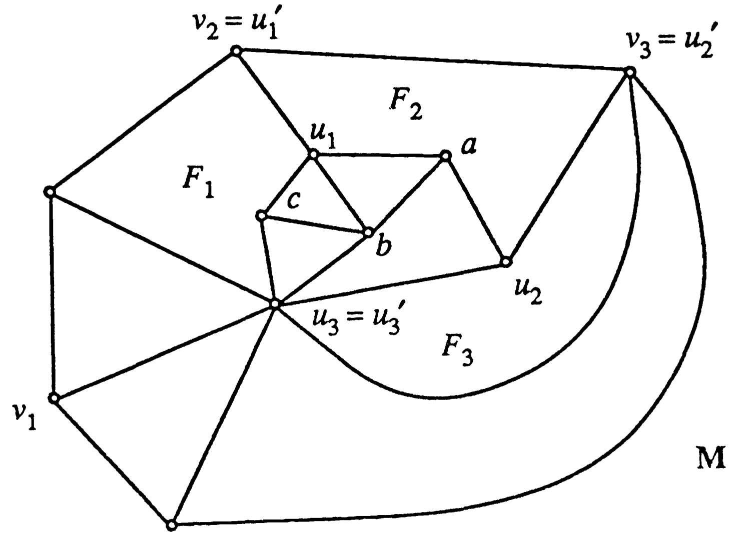


FIG. 7

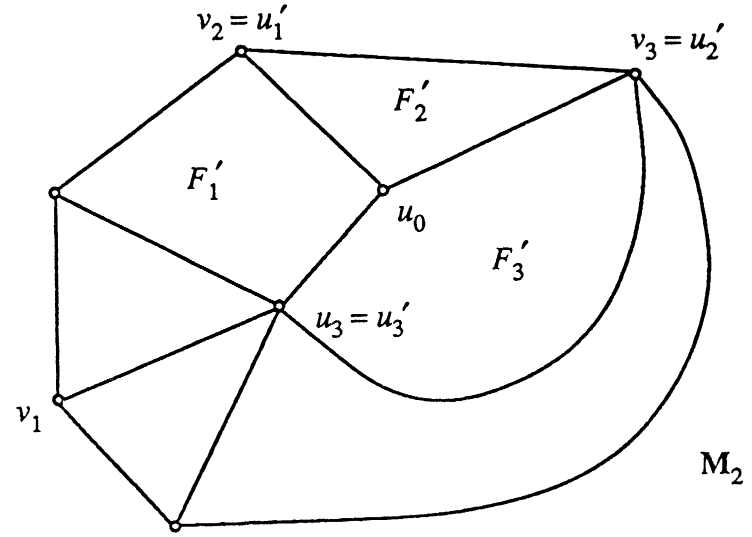


FIG. 8

We now show that (v_1, v_2, v_3) is a triad for M_2 and that M_2 satisfies the star-property for (v_1, v_2, v_3). It is obvious that (v_1, v_2, v_3) is a triad for M_2 if F_1, F_2, and F_3 are interior faces. Now suppose that one of them, say F_3, is the exterior face. In this case, the path $\mathrm{M}[u_2', u_3']$ is a portion of the boundary of M. In M_2, this path is replaced by $u_2' \oplus (u_2, u_0, u_3) \oplus u_3'$, so that (v_1, v_2, v_3) is also a triad for M_2.

Next, we show that $\mathbf{M}_2$ satisfies the star-property for (v_1, v_2, v_3). To the contrary, suppose that there exists a vertex $x \in V_2 - \{v_1, v_2, v_3\}$ for which there are two vertices y, z in $V_2 - \{x\}$ that separate x from $\{v_1, v_2, v_3\}$ in $\mathbf{M}_2$.

First, consider the case where $x = u_0$. Choose $\alpha \in \{1, 2, 3\}$ so that $u'_\alpha \notin \{y, z\}$. Clearly, $u'_\alpha \notin \{v_1, v_2, v_3\}$, so that in $\mathbf{M}$, there exist paths P_1, P_2, P_3, so that P_β is a path from u'_α to v_β and $P_\beta \cap P_{\beta+1} = \{u'_\alpha\}$ for each $\beta = 1, 2, 3$. Since $y, z \in V - \{u'_\alpha\}$, at least one of these paths, say P_γ, misses y and z in $\mathbf{M}$. If P_γ is a path in $\mathbf{M}_2$, we are done. Otherwise, P_γ contains at least two elements of $\{u_1, u_2, u_3\}$. Let P'_γ be the terminal segment of P_γ beginning with the last occurrence of an element of $\{u_1, u_2, u_3\}$ in P_γ. Then $u_0 \oplus P'_\gamma$ is a path from u_0 to v_γ in $\mathbf{M}_2 - \{y, z\}$.

Next, consider the case where $x \in V - \{u'_1, u'_2, u'_3\}$. Since $\mathbf{M}$ satisfies the star-property for (v_1, v_2, v_3), there exist paths P_1, P_2, P_3, so that P_α is a path from x to v_α and $P_\alpha \cap P_{\alpha+1} = \{x\}$ for each $\alpha = 1, 2, 3$. Any one of these three paths that is not a path in $\mathbf{M}_2$ must contain at least two elements of $\{u_1, u_2, u_3\}$, so at least two of P_1, P_2, and P_3 are paths in $\mathbf{M}_2$. So we may assume that P_α and $P_{\alpha+1}$ are paths in $\mathbf{M}_2$ with $y \in P_\alpha$ and $z \in P_{\alpha+1}$. We may also assume that $P_{\alpha+2}$ contains at least two elements from $\{u_1, u_2, u_3\}$. Let u_β be the first element from this set that belongs to $P_{\alpha+2}$ and let u_γ be the last. Then replace the portion of $P_{\alpha+2}$ beginning at u_β and ending with u_γ with (u_β, u_0, u_γ) to obtain a path from x to $v_{\alpha+2}$ in $\mathbf{M}_2 - \{y, z\}$.

Now suppose $x \in \{u'_1, u'_2, u'_3\}$. If neither y nor z is u_0, then y and z are vertices in $\mathbf{M}$, so there is a path P in $\mathbf{M}$ from x to $\{v_1, v_2, v_3\}$, with P avoiding y and z. If P is a path in $\mathbf{M}_2$, we are done. So we conclude that P contains at least two vertices from $\{u_1, u_2, u_3\}$. Let u_α be the last vertex from $\{u_1, u_2, u_3\}$, which belongs to P, and let P' be the terminal segment of P beginning at u_α. Then $(x, u_0, u_\alpha) \oplus P'$ is a path from x to $\{v_1, v_2, v_3\}$ in $\mathbf{M}_2$, which avoids y and z.

It remains only to consider the case where $x = u'_\alpha$ and one of the separating vertices, say y, is equal to u_0. If x has a neighbor w in $\mathbf{M}_2 - \{y, z, u'_1, u'_2, u'_3\}$, then we have that there is a path P in $\mathbf{M}_2$ from w to $\{v_1, v_2, v_3\}$ avoiding y and z. Then $(x, w) \oplus P$ is the desired path in $\mathbf{M}_2$. On the other hand, if x has no neighbor in $\mathbf{M}_2$ outside the set $\{y, z, u'_1, u'_2, u'_3\}$, then it is adjacent to one of the other $u'_1 \notin \{y, z\}$, say u'_β. Now if u'_β has a neighbor w in $\mathbf{M}_2 - \{y, z, u'_1, u'_2, u'_3\}$, then again there is a path P from w to $\{v_1, v_2, v_3\}$ in $\mathbf{M}_2$ avoiding y and z, which yields a path $(x, u'_\beta) \oplus P$ as required. Finally, if u'_β also has no neighbor outside $\{y, z, u'_1, u'_2, u'_3\}$, then the two vertices z and u'_γ, where $\gamma \notin \{\alpha, \beta\}$, separate u'_α (and also u'_β) from $\{v_1, v_2, v_3\}$ in $\mathbf{M}$, a contradiction.

This completes the argument that $\mathbf{M}_2$ satisfies the star-property for $\{v_1, v_2, v_3\}$. Now let $\mathcal{F}_2$ be a normal family of paths in $\mathbf{M}_2$ for the triad (v_1, v_2, v_3). We may assume without loss of generality that $u'_\alpha \in P(u_0, v_\alpha)$ for each $\alpha = 1, 2, 3$. We use $\mathcal{F}_0$ and $\mathcal{F}_2$ to construct a normal family for $\mathbf{M}$ as follows. Let $x \in V$. If $x \in V_0$, set $P(x, v_\alpha) = P_0(x, u_\alpha) \oplus P_2(u'_\alpha, v_\alpha)$. If $x \in V - V_0$, set $P(x, v_\alpha) = P_2(x, v_\alpha)$ when $u_0 \notin P_2(x, v_\alpha)$. If $x \in V - V_0$ and $u_0 \in P_2(x, v_\alpha)$, choose the unique elements u'_β, u'_γ for which u'_β precedes u_0 and u'_γ follows u_0 in $P_2(x, v_\alpha)$. Replace this portion of the path by $u'_\beta \oplus P_0(u_\beta, u_\gamma) \oplus u'_\gamma$. Verification that the resulting family of paths is normal is straightforward. Accordingly, we will assume in what follows that $\mathbf{M}$ does not contain a ring.

Case 5. We now present the closing argument.

Let v'_1 be the second vertex on the path $\mathbf{M}[v_1, v_2]$. Let $\mathbf{M}_0$ be the submap of $\mathbf{M}$ obtained by deleting v_1. On the path $P = \mathbf{M}_0[v_3, v'_1]$, rub out all vertices of degree 2 that are strictly between the end points of P. Call the resulting map $\mathbf{M}_1$.

Suppose there is a face G interior to $\mathbf{M}_1$ whose intersection with P does not form a single subpath. Let y and z be distinct vertices of P on G, with y on $\mathbf{M}_0[v_3, z]$, such that $\mathbf{M}_0[y, z] \cap G = \{y, z\}$.

There are two paths from z to y around the face G. Let P_0 be the one "nearer" to $\mathbf{M}_0[y, z]$, and let T be the region bounded by P_0 and $\mathbf{M}_0[y, z]$. This region has nonempty interior and contains some vertex w of $\mathbf{M}$ other than y and z, since $\mathbf{M}$ has no multiple edges. Clearly, $w \notin \{v_1, v_2, v_3\}$, so there is a path from w to $\{v_1, v_2, v_3\}$ in $\mathbf{M}$ avoiding $\{y, z\}$. There is no such path in $\mathbf{M}_1$, so this path must go to v_1 via an edge from some vertex strictly between y and z on $\mathbf{M}_0$.

Let $w_1, \ldots, w_k$ be the vertices strictly between y and z on P, in the order they occur on P: we have just shown that $k \geq 1$. If $k = 1$, let F_1 and F_2 be the two faces incident with the edge $v_1 w_1$ in $\mathbf{M}$. Then (F_1, F_2, G) forms a ring, with the cycle C being the boundary of T, contradicting our assumption that $\mathbf{M}$ has no ring. If $k > 1$, let F_1 be the face incident with edge $v_1 w_1$ and not including w_2; and let F_2 be the face incident with $v_1 w_k$ and not including w_{k-1}. Again, (F_1, F_2, G) forms a ring, with the cycle C consisting of the boundary of T, with $\mathbf{M}_0[w_1, w_k]$ replaced by the two edges $w_1 v_1$ and $v_1 w_k$. Again, this is a contradiction, so there is no such face G.

In particular, $\mathbf{M}_1$ has no multiple edges.

Now it is easy to see that (v_1', v_2, v_3) is a triad for $\mathbf{M}_1$. We next show that $\mathbf{M}_1$ satisfies the star-property for (v_1', v_2, v_3); suppose not. Choose $x \in V_1 - \{v_1', v_2, v_3\}$ for which there exist two vertices $y, z \in V_1 - \{x\}$ that separate x from $\{v_1', v_2, v_3\}$ in $\mathbf{M}_1$. Since $\mathbf{M}$ satisfies the star-property, there exists a path P' from x to one of $\{v_1, v_2, v_3\}$ with P' missing y and z. It is obvious that P' terminates at v_1. Thus P' contains a vertex w from the path $P = \mathbf{M}_0[v_3, v_1']$.

Hence y and z both lie on P, one either side of w. We may suppose that $y \in \mathbf{M}_1[v_3, w]$ and $z \in \mathbf{M}_1[w, v_1']$. If y and z do not share a face inside $\mathbf{M}_1$, then $\mathbf{M}_1 - y - z$ is connected, a contradiction. Thus, y and z do share a face G inside $\mathbf{M}_1$, which therefore contains the whole of $\mathbf{M}_1[y, z]$. So w lies on $\mathbf{M}_1[y, z]$ and has degree 2 in $\mathbf{M}_1$, a contradiction.

Thus, $\mathbf{M}_1$ satisfies the star-property for (v_1', v_2, v_3). Now let $\mathcal{F}_1$ be a normal family of paths in $\mathbf{M}_1$ for (v_1', v_2, v_3). We construct a family $\mathcal{F}$ of paths in $\mathbf{M}$ as follows:

1. For every vertex $x \in V_1$, $P(x, v_\alpha) = P_1(x, v_\alpha)$ for $\alpha = 2, 3$.

2. For every vertex $x \in V_1$, let y be the first vertex on $P_1(x, v_1')$, which is adjacent to x in M and let $P_1(x, y)$ be the initial segment of this path ending at g. Then set $P(x, v_1) = P_1(x, y) \oplus v_1$.

3. For every vertex$\in V - V_1$ with $x \neq v_1$, set $P(x, v_1) = (x, v_1)$, $P(x, v_2) = \mathbf{M}_0[x, v_2]$ and $P(x, v_3) = \mathbf{M}_0^r[x, v_3]$.

It is an easy exercise to verify that $\mathcal{F}$ is a normal family of paths. This completes the proof of Lemma 4.1. □

5. The lower bound. For the sake of completeness, we include a proof of the following result, which is also proved in [7].

THEOREM 5.1. *If* $\mathbf{M}$ *is a convex polytope in* $\mathbb{R}^3$, *then* $\dim(\mathbf{P_M}) \geq 4$.

Proof. Suppose to the contrary that $\dim(\mathbf{P_M}) \leq 3$. Choose linear extensions L_1, L_2, L_3 of $\mathbf{P_M}$, so that $\mathbf{P_M} = L_1 \cap L_2 \cap L_3$. Of all the faces, let F_0 be the L_3-least. Then let $x_1, x_2, \ldots, x_t$ be the vertices of F_0 and let $G_1, G_2, \ldots, G_t$ be the faces that share an edge with F_0. We may assume that these vertices and faces have been labeled so that $x_i \in G_i \cap G_{i+1}$ for $i = 1, 2, \ldots, t$.

Now $x_i < F_0 < G_j$ in L_3 for each i, j with $1 \leq i, j \leq t$. However, the subposet $\mathbf{P}_0$ of $\mathbf{P_M}$ generated by $\{x_i : 1 \leq i \leq t\} \cup \{G_i : 1 \leq i \leq t\}$ is isomorphic to a three-dimensional crown. The linear extension L_3 reverses no critical pairs of $\mathbf{P}_0$, which means they must all be reversed by L_1 and L_2. Since $\dim(\mathbf{P}_0) = 3$, this is impossible. □

Note that this argument actually shows that the subposet of $\mathbf{P}_\mathbf{M}$ consisting of vertices and faces has dimension at least 4.

6. Irreducible posets and duality. For $t \geq 2$, a poset $\mathbf{P}$ is said to be t irreducible if $\dim(\mathbf{P}) = t$ and $\dim(\mathbf{P} - x) < t$ for every $x \in \mathbf{P}$. The only 2-irreducible poset is a 2-element antichain. In [5] and [17], the collection of all 3-irreducible posets is determined. The posets in this collection can be grouped into seven infinite families with an additional eleven sporadic examples. For $t \geq 4$, constructions of t-irreducible posets are given in [4], [8], [9], and [14].

We find it interesting to note that each convex polytope in $\mathbb{R}^3$ determines a 4-irreducible poset in a natural manner.

THEOREM 6.1. *Let* M *be a convex polytope in* $\mathbb{R}^3$ *and let* F_0 *be an arbitrary face of* M. *Then the subposet* $\mathbf{Q}_0 = \mathbf{P}_\mathbf{M} - \{F_0\}$ *is three-dimensional.*

Proof. Consider a plane drawing of the map M so that F_0 is the exterior face. Choose vertices v_1, v_2, v_3 on F_0 so that (v_1, v_2, v_3) is a triad. Then let $\mathcal{F}$ be a normal family of paths for (v_1, v_2, v_3).

Now consider the critical pairs in $\mathbf{Q}_0$. In addition to the Type 1 critical pairs of the form (y, F), where F is an interior face and $y \notin F$, we also have Type 2 critical pairs of the following form.

Type 2: (x, e), where x is a vertex on an interior face F, e is an edge common to F and the exterior face F_0, and x is not an end point of e.

Let L_1, L_2, and L_3 be the linear orders on V defined in the proof of Theorem 3.6. Extend L_1, L_2, and L_3 to linear extensions of $\mathbf{Q}_0$ by inserting the edges and faces as low as possible in each of the three orders. Call the resulting orders L_1^*, L_2^*, L_3^*. We show $\mathbf{Q}_0 = L_1^* \cap L_2^* \cap L_3^*$. It suffices to show that each Type 2 critical pair is reversed in some L_i^*. (We know from 3.6 that the Type 1 critical pairs are automatically reversed.)

Let (x, e) be a Type 2 critical pair. Let y and z denote the two end points of e. Choose α so that $F \subseteq S(x, \alpha)$. Then $y, z \in \mathbf{M}[v_{\alpha+1}, v_{\alpha+2}]$, $\emptyset = S(y, \alpha) = S(z, \alpha) \subsetneq S(x, \alpha)$. So it follows that (y, x) and (z, x) belong to Q_α. Thus, $x > e$ in L_α^*. □

When M is a planar 3-connected map, the planar dual $\mathbf{M}^d$ of M is also 3-connected. Furthermore, it is easy to see that the poset associated with the dual of M is the dual of the poset associated with M. With this observation, we obtain the following dual form of the preceding theorem as well as the corollary summarizing the net effect of the two.

THEOREM 6.2. *Let* M *be a convex polytope in* $\mathbb{R}^3$ *and let* x *be an arbitrary vertex of* M. *Then the subposet* $\mathbf{Q}_1 = \mathbf{P}_\mathbf{M} - \{x\}$ *is three-dimensional.*

COROLLARY 6.3. *Let* M *be a convex polytope in* $\mathbb{R}^3$. *Then the subposet of* M *determined by the vertices and faces is* 4*-irreducible.*

7. Concluding remarks. As mentioned earlier, we have been able to establish the upper bound $\dim(\mathbf{P}_\mathbf{M}) \leq 4$, on the dimension of $\mathbf{P}_\mathbf{M}$ when M is an arbitrary planar map. In the most general setting, we allow disconnected maps, loops, and multiple edges. However, we do not have an independent proof of this result. Our argument depends heavily on having the results and techniques of this paper in hand.

It is perhaps interesting to note here that the analogue of Theorem 6.1 does not hold for general planar maps. In the map M shown below (see Fig. 9), each critical pair (x_i, F_i) must be reversed in a different linear extension of P_M.

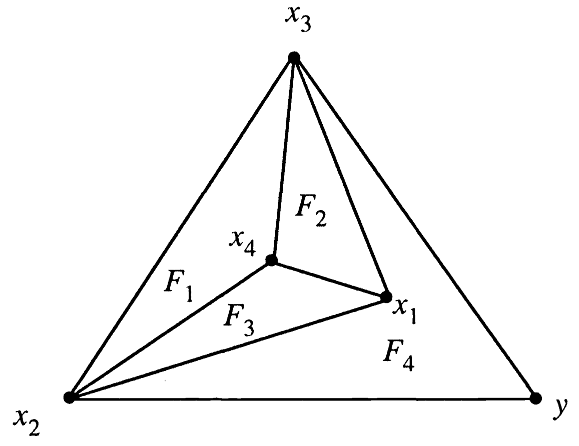


FIG. 9

It is relatively straightforward to show that for maps drawn on a surface of genus n, there is an upper bound of the form $\dim(\mathbf{P_M}) \leq f(n)$. It would be of some interest to determine $f(n)$. Perhaps the correct answer is $f(n) = n + 4$.

Acknowledgment. The authors gratefully acknowledge the assistance of Klaus Reuter, who posed this problem to us and encouraged us in this research.

REFERENCES

[1] G. BIRKHOFF, *Lattice Theory*, 3rd ed., Amer. Math. Soc. Colloq. Publ. 25, Providence, RI, 1967.
[2] B. DUSHNIK AND E. W. MILLER, *Partially ordered sets*, Amer. J. Math., 63 (1941), pp. 600–610.
[3] M. C. GOLUMBIC, *Algorithmic Graph Theory and Perfect Graphs*, Chap. 5, Academic Press, New York, 1980.
[4] D. KELLY, *On the dimension of partially ordered sets*, Discrete Math., 35 (1981), pp. 135–156.
[5] ———, *The 3-irreducible partially ordered sets*, Canad. J. Math., 29 (1977), pp. 367–383.
[6] D. KELLY AND W. T. TROTTER, *Dimension theory for ordered sets*, in Proc. Sympos. Ordered Sets, I. Rival et al., eds., Reidel Publishing, Dordrecht, 1982, pp. 171–212.
[7] K. REUTER, *On the order dimension of convex polytopes*, preprint.
[8] J. A. ROSS AND W. T. TROTTER, *Every t-irreducible partial order is a subposet of a $t+1$-irreducible partial order*, Annal. Discrete Math., 17 (1983), pp. 613–621.
[9] ———, *For $t \geq 3$, Every t-dimensional partial order can be embedded in a $t+1$-irreducible partial order*, in Finite and Infinite Sets, A. Hajnal, L. Lovász, and V. T. Sös, eds., Colloq. Math. Soc. J. Bolyai, 37 (1984), pp. 711–732 (with J. Ross).
[10] W. SCHNYDER, *Planar graphs and poset dimension*, Order, 5 (1989), pp. 323–343.
[11] V. SEDMAK, *Sur les réseaux de polyèdres n-dimensionnels*, C. R. Acad. Sci. Paris, 248 (1959), pp. 350–352.
[12] J. SPENCER, *Minimal scrambling sets of simple orders*, Acta. Math. Acad. Sci. Hungar., 22 (1971), pp. 349–353.
[13] E. STEINITZ, *Vorlesungen über die Theorie der Polyeder*, Springer, Berlin, 1934.
[14] W. T. TROTTER, *Dimension of the crown S_n^k*, Discrete Math., 8 (1974), pp. 85–103.
[15] ———, *Graphs and partially ordered sets*, in Selected Topics in Graph Theory II, R. Wilson and L. Beineke, eds., Academic Press, New York, 1983, pp. 237–268.
[16] ———, *Partially ordered sets*, in Handbook of Combinatorics, R. Graham, M. Groetschel, and L. Lovász, eds., to appear.
[17] W. T. TROTTER AND J. MOORE, *Characterization problems for graphs, partially ordered sets, lattices, and families of sets*, Discrete Math., 16 (1976), pp. 361–381.

SIAM J. DISC. MATH.
Vol. 6, No. 2, pp. 246–269, May 1993

MINKOWSKI ADDITION OF POLYTOPES: COMPUTATIONAL COMPLEXITY AND APPLICATIONS TO GRÖBNER BASES*

PETER GRITZMANN† AND BERND STURMFELS‡

Abstract. This paper deals with a problem from computational convexity and its application to computer algebra. This paper determines the complexity of computing the Minkowski sum of k convex polytopes in $\mathbb{R}^d$, which are presented either in terms of vertices or in terms of facets. In particular, if the dimension d is fixed, the authors obtain a polynomial time algorithm for adding k polytopes with up to n vertices. The second part of this paper introduces dynamic versions of Buchberger's Gröbner bases algorithm for polynomial ideals. Using the Minkowski addition of Newton polytopes, the authors show that the following problem can be solved in polynomial time for any finite set of polynomials $\mathcal{T} \subset K[x_1, \dots, x_d]$, where d is fixed: Does there exist a term order τ such that $\mathcal{T}$ is a Gröbner basis for its ideal with respect to τ? If not, find an optimal term order for $\mathcal{T}$ with respect to a natural Hilbert function criterion.

Key words. computational complexity, Minkowski sum, polytope, hyperplane arrangement, mathematical programming, computational convexity, computer algebra, Gröbner bases, polynomial, term order, Newton polytope, Hilbert function

AMS(MOS) subject classifications. 68U05, 68Q15, 68Q40, 52A25

1. Introduction.

1.1. General introduction. Although geometric and combinatorial properties of Minkowski sums of convex polytopes have been studied for a long time [7], these properties and techniques have only recently been applied to questions in mathematical programming and computer science [2], [6], [13]. An important special class of polytopes to be mentioned in this context are the *zonotopes* [20], which are obtained as the Minkowski sum of line segments.

The present paper has the twofold objective of investigating the computational aspects of Minkowski addition of polytopes and of applying the results, via Newton polytopes of polynomials, to a class of problems in computer algebra. The main emphasis of our complexity analysis lies on deciding polynomial-time computability.

This paper addresses readers from both (computational) convexity and (computer) algebra, aiming to provide a bridge between both subjects. It is organized in such a way that either of the two parts can be accessed separately.

In §§1.2 and 1.3 we start out with two introductions containing the basic background, sufficiently detailed for understanding the application of Minkowski sums of polytopes to Gröbner bases theory. In §1.4 we summarize our main results. Section 2 deals with the Minkowski addition of polytopes, first, in §2.1 from a geometric and combinatorial point of view. Emphasis will lie on the question of how many faces of given dimension such Minkowski sums can have. In §2.2 we will briefly discuss algorithmic aspects of zonotopes and their relation to arrangements of hyperplanes. This relation enables us to utilize the algorithm in [14], [15] for constructing arrangements. Section 2.3 deals with the complexity of computing Minkowski sums of polytopes. Section 3 contains our

*Received by the editors March 19, 1990; accepted for publication (in revised form) April 20, 1992. This research was supported in part by the Deutsche Forschungsgemeinschaft. Both authors acknowledge travel support from the Center for Discrete Mathematics and Theoretical Computer Science (DIMACS).

†Universität Trier, Fb IV, Mathematik, Postfach 3825, W-5500 Trier, Germany. Part of this paper was written while this author was visiting the University of Washington.

‡Cornell University, Department of Mathematics, Ithaca, New York 14853.

new results on Gröbner bases. We will give a dynamic version of Buchberger's algorithm (§3.1), study in detail the relation between term orders and the Minkowski sum of Newton polytopes (§3.2), and introduce methods based on Hilbert functions for finding optimal term orders (§3.3).

We recommend that readers who are only interested in one of the two subjects read §1 and the respective §2 or §3. General references for the topics of our paper are [7] for Minkowski addition, [21] for the theory of polytopes, [13] for computational geometry, [17] for computational complexity, and [9] for Gröbner bases and computer algebra in general.

1.2. Preliminaries on the Minkowski addition of polytopes. A *polytope* is the convex hull of finitely many points in $\mathbb{R}^d$. In particular, polytopes are convex compact subsets of $\mathbb{R}^d$, but we do not require that they be full-dimensional. The *Minkowski sum* $P_1 + P_2$ of two polytopes P_1 and P_2 in $\mathbb{R}^d$ is the polytope

$$P_1 + P_2 = \{ x \in \mathbb{R}^d \mid \exists x_1 \in P_1, x_2 \in P_2 : x = x_1 + x_2 \}.$$

Here P_1 and P_2 are called *summands* of $P_1 + P_2$, and the binary operation $+$ is called *Minkowski addition* (*of polytopes*). Minkowski addition is commutative and associative and thus generalizes naturally to more than two polytopes.

A special case that has received considerable attention in the applied mathematics literature is the case where the polytopes degenerate to line segments: Their Minkowski sum is a *zonotope*. Zonotopes turn up—explicitly or implicitly—in linear programming [20], in the problem of *maximizing quasiconvex functionals* [6], in the *flow shop problem* [2], and in the *minsum problem* [20], [23].

Another reason for the importance of zonotopes is the fact that they are equivalent under polarity to arrangements of hyperplanes (see §2.2), which play a central role in computational geometry (cf. [13]). Arrangements of hyperplanes are the geometric cell complexes induced by the dissection of $\mathbb{R}^d$ by a given set of hyperplanes. In the Gröbner bases application to be discussed in §3.2, the special case of zonotopes corresponds to the word problem for commutative semigroups.

In §2 of this paper, we study computational aspects of the general problem of Minkowski addition of k rational polytopes in d-space. For our complexity analysis, we distinguish the case of the dimension d and the number k of polytopes being part of the input from the cases in which one (or both) of these numbers is regarded as a constant. Special emphasis will be placed on the case when the dimension d is fixed and k is large. It turns out that (unless d is a constant) the results depend on how each input polytope is presented, namely, either as the convex hull of finitely many points or as the intersection of finitely many closed half-spaces.

A $\mathcal{V}$-*presentation* of a polytope $P \subset \mathbb{R}^d$ consists of integers n and d with $n > d \geq 1$, and n points $v_1, \ldots, v_n$ in $\mathbb{R}^d$ such that $P = \operatorname{conv}\{v_1, \ldots, v_n\}$. The number nd is called the *size in the real model*, or for short the *real size*, of this presentation. An $\mathcal{H}$-*presentation* of a polytope P consists of integers n and d with $n > d \geq 1$, a real $n \times d$ matrix A, and a vector $b \in \mathbb{R}^d$ such that $P = \{x \in \mathbb{R}^d | Ax \leq b\}$. The number $nd + n$ is called the *real size* of this presentation.

The *binary size* of the given presentation of a rational polytope P (usually denoted by L) is the number of binary digits needed to encode the data of the presentation. Here P being *rational* means $v_1, \ldots, v_n \in \mathbb{Q}^d$ if P is $\mathcal{V}$-presented, or the matrix A has rational entries and $b \in \mathbb{Q}^d$ if P is $\mathcal{H}$-presented. When speaking of binary size, we always assume that the polytope is rational. Note that the Newton polytopes to be considered in our Gröbner basis application are $\mathcal{V}$-presented and have integer vertices.

A $\mathcal{V}$- ($\mathcal{H}$-) presentation of a polytope is called *irredundant* if the omission of any of the points $v_1, \ldots, v_n$ (of any of the inequalities in $Ax \leq b$) changes the polytope. Geometrically, a $\mathcal{V}$-presentation is irredundant if each point v_i is a vertex of P, and, if P is d-dimensional, an $\mathcal{H}$-presentation is irredundant if each inequality induces a facet of P.

Each polytope $P \subset \mathbb{R}^d$ admits a $\mathcal{V}$-presentation and also admits an $\mathcal{H}$-presentation, and we refer to [12], [33], [34] for algorithms that convert one presentation into the other. However, because P may have many more vertices than facets (and vice versa) [26], it can happen that the minimum size of one presentation is much larger than the minimum size of the other presentation.

In the following, we assume that the desired (input and output) presentations of our polytopes are specified beforehand. More precisely, we assume that we are given a sequence

$$\Pi = (\mathcal{W}; \mathcal{W}_1, \mathcal{W}_2, \ldots), \quad \text{where} \quad \mathcal{W}, \mathcal{W}_i \in \{\mathcal{V}, \mathcal{H}\} \qquad (i = 1, 2, \ldots).$$

With this notation, we can formalize the problem to be studied in §2.

Π-MINKADD.

Input: $d, k \in \mathbb{N}$, *and a* $\mathcal{W}_i$*-presented polytope* P_i *in* $\mathbb{R}^d$, *for each* $i = 1, \ldots, k$.

Output: *An irredundant* $\mathcal{W}$*-presentation of the Minkowski sum* $P_1 + \cdots + P_k$.

The case of fixed k or/and fixed d will be denoted by FIXED-K-Π-MINKADD, FIXED-D-Π-MINKADD, and FIXED-K-D-Π-MINKADD, respectively. In the case that the sequence Π is constant, say $\Pi = (\mathcal{W}; \mathcal{W}, \mathcal{W}, \ldots)$, we will sometimes use the abbreviation $\mathcal{W}$ for Π and write, for example, $\mathcal{W}$-MINKADD instead of Π-MINKADD.

1.3. Preliminaries on Gröbner bases and Newton polytopes. Gröbner bases are a unifying method in computer algebra that simultaneously generalize the following well-known algorithms:

(1) the Euclidean algorithm, in the case of univariate polynomials,
(2) Gaussian elimination, in the case of linear polynomials, and
(3) classical elimination theory, in the case of d homogeneous polynomials in d variables.

The basic procedure (Algorithm 1.3.3) for computing a Gröbner basis of a polynomial ideal is due to Buchberger [8]. It is implemented in all major computer algebra systems (e.g., MAPLE, MACSYMA, REDUCE, MATHEMATICA), and calculating examples with one of these systems is a good way of familiarizing ourselves with the subject. In spite of their striking simplicity, Gröbner bases can be used to solve a wide range of problems from computational algebraic geometry. Examples are solving algebraic equations (over the complex numbers), computing dimension, singularities and irreducible decompositions of algebraic varieties, implicitization of parametric representations of curves and surfaces, or symbolic inversion of polynomial mappings (e.g., inverse kinematics in robot programming) [10].

Let $K[\mathbf{x}]$ denote the polynomial ring in d variables $\mathbf{x} = (x_1, \ldots, x_d)$ over a field K. Via the usual identification of a monomial $\mathbf{x}^\alpha$ with its exponent vector α, we can think of (the underlying vector space of) $K[\mathbf{x}]$ as the (infinite-dimensional) K-vector space spanned by the basis $\mathbb{N}^d$. A linear order "$\prec$" on the set of all monomials is called a *term order* if it respects the semigroup structure of $\mathbb{N}^d$, that is, if $1 \preceq \mathbf{x}^\alpha$ and ($\mathbf{x}^\alpha \prec \mathbf{x}^\beta \Rightarrow \mathbf{x}^\alpha \mathbf{x}^\gamma \prec \mathbf{x}^\beta \mathbf{x}^\gamma$) for all monomials $\mathbf{x}^\alpha, \mathbf{x}^\beta, \mathbf{x}^\gamma \in K[\mathbf{x}]$. Note that here the monomial $1 = \mathbf{x}^0 = x_1^0 x_2^0 \ldots x_d^0$ corresponds to the basis element $(0, 0, \ldots, 0) \in \mathbb{N}^d$. The following representation lemma is due to Ostrowski [29], [30, Thm. IV] (see also [32]).

LEMMA 1.3.1. *Let $\prec$ be a term order on $K[\mathbf{x}]$ and let $R \in \mathbb{N}$. Then there exists a positive weight vector $w \in \mathbb{R}^d_+$ such that, for all monomials $\mathbf{x}^\alpha, \mathbf{x}^\beta$ of total degree $\leq R$, $\mathbf{x}^\alpha \prec \mathbf{x}^\beta$ if and only if $\langle \alpha, w\rangle < \langle \beta, w\rangle$.*

Here $\langle \alpha, w\rangle$ denotes the dot product. It is easy to see that, conversely, every positive weight vector $w \in \mathbb{R}^d_+$ defines a term order $\prec$ on $K[\mathbf{x}]$, provided it separates the monomials. For example, if $d = 3$, then the weight vector $w = (10000, 100, 1)$ represents the *purely lexicographic order* $1 \prec x_3 \prec x_3^2 \prec x_3^3 \prec \cdots \prec x_2 \prec x_2x_3 \prec x_2x_3^2 \prec x_2x_3^3 \prec \cdots \prec x_2^2 \prec x_2^2x_3 \prec x_2^2x_3^2 \prec \cdots \prec x_1 \prec x_1x_3 \prec \cdots$ for all monomials $x_1^ix_2^jx_3^k$ of total degree $i+j+k \leq 99$.

In the following, let $\prec$ be any fixed term order on $K[\mathbf{x}]$. The *initial monomial* $init_\prec(t)$ of a polynomial $t \in K[\mathbf{x}]$ is then defined as the largest monomial in $\mathbb{N}^d$ that appears in t with a nonzero coefficient. Given any ideal $\mathcal{I} \subset K[\mathbf{x}]$, then its *initial ideal* $init_\prec(\mathcal{I})$is generated by the monomials $init_\prec(t)$, where $t \in \mathcal{I}$. A finite subset $\mathcal{G} = \{g_1, g_2, \ldots, g_l\}$ of $\mathcal{I}$ is called a *Gröbner basis* for $\mathcal{I}$ provided $init_\prec(\mathcal{I})$ is generated by $init_\prec(\mathcal{G}) = \{init_\prec(g_1), \ldots, init_\prec(g_l)\}$. Note that minimality is not required here. It follows as a consequence that $\mathcal{I}$ is generated by $\mathcal{G}$.

EXAMPLE 1.3.2. Consider the ideal $\mathcal{I} \subset K[x_1, x_2]$, which is generated by $t_1 := x_1^2 + x_2^2 - 1$ and $t_2 := 3x_1x_2 - 1$. Let $\prec$ be the purely lexicographic order induced by $x_2 \prec x_1$. Then $init_\prec(\mathcal{I})$ is generated by the monomials x_1 and x_2^4, and a Gröbner basis for $\mathcal{I}$ is given by $\mathcal{G} = \{x_1 + 3x_2^3 - 3x_2, 9x_2^4 - 9x_2^2 + 1\}$. Note that from the Gröbner basis $\mathcal{G}$ we can easily compute coordinates for the four intersection points of the unit circle given by $t_1 = 0$ and the hyperbola $t_2 = 0$. We remark that lexicographic Gröbner bases are always *triangularized* in the sense that $\mathcal{I} \cap K[x_1, \ldots, x_i] =< \mathcal{G} \cap K[x_1, \ldots, x_i] >$ for $i = 1, \ldots, d$ [9, Lemma 6.8].

THEOREM AND ALGORITHM 1.3.3 (Buchberger). *The following procedure transforms any generating set $\mathcal{G}_0$ of $\mathcal{I}$ into a Gröbner basis:*

```
$i := -1$
REPEAT
  $i := i + 1$
  $\mathcal{G}_{i+1} := \mathcal{G}_i \cup \big(\{\ \texttt{normalform}_{\mathcal{G}_i,\prec}(\texttt{S-polynomial}_\prec(p_1, p_2)) | p_1, p_2 \in \mathcal{G}_i\} \setminus \{0\}\big)$
UNTIL $\mathcal{G}_{i+1} = \mathcal{G}_i$.
```

We must explain the two abbreviations used in this algorithm. Given two polynomials $p_1, p_2 \in K[\mathbf{x}]$ (with leading coefficient 1—otherwise the following has to be modified in an obvious way), then S-polynomial$_\prec(p_1, p_2)$ denotes the polynomial $m_1 \cdot p_1 - m_2 \cdot p_2$, where m_1 and m_2 are the unique monomials satisfying

$$m_1 \cdot init_\prec(p_1) = m_2 \cdot init_\prec(p_2) = \text{least common multiple}\big(init_\prec(p_1), init_\prec(p_2)\big).$$

For instance, in Example 1.3.2 we have

$$\text{S-polynomial}_\prec(t_1, t_2) = 3x_2 \cdot t_1 - x_1 \cdot t_2 = x_1 + 3x_2^3 - 3x_2.$$

A *normal form* of a polynomial p with respect to a polynomial set $\{p_1, \ldots, p_l\}$ is obtained by successively replacing occurrences of $init_\prec(p_i)$ (as factors of terms) in p by $p_i - init_\prec(p_i)$. The following example shows that these normal forms are usually not unique (but any choice will do in Algorithm 1.3.3). Given $p = x_1^2x_2$ and $t_1, t_2, \prec$ as in Example 1.3.2 (and assuming $char(K) \neq 3$), then both $-x_2^3 + x_2$ and $\frac{1}{3}x_1$ are normal forms of p with respect to $\{t_1, t_2\}$.

COROLLARY 1.3.4. *A set $\mathcal{G} \subset K[\mathbf{x}]$ is a Gröbner basis if and only if the* S*-polynomial of any two elements of $\mathcal{G}$ reduces to* 0.

We next summarize some important results about Gröbner bases; see [3], [9], [25], [32], [36] for details and further references. As before, we fix a term order $\prec$.

(1) A set $\mathcal{G} \subset K[\mathbf{x}]$ is a Gröbner basis (for the ideal it generates) if and only if every $p \in K[\mathbf{x}]$ has a unique normal form with respect to $\mathcal{G}$.

(2) If the initial monomials of elements in $\mathcal{G}$ are pairwise relatively prime, then $\mathcal{G}$ is a Gröbner basis.

(3) Every ideal $\mathcal{I} \subset K[\mathbf{x}]$ has a unique *reduced* Gröbner basis $\mathcal{G} = \{g_1, g_2, \ldots, g_l\}$. This means that no monomial of g_i is a multiple of $init(g_j)$ for $i \neq j$ and that all g_i have leading coefficient 1.

(4) Let $\mathcal{I} \subset K[\mathbf{x}]$ be an ideal generated by polynomials of (total) degree $\leq R$ and let $\mathcal{G}$ be its reduced Gröbner basis. Then all polynomials in $\mathcal{G}$ have degree $\leq R^{2^d}$, and all intermediate computations involve only polynomials of at most that degree.

(5) While this doubly-exponential degree bound is optimal in general, for many important special cases (e.g., zero-dimensional ideals) singly exponential degree bounds are known.

(6) Every ideal $\mathcal{I} \subset K[\mathbf{x}]$ contains a *finite* subset $\mathcal{U}$ (called a *universal Gröbner basis*), which is a Gröbner basis for $\mathcal{I}$ with respect to *every* term order on $K[\mathbf{x}]$.

Throughout this paper, we assume that the degrees of all monomials can be bounded beforehand. When dealing with a fixed ideal, this is a legitimate assumption by (4). We therefore identify term orders $\prec$ with their representing weight vectors $w \in \mathbb{R}^d_+$.

The following definitions are fundamental for our algebraic application of Minkowski sums. The *Newton polytope* $N(t)$ of a polynomial $t = \sum_{i=1}^n c_i \mathbf{x}^{\alpha_i}$ is the convex hull of its monomials in $\mathbb{R}^d$, that is, $N(t) := \text{conv}\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$. Its Minkowski sum $N_{\text{aff}}(t) := N(t) + \mathbb{R}^d_-$ with the negative orthant is called the *affine Newton polyhedron* of t. We remark that Ostrowski in his work on factorization of polynomials [29], [30] uses the term "baric polyhedron of t" for $N(t)$.

PROPOSITION 1.3.5. *If $t \in K[\mathbf{x}]$ is a homogeneous polynomial, then the vertices of its Newton polytope $N(t)$ are the initial monomials of t with respect to all possible term orders. For general $t \in K[\mathbf{x}]$, the initial monomials are the vertices of the affine Newton polyhedron $N_{\text{aff}}(t)$.*

Proposition 1.3.5 is a special case of the results to be proved in §3.2. Let us point out that taking the Minkowski sum of Newton polytopes corresponds to the algebraic operation of multiplication (cf. [30, Thm. VI]).

REMARK 1.3.6. $N(t_1 t_2 \cdots t_k) = N(t_1) + N(t_2) + \cdots + N(t_k)$ *for all* $t_1, t_2, \ldots, t_k \in K[\mathbf{x}]$.

Minkowski sums of Newton polytopes play a crucial role in determining the computational complexity of the following decision problem.

Gröbner Basis Detection.

Input: *A set $\mathcal{T} \subset K[\mathbf{x}]$ of polynomials.*

Output: *A term order $w \in \mathbb{R}^d$ such that $\mathcal{T}$ is a Gröbner basis with respect to w, if such w exists*; *"NO" otherwise.*

This problem is not well defined unless we specify the representation of the polynomials in $\mathcal{T}$. Throughout this paper, we assume that any multivariate polynomial $t = c_1\mathbf{x}^{\alpha_1} + \cdots + c_n\mathbf{x}^{\alpha_n}$ is presented by its nonzero scalar coefficients $c_1, \ldots, c_n \in K$ and its corresponding nonnegative exponent vectors $\alpha_1, \ldots, \alpha_n \in \mathbb{Z}^d$. In this *sparse representation*, the cardinality of $\mathcal{T}$ and the numbers n are regarded as part of the input. Moreover, the logarithm of the total degree is part of the input via the exponent vectors. It is important to note that (in this sparse representation) the total degree itself is *not* part of the input. When dealing with the binary model of computation, we will further assume that the coefficient field K is the field $\mathbb{Q}$ of rational numbers.

1.4. The main results. We will now outline the main results of this paper (with numbers referring to the theorems in the corresponding section). Section 2.1 contains some preliminary geometric and combinatorial results. In particular, we give the following upper bound on the number of l-dimensional faces of $P_1 + \cdots + P_k$; this bound is computationally relevant for FIXED-D-Π-MINKADD.

THEOREM 2.1.10′. *Let $P_1, \ldots, P_k$ be polytopes in $\mathbb{R}^d$, let n be the number of nonparallel edges of $P_1, \ldots, P_k$, and let $l \in \{0, \ldots, d-1\}$. Then the number of l-dimensional faces of $P_1 + \cdots + P_k$ does not exceed*

$$2\binom{n}{l}\sum_{j=0}^{d-1-l}\binom{n-l-1}{j}.$$

In the case of special interest, where the number of vertices of the P_i are given and we would like to know an upper bound on the number of vertices of $P_1 + \cdots + P_k$, we obtain the following asymptotic result.

COROLLARY 2.1.11′. *Let $P_1, \ldots, P_k$ be polytopes in $\mathbb{R}^d$ with at most n vertices each. Then the number of vertices of $P_1 + \cdots + P_k$ is in $O(k^{d-1}n^{2(d-1)})$.*

Section 2.2 deals with zonotopes and hyperplane arrangements, whereas in §2.3 the complexity classes of computing Minkowski sums of polytopes under various assumptions are (almost completely) characterized.

The problem of computing Minkowski sums of polytopes has been addressed previously by Guibas and Seidel [22], who show that the Minkowski sum of two polytopes in $\mathbb{R}^3$ can be computed in time proportional to the size of the input plus the size of the output. Here we are dealing with Minkowski sums of an arbitrary (finite) number of polytopes in arbitrary (finite-)dimensional real space and the complexity is measured as a function of the input size. We prove that, for no choice of Π, the problem Π-MINKADD can be solved in polynomial-time in either of the two models (Remark 2.3.4). This result persists even for fixed k if at least one of the polytopes is $\mathcal{H}$-presented (Proposition 2.3.1). However, there exist polynomial-time algorithms for FIXED-K-$\mathcal{V}$-MINKADD in the binary model of computation (Proposition 2.3.2) and for FIXED-K-D-$\mathcal{V}$-MINKADD in both models of computation (Proposition 2.3.5). As the main result of §2.3 we show that the problem FIXED-D-$\mathcal{V}$-MINKADD can be solved in polynomial time, both in the binary and in the real model of computation. An improved algorithm for $d = 2$ will be given in Proposition 2.3.9.

THEOREM 2.3.7′, 2.3.11. *For $d \geq 2$ the problem* FIXED-D-$\mathcal{V}$-MINKADD *can be solved in $O(k^d n^{2d-1})$ arithmetic operations, where n denotes the maximum numbers of points in the given $\mathcal{V}$-presentations. Likewise, it can be solved in polynomial-time in the binary model of computation.*

It is this algorithm for FIXED-D-$\mathcal{V}$-MINKADD that plays the crucial role for the applications to Gröbner bases. In §3.1 we will deal with general dynamic versions of Buchberger's Gröbner basis algorithm. We will show (Algorithm 3.1.3) that it is possible to change the term order in every single step of Buchberger's algorithm without harming the termination and correctness. This gives enough margin to change term orders whenever it seems profitable. The main result of §3.2 is Theorem 3.2.6.

This result means that we can detect a Gröbner basis in polynomial time for d fixed. It will also give rise to criteria for deciding how to dynamically change term orders in order to speed up the Gröbner basis algorithm. In §3.3 we define a natural Hilbert function measure for "closeness to being a Gröbner basis." With respect to this measure, we can compute optimal term orders in polynomial time (Theorem 3.3.6).

2. Minkowski addition of polytopes. This section contains geometric, combinatorial, and computational results for the Minkowski addition of polytopes.

2.1. Geometric and combinatorial results. In the following, we derive some geometric and combinatorial results that are important for the subsequent sections. In particular, we give bounds for the number of vertices of the Minkowski sum of k polytopes in $\mathbb{R}^d$ in terms of the vertex numbers of the input polytopes. The question most relevant for the complexity of our main problem FIXED-D-Π-MINKADDand for the applications in §3 is to determine the asymptotic behavior of these numbers for fixed d and $k \to \infty$. However, in view of possible future applications, we keep the exposition more general in this section.

Given a polytope P in $\mathbb{R}^d$, we write $\mathcal{F}_i(P)$ for the set of i-dimensional faces of P and $f_i(P)$ for its cardinality. The set of all faces of P is abbreviated $\mathcal{F}(P)$.

PROBLEM 2.1.1. Let $d, k \in \mathbb{N} \setminus \{0, 1\}$, $l, m \in \{0, \dots, d-1\}$ and $n_1, \dots, n_k \in \mathbb{N} \setminus \{0\}$. Furthermore, let, for $i = 1, \dots, k$, $n_i = 1$ or $n_i \geq m+2$. Determine the number

$$v_{l,m}(d; k; n_1, \dots, n_k) := \max\{f_l(P_1 + \cdots + P_k) | P_1, \dots, P_k \text{ are polytopes in } \mathbb{R}^d \\ \text{with } f_m(P_1) = n_1, \dots, f_m(P_k) = n_k\}.$$

The restrictions in the hypothesis of this problem only rule out trivial cases. In particular, the numbers $v_{l,m}(d; k; n_1, \dots, n_k)$ are all finite (and thus the "maximum" is attained). Observe, however, that some of these numbers are 0, namely, if $\sum_{i=1}^k n_i$ is too small compared to l. The numbers $v_{l,0}(d; k; 2, \dots, 2)$, which are also the face numbers of generic zonotopes (and upper bounds for the face numbers of general zonotopes), are well known and can be found in [11], [21], [37]. Usually these are stated as the face numbers of simple hyperplane arrangements (see §2.2).

PROPOSITION 2.1.2 (see [11]). *The number of l-faces of a d-zonotope generated by k line segments in general position equals*

$$v_{l,0}(d; k; 2, \dots, 2) = v_{l,1}(d; k; 1, \dots, 1) = 2\binom{k}{l} \sum_{j=0}^{d-1-l} \binom{k-1-l}{j}.$$

In the following, we are mainly interested in nontrivial upper bounds that are—whenever possible—polynomial in k as $k \to \infty$. For small k, the trivial upper bounds can in general not be improved. As an example, we consider the case where $l = m = 0$. Because every vertex of a Minkowski sum of polytopes is a sum of vertices of the summands, the number $v_{0,0}(d; k; n_1, \dots, n_k)$ is bounded above by $\prod_{i=1}^k n_i$. This bound is sharp if the number k of polytopes is small relative to the dimension d.

REMARK 2.1.3. *Let $2k \leq d$. Then $v_{0,0}(d; k; n_1, \dots, n_k) = \prod_{i=1}^k n_i$.*

Proof. Let $L_1, \dots, L_k$ be pairwise orthogonal linear subspaces of $\mathbb{R}^d$ of dimension at least 2 and consider for $i = 1, \dots, k$ a polytope P_i in L_i with n_i vertices. In this situation, each such sum of vertices is extreme, and we have $f_0(P_1 + \cdots + P_k) = \prod_{i=1}^k n_i$. □

This argument can be modified to prove a result similar to Remark 2.1.3 also for the situation where all k polytopes have full dimension d. In that case, we replace each P_i by a suitable multiple pyramid over P_i. A similar construction can also be given for other pairs (l, m).

To identify the faces of the Minkowski sum of polytopes, we must consider the cones of outer normals (at relatively interior points of faces) of the participating polytopes. Let P be a polytope in $\mathbb{R}^d$. Given any face F of P, we let $\mathcal{N}(F; P)$ denote the relatively open polyhedral cone (with apex 0) of outer normals of P at F. The collection $\{\mathcal{N}(F; P) | F \in \mathcal{F}(P)\}$ forms a polyhedral cell complex whose underlying point set is $\mathbb{R}^d$. This complex will be denoted by $\mathcal{N}(P)$ and called the *normal fan* of P. Note that, if F is an i-dimensional face of P, then its *normal cone* $\mathcal{N}(F; P)$ is a $(d-i)$-dimensional cell of the normal fan $\mathcal{N}(P)$. The cells of $\mathcal{N}(P)$ are partially ordered by inclusion of their closures, and the assignment $F \mapsto \mathcal{N}(F; P)$ defines an order-reversing bijection between $\mathcal{F}(P)$ and $\mathcal{N}(P)$.

Let us introduce one more definition. Given a direction vector $z \in \mathbb{R}^d \setminus \{0\}$, we write

$$S(P; z) := \{x \in P | \langle x, z\rangle = \max_{y \in P} \langle y, z\rangle\},$$

where $\langle y, z\rangle$ denotes the Euclidean inner product of y and z. Thus $S(P; z)$ is the face of P consisting of all maximal points with respect to the linear functional $\langle z, \cdot\rangle$. Equivalently, $S(P; z)$ is the intersection of P with its supporting hyperplane in direction z.

LEMMA 2.1.4. *Let $P_1, \ldots, P_k$ be polytopes in $\mathbb{R}^d$ and let $z \in \mathbb{R}^d \setminus \{0\}$. Then, for any nonzero direction vector $z \in \mathbb{R}^d$, we have the relation*

$$S(P_1 + P_2 + \cdots + P_k; z) = S(P_1; z) + S(P_2; z) + \cdots + S(P_k; z).$$

Proof. The assertion follows directly from the definition. □

As a consequence of this lemma, we obtain a characterization of the faces of the Minkowski sum $P_1 + \cdots + P_k$ in terms of the normal fans $\mathcal{N}(P_1), \ldots, \mathcal{N}(P_k)$. The *common refinement* $\mathcal{N}(P_1) \wedge \cdots \wedge \mathcal{N}(P_k)$ of the complexes $\mathcal{N}(P_1), \ldots, \mathcal{N}(P_k)$ is defined as the smallest complex $\mathcal{N}$ such that the closure of each cell in one of the complexes $\mathcal{N}(P_1), \ldots, \mathcal{N}(P_k)$ is the union of the closure of cells of $\mathcal{N}$. In other words, the cells of $\mathcal{N}(P_1) \wedge \cdots \wedge \mathcal{N}(P_k)$ are obtained by taking all possible intersections $\cap_{i=1}^{k} \mathcal{N}(F_i; P_i)$, where $F_1 \in \mathcal{F}(P_1), \ldots, F_k \in \mathcal{F}(P_k)$.

LEMMA 2.1.5. $\mathcal{N}(P_1 + \cdots + P_k) = \mathcal{N}(P_1) \wedge \cdots \wedge \mathcal{N}(P_k)$.

Proof. By Lemma 2.1.4, each cell of $\mathcal{N}(P_1 + \cdots + P_k)$ is of the form

$$\{y \in \mathbb{R}^d \setminus \{0\} | S(P_1; y) + \cdots + S(P_k; y) = S(P_1; z) + \cdots + S(P_k; z)\}$$

for some fixed z. This cell is equal to

$$\{y \in \mathbb{R}^d \setminus \{0\} | S(P_1; y) = S(P_1; z), \ldots, S(P_k; y) = S(P_k; z)\},$$

which is a cell of $\mathcal{N}(P_1) \wedge \cdots \wedge \mathcal{N}(P_k)$. □

This result implies that the number of i-faces of $P_1 + \cdots + P_k$ is equal to the number of $(d-i)$-dimensional cells of $\mathcal{N}(P_1) \wedge \cdots \wedge \mathcal{N}(P_k)$. Given two polyhedral cell complexes $\mathcal{C}_1$ and $\mathcal{C}_2$, then we write $\mathcal{C}_1 \preceq \mathcal{C}_2$ if $\mathcal{C}_2$ is a refinement of $\mathcal{C}_1$.

LEMMA 2.1.6. *Let P, P_1 be polytopes in $\mathbb{R}^d$. Then $\mathcal{N}(P_1) \preceq \mathcal{N}(P)$ if and only if there exists a $\lambda \in \mathbb{R}_+$ such that λP_1 is a Minkowski summand of P.*

Proof. See [21, pp. 318–319]. □

The idea of a construction leading to upper bounds used in §2.3 is as follows. We replace each polytope P_i by the "smallest possible" zonotope Z_i whose normal fan $\mathcal{N}(Z_i)$ is a refinement of $\mathcal{N}(P_i)$. This will imply that the face numbers $f_i(Z_1 + \cdots + Z_k)$ are upper bounds for the face numbers $f_i(P_1+\cdots+P_k)$. More precisely, given any polytope P in $\mathbb{R}^d$, we define its *edgotope* $Z(P) = \sum_{E\in\mathcal{F}_1(P)} E$ to be the zonotope that is generated by all edges of P.

Note, as a side remark, that $Z(P)$ is a Minkowski-summand of P's *shellotope*

$$\Lambda(P) = \sum_{\substack{v_1,v_2\in\mathcal{F}_0(P)\\ v_1\neq v_2}} \operatorname{conv}\{v_1, v_2\}.$$

Observe that each vertex of $\Lambda(P)$ corresponds to an orientation of the edges and the diagonals of P according to the values of a linear functional. Hence the vertices of $\Lambda(P)$ are in one-to-one correspondence with the *line shellings* of P^*, the polar of P, which are induced by all lines through the center of polarity. If P is a d-simplex, then the line shellings of P^* correspond to the permutations of $\{0,\dots,d\}$, and the edgotope coincides with the shellotope and is isomorphic to the *permutohedron*

$$\operatorname{conv}\{(\pi(0),\dots,\pi(d))|\pi \text{ is a permutation of } \{0,\dots,d\}\}.$$

Note that—with respect to normal fans—taking edgotopes commutes with Minkowski addition.

REMARK 2.1.7. *Let $P_1,\dots,P_k$ be polytopes in $\mathbb{R}^d$. Then*

$$\mathcal{N}(Z(P_1+\cdots+P_k)) = \mathcal{N}(Z(P_1)+\cdots+Z(P_k)).$$

PROPOSITION 2.1.8. *Let P be a polytope in $\mathbb{R}^d$ and $Z(P)$ its edgotope. Then $\mathcal{N}(P) \preceq \mathcal{N}(Z(P))$ with equality if and only if P is a zonotope.*

Proof. Let $C \in \mathcal{N}(Z(P))$ and $z_1, z_2 \in C$. We must show that z_1 and z_2 lie in the same cell of $\mathcal{N}(P)$. Suppose that this were not the case, which is equivalent to $S(P, z_1) \neq S(P, z_2)$. After replacing z_1 by a convex combination $\mu z_1 + (1-\mu)z_2$, if necessary, we may assume that the face $S(P, z_1)$ has dimension ≥ 1 and that $S(P, z_1)$ is not a face of $S(P, z_2)$. Then there exists an edge E of $S(P, z_1)$ that is not an edge of $S(P, z_2)$. Hence E is a Minkowski summand of $S(Z(P), z_1) = Z(S(P, z_1))$ but not of $S(Z(P), z_2) = Z(S(P, z_2))$. These two subedgotopes being distinct is a contradiction to the choice of z_1, z_2. We omit the easy proof of the "if and only if" part of Proposition 2.1.8. □

Using Lemma 2.1.6 and Proposition 2.1.8, we see that the edgotope $Z(P)$ is the smallest zonotope that has P as a Minkowski summand. Furthermore, we obtain the following bound on the numbers of i-dimensional faces of Minkowski sums of polytopes.

COROLLARY 2.1.9. *Given polytopes $P_1,\dots,P_k$ in $\mathbb{R}^d$, we have*

$$f_i(P_1+\cdots+P_k) \leq f_i(Z(P_1)+\cdots+Z(P_k)) \quad \textit{for} \quad i = 0, 1, \dots, d-1.$$

Proof. The proof follows immediately from Remark 2.1.7 and Proposition 2.1.8. □

THEOREM 2.1.10. *Let $P_1,\dots,P_k$ be polytopes in $\mathbb{R}^d$, let n denote the number of non-parallel edges of $P_1,\dots,P_k$, and let $l \in \{0,\dots,d-1\}$. Then*

$$\begin{aligned} f_l(P_1+\cdots+P_k) &\leq v_{l,1}(d; n; 1, \dots, 1) \\ &= 2\binom{n}{l}\sum_{j=0}^{d-1-l}\binom{n-1-l}{j}, \end{aligned}$$

with equality if all polypopes $P_1, \dots, P_k$ are zonotopes and their generating edges are in general position.

Proof. The proof follows from Corollary 2.1.9 and Proposition 2.1.2. □

The special case to be considered in §§2.3 and 3 is the case where $l = m = 0$. By first replacing n by $f_1(P_1) + \cdots + f_1(P_k)$ and then each $f_1(P_i)$ by its upper bound $\binom{f_0(P_i)}{2}$, we obtain the following asymptotic result.

COROLLARY 2.1.11. *For fixed d, we have*

$$v_{0,0}(d; k; n_1, \dots, n_k) = O(k^{d-1}(\max\{n_1, \dots, n_k\})^{2(d-1)}).$$

2.2. On zonotopes and hyperplane arrangements. As was mentioned earlier, zonotopes and arrangements of hyperplanes are equivalent structures (see [13], [21]). This correspondence is particularly transparent in the case of *linear* arrangements, where all hyperplanes contain the origin. Consider a linear arrangement $\mathcal{A} = \{H_1, \dots, H_n\}$ of hyperplanes $H_i = \{x \in \mathbb{R}^d | \langle x, z_i \rangle = 0\}$ defined by the normal vectors $z_1, \dots, z_n \in \mathbb{R}^d$. The face lattice of (the cell complex induced by) $\mathcal{A}$ is antiisomorphic to the face lattice of the zonotope $Z = \sum_{i=1}^{n} [-1, 1] z_i$. In fact, the geometric polarity between arrangements and zonotopes follows as a special case from our arguments in §2.1.

Thus, our edgotope construction can also be expressed in the framework of hyperplane arrangements. This fact is of some algorithmic interest, since it allows us to utilize an algorithm given by Edelsbrunner, O'Rourke, and Seidel [14] (cf. [15] for a correction) to determine the face lattice of an arrangement of hyperplanes. (This algorithm can, in turn, be rephrased in terms of zonotopes.)

PROPOSITION 2.2.1. *Let $P_1, \dots, P_k$ be polytopes in $\mathbb{R}^d$ and let $Z = Z(P_1) + \cdots + Z(P_k)$ be the Minkowski sum of the associated edgotopes. Then the normal fan $\mathcal{N}(Z)$ equals the linear hyperplane arrangement $\mathcal{A} = \{\mathrm{lin}\mathcal{N}(F_i; P_i) | i = 1, \dots, k, F_i \in \mathcal{F}_1(P_i)\}$.*

Proof. The assertion follows immediately from Lemma 2.1.5 and the fact that the normal cones at the two vertices of a segment $P = \mathrm{conv}\,\{v_1, v_2\}$ are the open half-spaces with boundary hyperplane perpendicular to $\mathrm{aff}\{v_1, v_2\}$. □

Proposition 2.2.1 shows that, to estimate the number of i-faces of $P_1 + \cdots + P_k$, we may, in principle, proceed as follows.

ALGORITHM 2.2.2.

```
(1) Determine the complexes $\mathcal{N}(P_1), \dots, \mathcal{N}(P_k)$.
(2) Take for each $(d-1)$-cell of any of the complexes $\mathcal{N}(P_1), \dots, \mathcal{N}(P_k)$
    its linear hull. (This determines a linear arrangement of hyperplanes $\mathcal{A}$.)
(3) Determine the number of $(d-i)$-cells of $\mathcal{A}$.
```

This algorithmic scheme will be extended in the next section (Algorithm 2.3.6).

2.3. The complexity of computing Minkowski sums of polytopes. In this section, we determine the complexity of computing the Minkowski sum $P_1 + \cdots + P_k$ of k polytopes P_i in real d-space. The emphasis is, generally, not on giving best bounds but on deciding whether there exist polynomial time algorithms. We will deal with the various problems that occur if we regard none, one, or both of the parameters d and k as constants. Some of the results below are (at least implicitly) known, and some of the proofs are quite standard—the full material is included to give the complete picture here.

2.3.1. The case of varying *d* but fixed *k*. The following result shows that in most cases there is no polynomial-time algorithm for the problem FIXED-K-Π-MINKADD.

PROPOSITION 2.3.1. *Let P_i be a $\mathcal{W}_i$-presented polytope for $i = 1, 2$, let $P = P_1 + P_2$ be (irredundantly) $\mathcal{W}$-presented and suppose that $\mathcal{H} \in \{\mathcal{W}, \mathcal{W}_1, \mathcal{W}_2\}$. Then there is no* *polynomial in the (binary or real) sizes of P_1, P_2 that is an upper bound for the (binary or real) size of P.*

Proof. We first note that an $\mathcal{H}$-presentation of a simplex can be converted into a $\mathcal{V}$-presentation in polynomial time, and vice versa; this is the case for the (binary) Turing-machine model as well as for the real RAM model of computation. Now, let P_1 be a d-simplex and let $P_2 = -P_1$. Then the Minkowski sum $P_1 + P_2$ is a centrally symmetric polytope that has $2^{d+1} - 2$ facets. Thus there is no polynomial-time algorithm for computing an $\mathcal{H}$-presentation of $P_1 + P_2$. This settles all cases with $\mathcal{W} = \mathcal{H}$.

Let us now deal with the cases where $\mathcal{W} = \mathcal{V}$. We define P_1 to simply be the singleton $\{0\}$, which has the $\mathcal{H}$-presentation $P_1 = \bigcap_{j=1,\dots,d}\{x | \langle \pm e_j, x\rangle \leq 0\}$, where e_j denotes the jth standard unit vector in $\mathbb{R}^d$. Furthermore, let P_2 be the unit cube $[-1, 1]^d$, which has the $\mathcal{H}$-presentation $P_2 = \bigcap_{j=1,\dots,d}\{x | \langle \pm e_j, x\rangle \leq 1\}$. Then the Minkowski sum $P_1 + P_2 = P_2$ is the cube again, and hence, $P_1 + P_2$ has 2^d vertices. This settles the cases where $\mathcal{W}_1 = \mathcal{H}$ or $\mathcal{W}_2 = \mathcal{H}$. □

So the only remaining case is FIXED-K-$\mathcal{V}$-MINKADD. Here we have the following result.

PROPOSITION 2.3.2. *In the binary model of computation*, FIXED-K-$\mathcal{V}$-MINKADD *can be solved in polynomial time. More precisely, if L is an upper bound for the binary sizes of the polytopes P_i, then there is an algorithm that solves this problem in $O(((s + d)s^2 + (s + d)^{1.5}s)L)$ arithmetic operations and $O(((s + d)s^2 + (s + d)^{1.5}s)L^2(\log L)(\log\log L))$ bit operations, where $s = n_1 \cdot \ldots \cdot n_k$ is the product of the vertex numbers of the input polytopes.*

Proof. Let, for $i = 1, \dots, k$, $P_i = \operatorname{conv}\{v_{i,1}, \dots, v_{i,n_i}\}$. Set, for $i = 1, \dots, k$, $J_i := \{1, \dots, n_i\}$. Then

$$P_1 + \cdots + P_k = \operatorname{conv}\{v_{1,j_1} + \cdots + v_{k,j_k} | j_1 \in J_1, \dots, j_k \in J_k\}.$$

Now, observe, that from this $\mathcal{V}$-presentation we can obtain an irredundant $\mathcal{V}$-presentation in polynomial time. To this end, we solve, for each of the $n_1 \cdot \ldots \cdot n_k$-tuples $(v_{1,j_1^*}, \dots, v_{k,j_k^*})$, the feasibility problem

$$\sum_{j_1 \in J_1 \setminus \{j_1^*\}} \cdots \sum_{j_k \in J_k \setminus \{j_k^*\}} \lambda_{j_1 \dots j_k}(v_{1,j_1} + \cdots + v_{k,j_k}) = v_{1,j_1^*} + \cdots + v_{k,j_k^*}$$

$$\sum_{j_1 \in J_1 \setminus \{j_1^*\}} \cdots \sum_{j_k \in J_k \setminus \{j_k^*\}} \lambda_{j_1 \dots j_k} = 1,$$

$$\lambda_{j_1 \dots j_k} \geq 0; \quad j_i \in J_i \setminus \{j_i\} \text{ for i=1,...,k.}$$

If this problem is feasible, we remove $(v_{1,j_1^*}, \dots, v_{k,j_k^*})$ from our set of candidates of vertices of $P_1 + \cdots + P_k$.

The linear programming problem can be solved in polynomial time (recall that here k is a constant). In fact, since we have systems in at most $s = n_1 \cdot \ldots \cdot n_k$ variables and with $d + 1$ equality constraints, the total algorithm requires no more than $O(((s + d)s^2 + (s + d)^{1.5}s)L)$ arithmetic operations and no more than $O(((s + d)s^2 + (s + d)^{1.5}s)L^2(\log L)(\log\log L))$ bit operations [19], [31], [35]. □

An obvious method for improving the above algorithm is the following.

```
(1) split the set of polytopes into subsets $C_1, \dots, C_{\lceil k/c \rceil}$ of size at
    most $c$, where $c$ is a positive integer;
(2) construct the Minkowski sums $P_{C_i} = \sum_{P \in C_i} P$ for $i = 1, \dots, \lceil k/c \rceil$;
(3) replace original polytopes by $P_{C_1}, \dots, P_{C_{\lceil k/c \rceil}}$; replace $k$ by $\lceil k/c \rceil$;
(4) repeat steps (1)–(3) as long as $\lceil k/c \rceil \geq 2$.
```

Clearly, this general procedure will be more efficient than the approach given in Proposition 2.3.2 if $P_1 + \cdots + P_k$ has less vertices than $s = n_1 \cdot \ldots \cdot n_k$. This is, in particular, the case when k is very large compared to d. In the case of fixed k and $d \to \infty$, on the other hand, the number of vertices of $P_1, \ldots, P_k$ can be as large as $n_1 \cdot \ldots \cdot n_k$ (see Remark 2.1.3). This means that the last linear program we have to solve in this modified approach has s variables and $d + 1$ equality constraints, which is exactly the same as in the program that we solve in the proof of Proposition 2.3.2.

Note that Proposition 2.3.2 does not settle the problem for the real RAM model since the number of arithmetic operations depends on L. In fact, it is one of the most prominent unsolved problems in mathematical programming whether or not there is an algorithm that solves linear programs in a number of arithmetic operations that is bounded by a polynomial in the dimension and the number of constraints.

PROBLEM 2.3.3. In the real RAM model of computation, can the problem FIXED-K-$\mathcal{V}$-MINKADD be solved in polynomial time?

2.3.2. The case of varying *d* and varying *k*. Note that if FIXED-K-Π-MINKADD cannot be solved in polynomial time, then there is no polynomial time algorithm for Π-MINKADD, either. By the results of the previous analysis, this is the case whenever $\mathcal{H}$ appears in the string Π. The remaining case, $\mathcal{V}$-MINKADD, is settled by the following remark.

REMARK 2.3.4. *Neither in the binary nor in the* (*real*) RAM *model of computation there is a polynomial time algorithm for* $\mathcal{V}$-MINKADD.

Proof. For $i = 1, \ldots, d$, let $P_i = [0, 1]e_i$ be the unit segment on the ith coordinate axis. Then $f_0(\sum_{i=1}^{d} P_i) = 2^d$, whereas the total (binary or real) size of the problem $\mathcal{V}$-MINKADD is of order $O(d^2)$. □

2.3.3. The case of fixed *d* and fixed *k*. For fixed dimension, the crucial difference between $\mathcal{V}$- and $\mathcal{H}$-presented polytopes disappears. In fact, we can pass from one presentation to the other in polynomial time. Thus, in the following we deal with the problem FIXED-K-D-$\mathcal{V}$-MINKADD.

Let $P_i = \operatorname{conv}\{v_{i,1}, \ldots, v_{i,n_i}\}$ for $i = 1, \ldots, k$, let $P = P_1 + \cdots + P_k$, and let

$$S = \{v_{1,j_1} + \cdots + v_{k,j_k} | j_1 = 1, \ldots, n_1; \ldots j_k = 1, \ldots, n_k\}.$$

To give an irredundant $\mathcal{V}$-presentation of P, we must identify the extreme points of S.

We apply a result of Megiddo [27] to see that this can be done in $O(s^2)$ arithmetic operations, where $s = n_1 \cdot \ldots \cdot n_k$. Suppose without loss of generality that the barycenter of the point set S is the origin. Because d is constant, we can find aff(S) in $O(s)$ operations. Hence, we may assume that P is d-dimensional, and thus, $0 \in \operatorname{int}(P)$. Under polarity the points of S transform to (oriented) hyperplanes and the problem is equivalent to reducing this set to an irredundant $\mathcal{H}$-presentation. This tasks splits, again, into at most s linear programs in dimension d and with s constraints. Hence, using Megiddo's [27] linear programming algorithm, this problem can be solved in time $O(s^2)$.

Note that FIXED-K-D-$\mathcal{V}$-MINKADD can also be solved by any convex hull algorithm applied to S. This gives an algorithm for the complete face lattice of P that runs in time $O(s^{\lfloor (d+1)/2 \rfloor})$ (see, e.g., [13]).

However, the extreme points of a finite point set in $\mathbb{R}^d$ can be computed more quickly. Indeed, Matoušek and Schwarzkopf [24] combined a multidimensional version of Megiddo's parametric search technique with appropriate data structures for so-called half-space-emptiness queries to show that for fixed d the extreme points of an arbitrary point set in $\mathbb{R}^d$ of cardinality n can be found in time $O(n^{2-(2/(1+\lfloor d/2 \rfloor))+\delta})$ for any fixed positive δ. This implies the following result.

PROPOSITION 2.3.5. FIXED-K-D-$\mathcal{V}$-MINKADD *can be solved in* $O(s^{2-(2/(1+\lfloor d/2 \rfloor))+\delta})$ *arithmetic operations, where* $s = n_1 \cdot \ldots \cdot n_k$ *and* δ *is a fixed positive real.*

Proposition 2.3.5 implies as a corollary that regardless of the presentation of the input polytopes and the desired presentation ($\mathcal{V}$ or $\mathcal{H}$) of the output polytope, the Minkowski sum of k polytopes can be computed in a polynomial number of arithmetic operations. The same is true for the binary model of computation.

2.3.4. The case of fixed d and varying k. In the following, we apply the technique used to prove Corollary 2.1.11 to give a polynomial time algorithm (in the binary and in the real model) for our problem in fixed dimension. Again, since for fixed dimension d any $\mathcal{H}$-presentation of a polytope can be converted to a $\mathcal{V}$-presentation in polynomial time, we only deal with the problem FIXED-D-$\mathcal{V}$-MINKADD. The structure of the following algorithms is an extension of Algorithm 2.2.2.

ALGORITHM 2.3.6.
Input: *Polytopes* $P_1, \ldots, P_k$.
Output: *A complete list* $\mathcal{L}$ *of all faces of* $P_1 + \cdots + P_k$.

```
(1) Determine the edges of $P_1, \ldots, P_k$.
(2) Compute the hyperplanes through 0 that are perpendicular to these
    edges; let $\mathcal{A}$ denote the corresponding linear arrangement.
(3) Determine the (relatively open) cells of $\mathcal{A}$.
(4) For each cell $F$ of $\mathcal{A}$ compute a sample point $z \in F$.
(5) Compute $S(P_1 + \cdots + P_k; z)$.
(6) Store $S(P_1 + \cdots + P_k; z)$ in $\mathcal{L}$ if it is not already contained there.
```

We remark that in step (3) we need only consider the maximal cells of $\mathcal{A}$. Then the algorithm will generate precisely all vertices of $P_1 + \cdots + P_k$. Because the dimension d is fixed, we can then use any convex hull algorithm to compute the entire face lattice of $P_1 + \cdots + P_k$.

The following theorem is the main result of §3.3.

THEOREM 2.3.7. *For* $d \geq 3$, *the problem* FIXED-D-$\mathcal{V}$-MINKADD *can be solved in* $O(k^{d-1}n^{2d-1})$ *arithmetic operations*, *where* n *denotes the maximum numbers of points in the given* $\mathcal{V}$*-presentations.*

Proof. By use of any algorithm that computes the convex hull of a point set, we can determine the entire face lattice of the polytopes P_i. It is known that this can be done in time $O(n^{\lfloor (d+1)/2 \rfloor})$ (recall that $d \geq 3$) for each polytope (see, e.g., [13]). In this way, we reduce the given $\mathcal{V}$-presentations to irredundant $\mathcal{V}$-presentations. In the following, we will therefore assume that the given presentations are irredundant.

Let n_i denote the number of vertices of P_i, let $\mathcal{A}_i$ denote the linear arrangement induced by $\mathcal{N}(P_i)$, and let $\mathcal{A}$ be the common refinement of all the $\mathcal{A}_i$'s. By Corollary 2.1.11, $\mathcal{A}$ is a linear arrangement of at most $\frac{1}{2}\sum_{i=1}^{k} n_i(n_i - 1)$ hyperplanes, which has at most $O(k^{d-1}n^{2(d-1)})$ maximal cells. Using the algorithm in [14] and [15], the arrangement $\mathcal{A}$ can be constructed in time $O(k^{d-1}n^{2(d-1)})$, once the normals of the hyperplanes are given. However, these normals are the edges of P_i, which have already been computed. By "constructing an arrangement," we mean computing the entire face lattice of the arrangement, as well as a reference point in the relative interior of every face.

The last step is to find for each reference vector z in a d-dimensional cell of $\mathcal{A}$ the (uniquely determined) vertex $v_{i;z}$ of P_i that is maximal with respect to the linear functional induced by z. Observe that this can be done by computing the inner product $\langle v, z\rangle$ for each vertex v of P_i. Then $\sum_{i=1}^{k} v_{i;z}$ is a vertex of $\sum_{i=1}^{k} P_i$, and all vertices of the Minkowski sum are obtained that way. The total number of required arithmetic operations is of order $O(k^d n^{2d-1})$. □

COROLLARY 2.3.8. *Let $d \geq 3$ and let $P_1, \ldots, P_k$ be $\mathcal{V}$- or $\mathcal{H}$-presented polytopes in $\mathbb{R}^d$. Then the face lattice of $P_1 + \cdots + P_k$ can be determined in $O(k^d n^{2d-1})$ arithmetic operations, where n denotes the maximum numbers of vertices of $P_1, \ldots, P_k$.*

In the plane, this bound can easily be improved as follows.

PROPOSITION 2.3.9. *In dimension $d = 2$, the problem* FIXED-D-$\mathcal{V}$-MINKADD *can be solved in $O(kn \log n)$ arithmetic operations.*

Proof. Let $P_1, \ldots, P_k$ be the given $\mathcal{V}$-polygons. In $O(kn \log n)$ arithmetic operations, we can compute the vertices of all polygons and their adjacent vertices. Thus, we can compute all normal fans $\mathcal{N}(P_1), \ldots, \mathcal{N}(P_k)$ in a total of $O(kn \log n)$ arithmetic operations. We can assume that for each $i = 1, \ldots, k$ the vertices of P_i and the edges of $\mathcal{N}(P_i)$ are ordered with respect to increasing angle to the positive x-axis, respectively. Furthermore, the edges of $\mathcal{N}(P_i)$ are labeled with the corresponding pair (v^+, v^-) of vertices of P_i in the given order.

Next, we amalgamate the ordered lists of edges of $\mathcal{N}(P_1), \ldots, \mathcal{N}(P_k)$ so as to obtain the corresponding ordering of the edges of $\mathcal{N}(P_1) \wedge \cdots \wedge \mathcal{N}(P_k)$. This can be done in time $O(kn)$.

We assume that the edges of $\mathcal{N}(P_1), \ldots, \mathcal{N}(P_k)$ are all different. Otherwise, the following construction can be carried through after suitably perturbing the polygons.

In the last step, we compute all vertices of $P_1 + \cdots + P_k$. We do this in a counterclockwise order. First, we compute a reference point z in the interior of the 2-cell (or possibly one of the two 2-cells) that contains the positive x-axis. By solving a point location problem for this reference point in each of the complexes $\mathcal{N}(P_i)$, we can identify the vertices of $P_1, \ldots, P_k$ whose sum w is the vertex of $P_1 + \cdots + P_k$ that corresponds to z. Then we move in counterclockwise order to the next 2-cell of $\mathcal{N}(P_1) \wedge \cdots \wedge \mathcal{N}(P_k)$. We pass exactly one of the edges of $\mathcal{N}(P_1) \wedge \cdots \wedge \mathcal{N}(P_k)$—say the one with label (v, v'). Then $w' = w - v + v'$ is the next vertex of $P_1 + \cdots + P_k$.

Hence, all vertices of $P_1 + \cdots + P_k$ can be computed in $O(kn \log n)$ arithmetic operations. □

Proposition 2.3.9 can also be obtained by combining a convex hull algorithm with the topological sweeping method given in Guibas and Seidel [22]. The approach of [22] leads to remarkable output sensitive algorithms for various problems, including the construction of the Minkowski sum of two polytopes in $\mathbb{R}^3$. Note that the methods in [22] cannot be applied directly for further improving our complexity bounds because here we are allowing redundant $\mathcal{V}$-presentations of our input polygons.

The algorithms given in the proofs of Corollary 2.3.8 and Proposition 2.3.9 lead likewise to a polynomial time algorithm in the binary model. Here we may apply an LP-approach similar to the one used in the proof of Proposition 2.3.2 to reduce the given $\mathcal{V}$-presentations to irredundant $\mathcal{V}$-presentations.

COROLLARY 2.3.10. *The problem* FIXED-D-$\mathcal{V}$-MINKADD *can be solved in polynomial-time in the binary model of computation.*

3. Gröbner bases and Minkowski sums of Newton polytopes. We propose to use techniques from computational convexity to improve the performance of Buchberger's Gröbner bases algorithm (Algorithm 1.3.3). The key idea is to find good term orders by analyzing the Newton polytopes of the input (or intermediate) polynomials. Our approach is motivated by the observation that, for specific classes of *sparse* polynomial sets, varying the term order before or during the Buchberger completion may result in substantial time savings. It needs to be pointed out that the methods presented here will hardly be practical for input polynomials that are dense and sufficiently generic, for Bayer and Stillman [4] proved that in the generic situation the reverse lexicographic order is always optimal (with respect to the total degree of the output). However, even in this worst case, our polyhedral computations do not cause significant overhead for the Buchberger algorithm as they require only polynomial time when the number d of variables is fixed.

3.1. Dynamic versions of the Buchberger algorithm. Several papers in the computer algebra literature have addressed the question of how the specific choice of term order effects the computation of a Gröbner basis for a polynomial ideal. We start out with a brief summary of some results of Mora and Robbiano [28] and Bayer and Morrison [3].

Let $\mathcal{I} \subset K[\mathbf{x}]$ be a fixed ideal. Two term orders w_1 and w_2 are said to be *equivalent* (with respect to $\mathcal{I}$) provided $init_{w_1}(\mathcal{I}) = init_{w_2}(\mathcal{I})$. It is shown in [28] that there are only finitely many equivalence classes, and under the identification of Lemma 1.3.1, each equivalence class corresponds to an open, convex, polyhedral cone in $\mathbb{R}^d$. The polyhedral cell complex $\mathcal{G}(\mathcal{I})$ defined by these cones is called the *Gröbner fan* of the ideal $\mathcal{I}$. Suppose now that $\mathcal{I}$ is *homogeneous* (i.e., generated by homogeneous polynomials). In that case, also negative weights are permitted, and the Gröbner fan $\mathcal{G}(\mathcal{I})$ covers all of $\mathbb{R}^d$. In [3] we find the following convexity theorem.

THEOREM 3.1.1 (see [3]). *There exists a lattice polytope $P_\mathcal{I}$ in $\mathbb{R}^d$ whose normal fan $\mathcal{N}(P_\mathcal{I})$ equals the Gröbner fan $\mathcal{G}(\mathcal{I})$.*

Every polytope $P_\mathcal{I}$ with the above property will here be called a *state polytope* of the ideal $\mathcal{I}$. (This definition is slightly more general than the usual one from geometric invariant theory, where the "state polytope" is the convex hull of the weights of a certain $GL_d(K)$-module; see [3].) The vertices of any state polytope $P_\mathcal{I}$ are in one-to-one correspondence with all possible (equivalence classes of) Gröbner bases of $\mathcal{I}$. Note that state polytopes are a direct generalization of Newton polytopes: If $\mathcal{I} =< t >$ is a principal ideal, then its Gröbner fan $\mathcal{G}(\mathcal{I})$ equals the normal fan of the Newton polytope $N(t)$ (cf. Proposition 1.3.5).

For readers who are familiar with combinatorial optimization, we mention parenthetically that Gröbner bases algorithms also generalize the *greedy algorithm* for representable matroids. Suppose that $\mathcal{I} \subset K[\mathbf{x}]$ is a homogeneous ideal generated by linear polynomials, then the linear relations modulo $\mathcal{I}$ define a matroid on the set of variables $\{x_1, x_2, \ldots, x_d\}$, and every representable matroid arises in this manner. In this case, two weight vectors $w_1, w_2 \in \mathbb{R}^d$ are equivalent whenever they give rise to the same weightiest matroid base. Here the state polytope $P_\mathcal{I}$ equals the *matroid polytope*, which was introduced by Edmonds [16]. Of course, the situation is much more complicated for nonlinear ideals, and it is one objective of the present paper to stimulate further research on the applicability of techniques from polyhedral combinatorics to computer algebra algorithms.

We recall that the classical Buchberger algorithm is *static* in the following sense. Its input consists of a set of polynomials $\mathcal{T}$ *and* a term order w, and the output is a Gröbner basis $\mathcal{G}$ of the ideal $< \mathcal{G} >$ with respect to w. In a *dynamic* version, on the other hand, the term order becomes part of the output.

Dynamic Gröbner basis.

Input: *A set $\mathcal{T} \subset K[\mathbf{x}]$ of polynomials.*

Output: *A term order w and a Gröbner basis $\mathcal{G}$ for the ideal $< \mathcal{T} >$ with respect to w.*

The following example illustrates the advantages of this dynamic point of view.

EXAMPLE 3.1.2. Suppose that we are interested in finding the complex zeros of the (nonhomogeneous) polynomial ideal generated by

$$\mathcal{T} = \{x_1^5 + x_2^3 + x_3^2 - 1,\ x_1^2 + x_2^2 + x_3 - 1,\ x_1^6 + x_2^5 + x_3^3 - 1\} \quad \subset \quad \mathbb{C}[x_1, x_2, x_3].$$

For such a task computer algebraists usually recommend computing a purely lexicographic Gröbner basis [9, §6.6]. In our example, there are six lexicographic Gröbner bases, one for each ordering of the variables.

Unfortunately, each of the six lexicographic Gröbner bases for $< \mathcal{T} >$ contains high-degree polynomials with very large coefficients. For instance, a typical polynomial in the lexicographic Gröbner basis induced by $x_1 \prec x_2 \prec x_3$ has degree 21 and the maximal appearing integer coefficient is the 19-digit number 1553067597584776499. This means that the Gröbner basis computation is rather slow, and subsequent numerical approximations of the zeros, if desired, are difficult for reasons of numerical instability.

For this specific example, the problem DYNAMIC GRÖBNER BASIS has a much nicer solution. Suppose that the output term order is defined by the weight vector $w = (3, 4, 7)$. Our input set $\mathcal{T}$ has the leading monomials x_1^5, x_2^2, x_3^3 with respect to this term order. These monomials are relatively prime, and using [9, Lemma 6.4] we conclude that the set $\mathcal{T}$ is already a Gröbner basis; no algebraic computation was necessary. In §§3.2 and 3.3 it will be explained how "lucky" orders can be detected systematically, namely, by first computing the Minkowski sum of the corresponding Newton polytopes, here the three tetrahedra $N(t_i)$.

From the three leading monomials, we can read off that our ideal $< \mathcal{T} >$ is zero-dimensional [9, Method 6.9]. More precisely, there are $30 = 5 \cdot 2 \cdot 3$ zeros up to multiplicities in affine 3-space $\mathbb{C}^3$. If we wish to compute some or all of these zeros, then we can do so either symbolically or numerically. On the numerical side, we can use the method suggested by Auzinger and Stetter [1], which amounts to solving a joint eigenvector problem for three integer (30×30)-matrices. The largest integer entry in these three matrices is found to be 128 (in comparison to 1553067597584776499). If a symbolic solution is preferred, we may now apply the algorithm of Gianni [18] for transforming our Gröbner basis $\mathcal{T}$ into a lexicographic Gröbner basis. As in shown in [18], this detour toward a lexicographic Gröbner basis will often be significantly faster than starting with the lexicographic order in the first place.

As we have seen, the following is an important special case of DYNAMIC GRÖBNER BASIS.

Gröbner basis detection.

Input: *A set $\mathcal{T} \subset K[\mathrm{x}]$ of polynomials.*

Output: *A term order $w \in \mathbb{R}^d$ such that $\mathcal{T}$ is a Gröbner basis with respect to w, if such w exists; "NO" otherwise.*

In the next section, we will investigate the computational complexity of GRÖBNER BASIS DETECTION. This problem amounts to enumerating all (equivalence classes of) term orders for a given polynomial set $\mathcal{T} = \{t_1, \ldots, t_k\}$, which, in turn, is equivalent to computing the Minkowski sum of the Newton polytopes of the t_i, thus reducing GRÖBNER BASIS DETECTION to FIXED-D-$\mathcal{V}$-MINKADD, the problem studied in §2.3.

As the reader will have noted by now, the polynomial set $\mathcal{T}$ we chose for Example 3.1.2 was rather special and artificial, and, in most instances we will have to expect the output "NO" in the GRÖBNER BASIS DETECTION problem. However, an enumeration of possible term orders will not be wasted computation time if it enables us to make a good choice among these term orders. Therefore, we need to find an easy-to-compute measure for "closeness to being a Gröbner basis" for pairs $(\mathcal{T}, w)$ consisting of a polynomial set $\mathcal{T}$ and a term order w. In §3.3 we will derive such measures from the Hilbert function characterization of Gröbner bases in the homogeneous case. These enumerative measures satisfy the important requirements that they solve GRÖBNER BASIS DETECTION as a special case, and that (for fixed d) optimal term orders can be computed in polynomial time.

As the main new result in this section we prove the termination and correctness of the following general algorithm for DYNAMIC GRÖBNER BASIS. The important point here is that this procedure will not end up in an infinite loop, even for a succession of worst possible choices of term orders.

THEOREM AND ALGORITHM 3.1.3. *Given any finite generating set $\mathcal{G}_0$ of an ideal $\mathcal{I} \subset K[\mathbf{x}]$, then the following algorithm terminates with a Gröbner basis $\mathcal{G}_i$ for $\mathcal{I}$ with respect to the last term order w_i :*

```
$i = -1$
REPEAT
  $i := i + 1$;
  Choose a term order $w_i$
  $\mathcal{G}_{i+1} := \mathcal{G}_i \cup (\{\mathtt{normalform}_{\mathcal{G}_i, w_i}(\mathtt{S\text{-}polynomial}_{w_i}(p_1, p_2)) | p_1, p_2 \in \mathcal{G}_i\} \setminus \{0\})$
UNTIL $\mathcal{G}_{i+1} = \mathcal{G}_i$.
```

Before proving this theorem we need to make a few comments. For clarity of exposition, our generic algorithm is formulated as a trivial extension to Buchberger's Algorithm 1.3.3. It follows from the proof below that Algorithm 3.1.3 remains valid if the term order is allowed to be changed even more often than stated above, namely, after every individual nonzero S-polynomial reduction. Note that our extension is completely independent from other known speed-up techniques (such as intermediate reductions, predicting unnecessary S-polynomials, throwing in resultants, factoring, and so forth; cf. [9, §6.4]). Any practical implementation will contain some of these techniques. Naturally, we also wish to make the choices of w_i as effective as possible, and at this point we can include a subroutine based upon Corollary 3.3.6.

Proof of Algorithm 3.1.3. Recall that two term orders v_1 and v_2 are *equivalent* with respect to our ideal $\mathcal{I} :=< \mathcal{G}_\prime >$ if the corresponding initial ideals $init_{v_1}(\mathcal{I})$ and $init_{v_2}(\mathcal{I})$ are equal, and by [28, Lemma 2.6] the number of equivalence classes is finite. Let $\{v_1, v_2, \ldots, v_r\}$ be a system of representative term orders for $\mathcal{I}$.

Let M denote the set of all monomial ideals in $K[\mathbf{x}]$. We partially order the set M by inclusion. By Dickson's lemma or by Hilbert's basis theorem [9, p. 192], the poset M is *Noetherian*, that is, there are no infinite ascending chains $M_1 \subsetneq M_2 \subsetneq M_3 \subsetneq \cdots$ of monomial ideals. This implies that also the product poset $\mathbf{M}^r := \mathbf{M} \times \mathbf{M} \times \cdots \times \mathbf{M}$, defined by componentwise inclusion, is Noetherian.

Now suppose that Algorithm 3.1.3 did not terminate. This means the operation $\mathcal{G}_{i+1} := \mathcal{G}_i \cup \cdots$ produces an infinite sequence $\{\mathcal{G}_i\}_{i \in \mathbb{N}}$ of strictly increasing generating sets for $\mathcal{I}$. Let h_i denote one of the new nonzero reduced S-polynomials in $\mathcal{G}_{i+1}$ computed with respect to the term order w_i. The leading monomial of h_i is contained in the monomial ideal $< init_{w_i}(\mathcal{G}_{i+1}) >$ but not in $< init_{w_i}(\mathcal{G}_i) >$.

With every $\mathcal{G}_i$, we associate an element in the product poset $\mathbf{M}^r$, namely, the vector

$$(< init_{v_1}(\mathcal{G}_i) >, < init_{v_2}(\mathcal{G}_i) >, \cdots, < init_{v_r}(\mathcal{G}_i) >)$$

of initial ideals with respect to the term orders v_j. Because $\{v_j\}$ forms a system of representatives, there exists an index j_i such that w_i is equivalent to v_{j_i}. This implies that the vector $(< init_{v_1}(\mathcal{G}_{i+1}) >, \cdots, < init_{v_r}(\mathcal{G}_{i+1}) >)$ is larger in $\mathbf{M}^r$ than the previous vector $(< init_{v_1}(\mathcal{G}_i) >, \cdots, < init_{v_r}(\mathcal{G}_i) >)$. This is a contradiction to the Noetherianess of the poset $\mathbf{M}^r$, which implies that Algorithm 3.1.3 terminates.

The correctness of Algorithm 3.1.3 is easily seen. The termination condition $\mathcal{G}_{i+1} = \mathcal{G}_i$ is satisfied only if all S-polynomials of pairs in $\mathcal{G}_i$ reduce to zero with respect to w_i. Hence, $\mathcal{G}_i$ is a Gröbner basis with respect to w_i by Buchberger's criterion (Corollary 1.3.4).

3.2. Newton polytopes and the complexity of Gröbner basis detection. The Newton polytope $N(t)$ of a single polynomial $t = \sum_{i=1}^{n} c_i \mathbf{x}^{\alpha_i}$ is the convex hull of its monomials, that is, $N(t) = \text{conv}\{\alpha_1, \alpha_2, \ldots, \alpha_n\} \subset \mathbb{R}^d$. We now define the *Newton polytope* of a set of polynomials $\mathcal{T} = \{t_1, t_2, \ldots, t_k\}$ to be the *Minkowski sum*

$$N(\mathcal{T}) = N(t_1) + N(t_2) + \cdots + N(t_k)$$

of the respective Newton polytopes, or, equivalently (by Remark 1.3.6), as the Newton polytope of the product $t_1 t_2 \cdots t_k$. To generalize Proposition 1.3.5 to this situation, we define the *affine Newton polyhedron* of $\mathcal{T}$ to be the Minkowski sum

$$N_{\text{aff}}(\mathcal{T}) := N(\mathcal{T}) + \mathbb{R}^d_-$$

of the Minkowski polytope with the negative orthant.

Two term orders w_1 and w_2 on the polynomial ring $K[\mathbf{x}]$ are said to be *equivalent* with respect to a finite subset $\mathcal{T} \subset K[\mathbf{x}]$, provided that $init_{w_1}(t) = init_{w_2}(t)$ for all $t \in \mathcal{T}$. Note that this is a finer equivalence relation than the one defined for ideals in the previous section. The proof of the following theorem is self-contained; however, we wish to draw the reader's attention to the close connection with Lemma 2.1.5. In fact, two term orders are equivalent if and only if the corresponding weight vectors are contained in the same maximal cell of $\mathcal{N}(N(\mathcal{T})) = \mathcal{N}(N(t_1) \wedge \cdots \wedge \mathcal{N}(N(t_k))$.

PROPOSITION 3.2.1. *Let $\mathcal{T}$ be a finite set of polynomials in $K[\mathbf{x}]$. Then the vertices of the affine Newton polyhedron $N_{\text{aff}}(\mathcal{T})$ are in one-to-one correspondence with the equivalence classes of term orders with respect to $\mathcal{T}$.*

Proof. Suppose that $t_i = \sum_{j=1}^{n_i} c_{ij} \mathbf{x}^{\alpha_{ij}}$ for $i = 1, 2, \ldots, k$. By Lemma 1.3.1, we may identify each term order with a weight vector $w \in \mathbb{R}^d_+$. In this identification, only those vectors w appear that do not produce a tie among two monomials α_{ij}. Two term orders w_1 and w_2 are equivalent with respect to $\mathcal{T}$ if and only if

$$\max\{\langle \alpha_{ij}, w_1 \rangle | 1 \leq j \leq n_i\} = \max\{\langle \alpha_{ij}, w_2 \rangle | 1 \leq j \leq n_i\}$$

for all $i \in 1, \ldots, k$. Consider the set of (transversal) index vectors

$$\mathbf{J} := \{\mathbf{j} = (j_1, j_2, \ldots, j_k) \in \mathbb{N}^k | 1 \leq j_i \leq n_i \quad \text{for all } i \in 1, \ldots, k\}.$$

With each element of $\mathbf{J}$, we associate the (possibly empty) open polyhedral cone

$$C_{\mathbf{j}} := \{w \in \mathbb{R}^d_+ | \langle \alpha_{ij_i}, w \rangle > \langle \alpha_{ij}, w \rangle \quad \text{for all } i \in \{1, \ldots, k\}, j \in \{1, \ldots, n_i\} \setminus \{j_i\}\}.$$

A weight vector w is contained in $C_{\mathbf{j}}$ if and only if monomials indexed by $\mathbf{j}$ are the leading terms of $\mathcal{T}$ with respect to w. Hence the equivalence classes of term orders with respect to $\mathcal{T}$ are in one-to-one correspondence with the nonempty $C_{\mathbf{j}}$'s.

The Newton polytope $N(\mathcal{T})$ is the convex hull of the points $\alpha_{\mathbf{j}} := \sum_{i=1}^{k} \alpha_{ij_i}$, where $\mathbf{j} = (j_1, \ldots, j_k)$ ranges over all elements of $\mathbf{J}$. The set of vertices of the affine Newton polyhedron $N_{\text{aff}}(\mathcal{T})$ is a subset of the vertices of $N(\mathcal{T})$. A point $\alpha_{\mathbf{j}}$ is a vertex of $N_{\text{aff}}(\mathcal{T}) =$

$N(\mathcal{T}) + \mathbb{R}^d_-$ if and only if $\alpha_\mathbf{j}$ is the maximum of some linear functional from $\mathbb{R}^d_+$. This means that there exists a positive vector w such that $\langle \alpha_\mathbf{j}, w\rangle = \sum_{i=1}^k \langle \alpha_{j_i}, w\rangle$ is larger than $\langle \alpha_{\mathbf{j}'}, w\rangle$ for every other $\mathbf{j}' \in \mathbf{J}$. This condition, however, is equivalent to $w \in C_\mathbf{j}$. We have shown that the nonempty $C_\mathbf{j}$'s are the normal cones to the vertices of the polyhedron $N_{\text{aff}}(\mathcal{T})$. This proves the claim. □

For the important special case of *homogeneous* polynomials, we get the following result.

COROLLARY 3.2.2. *Let $\mathcal{T}$ be a finite set of homogeneous polynomials in $K[\mathbf{x}]$. Then the vertices of the Newton polytope $N(\mathcal{T})$ are in one-to-one correspondence with the equivalence classes of term orders with respect to $\mathcal{T}$.*

Proof. Suppose that t_i is homogeneous of degree R_i and let $R := R_1 + R_2 + \cdots + R_k$. Then the Newton polytope $N(\mathcal{T})$ is contained in the affine hyperplane $\{y \in \mathbb{R}^d | \sum_{j=1}^d y_j = R\}$. If a vertex $\alpha_\mathbf{j}$ of $N(\mathcal{T})$ is extremal with respect to some direction vector w, then it is also extremal in the direction $w + (c, c, \dots, c)$ for every $c \in \mathbb{R}_+$. Choosing c sufficiently large, we find that $\alpha_\mathbf{j}$ is also a vertex of $N_{\text{aff}}(\mathcal{T})$. □

In Proposition 3.2.1 and Corollary 3.2.2, we have seen that enumerating all possible term orders for a set of multivariate polynomials reduces to the problem of constructing the Minkowski sum of convex polytopes in $\mathcal{V}$-presentation. We obtain a system of representatives for the term orders relative to a set $\mathcal{T}$ of homogeneous polynomials by choosing one weight vector in the open normal cone of each vertex of $N(\mathcal{T})$.

COROLLARY 3.2.3. *Let $P_\mathcal{I}$ be a state polytope of a homogeneous ideal $\mathcal{I}$ and let $\mathcal{U} \subset \mathcal{I}$ be any universal Gröbner bases of $\mathcal{I}$. Then there is a $\lambda \in \mathbb{R}_+$ such that $\lambda P_\mathcal{I}$ is a Minkowski summand of $N(\mathcal{U})$.*

Proof. By Theorem 3.1.1 and Corollary 3.2.2, the maximum cells of the normal fan $\mathcal{N}(P_\mathcal{I})$ are unions of closures of maximum cells of the normal fan $\mathcal{N}(N(\mathcal{U}))$. This implies that $\mathcal{N}(N(\mathcal{U})) \preceq \mathcal{N}(P_\mathcal{I})$. The assertion follows from Lemma 2.1.6. □

The special case where $N(\mathcal{T})$ is a zonotope is of considerable interest for theoretical computer science. Suppose that all input polynomials t_i are differences of monomials, that is, $t_i = \mathbf{x}^{\alpha_i} - \mathbf{x}^{\beta_i}$ for $i = 1, \dots, k$. In this case, the ideal membership problem is the *word problem for commutative semigroups* [9, §6.10]. Besides the practical importance of this problem, it is noteworthy that the doubly-exponential lower-bound construction of Mayr and Meyer [25] has this special form. Here the Newton polytope $N(\mathcal{T})$ equals the Minkowski sum of k line segments $\text{conv}\{\alpha_i, \beta_i\}$, and, by our discussion in §2.2, the vertices of this zonotope are in one-to-one correspondence with the maximal cells in the hyperplane arrangement $\{\{x \in \mathbb{R}^d | \langle \alpha_i - \beta_i, x\rangle = 0\} | i = 1, 2, \dots, k\}$.

Returning to the general case, we will now prove a complexity result for GRÖBNER BASIS DETECTION. To this end, we need two lemmas. The first lemma deals with the the size of weight vectors that correspond to term orders with respect to a given set of polynomials. It is formulated in the language of lattice polytopes.

LEMMA 3.2.4. *Let $P = \text{conv}\{v_1, \dots, v_n\}$ be an irredundantly $\mathcal{V}$-presented polytope in $\mathbb{R}^d$ and let $v_1, \dots, v_n \in \mathbb{Z}^d$ with absolute value of coordinates bounded by a constant R. Then, for every index $i = 1, \dots, n$, there is a vector $w_i \in \mathcal{N}(\{v_i\}, P)$, which has integer coordinates whose absolute values are bounded by $(2dR)^{2d}$.*

Proof. Let $a \in V = \{v_1, \dots, v_n\}$. Consider the following system of linear inequalities:

$$\langle a, w\rangle - \langle v, w\rangle \geq 1 \quad \text{for } v \in V \setminus \{a\}.$$

Clearly, this system is feasible, and all solutions lie in $\mathcal{N}(\{a\}, P)$. Hence, the set of all such solutions is a polyhedron contained in the cone $\mathcal{N}(\{a\}, P)$ and therefore has a vertex. Any such vertex w_0 is given as the solution to a system of d linear equations

$$\langle a, w\rangle - \langle v, w\rangle = 1 \quad \text{for some } d \text{ points } v \in V \setminus \{a\}.$$

Hence, by Cramer's rule, w_0 is a rational vector (with common denominator) such that the absolute values of the numerators and the denominator of its coordinates is bounded by $d!(2R)^d$. Thus, there is a vector in $\mathcal{N}(\{a\}, P) \cap \mathbb{Z}^d$ with coordinates bounded by $(2dR)^{2d}$. □

The next lemma is concerned with the number of different monomials that are smaller than the leading monomial (with respect) to a given weight vector. Again, it is formulated in term of solutions of a system of diophantine inequalities.

LEMMA 3.2.5. *Let $v, w \in \mathbb{Z}^d$, $v, w > 0$, let the coordinates of v be bounded by r and let the coordinates of w be at most s. Then the number of integer solutions x of the system*

$$\begin{aligned} \langle w, x\rangle - \langle w, v\rangle &\leq 0, \\ x &\geq 0 \end{aligned}$$

is bounded above by $(drs + 1)^d$.

Proof. The hyperplane $\langle w, x\rangle = \langle w, v\rangle$ intersects the ith coordinate axis at distance $\lambda_i := \langle w, v\rangle / \langle w, e_i\rangle$ from the origin. Hence, the set of solutions to the above system is contained in the box $\sum_{i=1}^{d} [0, \lambda_i] e_i$. The number of integer solutions is, therefore, bounded by $\Pi_{i=1}^{d} (\lambda_i + 1)$. This implies the bound stated in the lemma. □

THEOREM 3.2.6. *Let $\mathcal{T} = \{t_1, t_2, \ldots, t_k\} \subset K[x_1, \ldots, x_d]$, where d is fixed, and suppose that each t_i has at most n monomials of total degree at most R. Then* GRÖBNER BASIS DETECTION *can be solved in at most $O(k^{d+2} n^{2d-1} R^{(2d+1)d})$ arithmetic operations.*

Proof. The Newton polytopes $N(t_1), \ldots, N(t_k)$ have at most n vertices with integer coordinates whose ℓ_1-norms are bounded above by R. To deal with affine Newton polytopes, we consider the unit cube $C^d = \operatorname{conv}\left(\{-1, 1\}^d\right)$. Observe, that the size of C^d (in any model under consideration) is constant. Then, by Lemmas 3.2.5 and 2.1.5, the equivalence classes of term orders with respect to $\mathcal{T}$ correspond to those vertices of

$$P := N(\mathcal{T}) + C^d = N(t_1) + \cdots + N(t_k) + C^d,$$

whose normal cone is contained in $\mathbb{R}^d_+$. By Theorem 2.3.7, we can compute P in at most $O(k^d n^{2d-1})$ arithmetic operations. Furthermore, we compute for each vertex of P a reference point in the interior of the cone of outer normals. With the aid of Lemma 3.2.4, we see that we can assume that these vectors are integral and have coordinates whose absolute values are bounded from above by $(2dR)^{2d}$. In the following, this fact will only be used for estimating the number of reduction steps. Because the reduction algorithm is invariant under changes to equivalent weight vectors, the weight vectors stemming from Theorem 2.3.7 are fine for the reduction. Let $w_1, \ldots, w_m$ be those reference points whose coordinates are all positive. Clearly, $m = O(k^d n^{2d-1})$. The vectors $w_1, \ldots, w_m$ correspond to the equivalence classes of term orders with respect to $\mathcal{T}$. In the following, let $w \in \{w_1, \ldots, w_m\}$.

According to Corollary 1.3.4, $\mathcal{T}$ is a Gröbner basis with respect to w if and only if the S-polynomial of any two elements of $\mathcal{T}$ reduces to 0. Let $p_1, p_2 \in \mathcal{T}$ and let $s =$ S-polynomial$_w(p_1, p_2)$. The total degree of s is at most $2R$, whence $N(s)$ is contained in $\mathbb{R}^d_+ \cap \{x | x_1 + \cdots + x_d \leq 2R\}$. After a reduction step with respect to any of the elements in $\mathcal{T}$, the new polynomial s' might have total degree higher than $2R$. The maximum of the linear functional $\langle w, \cdot\rangle$ over $N(s')$ is, however, strictly less than over $N(s)$. Therefore the number of reductions is bounded by the "lattice breadth" of $N(s)$ in direction w. By Lemma 3.2.5, this number is bounded by $(2dR(2dR)^{2d} + 1)^d$, which is in $O(R^{(2d+1)d})$.

The reduction must be carried out for each pair of polynomials in $\mathcal{T}$, and we must consider each of the vectors w_i separately. Hence, the total number of arithmetic operations is of order $O(k^{d+2}n^{2d-1}R^{(2d+1)d})$, as claimed in the statement of our theorem. □

It is not hard to see that a similar result holds also for the binary model of computation.

COROLLARY 3.2.7. *In the binary model of computation,* GRÖBNER BASIS DETECTION *can be solved in time that is polynomial in* L*, the size of the input, and in* R*, the total degree of the participating polynomials.*

Proof. The only thing that remains to be shown is that the coefficients of our polynomial do not grow exponentially. These coefficient are only changed in the S-polynomial step and in the subsequent reductions. Because the number of reductions is at most $O(R^{(2d+1)d})$, the size of the constants is at most $O(R^{(2d+1)d}L)$. □

We remark that Corollary 3.2.7 does *not* imply that GRÖBNER BASIS DETECTION can be solved in polynomial time in the binary model of computation. For only the logarithm of R and not R itself is part of the input (cf. §1.3). The following simple example shows that in the binary model, even for fixed $d = 1$, normal form reductions require exponential time: Let $s = x^{2R}$ and $\mathcal{T} = \{x^{R+1} - x^R\}$. The normal form reduction of s with respect to $\mathcal{T}$ requires R steps, while the binary size of the data is in $O(\log(R))$.

3.3. Measures for closeness to being a Gröbner basis. Let $\mathcal{T}$ be a fixed generating set of a polynomial ideal $\mathcal{I} \subset K[\mathbf{x}]$ and let w be any term order. In this section, we introduce a function $\Theta_{\mathcal{T}}(w)$, which provides an a priori measure for the deviation of $\mathcal{T}$ from being a Gröbner basis with respect to w. Extending our results from the previous sections, we show that $\Theta_{\mathcal{T}}(\cdot)$ also solves GRÖBNER BASIS DETECTION and can be computed in polynomial time for fixed d.

For simplicity of exposition, we assume throughout this section that $\mathcal{I}$ is homogeneous. The reader familiar with Gröbner bases theory will note that our techniques can be generalized to affine ideals by introducing an extra homogenizing variable.

We begin with a brief review of some standard commutative algebra techniques for measuring the "size" of a homogeneous ideal $\mathcal{I} \subset K[\mathbf{x}]$. Let $\mathcal{I}_r$ denote the set of homogeneous polynomials of degree r in $\mathcal{I}$. As a K-vector space, $\mathcal{I}$ is the direct sum of the finite-dimensional K-vector spaces $\mathcal{I}_r$. The *Hilbert function* of $\mathcal{I}$ is the numerical function $h_{\mathcal{I}} : \mathbb{N} \to \mathbb{N}, r \mapsto \dim_K(\mathcal{I}_r)$, which measures the dimensions of these K-vector spaces. For instance, the Hilbert function of the full polynomial ring $K[\mathbf{x}]$ counts the number of all monomials of a given degree r, and we abbreviate this number by $h(r) := h_{K[\mathbf{x}]}(r) = \binom{d+r-1}{d-1}$.

For many purposes, it is convenient to express the Hilbert function in the form of a generating function $H_{\mathcal{I}}(z) := \sum_{r=0}^{\infty} h_{\mathcal{I}}(r)z^r$. The formal power series $H_{\mathcal{I}}$ is called the *Hilbert series* of $\mathcal{I}$. For instance, the Hilbert series of the full polynomial ring $K[\mathbf{x}]$ equals $H(z) := H_{K[\mathbf{x}]}(z) = (1-z)^{-d}$.

Computing the Hilbert function (or Hilbert series) of an ideal $\mathcal{I}$ is a typical application of the Buchberger algorithm (cf. [9]). More generally, once we know a Gröbner basis for $\mathcal{I}$, we also get an easy explicit K-linear basis for $\mathcal{I}_r$. For the purposes of this section, the following observation is sufficient.

REMARK 3.3.1. *Let* $\mathcal{I}$ *be a homogeneous ideal in* $K[\mathbf{x}]$ *and* w *any term order. Then* $\mathcal{I}$ *and its initial ideal* $\mathit{init}_w(\mathcal{I})$ *have the same Hilbert function.*

This reduces the computation of general Hilbert functions to the problem of computing the Hilbert function $h_{\mathcal{M}}$ of an ideal $< \mathcal{M} >$ that is generated by a set of monomials $\mathcal{M} = \{m_1, m_2, \ldots, m_k\}$. Note that, because $\mathcal{M}$ can be regarded as a subset of

$\mathbb{N}^d$, this is a purely combinatorial problem, and, in fact, the principle of inclusion and exclusion implies the following explicit formula.

PROPOSITION 3.3.2. *The number of monomials of degree r in* $< \mathcal{M} >$ *equals*

$$h_{\mathcal{M}}(r) = \sum_{s=1}^{k} \sum_{\{m_{i_1},\ldots,m_{i_s}\}\subset\mathcal{M}} (-1)^{s+1} h\Big(r - \deg\big(\mathrm{lcm}(m_{i_1},\ldots,m_{i_s})\big)\Big).$$

Here lcm *denotes the least common multiple, and, by the usual conventions for binomial coefficients,* $h(\cdot)$ *is zero for negative values.*

The combination of Remark 3.3.1 and Proposition 3.3.2 is an easy Gröbnerian proof for the well-known fact that, for all sufficiently large integers $r \gg 0$, the Hilbert function $h_{\mathcal{I}}(r)$ is equal to a polynomial $p_{\mathcal{I}}(r)$, called the *Hilbert polynomial* of $\mathcal{I}$. For computational purposes, on the other hand, Proposition 3.3.2 is not very useful. Because the formula involves 2^k summands, its evaluation requires exponential time in the crucial parameter k, regardless of whether the dimension d is fixed.

A substantially better method for computing the Hilbert polynomial and the Hilbert series has been given by Bayer and Stillman [5]. Here the Hilbert series in the output is represented as the quotient of two polynomials. Their algorithm is implemented in MACAULAY and has proved to be very efficient even for fairly large sets of monomials.

THEOREM 3.3.3 (see [5]). *The Hilbert series* $H_{\mathcal{M}}$ *of the ideal generated by a set* $\mathcal{M}$ *of k monomials in d variables can be computed in* $O(k^d)$ *arithmetic operations.*

We will now apply these results to the problem DYNAMIC GRÖBNER BASIS. Let $\mathcal{T} = \{t_1, t_2, \ldots, t_k\}$ be a fixed (homogeneous) generating set of a homogeneous ideal $\mathcal{I} \subset K[\mathrm{x}]$. Given any term order w, then we write $init_w(\mathcal{T}) = \{init_w(t_1), \ldots, init_w(t_k)\}$ for the set of initial monomials. Its monomial ideal $< init_w(\mathcal{T}) >$ is a subset of the monomial ideal $init_w(\mathcal{I})$, and equality holds if and only if $\mathcal{T}$ is a Gröbner basis for $\mathcal{I}$ with respect to w. This implies the following enumerative characterization of Gröbner bases.

THEOREM 3.3.4. *A set* $\mathcal{T}$ *of homogeneous polynomials of degree* $\leq R$ *is a Gröbner basis of its ideal* $\mathcal{I}$ *with respect to a term order* w *if and only if* $h_{\mathcal{I}}(r) = h_{<init_w(\mathcal{T})>}(r)$ *for all* $r = 1, 2, \ldots, 2R-1$.

Proof. The "only if" part follows immediately from Remark 3.3.1. To prove the "if" direction, let us assume that $\mathcal{T}$ is not a Gröbner basis. By Buchberger's criterion (Corollary 1.3.4), there exists an S-polynomial $S(t_i, t_j)$ that does not reduce to zero. Let t be a (nonzero) normal form of $S(t_i, t_j)$ modulo $\mathcal{T}$ with respect to w. Our degree hypothesis implies that both t and $S(t_i, t_j)$ are homogeneous polynomials of degree $r \leq 2R-1$. (Here $r = 2R$ would imply that the initial monomials of t_i and t_j were relatively prime.) The K-vector space $< init_w(\mathcal{T}) >_r$ is a proper linear subspace of $(init_w(\mathcal{I}))_r$ because the monomial $init_w(t)$ is contained in their difference. Both K-vector spaces being finite-dimensional, this implies strict inequality between their dimensions

$$h_{<init_w(\mathcal{T})>}(r) = \dim_K\big(< init_w(\mathcal{T}) >_r\big) < \dim_K\big(init_w(\mathcal{I})\big)_r = h_{<init_w(\mathcal{I})>}(r).$$

By Remark 3.3.1, the right-hand side is equal to $h_{\mathcal{I}}(r)$. This completes the proof. □

This result suggests the following definitions. As before, $\mathcal{T}$ is a fixed set of polynomials of degree at most R. The *tentative Hilbert function* $\Theta_{\mathcal{T}}(w)$ depends on the term order w and is defined by $\Theta_{\mathcal{T}}(w, r) := h_{<init_w(\mathcal{T})>}(r)$ for $r \in \mathbb{N}$. By the above observations, the tentative Hilbert function is bounded above by the "true" Hilbert function $h_{\mathcal{I}}$ of the ideal $\mathcal{I} =< \mathcal{T} >$. Theorem 3.3.4 states in other words that equality holds if and only if $\mathcal{T}$ is a Gröbner basis with respect to w.

In practice we will not know the true Hilbert function $h_{\mathcal{I}}$ beforehand, but using the procedure of Bayer and Stillman (Theorem 3.3.3) we can compare tentative Hilbert functions for different term orders.

Here a term order w_1 is said to be *preferable* to w_2 if the first nonzero entry in the vector $\left(\Theta_{\mathcal{T}}(w_1, r) - \Theta_{\mathcal{T}}(w_2, r)\right)_{r=1,2,\ldots,2R-1}$ is positive. A term order w is said to be *optimal* if it is preferable to all other term orders.

Optimal term order.

Input: *A set $\mathcal{T} \subset K[\mathbf{x}]$ of homogeneous polynomials.*

Output: *An optimal term order $w \in \mathbb{R}^d$ for $\mathcal{T}$.*

The next corollary follows directly from Theorem 3.3.4. It shows that our earlier problem GRÖBNER BASIS DETECTION is just a special case of OPTIMAL TERM ORDER, and every algorithm for the latter automatically solves the first.

COROLLARY 3.3.5. *Let $\mathcal{T}$ be a set of homogeneous polynomials in $K[\mathbf{x}]$. If $\mathcal{T}$ is a Gröbner basis with respect to a term order w, then every such term order is optimal.*

Combining the computational results of the previous section (Corollary 3.2.3) with the Hilbert series algorithm of Bayer and Stillman (Theorem 3.3.3), we obtain the desired polynomial complexity bound for optimizing term orders.

THEOREM 3.3.6. *Let $\mathcal{T} = \{t_1, t_2, \ldots, t_k\} \subset K[x_1, \ldots, x_d]$, where d is fixed, and suppose that each t_i has at most n monomials of total degree at most R. Then* OPTIMAL TERM ORDER *can be solved in at most $O(k^{d(d+2)} n^{2d-1} R^{(2d+1)d})$ arithmetic operations.*

REFERENCES

[1] W. AUZINGER AND H. J. STETTER, *An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations*, Internat. Ser. Numer. Math., 86 (1988), pp. 11–30.

[2] I. BÁRÁNY, *A vector-sum theorem and its application to improving flow shop guarantees*, Math. Oper. Res., 6 (1981), pp. 445–452.

[3] D. BAYER AND I. MORRISON, *Standard bases and geometric invariant theory* I. *Initial ideals and state polytopes*, J. Symbolic Comput., 6 (1988), pp. 209–217.

[4] D. BAYER AND M. STILLMAN, *A criterion for detecting m-regularity*, Inventiones Math., 87 (1987), pp. 1–11.

[5] ———, *Computation of Hilbert functions*, J. Symbolic Comput., 14 (1992).

[6] H. L. BODLAENDER, P. GRITZMANN, V. KLEE, AND J. VAN LEEUWEN, *Computational complexity of norm maximization*, Combinatorica, 10 (1990), pp. 203–225.

[7] T. BONNESER AND W. FENCHEL, *Theorie der konvexen Körper*, Springer, Berlin, 1934.

[8] B. BUCHBERGER, *Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystemes*, Aequationes Math., 4 (1970), pp. 374–383.

[9] ———, *Gröbner bases—an algorithmic method in polynomial ideal theory*, in Multidimensional Systems Theory, N. K. Bose, ed., D. Reidel, 1985, Chapter 6.

[10] ———, *Applications of Gröbner bases in non-linear computational geometry*, in I.M.A. Volumes in Mathematics and Its Applications, Vol. 14, Scientific Software, J. R. Rice, ed., Vol. 14, Springer, New York, 1988.

[11] R. C. BUCK, *Partition of space*, Amer. Math. Monthly, 50 (1943), pp. 541–544.

[12] M. E. DYER, *The complexity of vertex enumeration methods*, Math. Oper. Res., 8 (1983), pp. 381–402.

[13] H. EDELSBRUNNER, *Algorithms in Combinatorial Geometry*, Springer, New York, 1987.

[14] H. EDELSBRUNNER, J. O'ROURKE, AND R. SEIDEL, *Constructing arrangements of lines and hyperplanes with applications*, SIAM J. Comput., 15 (1986), pp. 341–363.

[15] H. EDELSBRUNNER, R. SEIDEL, AND M. SHARIR, *On the zone theorem for hyperplane arrangements*, in New Results and Trends in Computer Science, H. Maurer, ed., Springer Lecture Notes in Computer Science, Berlin, 1991, pp. 108–123.

[16] J. EDMONDS, *Matroids and the greedy algorithm*, Math. Programming, 1 (1971), pp. 127–136.

[17] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability*, W. H. Freeman, San Francisco, 1979.

[18] P. GIANNI, *Efficient computation of zero-dimensional Gröbner bases by change of ordering*, presented at the Conference on Computers and Commutative Algebra (COCOA II), Genova, Italy, June 1989.

[19] C. C. GONZAGA, *An algorithm for solving linear programming problems in $O(n^3 L)$ operations*, in Progress in Mathematical Programming, N. Megiddo, ed., Springer, New York, 1989, pp. 1–28.

[20] P. Gritzmann and V. Klee, *Computational aspects of zonotopes and their polars*, in preparation.

[21] B. Grünbaum, *Convex Polytopes*, Wiley-Interscience, London, 1967.

[22] L. J. Guibas and R. Seidel, *Computing convolutions by reciprocal search*, Discrete Comput. Geom., 2 (1987), pp. 175–193.

[23] N. M. Korneenko and H. Martini, *The minsum hyperplane problem*, in New Trends in Discrete and Computational Geometry, J. Pach, ed., Springer, New York, 1992.

[24] J. Matoušek and O. Schwarzkopf, *Linear optimization queries*, Math. Tech. Report FU Berlin, B91-19, 1991.

[25] E. Mayr and A. Meyer, *The complexity of the word problem for commutative semigroups and polynomial ideals*, Adv. Math., 46 (1982), pp. 305–329.

[26] P. McMullen, *The maximum number of faces of a convex polytope*, Mathematika, 17 (1970), pp. 179–184.

[27] N. Megiddo, *Linear programming in linear time when the dimension is fixed*, J. Assoc. Comput. Mach., 31 (1984), pp. 114–127.

[28] T. Mora and L. Robbiano, *The Gröbner fan of an ideal*, J. Symbolic Comput., 6 (1988), pp. 183–208.

[29] A. Ostrowski, *Über die Bedeutung der Theorie der konvexen Polyeder für die formale Algebra*, Jahresberichte Deutsche Math. Verein., 30 (1921), pp. 98–99.

[30] ———, *On multiplication and factorization of polynomials* I. *Lexicographic orderings and extreme aggregates of terms*, Aequationes Math., 13 (1975), pp. 201–228.

[31] J. Renegar, *A polynomial-time algorithm based on Newton's method for linear programming*, Math. Programming, 40 (1988), pp. 59–93.

[32] L. Robbiano, *On the theory of graded structures*, J. Symbolic Comput., 2 (1986), pp. 139–170.

[33] R. Seidel, *Output-Size Sensitive Algorithms for Constructive Problems in Computational Geometry*, Ph.D. dissertation, Department of Computer Science, Cornell University, Ithaca, NY, 1987.

[34] G. Swart, *Finding the convex hull facet by facet*, J. Algorithms, 6 (1985), pp. 17–38.

[35] P. Vaidya, *An algorithm for linear programming which requires $O((m+n)n^2+(m+n)^{1.5}n)L)$ arithmetic operations*, Math. Programming, 41 (1990), pp. 175–201.

[36] V. Weispfenning, *Constructing universal Gröbner bases*, Proceedings AAECC-5, Menorca 1987, Lecture Notes Comput. Sci., 356 (1987), pp. 408-417.

[37] T. Zaslavsky, *Facing up to arrangements: Face-count formulas for partitions of space by hyperplanes*, Mem. Amer. Math. Soc., 154 (1975).

SIAM J. DISC. MATH.
Vol. 6, No. 2, pp. 270–273, May 1993

FINDING A LONGEST PATH IN A COMPLETE MULTIPARTITE DIGRAPH*

G. GUTIN†

Abstract. A digraph obtained by replacing each edge of a complete m-partite graph with an arc or a pair of mutually opposite arcs with the same end vertices is called a complete m-partite digraph. An $O(n^3)$ algorithm for finding a longest path in a complete m-partite ($m \geq 2$) digraph with n vertices is described in this paper. The algorithm requires time $O(n^{2.5})$ in case of testing only the existence of a Hamiltonian path and finding it if one exists. It is simpler than the algorithm of Manoussakis and Tuza [*SIAM J. Discrete Math.*, 3 (1990), pp. 537–543], which works only for $m = 2$. The algorithm implies a simple characterization of complete m-partite digraphs having Hamiltonian paths that was obtained for the first time in Gutin [*Kibernetica* (*Kiev*), 4 (1985), pp. 124–125] for $m = 2$ and in Gutin [*Kibernetica* (*Kiev*), 1 (1988), pp. 107–108] for $m \geq 2$.

Key words. digraph, longest path, polynomial algorithm

AMS(MOS) subject classifications. 05C38, 05C45, 68R10

1. Introduction and terminology. In this note we consider only digraphs without loops, unless otherwise specified. A digraph D on m disjoint vertex classes is called a complete m-partite (multipartite) digraph (CMD) if for any two vertices u, v in different classes either (u, v) or (v, u) (or both) is an arc of D.

In [1] a characterization was given of complete bipartite digraphs (CBD) containing Hamiltonian paths. This characterization was generalized to CMD in [2]. Using another approach, Häggkvist and Manoussakis gave in [3] analogous characterization of CBD having a Hamiltonian path. The results in [1] and [2] supply an $O(n^{2.5})$ algorithm for checking if a given CMD with n vertices has a Hamiltonian path.

Manoussakis and Tuza obtained in [4] an $O(n^{2.5})$ algorithm for finding a Hamiltonian path in a complete oriented bipartite graph B (if B has a Hamiltonian path). In this work, we describe an $O(n^3)$ algorithm for finding a longest path in a CMD. This algorithm requires time $O(n^{2.5})$ in the case of testing only the existence of a Hamiltonian path and finding it, if one exists. It is simpler than the algorithm of Manoussakis and Tuza [4] (in the case where $m = 2$ particularly, see §3) and does not require an algorithm for finding a Hamiltonian cycle (as in [4]). Our algorithm implies a simple characterization of CMD, having Hamiltonian paths [2].

$V(D)$, $A(D)$ are the sets of vertices and arcs of a digraph D. A digraph D is called 1-diregular if $d^+(x) = d^-(x) = 1$ for any $x \in V(D)$. A digraph D is called almost 1-diregular if there exist vertices x, y (possibly $x = y$), such that $d^+(x) = d^-(y) = 0$, and $d^+(z) = 1$ for $z \in V(D)\backslash x$, $d^-(v) = 1$ for $v \in V(D)\backslash y$. It is easy to see that a 1-diregular digraph F represents a collection of vertex disjoint cycles $C_1, C_2, \ldots, C_t$ ($t \geq 1$), i.e., $F = C_1 \cup C_2 \cup \cdots \cup C_t$. Similarly, an almost 1-diregular digraph $S = C_0 \cup C_1 \cup C_2 \cup \cdots \cup C_q$, where C_0 is a path (which may have only 1 vertex); $C_1, C_2, \ldots, C_q$ are cycles; $V(C_i) \cap V(C_j) = \emptyset$ for $0 \leq i \neq j \leq q$, $q \geq 0$.

If $C = (x_1, x_2, \ldots, x_p, x_1)$ is a cycle and $P = (y_1, y_2, \ldots, y_q)$ is a path, then

$$PT(y_i, y_j, P) \quad \text{is the path} \quad (y_i, y_i + 1, \ldots, y_j) \quad (i \leq j),$$

$$PT(x_i, C) = (x_i, x_{i+1}, \ldots, x_p, x_1, \ldots, x_{i-1}),$$

*Received by the editors February 6, 1992; accepted for publication (in revised form) April 28, 1992.

†Raymond and Beverly Sackler Faculty of Exact Sciences, School of Mathematical Sciences, Tel Aviv University, Ramat-Aviv 69978, Israel.

$$PT(C,x_i) = PT(x_{i+1},C), PT(x_i,x_j,C) = PT(x_i,x_j,PT(x_i,C)).$$

Let D be a digraph, and let x be a vertex of D; then

$$\Gamma^+(x) = \{y \in V(D) : (x,y) \in A(D)\}\ , \quad \Gamma^-(x) = \{z \in V(D) : (z,x) \in A(D)\}.$$

A digraph containing loops is called a general digraph.

2. Main results. At first, we consider a construction (due to N. Alon) that allows us to find a 1-diregular subgraph with maximum order of a given digraph D. We add to D a loop in each vertex, associate with any loop a weight equals 2, and with any other arc of D a weight equals 1. We obtain a weighted general digraph L. Let $B = B(D)$ be a bipartite undirected graph, such that (X, X') is the partition of B, where $X = V(L)$, $X' = \{x' : x \in X\}$, $xy' \in E(B)$, if and only if $(x,y) \in A(L)$ (including the case where $x = y$) and the weight of an edge xy' of B equals the weight of the arc (x,y). Obviously, a minimum weight 1-factor of B corresponds to a minimum weight 1-diregular spanning general subdigraph Q of L (i.e., a union of disjoint cycles and loops covering $V(L)$). It is easy to see that, removing all loops from Q, we obtain some 1-diregular subgraph F of D of maximum order. Since Q can be found by solving an assignment problem, we may find a 1-diregular subgraph of D of maximum order in time $O(n^3)$, (cf. [5]) where, here and below, $n = |V(D)|$. Now we are ready to consider the main algorithm.

Algorithm ALP

Input. A complete multipartite digraph D.

Output. A longest path H of D.

Step 1. Construct the digraph D' with

$$V(D') = \{x\} \cup V(D) \quad (x \notin V(D)),$$

$$A(D') = A(D) \cup \{(x,y),\ (y,x) : y \in V(D)\}.$$

Find a 1-diregular subgraph F' of D' of maximum order. Let $C_0, C_1, \ldots, C_t (t \geq 0)$ be the cycles of F', and suppose that $x \in V(C_0)$. (It is easy to see that $x \in F'$.) Find $P = C_0 - x$ and put

$$F := P \cup C_1 \cup \cdots \cup C_t.$$

Note that F is almost a 1-diregular subgraph of D of maximum order. We will construct a path on all the vertices of F—this will clearly be a longest path.

Step 2. If $t = 0$, then $H := P$, and we have finished. Otherwise, put $C := C_t$, $t := t - 1$. Let

$$P = (x_1, x_2, \ldots, x_m), \qquad C = (y_1, y_2, \ldots, y_k, y_1).$$

Step 3. If $\Gamma^-(x_1) \cap V(C) \neq \emptyset$, then pick any $x \in \Gamma^-(x_1) \cap V(C)$, put $P := (PT(C,x), P)$, and go back to Step 2. Analogously, if there exists $y \in \Gamma^+(x_m) \cap V(C)$, put $P := (P,\ PT(y,C))$ and go back to Step 2.

Step 4. For $i = 1, 2, \ldots, m-1$; $j = 1, 2, \ldots, k$, if $(y_j,\ x_{i+1})$, $(x_i,\ y_{j+1}) \in A(D)$, then

$$P := (PT(x_1,x_i,P),\ PT(y_{j+1},\ C),\ PT(x_{i+1},x_m,P)),$$

and go to Step 2.

If none of Steps 2–4 can be applied, we go to Step 5 below.

Step 5. For $j = 1, 2, \ldots, k$; $i = 1, 2, \ldots, m-1$, if i is minimal, such that there exists $j = j(i)$ for which

$$(y_j,\ x_{i+1}),\ (y_{j+1}, x_i) \in A(D), \tag{1}$$

then let P be a directed path containing

$$PT(x_1,\ x_{i-1}, P)\ ,\quad y_{j+1},\ x_i\ ,\quad PT(y_{j+2},\ y_j,\ C),\ PT(x_{i+1},\ x_m,\ P) \tag{2}$$

and three additional arcs and go to Step 2. (We prove below that such a directed path indeed exists.)

LEMMA 2.1. *Algorithm* ALP *finds a longest path in a* CMD D *in time* $O(n^3)$.

Proof. We claim that during Algorithm ALP P is always a path in D. It is obvious that this is the case after each execution of Steps 1, 2, 3, or 4 (provided that this was the case before starting such a step). Hence, we consider only Step 5. When Algorithm ALP executes Step 5, none of the conditions of Steps 3 and 4 hold for the current P and C. Hence,

$$\Gamma^-(x_1) \cap V(C) = \Gamma^+(x_m) \cap V(C) = \emptyset, \tag{3}$$

and there are no indices $i \in \{1, 2, \ldots, m-1\}, j \in \{1, 2, \ldots, k\}$ such that both $(y_j,\ x_{i+1})$ and $(x_i,\ y_{j+1})$ belong to D, i.e.,

$$\{(y_j,\ x_{i+1})\ ,\ (x_i,\ y_{j+1})\} \not\subseteq A(D). \tag{4}$$

We must prove that, if Algorithm ALP is at Step 5, then there exist arcs satisfying (1), and in this case there exists the path (2).

At first, assume that there are no arcs satisfying (1). By (3) $(y_s, x_m) \in A(D)$ for some s. Then x_{m-1} and y_{s+1} are nonadjacent. Indeed, by (4) $(x_{m-1},\ y_{s+1}) \notin A(D)$, and by the assumption $(y_{s+1},\ x_{m-1}) \notin A(D)$. Since y_{s+1} is not adjacent with x_{m-1}, it is adjacent with x_m. Therefore, $(y_{s+1},\ x_m) \in A(D)$. Hence, x_{m-1} and y_{s+2} are nonadjacent, and x_{m-1} is not adjacent with any of y_{s+1} and y_{s+2} . Since y_{s+1} and y_{s+2} are adjacent (and hence do not belong to the same part), this is a contradiction. We conclude that there exist arcs satisfying (1). Let i be the minimum possible index in (1), and put $j = j(i)$.

Now we prove that D has the path (2). By (3), $i > 1$. By the minimality of i and by (4), the vertices $x_{i-1},\ y_{j+2}$ are nonadjacent. If $(y_{j+2},\ x_i) \in A(D)$, then, again, by the minimality of i (and by (4)) the vertices $x_{i-1},\ y_{j+3}$ are nonadjacent, but this is impossible. Hence,

$$(x_i,\ y_{j+2}) \in A(D). \tag{5}$$

If $i = 2$, we have (by (3)) that $(x_{i-1},\ y_{j+1}) \in A(D)$. If $i > 2$ and $(y_{j+1},\ x_{i-1}) \in A(D)$, it follows that $x_{i-2},\ y_{j+2}$ are nonadjacent; that is impossible because x_{i-1} and y_{j+2} are nonadjacent. Hence,

$$(x_{i-1},\ y_{j+1}) \in A(D), \tag{6}$$

in any case. Therefore, using the arcs from (5), (6), we may form path (2).

Note that the number of operations we need for executing Steps 3–5 is $O(|V(P)| \cdot |V(C)|)$ for the current pair P, C. Hence, the total number of operations at Steps 2–5 is

$$O(|V(P)| \cdot |V(C_1)|) + \sum_{j=1}^{t-1} (|V(P)| + \cdots + |V(C_j)|)(|V(C_{j+1})|) = O(n^2).$$

At last, note that the execution of Step 1 takes time $O(n^3)$. □

Algorithm ALP and the proof of Lemma 1 imply immediately the following result.

THEOREM. *Let D be a CMD. Then, for any almost 1-diregular subgraph F of D, there is a path P of D satisfying $V(P) = V(F)$. If F is a maximum 1-diregular subgraphs, each such path is a longest path of D. There exists an algorithm for finding a longest path in D in time $O(n^3)$.*

3. Modifications of the main results. Using any maximum matching algorithm (see [5] and [6]), we can test whether a digraph contains a 1-diregular spanning subgraph F' and find some F' in time $O(n^{2.5})$. Note that $F = F' - x(x \in V(F))$ is an almost 1-diregular spanning subgraph. Hence, after a trivial modification of Step 1 in Algorithm ALP, we obtain an $O(n^{2.5})$ algorithm allowing to test whether a CMD D has a Hamiltonian path (and to construct one of them in the case that it exists). This implies the following corollary.

COROLLARY 1. *A CMD D has a Hamiltonian path if and only if it has an almost 1-diregular spanning subgraph. Testing whether D has a Hamiltonian path (and finding one of them) requires, at most, time $O(n^{2.5})$.*

Let D be a CBD. Then we can remove Step 5 from the algorithm, since the algorithm does not use Step 5 in this case. To prove this, we must show that the algorithm never goes to Step 5 (from Step 4); i.e., it always constructs a new path P in Step 3 or Step 4. If the algorithm reaches Step 4 after executing Step 3 for the current P and C, then (3) holds. Therefore, there exists $i \in \{1, 2, \ldots, m-1\}$ such that $\Gamma^-(x_i) \cap V(C) = \emptyset$, but $\Gamma^- (x_{i+1}) \cap V(C) \neq \emptyset$; $(y_j,\ x_{i+1}) \in A(D)$. Since D is bipartite, the vertices $x_i,\ y_{j+1}$ are adjacent. Hence $(x_i,\ y_{j+1}) \in A(D)$, and the algorithm can construct a new path P.

Acknowledgment. This paper forms part of a Ph.D. thesis written by the author under the supervision of Professor N. Alon. I would like to thank Professor Alon for fruitful comments and suggestions.

REFERENCES

[1] G. GUTIN, *Effective characterization of complete bipartite digraphs that have a Hamiltonian path*, Kibernetica (Kiev), 4 (1985) pp. 124–125. (In Russian.)

[2] ———, *A characterization of complete n-partite digraphs that have a Hamiltonian path*, Kibernetica (Kiev), 1 (1988), pp. 107–108. (In Russian.)

[3] R. HÄGGKVIST AND Y. MANOUSSAKIS, *Cycles and paths in bipartite tournaments with spanning configurations*, Combinatorica, 9 (1989), pp. 33–38.

[4] Y. MANOUSSAKIS AND Z. TUZA, *Polynomial algorithms for finding cycles and paths in bipartite tournaments*, SIAM J. Discrete Math., 3 (1990), pp. 537–543.

[5] H. PAPADIMITRIOU AND K. STEIGLITZ, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[6] J. HOPCROFT AND R. KARP, *A $n^{5/2}$ algorithm for maximum matching in bipartite graphs*, SIAM J. Comput., 2 (1973), pp. 225–231.

SIAM J. DISC. MATH.
Vol. 6, No. 2, pp. 274–283, May 1993

AN EXACT CHARACTERIZATION OF GREEDY STRUCTURES*

PAUL HELMAN†, BERNARD M. E. MORET†, AND HENRY D. SHAPIRO†

Abstract. The authors present exact characterizations of structures on which the greedy algorithm produces optimal solutions. Our characterization, which are called matroid embeddings, complete the partial characterizations of Rado [*A note on independent functions*, Proc. London Math. Soc., 7 (1957), pp. 300–320], Gale [*Optimal assignments in an ordered set*, J. Combin. Theory, 4 (1968), pp. 176–180], and Edmonds [*Matroids and the greedy algorithm*, Math. Programming, 1 (1971), pp. 127–136], (matroids), and of Korte and Lovasz [*Greedoids and linear object functions*, SIAM J. Alg. Discrete Meth., 5 (1984), pp. 239–248] and [*Mathematical structures underlying greedy algorithms*, in Fundamentals of Computational Theory, LNCS 177, Springer-Verlag, 1981, pp. 205–209] (greedoids). It is shown that the greedy algorithm optimizes all linear objective functions if and only if the problem structure (phrased in terms of either accessible set systems or hereditary languages) is a matroid embedding. An exact characterization of the objective functions optimized by the greedy algorithm on matroid embeddings is also presented. Finally, the authors present an exact characterization of the structures on which the greedy algorithm optimizes all bottleneck functions, structures that are less constrained than matroid embeddings.

Key words. algorithmic paradigm, bottleneck objective function, greedy algorithm, greedoid, linear objective function, matroid, matroid embedding, set system

AMS(MOS) subject classifications. 68Q20, 05B35, 90C27, 68R05

1. Introduction. Obtaining an exact characterization of the class of problems for which the greedy algorithm returns an optimal solution has been an open problem. Rado [9], Gale [3], and Edmonds [1] have independently shown that matroids characterize a subclass of problems on which the greedy algorithm always optimizes linear objectives; their results are limited by the assumption that the greedy algorithm operates on a hereditary set system, whereas most common greedy algorithms operate on set systems that do not obey the heredity axiom. Faigle [2] has provided an exact characterization of the partially ordered set systems on which the greedy algorithm optimizes linear objectives, but the assumption of a partial order, which constrains the choices of the greedy algorithm, limits the characterization. Korte and Lovasz [6], [7] have defined greedoids, a generalization of matroids, and have provided necessary and sufficient conditions for the greedy algorithm to be optimal with respect to linear objectives when run on greedoids. However, greedoids are both too general (the greedy algorithm need not return an optimal solution on a greedoid) and too constraining: there exist set systems on which the greedy algorithm always optimizes linear objectives, but that are not greedoids. Goecke [4] has given necessary and sufficient conditions for the optimization of linear objectives over set systems by a variant of the greedy algorithm, but his variant of the greedy algorithm (find any solution, partial or complete, which optimizes the objective) does not fit well in many standard applications of the greedy algorithm, in particular, applications where the objective function is to be minimized.

We solve the open problem by presenting the following three exact characterizations, all based on a very general model of the problem structure:

1. An exact characterization, which we call a *matroid embedding*, of the structure of problems on which the greedy algorithm optimizes all linear objectives;
2. A similar characterization for bottleneck objectives; and

*Received by the editors September 9, 1990; accepted for publication (in revised form) April 9, 1992. A preliminary version of this paper appears in the Proceedings of the Second Conference on Integer Programming and Combinatorial Optimization, Carnegie-Mellon University, May 25–27, 1992.

†Department of Computer Science, University of New Mexico, Albuquerque, New Mexico 87131-1386 (moret@unmvax.cs.unm.edu).

3. An exact characterization of the objective functions optimized by the greedy algorithm on matroid embeddings.

Our presentation is in four parts. First, we set the stage by recalling briefly the definitions and main existing results pertaining to set systems and the greedy algorithm. Second, we introduce additional properties, relate them to the existing structures, and prove our main result. In a third part, we extend these results to a family of objective functions and then examine one particular class, the bottleneck functions, and give an exact characterization of the problems on which the greedy algorithm optimizes these objectives. The fourth part extends our results from set systems to languages. We conclude with some general observations and a number of open questions.

2. Preliminaries. We include this section for readers unfamiliar with the terminology; other readers may wish to skip to the next section.

Let S be a set and $\mathcal{C}$ a collection of subsets of S; the pair $(S, \mathcal{C})$ is called a *set system*. To simplify the notation, we let $\operatorname{ext}(X) = \{\, x \mid X \cup \{x\} \in \mathcal{C}\,\}$. A set system is an *accessible set system* if it obeys the following two axioms:

(*trivial axiom*) $\emptyset \in \mathcal{C}$,

(*accessibility axiom*) If $X \in \mathcal{C}$ and $X \neq \emptyset$, then $\exists x \in X$ such that $X - \{x\} \in \mathcal{C}$.

In an accessible set system $(S, \mathcal{C})$, the elements of $\mathcal{C}$ are called *feasible sets*; a maximal feasible set (i.e., one that is not contained in any other) is called a *basis*. A set system is a *hereditary set system* (also known as a *simplicial complex* or an *independence structure*) if it obeys the trivial axiom and

(*heredity axiom*) If $X \in \mathcal{C}$ and $Y \subseteq X$, then $Y \in \mathcal{C}$.

Given an arbitrary, but nonempty set system $(S, \mathcal{C})$, we define its *hereditary closure* as the set system $(S, \mathcal{C}^*)$, where $\mathcal{C}^* = \{Y \subseteq X \mid X \in \mathcal{C}\}$.

Let $(S, \mathcal{C})$ be a set system. An *objective function* is an assignment of values to the subsets of S, $f : 2^S \to \mathbb{R}$. We define the *optimization problem* for f over $(S, \mathcal{C})$ as the problem of finding a basis B such that $f(B) = \max\{\, f(X) \mid X \text{ is a basis of } (S, \mathcal{C})\,\}$. (Note that only bases are candidates for solution. Further note that we restrict our discussion to maximization problems mutatis mutandis, identical results hold for minimization problem.) Given a weight assignment to the elements of S, $w : S \to \mathbb{R}$, the induced *linear objective function* is defined by $f(X) = \sum_{x \in X} w(x)$, for $X \subseteq S$, and the induced *bottleneck objective function* is defined by $f(X) = \min_{x \in X} w(x)$, for $X \subseteq S$.

Informally, the greedy algorithm, when run on a set system, builds a solution by beginning with the empty set and successively adding the best remaining element while maintaining feasibility. (Korte and Lovasz [7] have considered a variant known as the *worst-out greedy algorithm*; we do not pursue it further here.) The accessibility of a set system allows any feasible set, and in particular any basis, to be built one element at a time from the empty set—a necessary condition for the greedy algorithm to succeed. Formally, we define the *best-in greedy algorithm* on an accessible set system $(S, \mathcal{C})$, with objective function $f : 2^S \to \mathbb{R}$, as follows. The algorithm starts with the empty set; at each step, i, it chooses an element $x_i \in S$ such that

1. $\{x_1, x_2, \ldots, x_i\} \in \mathcal{C}$; and
2. $f(\{x_1, \ldots, x_i\}) = \max\{f(\{x_1, \ldots, x_{i-1}, y\}) \mid \{x_1, \ldots, x_{i-1}, y\} \in \mathcal{C}\}$.

The algorithm terminates when it can no longer incorporate another element into its partial solution, i.e., when $\operatorname{ext}\{x_1, \ldots, x_{i-1}\} = \emptyset$.

DEFINITION 2.1. A feasible set X is a *greedy set* under f if there exists a sequence $\emptyset, \{x_1\}, \{x_1, x_2\}, \ldots, \{x_1, \ldots, x_i\}, \ldots, \{x_1, \ldots, x_i, \ldots, x_k\} = X$ of feasible subsets of X such that, for each i, $f(\{x_1, \ldots, x_{i-1}, x_i\}) = \max\{f(\{x_1, \ldots, x_{i-1}, y\}) \mid \{x_1, \ldots, x_{i-1}, y\}$ is feasible$\}$. A basis with this property is called a *greedy basis.*

The greedy algorithm, for any objective function f, can construct only greedy sets under f; using the proper tie-breaking rule, it can construct any greedy set under f. We say that an accessible set system $(S, \mathcal{C})$ is *pathological* if there exist feasible sets A and B, with $A \subset B$, such that B is a basis and ext$(A) = \emptyset$. Due to the presence of pathologies, the greedy algorithm can terminate at a set that is not a basis. We could, as a result, redefine the optimization problem as the problem of finding a nonextensible set of maximal value. Our results hold under this interpretation as well.

A *greedoid* is an accessible set system $(S, \mathcal{C})$ that obeys the following axiom:

(*augmentation axiom*) If $X, Y \in \mathcal{C}$ and $|X| = |Y| + 1$, then $\exists x \in X - Y$ such that $Y \cup \{x\} \in \mathcal{C}$.

A *matroid* is a hereditary set system that obeys the augmentation axiom. (Note that the bases of a greedoid or matroid have equal cardinality and that pathologies cannot occur.) This axiom is often phrased more generally, as follows:

(*exchange axiom*) If $X, Y \in \mathcal{C}$ and $|Y| < |X|$, then $\exists x \in X - Y$ such that $Y \cup \{x\} \in \mathcal{C}$.

In the presence of the trivial axiom, the exchange axiom is equivalent to the combination of the accessibility and augmentation axioms.

Rado [9], Gale [3], and Edmonds [1] have independently proved that the best-in greedy algorithm optimizes all linear objective functions over a hereditary set system $(S, \mathcal{C})$ if and only if $(S, \mathcal{C})$ is a matroid. Korte and Lovasz [6], [7] have defined greedoids and proved that the best-in greedy algorithm optimizes all linear objective functions over a greedoid $(S, \mathcal{C})$ if and only if $(S, \mathcal{C})$ obeys the following axiom:

(*strong exchange axiom*) Let $A, B \in \mathcal{C}$, with B a basis and $A \subset B$. If $x \in S - B$ is such that $A \cup \{x\} \in \mathcal{C}$, then $\exists y \in B - A$ such that $A \cup \{y\} \in \mathcal{C}$ and $B \cup \{x\} - \{y\} \in \mathcal{C}$.

3. An exact characterization. We propose two new axioms to establish an exact characterization. The first is a strengthened version of accessibility for bases. An accessible set system $(S, \mathcal{C})$ is *extensible* if it obeys the following axiom:

(*extensibility axiom*) If X and B are feasible sets, with B a basis and $X \subset B$, then there exists $y \in B - X$ such that $X \cup \{y\}$ is feasible.

Note that every greedoid is extensible. An accessible set system $(S, \mathcal{C})$ is *closure congruent* if it obeys the following axiom:

(*closure congruence axiom*) $\forall X \in \mathcal{C}, \forall x, y \in \text{ext}(X), \forall E \subseteq S - X - \text{ext}(X), X \cup \{x\} \cup E \in \mathcal{C}^* \implies X \cup \{y\} \cup E \in \mathcal{C}^*$.

Even in the setting of extensible accessible set systems, closure congruence neither implies, nor is implied by, augmentation. Indeed, we observe that every hereditary set system, but not every greedoid, is closure congruent (because, in a hereditary set system, the empty set is the only choice for E in the definition of closure congruence). Greedoids that obey the strong exchange axiom are closure congruent (a corollary of Thm. 3.1), but there exist extensible, closure congruent accessible set systems that do not obey the augmentation axiom and hence do not define greedoids.

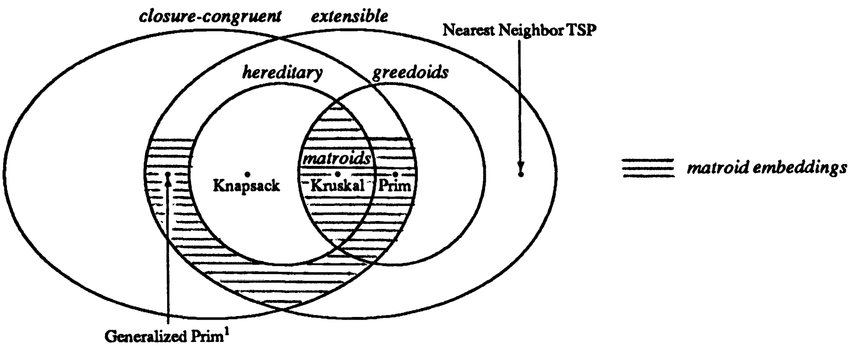


FIG. 1. *The relationships among varieties of accessible set systems.*

DEFINITION 3.1. A *matroid embedding* is an accessible set system, which is extensible, closure congruent, and the hereditary closure of which is a matroid.

Note that there exist matroid embeddings that are not greedoids; indeed, the three conditions defining a matroid embedding are independent. Figure 1 shows the relationships among our axioms and previously defined structures.

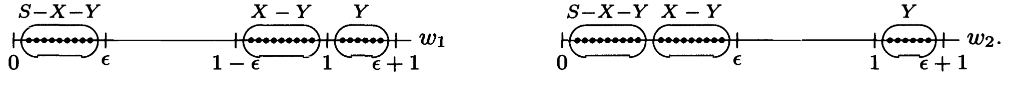
We can now prove our main result, which solves the open problem.

THEOREM 3.1. *Let $(S, \mathcal{C})$ be an accessible set system; then the following are equivalent:*

1. *For every positively weighted linear objective function, $(S, \mathcal{C})$ has an optimal greedy basis,*
2. *$(S, \mathcal{C})$ is a matroid embedding,*
3. *For every linear objective function, the greedy bases of $(S, \mathcal{C})$ are exactly its optimal bases.*

Proof. We prove the implications (1)⇒(2) and (2)⇒(3); the implication (3)⇒(1) is trivial.

(1)⇒(2) We begin by showing that $\mathcal{C}^*$ must be a matroid, a result first derived by Helman [5]. Assume two sets, X, $Y \in \mathcal{C}^*$, with $|X| = |Y| + 1$, between which augmentation fails in $\mathcal{C}^*$. Since augmentation fails, no basis that contains all of Y can contain any element of $X - Y$. We design a pair of weight assignments, w_1 and w_2, such that (i) the relative ordering of elements by weight is the same under both w_1 and w_2 and distinct elements get assigned distinct weights in each weight assignment—so that w_1 and w_2 share the same unique nonextensible greedy set; and (ii) w_1 and w_2 share no optimal basis, thereby contradicting (1) and proving the result. We choose the two weight assignments as illustrated below:



Observe that, under w_1, an optimal basis cannot contain all of Y, while, under w_2, an optimal basis must contain all of Y.

We now prove that $(S, \mathcal{C})$ is extensible. Let A be a feasible set and B be a basis, with $A \subset B$. Since $(S, \mathcal{C})$ is accessible, there exists a sequence of feasible sets $\emptyset, \{x_1\}, \{x_1, x_2\}, \ldots, \{x_1, x_2, \ldots, x_k\} = A$; denote the other elements of S by x_{k+1},

[1] The set system denoted as "generalized Prim's" resembles that for Prim's algorithm, in that both have subtrees of the ground graph as feasible sets; however, in Prim's algorithm, all such subtrees include the same designated vertex, whereas in the generalized version any subtree is feasible.

$x_{k+2}, \ldots, x_n$. We force the greedy algorithm to construct each set in the sequence leading to A by assigning weights as follows:

$$w(x_i) = \begin{cases} 1 + \epsilon/i & \text{for } 1 \leq i \leq k, \\ 1 & \text{for } x_i \in B - A, \\ \epsilon & \text{for } x_i \in S - B. \end{cases}$$

Thus, B is the unique optimal basis; since the greedy algorithm must start by constructing A, it can construct B only if A is extensible to B, as desired.

Finally, we show that $(S, \mathcal{C})$ is closure congruent. Let $A \in \mathcal{C}$, $x, y \in \text{ext}(A)$, and $E \subseteq S - A - \text{ext}(A)$, with $A \cup \{x\} \cup E \in \mathcal{C}^*$. In the style of the previous construction, we force the greedy algorithm to construct A followed by the set $A \cup \{y\}$ by using a weight assignment that gives very high weights to elements of A and E, high weights to x and y, and very low weights to all other elements. Since we have $A \cup \{x\} \cup E \in \mathcal{C}^*$, there is a basis B containing $A \cup \{x\} \cup E$. The greedy algorithm must begin by constructing $A \cup \{y\}$ and constructs some optimal basis B'; but then we must have $E \subseteq B'$ and thus $A \cup \{y\} \cup E \in \mathcal{C}^*$.

(2)⇒(3) (Note that the structure of a matroid embedding ensures that all nonextensible sets are bases.) Assume that, for some linear objective function f, some greedy basis B_g is not optimal. Let A be a greedy subset of B_g of maximal size, with the property that A is contained in some optimal basis B. Since A itself cannot be a basis, ext(A) is not empty. Let x be an element that the greedy algorithm can add to A. We have $A \cup \{x\} \not\subseteq B$, or else $A \cup \{x\}$ would be a greedy subset of B_g contained in B, contradicting the maximality of A. By extensibility, there exists $y \in B \cap \text{ext}(A)$; set $E = B - A - \text{ext}(A)$. Observe that $A \cup \{y\} \cup E$ is in $\mathcal{C}^*$, so that, by closure congruence, so is $A \cup \{x\} \cup E$. Because $(S, \mathcal{C}^*)$ is a matroid, we can apply the exchange axiom to $A \cup \{x\} \cup E$ with respect to B, yielding some basis B'. B and B' differ by one element: B' contains x at the expense of some other element in ext(A), say z. Since the greedy algorithm chose to augment A with x, we know that $w(x) \geq w(z)$, so that $f(B') \geq f(B)$ and thus B' is an optimal basis. Then, however, $A \cup \{x\} \subset B'$ is a greedy subset of B_g, which contradicts the maximality of A. A similar argument also shows that every optimal basis is greedy: if some nongreedy optimal basis B exists, let A be its largest greedy subset; note that we must have $|B| \geq |A| + 2$, since otherwise A can be extended to B and that extension must be greedy because B is optimal. However, then B', produced as above, has a larger objective value than B because, since $A \cup \{z\} \subset B$ is not greedy, we must have $w(x) > w(z)$; hence B is not optimal, yielding the desired contradiction. □

This theorem subsumes the results of Rado [9], Gale [3], and Edmonds [1], as well as Theorem 4.2 of Korte and Lovasz [7]; more importantly, unlike these results, it provides an exact structural characterization of the problems on which the best-in greedy algorithm works for all linear objectives.

4. Other classes of objective functions.

4.1. Consistent functions. We identify the largest class of functions to which the results of the previous section apply.

DEFINITION 4.1. An objective function $f(\ )$ is *consistent* if, given sets $T \subset T' \subset S$ and elements $x, y \in S - T'$,

$$f(T \cup \{x\}) \geq f(T \cup \{y\}) \Longrightarrow f(T' \cup \{x\}) \geq f(T' \cup \{y\});$$

$f(\ )$ is *strictly consistent* if we can further assert that

$$f(T \cup \{x\}) > f(T \cup \{y\}) \Longrightarrow f(T' \cup \{x\}) > f(T' \cup \{y\});$$

finally, $f(\ )$ is *weakly consistent* if we strengthen the hypothesis used in consistency to exclude equality, i.e., if

$$f(T \cup \{x\}) > f(T \cup \{y\}) \Longrightarrow f(T' \cup \{x\}) \geq f(T' \cup \{y\}).$$

Note that linear objectives are strictly consistent, while bottleneck objectives are consistent.

THEOREM 4.1. *Let S be a set and $f(\ )$ a function defined on 2^S.*

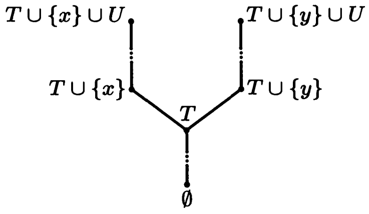
(1) *$f(\ )$ is strictly consistent if and only if, for each matroid embedding on S, the greedy bases are exactly the optimal bases.*

(2) *$f(\ )$ is consistent if and only if, for each matroid embedding on S, all greedy bases are optimal.*

(3) *$f(\ )$ is weakly consistent if and only if, for each matroid embedding on S, some greedy basis is optimal.*

Proof. 1. To show the only if part, it is enough to observe that, in the proof of Theorem 3.1, the inequalities involving $w(x)$ and $w(z)$ can be replaced by inequalities involving $f(A \cup \{x\})$ and $f(A \cup \{z\})$. Letting $T' = B \cap B'$, the strict consistency of f leads to the same contradictions as in the previous proof.

To prove the if part, assume that $f(\ )$ fails consistency (strict or not) on sets T, $T' = T \cup U$ and elements $x, y \notin T$, with $U \cap (T \cup \{x, y\}) = \emptyset$. Consider the matroid embedding depicted below.



If $f(\ )$ is not consistent, then we have $f(T \cup \{x\}) \geq f(T \cup \{y\})$ and yet also $f(T \cup \{x\} \cup U) < f(T \cup \{y\} \cup U)$. However, then $T \cup \{x\} \cup U$ is a suboptimal greedy basis, the desired contradiction. If $f(\ )$ is not strictly consistent, then we have $f(T \cup \{x\}) > f(T \cup \{y\})$ and yet also $f(T \cup \{x\} \cup U) \leq f(T \cup \{y\} \cup U)$. However, then $T \cup \{y\} \cup U$ is an optimal basis and yet is not greedy, the desired contradiction.

The same proof techniques apply in 2 and 3, with the obvious changes. □

4.2. An exact characterization of greedy structures for bottleneck functions. Bottlenecks functions form an important subclass of consistent functions. Formally, we define a (simple) *bottleneck function* to be an objective function of the form $f(A) = \min_{x \in A} w(x)$; by convention, we set $f(\emptyset) = 1 + \max_{x \in S} w(x)$. Our previous results show that the greedy algorithm is optimal for all bottleneck objective functions when run on a matroid embedding. Korte and Lovasz [6] considered a generalization of bottleneck objectives in which the weight of an element is a nondecreasing function of the size of the feasible set in which it could be included; they showed that the greedy algorithm is optimal for all such generalized bottleneck objectives only if the set system defines a greedoid. Since not every matroid embedding is a greedoid, this result does not hold when restricted to simple bottleneck objectives.

The matroid embedding structure is not necessary to ensure optimality of the greedy algorithm for accessible set systems with bottleneck objectives. In particular, it is easily verified that optimality for all bottleneck objectives on an accessible set system does not imply that the hereditary closure of the set system is a matroid. We introduce a more restricted property. An accessible set system $(S, \mathcal{C})$ is *strongly extensible* if it obeys the following axiom:

(*strong extensibility axiom*) For any $X, B \in \mathcal{C}$, with B a basis and $|X| < |B|$, there exists $x \in B - X$ such that $X \cup \{x\} \in \mathcal{C}$.

The bases of any strongly extensible accessible set system are of the same cardinality; in fact, a hereditary set system is strongly extensible if and only if it is a matroid.

LEMMA 4.2. *Let $(S, \mathcal{C})$ be an accessible set system. If, for every positive weighted bottleneck function f, there exists a greedy basis that is also optimal, then $(S, \mathcal{C})$ is extensible and all of its bases have equal cardinality.*

Proof. Assume that there exists a basis B and a feasible set $A \subset B$ such that either (i) A is nonextensible; or (ii) for all $x \in \text{ext}(A)$, $A \cup \{x\} \not\subseteq B$. We force the greedy algorithm to construct A by assigning suitable weights, also giving elements of $S - B$ very low weights. Now, however, the greedy algorithm must terminate with a suboptimal feasible set. If A is nonextensible, then it is the unique nonextensible greedy set, so that there does not exist an optimal greedy basis, contrary to the hypothesis of the lemma. Otherwise, because A cannot be extended with an element of B, any greedy basis contains an element of $S - B$ and thus is not optimal.

Assume that at least two sizes of bases exist; let C be an arbitrary nonminimal-size basis and let B be a minimal-size basis, such that among all minimal-size bases, B shares with C a largest size feasible subset A. Note that A is a proper subset of B and cannot itself be a basis. Since $(S, \mathcal{C})$ is extensible and since $A \subset C$, there exists some $y \in \text{ext}(A)$ such that $A \cup \{y\} \subset C$; note that, by our assumption of maximality of A, $y \notin B$. We force the greedy algorithm to construct the set $A \cup \{y\}$ and then to complete it in suboptimal manner by assigning suitable weights, including very low weights for elements of $S - B$. Under such an assignment, basis B is optimal. The greedy bases constructed must all contain $A \cup \{y\}$ and, by our assumption of the maximality of A, have size greater than $|B|$. Thus, at least $|B - A|$ elements must be added to $A \cup \{y\}$; since B is a basis, these elements cannot all come from B. Thus, all greedy bases are suboptimal, the desired contradiction. □

THEOREM 4.3. *Let $(S, \mathcal{C})$ be an accessible set system; then the following are equivalent:*

(1) *For every positive weighted bottleneck function, $(S, \mathcal{C})$ has an optimal greedy basis,*

(2) *$(S, \mathcal{C})$ is strongly extensible,*

(3) *For every bottleneck function, $(S, \mathcal{C})$ has at least one greedy basis and all its greedy bases are optimal.*

Proof. As before, only two of the three implications are nontrivial.

(1)⇒(2) Assume that there exist A, $B \in \mathcal{C}$, where B is a basis, with $|A| < |B|$ and $\text{ext}(A) \cap B = \emptyset$. By the previous lemma, A must be extensible. We force the greedy algorithm to construct A and to extend it to a suboptimal basis by assigning suitable weights, including very low weights to elements of $S - B - A$. Since A cannot be extended by an element of B, every greedy basis has very low value, while B is an optimal basis with higher value, the desired contradiction.

(2)⇒(3) (Note that any nonextensible set must be a basis, by strong extensibility.) For some assignment of weights, assume that some greedy basis B is suboptimal. Let A be a greedy subset of B of maximal size that (i) the greedy algorithm can extend to the greedy basis B; and (ii) is a subset of some optimal basis B'. Let $A \cup \{x\}$ be a greedy set extensible to B. Since the set system is strongly extensible, there also exists $y \in B' - A$ such that $A \cup \{y\}$ is feasible; note that $A \cup \{y\}$ cannot be a greedy set extensible to B, as this would contradict the maximality of A. Since $A \cup \{y\} \subseteq B'$, we have $f(A \cup \{y\}) \geq f(B')$; since the set system is strongly extensible (and thus allows $A \cup \{x\}$ to be extended with elements from some optimal basis), since A is maximal, and since the objective function is determined by the minimum weight of its arguments, we also have $f(B') > f(A \cup \{x\})$. Combining these two inequalities yields $f(A \cup \{y\}) > f(A \cup \{x\})$, which contradicts the fact that the greedy algorithm can choose x. □

5. Exact characterizations of greedy languages. While set systems have been the traditional setting for defining and studying greedy algorithms, several researchers have recognized the desirability of extending the results to more general settings (Helman [5], Korte and Lovasz [6]). In this section, we demonstrate that our exact characterizations extend directly to hereditary languages.

In the language world, feasible structures are ordered sets, or strings, generally called *simple words*. Let S be a set and $\mathcal{L}$ a collection of simple words on S; furthermore, let $s(\alpha)$ for each $\alpha \in \mathcal{L}$ denote the (unordered) subset of S corresponding to α and let $s(\mathcal{L})$ denote the collection of unordered subsets corresponding to the words of $\mathcal{L}$—i.e., $s(\mathcal{L}) = \{A \mid A = s(\alpha),\ \alpha \in \mathcal{L}\}$. We say that $(S, \mathcal{L})$ is a *hereditary word system* (or *hereditary language*) if it obeys the following two axioms:

(*trivial axiom*) $\mathcal{L} \neq \emptyset$,

(*heredity axiom*) If $\alpha \in \mathcal{L}$ and β is a prefix of α (i.e., $\alpha = \beta\gamma$ for some string γ), then $\beta \in \mathcal{L}$.

If $(S, \mathcal{L})$ is a hereditary language, we call the elements of $\mathcal{L}$ *feasible words*; any feasible word α with the property that there does not exist $x \in S$ with $\alpha x \in \mathcal{L}$ is called a *basic word*. For any word α, let $\text{ext}(\alpha)$ denote the set $\{x \mid \alpha x \in \mathcal{L}\}$. Note that, if $(S, \mathcal{L})$ is a hereditary language, then $(S, s(\mathcal{L}))$ is an accessible set system. We define the hereditary closure of the hereditary language $(S, \mathcal{L})$ to be the hereditary closure $(S, s^*(\mathcal{L}))$ of the corresponding accessible set system $(S, s(\mathcal{L}))$.

There is a very natural link between hereditary languages and the greedy algorithm, as hereditary languages record the full history of the execution of the algorithm. Formally, the best-in greedy algorithm on a hereditary language $(S, \mathcal{L})$ with objective function $f : \mathcal{L} \rightarrow \mathbb{R}$ starts with the empty string λ; at each step i, it chooses an element $x_i \in S$ such that

1. $x_1 x_2 \ldots x_i \in \mathcal{L}$; and

2. $f\big(x_1 x_2 \ldots x_i\big) = \max\big\{f\big(x_1 \ldots x_{i-1}\, y\big) \bigm| x_1 \ldots x_{i-1} y \in \mathcal{L}\big\}$; the algorithm terminates when it has constructed a basic word. A feasible word $x_1 x_2 \ldots x_k$ is a *greedy word* under f if, for each $1 \leq i \leq k$, $f(x_1 \ldots x_{i-1} x_i) = \max\{\, f(x_1 \ldots x_{i-1} y) \mid x_1 \ldots x_{i-1} y \in \mathcal{L}\,\}$. Given an objective function on S, an objective function f on words is, respectively, a linear, bottleneck, or consistent function if there is a linear, bottleneck, or consistent function g on sets such that $f(\alpha) = g(s(\alpha))$ for all words α. This implies that if β is a permutation of α, then $f(\beta) = f(\alpha)$, a property often called *stability*.

The necessary and sufficient conditions for hereditary languages are (essentially) the obvious language versions of the accessible set system conditions. Consider the following language version of each of our axioms. A hereditary language $(S, \mathcal{L})$ is *extensible* if it obeys the following axiom:

(*extensibility axiom*) If $\alpha, \beta \in \mathcal{L}$, β is a basic word and $s(\alpha) \subset s(\beta)$, then $\exists x \in s(\beta) - s(\alpha)$ such that $\alpha x \in \mathcal{L}$.

A hereditary language $(S, \mathcal{L})$ is *closure congruent* if it obeys the following axiom:

(*closure-congruence axiom*) $\forall \alpha \in \mathcal{L}, \forall x, y \in \text{ext}(\alpha), \forall E \subseteq S - s(\alpha) - \text{ext}(\alpha)$, $s(\alpha) \cup \{x\} \cup E \in s^*(\mathcal{L}) \Longrightarrow s(\alpha) \cup \{y\} \cup E \in s^*(\mathcal{L})$.

A hereditary language, $(S, \mathcal{L})$, is *strongly extensible* if it obeys the following axiom:

(*strong extensibility axiom*) If $\alpha, \beta \in \mathcal{L}$, β is a basic word, and $|\alpha| < |\beta|$, then $\exists x \in s(\beta) - s(\alpha)$ such that $\alpha x \in \mathcal{L}$.

DEFINITION 5.1. A (language) *matroid embedding* is a hereditary language that is extensible, closure congruent, and the hereditary closure of which is a matroid.

The situation regarding pathologies is more complex for hereditary languages than for accessible set systems; we say that a hereditary language is*pathological* if there exists a pair of basic words β_1 and β_2 such that $s(\beta_1) \subset s(\beta_2)$. In the language world, pathologies appear natural if α and β form a pathology, then this simply means that α cannot be a prefix of β.

In spite of these differences, all of our theorems hold in their obvious rephrasing.

THEOREM 5.1. *Let $(S, \mathcal{L})$ be a hereditary language; then the following are equivalent:*

(1) *For every positively weighted linear objective function f, there exists an optimal greedy basic word,*

(2) *$(S, \mathcal{L})$ is a matroid embedding,*

(3) *For every linear objective function, the greedy basic words are exactly the optimal basic words.*

The generalization to the language world is not trivial, in the sense that there exist distinct language-based matroid embeddings corresponding to the same set-based matroid embedding; i.e., there exists a language-based matroid embedding, which contains two feasible words that are equal as sets but have different extension sets. (This result should be contrasted with a theorem of Korte and Lovasz [6] showing that greedoids do not give rise to such situations.)

The results presented so far for hereditary languages completely parallel those for accessible set systems. The same is almost true for the class of bottleneck functions. However, certain pathologies (that can occur in hereditary languages but not in set systems) allow the greedy algorithm to optimize all bottleneck functions on languages that fail to obey even the weaker of the extensibility axioms. By proving results paralleling Lemma 4.2 and Theorem 4.3, we can establish the following exact characterization of greedy optimality for bottleneck functions on hereditary languages.

THEOREM 5.2. *The best-in greedy algorithm run on a hereditary language optimizes all bottleneck functions if and only if the language is strongly extensible (except with respect to pairs of sets that form pathologies).*

6. Conclusion. We have presented exact characterizations of problem structures on which the greedy algorithm optimizes linear, bottleneck, and, more generally, consistent objective functions. These exact characterizations apply both to accessible set systems and hereditary languages and answer questions raised, and only partially answered, by Edmonds [1], Korte and Lovasz [6], [7], and others.

Our results provide a framework for future research: what are additional structural properties of matroid embeddings? How can constraints about the objective function be traded against constraints on the language structure? A consequence of our results is that the linear objectives are the hardest of all consistent objectives to optimize by greedy methods on matroid embeddings, in the sense that, if they are optimized, then so too is any other consistent objective. This suggests a study of families of objective functions along much the same lines as classical complexity theory; in this direction, Lengauer and Theune [8] have demonstrated reductions among cost functions for path problems. Now that we have a proper setting for optimal greedy algorithms, we can investigate the complexity of such algorithms. This problem is harder than it may seem, since much of the structure used is given implicitly by a feasibility oracle or some such theoretical construct: an efficient greedy algorithm results from both a fast feasibility check and a fast identification of the best extension.

Acknowledgment. We thank an anonymous referee for very helpful comments, which led to major improvements in our presentation.

REFERENCES

[1] J. Edmonds, *Matroids and the greedy algorithm*, Math. Programming, 1 (1971), pp. 127–136.

[2] U. Faigle, *The greedy algorithm for partially ordered sets*, Discrete Math., 28 (1979), pp. 153–159.

[3] D. Gale, *Optimal assignments in an ordered set: an application of matroid theory*, J. Combin. Theory, 4 (1968), pp. 176–180.

[4] O. Goecke, *A greedy algorithm for hereditary set systems and a generalization of the rado-edmonds characterization of matroids*, Discrete Appl. Math., 20 (1988), pp. 39–49.

[5] P. Helman, *A theory of greedy structures based on k-ary dominance relations*, Tech. Report CS89-11, Dept. of Computer Science, University of New Mexico, 1989.

[6] B. Korte and L. Lovasz, *Mathematical structures underlying greedy algorithms*, in Fundamentals of Computation Theory, LNCS 177, Springer Verlag, 1981, pp. 205–209.

[7] ———, *Greedoids and linear objective functions*, SIAM J. Alg. Discrete Meth., 5 (1984), pp. 229–238.

[8] T. Lengauer and D. Theune, *Unstructured path problems and the making of semirings*, in Second Workshop on Data Structures and Algorithms, LNCS 519, Springer Verlag, 1991, pp. 189–200.

[9] R. Rado, *A note on independence functions*, Proc. London Math. Soc., 7 (1957), pp. 300–320.

SIAM J. DISC. MATH.
Vol. 6, No. 2, pp. 284–288, May 1993

009

THE NUMBER OF MAXIMAL INDEPENDENT SETS IN TRIANGLE-FREE GRAPHS*

MIHÁLY HUJTER† AND ZSOLT TUZA‡

Abstract. In this paper, it is proved that every triangle-free graph on $n \geq 4$ vertices has at most $2^{n/2}$ or $5 \cdot 2^{(n-5)/2}$ independent sets maximal under inclusion, whether n is even or odd. In each case, the extremal graph is unique. If the graph is a forest of odd order, then the upper bound can be improved to $2^{(n-1)/2}$.

Key words. independent vertices, triangle-free graphs, extremal graphs

AMS(MOS) subject classifications. 05C35, 05C70, 68R10

1. Introduction and results. Let $G = (V, E)$ be a simple graph (i.e., undirected, without loops and multiple edges). The set of vertices adjacent to any particular vertex $x \in V$ will be denoted by $\Gamma_G(x)$. The degree of x is $d_G(x) = |\Gamma_G(x)|$. The minimum and maximum degrees are denoted by δ_G and Δ_G, respectively. For any $Y \subseteq V$, the graph $G - Y$ is obtained from G by removing the elements of Y, and the edges incident to them. For a positive integer n, the complete graph, the path, and the cycle on n vertices, are denoted by K_n, P_n, and C_n, respectively, (in the latter, $n \geq 3$).

A set $X \subseteq V$ is *independent* if it consists of mutually nonadjacent vertices. The collection of independent sets *maximal under inclusion* in G is denoted by I_G. If G is the *null graph*, i.e., if $V = \emptyset$, then I_G is defined as $\{\emptyset\}$. Note that $|I_{G-Y}| \leq |I_G|$ holds for any graph $G = (V, E)$ and for any $Y \subseteq$V.

The first important result concerning the number of maximal independent sets was proved by Moon and Moser [6], who observed that $|I_G| \leq 3^{n/3}$ in every graph G on n vertices and characterized the extremal graphs. Their bound was slightly strengthened for connected graphs independently by Füredi [1] and Griggs, Grinstead, and Guichard [3]. It appears to be more interesting, however, that, in trees, far better upper bounds can be proved for $|I_G|$; namely, the base $3^{1/3}$ can be reduced to $2^{1/2}$. This fact was first shown by Wilf [9]; later, Sagan [8] and, independently, Griggs and Grinstead [2] gave shorter proofs and characterized all trees on n vertices that have the largest possible number of maximal independent sets.

The goal of this paper is to prove the somewhat surprising fact that the considerable decrease from $3^{n/3}$ to $2^{n/2}$ as an upper bound on $|I_G|$ is valid under a much weaker assumption. Instead of excluding *all* cycles (as in the case of trees) it suffices to exclude just the cycles C_3 of length 3.

THEOREM 1. *If G is a triangle-free graph on n vertices*, $n \geq 4$, *then*

$$|I_G| \leq \begin{cases} 2^{n/2} & \text{for } n \text{ even}, \\ 5 \cdot 2^{(n-5)/2} & \text{for } n \text{ odd}. \end{cases}$$

Moreover, equality holds if and only if G is isomorphic to the following graph, denoted by H_n. *If n is even, then* H_n *consists of* $n/2$ *disjoint copies of* K_2 *(i.e., it is a perfect matching). If n is odd, then one connected component of* H_n *is isomorphic to* C_5, *and the other components are isolated edges.*

*Received by the editors May 28, 1991; accepted for publication (in revised form) March 13, 1992.

†Institute of Mathematics, University of Miskolc, Miskolc-Egyetemváros, H-3515 Hungary.

‡Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, Kende u. 13-17, H-1111 Hungary.

We must note that, despite the same growth of $\mathcal{O}(2^{n/2})$ in their maximum number, the structure of the collection of maximal independent sets in a triangle-free graph can be far more complicated than in a tree. It is known, for example, that the *maximum independent set problem* is NP-complete on triangle-free graphs [7], while it is polynomially solvable not only on trees but also on bipartite graphs.

2. Proof of the main result. In this section, we prove Theorem 1. We apply the following observations.

LEMMA 1. *Let $G = (V, E)$ be an arbitrary graph. Then*

(a) $|I_G| = |I_{G-H}| \cdot |I_H|$, *if G is disconnected, and H is a connected component of G*;

(b) $|I_G| \leq |I_{G-\{x\}}| + |I_{G-(\{x\}\cup\Gamma_G(x))}|$, *if* $x \in V$;

(c) $|I_G| \leq |I_{G-\{x,y\}}| + |I_{G-(\{x\}\cup\Gamma_G(x))}|$, *if* $xy \in E$ *such that* $d_G(y) = 1$;

(d) $|I_G| \leq |I_{P_{n-k}}| + |I_{P_{n-3}}|$, *if G is a cycle or a path on* $n \geq 4$ *vertices and* $k = 3 - \delta_G$.

Proof. Lemma 1(a) is obvious. In (b), the first term on the right-hand side is an upper bound on the number of sets $X \in I_G$ with $x \notin X$, while $x \in X \in I_G$ holds if and only if $(X - \{x\}) \in I_{G-(\{x\}\cup\Gamma_G(x))}$. Lemma 1(c) follows from (b) since $X \in I_{G-\{x,y\}}$ if and only if $(X \cup \{y\}) \in I_{G-\{x\}}$. Finally, to prove (d), we apply (b) or (c) if G is a cycle or a path, respectively. □

It is convenient to abbreviate some values involved in the computations as follows:

$$r = 2^{1/2};$$

$$f_n = \begin{cases} r^n & \text{for } n \text{ even,} \\ 5r^{n-5} & \text{for } n \text{ odd;} \end{cases}$$

$$g_n = 7r^{n-6}.$$

Now a direct counting (based on Lemma 1(a) immediately implies that $|I_{H_n}| = f_n$ for $n \geq 4$. Moreover, since G cannot be a triangle, $|I_G| \leq f_n$ holds for $n \leq 3$; more precisely, $|I_G| < f_n$ is also valid for $n = 1$ and $n = 3$. (For $n = 2$, the equality $|I_G| = f_2$ holds only for $G = K_2 = H_2$.)

We prove the theorem by induction on n, but we note first that the following inequalities are valid for every integer n:

$$g_n < 5r^{n-5} \leq f_n \leq r^n,$$

$$f_n < \tfrac{5}{4} f_n \leq f_{n+1},$$

$$g_n < g_{n+1},$$

$$f_k f_{n-k} \leq f_n.$$

Thus, if G is disconnected, say it has a connected component on k vertices, then Lemma 1 implies that $|I_G| \leq f_n$ by induction. In addition, assuming that H_k and H_{n-k} are the unique extremal graphs on k and $n - k$ vertices, respectively, the equality $|I_G| = f_n$ implies that G is isomorphic to H_n. Hence, we may assume that G is connected.

The following observation will be useful.

LEMMA 2. *Let $n \geq 3$ be a fixed integer. Suppose that $|I_H| \leq f_m$ for every triangle-free graph H with $m < n$ vertices. If a connected triangle-free graph G' on n vertices contains a vertex of degree one, then $|I_{G'}| < g_n$.*

Proof. Let xy be an edge in G' with $\Gamma_{G'}(y) = \{x\}$. Note that $d_G(x) \geq 2$, as $n \geq 3$. Then, by Lemma 1(c) and the assumption for all $m < n$,

$$\begin{aligned} |I_{G'}| &\leq |I_{G'-\{x,y\}}| + |I_{G'-(\{x\}\cup\Gamma_{G'}(x))}| \\ &\leq f_{n-2} + f_{n-3} \leq r^{n-2} + r^{n-3} \\ &= (r^4 + r^3)r^{n-6} < 7r^{n-6} = g_n. \end{aligned}$$

□

Since $g_n < f_n$, we can assume that $\delta_G \geq 2$. Next, we settle the case where $\Delta_G \geq 4$. Say, $d_G(x) \geq 4$ for some vertex x of G. Then Lemma 1(b) yields

$$\begin{aligned} |I_G| &\leq |I_{G-\{x\}}| + |I_{G-(\{x\}\cup\Gamma_G(x))}| \\ &\leq f_{n-1} + f_{n-5} = f_{n-1} + r^{-4}f_{n-1} = \tfrac{5}{4}f_{n-1} \leq f_n, \end{aligned}$$

with equality only if $d_G(x) = 4, G - \{x\}$ is isomorphic to H_{n-1}, and $G - (\{x\} \cup \Gamma_G(x))$ is isomorphic to H_{n-5}. This implies that the subgraph induced by $\Gamma_G(x)$ in G is isomporhic to $H_4 = K_2$, contradicting the assumption that G is triangle-free. Thus, $|I_G| < f_n$ whenever $\Delta_G \geq 4$.

Hence, it remains to show that, if G is connected and triangle-free with $2 \leq \delta_G \leq \Delta_G \leq 3$, then $|I_G| \leq f_n$ holds (with strict inequality for $G \neq H_n$). If $\Delta_G = 2$, then G is a cycle on $n \geq 4$ vertices, and thus G is isomorphic to H_n if and only if $n = 5$. Hence, $|I_G| = f_n$ if $n = 5$. On the other hand, we observe that $|I_{C_n}| < f_n$ for every $n \neq 5$. Indeed, if $n = 4$, then $|I_{C_4}| = 2 < f_4$, and for $n \geq 6$, Lemmas 1(d) and 2 imply that

$$\begin{aligned} |I_G| &\leq |I_{P_{n-1}}| + |I_{P_{n-3}}| \leq \left(|I_{P_{n-3}}| + |I_{P_{n-4}}|\right) + |I_{P_{n-3}}| \\ &< 2g_{n-3} + f_{n-4} \leq 14r^{n-9} + r^{n-4} \\ &= (7 + 2r)r^{n-7} < 10r^{n-7} = 5r^{n-5} \leq f_n. \end{aligned}$$

Suppose that $\Delta_G = 3$ and let x be a vertex of degree 3. If n is even (and, in particular, if $\delta_G = 3$), then $f_n = r^n, f_{n-1} = 5r^{n-6}$, and $f_{n-4} = r^{n-4}$. In this case, Lemma 1(b) implies

$$\begin{aligned} |I_G| &\leq |I_{G-\{x\}}| + |I_{G-(\{x\}\cup\Gamma_G(x))}| \\ &\leq f_{n-1} + f_{n-4} = 5r^{n-6} + r^{n-4} = \tfrac{7}{8}r^n < f_n. \end{aligned}$$

Finally, suppose that n is odd and $2 = \delta_G < \Delta_G = 3$. Since G is connected, there is an edge xy in G such that $d_G(y) = 2$ and $d_G(x) = 3$. Let z be the unique neighbor of y in $G - \{x\}$. Then in $G - \{x\}$, the degree of y is one, and the degree of z is $d_{G-\{x\}}(z) = d_G(z) \geq \delta_G \geq 2$ since G is triangle-free. Let G' denote the connected component containing y in $G - \{x\}$, and let k denote the number of vertices in G'. Now Lemmas 1(a) and 2, together with the induction hypothesis, yield

$$|I_{G-\{x\}}| < f_{n-k-1}g_k \leq g_{n-1}.$$

Applying Lemma 1(b) and the assumption that n is odd, we obtain

$$\begin{aligned}|I_G| &< g_{n-1} + f_{n-4} = 7r^{n-7} + 5r^{n-9}\\ &= (14+5)r^{n-9} < 20r^{n-9} = 5r^{n-5} = f_n.\end{aligned}$$

This completes the proof of Theorem 1. □

3. Concluding remarks.

3.1. Forests. Theorem 1 and a repeated application of Lemma 1(c) (for n even and n odd, respectively) yields the following sharp result for the largest number of maximal independent sets in forests of given order.

COROLLARY 1 ((see [2])). *If G is a forest on n vertices, then*

$$|I_G| \leq \begin{cases} 2^{n/2} & \textit{for } n \textit{ even},\\ 2^{(n-1)/2} & \textit{for } n \textit{ odd}. \end{cases}$$

This inequality was first reported in the last section of [8]. However, that presentation is misleading because it suggests that the extremal forest is unique for each n. As a matter of fact, there is more than one extremal forest for each *odd* $n \geq 5$, and they have the following structural characterization: All but one of the connected components are isolated edges, and the last component is either an isolated vertex or a tree on $2k+1$ vertices ($0 < k < n/2$) that consists of k paths of length 2 starting from the same vertex. (This last component is the unique extremal tree of order $2k+1$, as shown in [2] and [7].)

An alternative proof of Corollary 1 is to apply Lemma 1(a) with the theorem of [2], [7], and [8].

3.2. Further problems. Comparing the results of [2], [7], and [8] with our Theorem 1, we see that, for any n, the maximum value of $|I_G|$ for triangle-free graphs G of order n is at most constant times more than for trees. There are several related problems that remain open. For example, it would be interesting to see how assumptions on *girth*, *minimum degree*, *k-connectivity* for some $k \geq 1$, *forbidden* (induced) *subgraphs*, or their combinations (possibly with some further basic properties of graphs) make $\max\{|I_G| : |V(G)| = n\}$ decrease.

3.3. Applications. One reason why upper bounds on $|I_G|$ are of interest is that better estimates on the size of I_G lead to improvements on the time analysis of algorithms determining several hard graph invariants. For example, the known exact graph-coloring algorithms can be executed in c^n steps in the worst case, where the value of the constant c depends on the assumptions on the graphs themselves as well as on the properties of colorings to be found. In particular, our results presented here lead to $\left(c_1 n^3 (2^{1/2}+1)^n\right)$-algorithms to find the chromatic number of a triangle-free graph of order n, while the general bound [5] without excluding K_3 is as large as $c_2 n^3 (3^{1/3}+1)^n$. (If a graph G has

n vertices and $K > 1$ is a constant such that for every $i \leq n$, each induced subgraph of G on i vertices has at most K^i maximal independent sets, then $c_3 n^3 (K+1)^n$ steps suffice to find the chromatic number of G.) Similar improved bounds are valid for more general coloring concepts, too, as we show in a forthcoming paper [4].

Acknowledgments. We thank the referees for some helpful observations.

REFERENCES

[1] Z. Füredi, *The number of maximal independent sets in connected graphs*, J. Graph Theory, 11 (1987), pp. 463–470.

[2] J. R. Griggs and C. M. Grinstead, unpublished result, 1986.

[3] J. R. Griggs, C. M. Grinstead, and D. R. Guichard, *The number of maximal independent sets in a connected graph*, Discrete Math., 68 (1988), pp. 211–220.

[4] M. Hujter and Zs. Tuza, *Precoloring Extension. IV. General bounds and list colorings*, in preparation.

[5] E. L. Lawler, *A note on the complexity of the chromatic number problem*, Inform. Process. Lett., 5 (1976), pp. 66–67.

[6] J. W. Moon and L. Moser, *On cliques in graphs*, Israel J. Math., 3 (1965), pp. 23–28.

[7] S. Poljak, *A note on stable sets and coloring of graphs*, Comment. Math. Univ. Carolin., 15 (1974), pp. 307–309.

[8] B. E. Sagan, *A note on independent sets in trees*, SIAM J. Discrete Math., 1 (1988), pp. 105–108.

[9] H. S. Wilf, *The number of maximal independent sets in a tree*, SIAM J. Alg. Discrete Meth., 7 (1986), pp. 125–130.

SIAM J. DISC. MATH.
Vol. 6, No. 2, pp. 289–293, May 1993

TRIANGULATING THREE-COLORED GRAPHS IN LINEAR TIME AND LINEAR SPACE*

RAMANA M. IDURY† AND ALEJANDRO A. SCHÄFFER‡

Abstract. Kannan and Warnow [*Triangulating Three-Colored Graphs*, Proc. 2nd SODA, 1991, pp. 337–343 and *SIAM J. Discrete Math.*, 5 (1992), pp. 249–258] describe an algorithm to decide whether a three-colored graph can be triangulated so that all the edges connect vertices of different colors. This problem is motivated by a problem in evolutionary biology. Kannan and Warnow have two implementation strategies for their algorithm: one uses slightly superlinear time, while the other uses linear time but quadratic space. We note that three-colored triangulatable graphs are always planar, and we use this fact to modify Kannan and Warnow's algorithm to obtain an algorithm that uses both linear time and linear space.

Key words. chordal graphs, perfect phylogeny problem, planar graphs, k-trees

AMS(MOS) subject classifications. 68Q20, 68R10

1. Introduction and definitions. In a series of recent papers, Bodlaender, Fellows, and Warnow [2]; Kannan and Warnow [6]; and McMorris, Warnow, and Wimer [7] consider the problem of whether a vertex-colored graph can be triangulated in a manner consistent with the coloring. The **Triangulating a Colored Graph (TCG) Problem** is: Given an undirected graph $G = (V, E)$ with a vertex coloring $c : V \to \{1, 2, \ldots, |V|\}$, determine whether it is possible to insert extra edges so that every cycle contains a chord and all edges connect only pairs of vertices with different colors. If a legal triangulation exists with respect to the coloring c, we refer to the augmented graph as a c-triangulation and say that G is c-triangulatable. If the coloring is known to be a 3-coloring, we say that G is 3-triangulatable, and the triangulation is a 3-triangulation.

The reason for studying the TCG problem is that it is polynomially equivalent to an important problem in evolutionary biology called the **Perfect Phylogeny (PP) Problem** [4]. Details about the PP problem can be found in [6]. Bodlaender, Fellows, and Warnow showed that TCG, and hence PP, are NP-complete [2]. In contrast, McMorris, Warnow, and Wimer [7] described an algorithm to solve TCG in $O((n + m(k-1))^{(k+1)})$ time, where n is the number of vertices, m is the number of edges, and k is the number of colors. Although the general problem is now known to be NP-complete, this does not rule out the possibility of small-degree polynomial-time algorithms for fixed parameter values. An indication that TCG may be difficult even for fixed parameters is given by the second main theorem of [2], which shows in a formal sense that the fixed-parameter problem resists all the standard bounded tree-width techniques when the number of colors is four or more.

Kannan and Warnow [6] focused on the three-color case, which is relevant to the tools that biologists use in practice to solve the PP problem. For three-colored graphs, the algorithm in [7] uses $O((n+2m)^4)$ time to decide if the graph can be c-triangulated. Kannan and Warnow [6] described an algorithm that runs in $O(n\alpha(n))$ time; this is the same slightly nonlinear asymptotic time that is required for n union-find operations. In the journal version of their paper, Kannan and Warnow showed how a different choice of data structures removes the need for union-find, making the running time linear. They

*Received by the editors September 23, 1991; accepted for publication (in revised form) June 12, 1992.

†Department of Computer Science, Rice University, P. O. Box 1892, Houston, Texas 77251. This research was partially supported by the W. M. Keck Foundation.

‡Department of Computer Science, Rice University, P. O. Box 1892, Houston, Texas 77251. This research was partially supported by National Science Foundation grant CCR-9010534.

used the trick of [1, Ex. 2.12] to convert a graph from adjacency list representation to adjacency matrix representation, so that adjacency queries can be done in constant time [6]. However, this trick requires quadratic space to store the matrix, even though only a linear number of the entries are actually examined.

In this note, we describe some simple modifications to Kannan and Warnow's algorithm [6] that yield an algorithm that runs in $O(n)$ time *and* uses $O(n)$ space. Bodlaender and Kloks [3] independently found a very different algorithm that is not based on Kannan and Warnow's algorithm [6] and also achieves these bounds. The constants hidden behind the O-notation seem to be small in both our algorithm and the algorithm of Bodlaender and Kloks [3]. The algorithm of Bodlaender and Kloks [3] may be suitable for implementing from scratch, whereas our algorithm is very simple to implement if we have access to a package that tests the planarity of a graph and provides a plane embedding if the graph is planar.

We now present some relevant definitions and lemmas and a high-level sketch of the Kannan–Warnow algorithm [6] to decide if a three-colored graph can be triangulated. Our modifications to the algorithm are described in §2.

DEFINITION (see [5, p. 100]). The set of *k-trees* is described by the following conditions:

1. A k-clique is a k-tree.

2. If G is a k-tree, C is a k-clique in G, and v is a new vertex, then the graph G' formed by inserting v in G and connecting v to precisely the vertices in C is a k-tree.

3. No other graphs are k-trees.

LEMMA 1.1 (see [7]). *A* $(k+1)$*-colored graph can be triangulated if and only if it can be triangulated into a k-tree.*

From the definition of k-trees, Lemma 1.2 follows.

LEMMA 1.2. *2-trees have at most* $2n-3$ *edges.*

The key idea in our modifications is the following fact.

LEMMA 1.3 (see [5, Ex. 4.8]). *2-trees are planar.*

(We note that in [5] it is stated incorrectly that 3-trees are planar. Actually, 2-trees are planar, but some 3-trees are not planar.)

Since planarity is inherited by subgraphs we have the following corollary.

COROLLARY 1.4. *Any 3-triangulatable graph is planar.*

DEFINITION. A vertex is *simplicial* if it and its neighbors induce a clique.

THEOREM 1.5 (see [5, Thm. 4.1]). *A graph is c-triangulated* (*or chordal*) *if and only if its vertices can be ordered* $v_1 < v_2 < \cdots < v_n$ *so that* v_i *is simplicial in the graph induced by* $\{v_i, v_{i+1}, \ldots, v_n\}$. *Such an ordering is called a* Perfect Elimination Ordering(PEO).

LEMMA 1.6 (see [6]). *G can be c-triangulated if and only if all its biconnected components can be c-triangulated. Furthermore, the c-triangulations of the biconnected components can be put together in linear time and linear space.*

The Kannan–Warnow algorithm repeatedly chooses cycles and attempts to triangulate them. The main difficulty is how to choose the cycle to work on next.

DEFINITION. A cycle is *feasible* if

1. all but two of its vertices have degree 2, and

2. the two vertices of higher degree have neighbors outside the cycle.

The vertices of higher degree in a feasible cycle are called *port* vertices.

Kannan and Warnow's algorithm can be summarized as follows. It uses a subroutine that decides whether cycles can be c-triangulated and c-triangulates cycles in time proportional to the length of the cycle.

```
if $|E| > 2n - 3$ then return NO
while $V \neq \emptyset$ do
    delete all simplicial vertices (1)
    if $V = \emptyset$ then return Yes
    if $G$ is a cycle then
        Call the cycle subroutine on $G$ and return Yes/No
        depending on whether it succeeds
    else
        find a feasible cycle with port vertices $a, b$
        if $c(a) \neq c(b)$ then
            insert the edge $(a, b)$ and use the cycle
            subroutine to see if the rest of the cycle
            can be $c$-triangulated
            if the subroutine fails then return No
            else store the new chords and delete all vertices
                in the cycle, except $a, b$, as they are now simplicial (2)
        else return No
        if no feasible cycle exists return No
endwhile
```

The key facts underlying the correctness of the algorithm are summarized by the following lemmas.

LEMMA 1.7 (see [6]). *If G is a 3-triangulatable biconnected graph with no simplicial vertices, either G is a cycle or G contains a feasible cycle.*

LEMMA 1.8 (see [6]). *Suppose γ is a feasible cycle with port vertices a, b. Then G can be c-triangulated if and only if $G \cup (a, b)$ can be c-triangulated.*

We will prove in Theorem 2.1 that the assumption that there are no simplicial vertices is unnecessary in Lemma 1.7. This assumption is the only reason for deleting simplicial vertices at step 1, so we omit step 1 in our modified algorithm.

The analysis of running time can be summarized by the following lemma.

LEMMA 1.9 (see [6]). *All the steps in the algorithm can be implemented to use linear time and linear space over the entire history of the algorithm, except the steps of deleting simplicial vertices at step* 1 *and of finding feasible cycles.*

In §2, we show how to use planarity to find all necessary feasible cycles in linear time and linear space.

2. Finding feasible cycles. In this section, we describe an algorithm that finds a feasible cycle in time linear in the size of the cycle and requires overall linear space. By Corollary 1.5 and Lemma 1.8, we can assume that the input graph G is planar and biconnected. Hence, in any planar embedding $\hat{G}$ of G, all faces of $\hat{G}$ are simple cycles of G. Any feasible cycle that is a face in $\hat{G}$ is called a *feasible face*. The following theorem is the basis for our algorithm.

THEOREM 2.1. *If G is a 3-triangulatable graph, but not a cycle, and $\hat{G}$ is* any *plane representation of G, then there is a feasible cycle that is a face in $\hat{G}$.*

Proof. The proof is by induction on the number of vertices, n. For $n = 3$, the only cycle possible is a triangle and it must be a face. Let $j > 3$ be given and assume the theorem is true for all graphs with fewer than j vertices. Let G be a 3-triangulatable graph with j vertices and let $\hat{G}$ be any plane representation of G. Since G is 3-triangulatable, let T be any 3-triangulation, and let P be a perfect elimination order for T. Let v_1 be the first vertex in P. Since G is a subgraph of T, which is three-colored and biconnected,

$degree(v_1)$ must be 2. Let v_r, v_s be its two neighbors in G. Since v_1 is of degree 2 it will be incident to two faces; let F_1 and F_2 be the faces on either side of the edges (v_1, v_r) and (v_1, v_s) in $\hat{G}$.

Case 1. $(v_r, v_s) \in E(G)$. If the three edges (v_r, v_s), (v_1, v_r), and (v_1, v_s) form a triangular face in $\hat{G}$ then that itself is a feasible face; in particular, if $\hat{G} \setminus \{v_1\}$ is a cycle, then the three edges form a triangular feasible face. Otherwise, let F' be the face in $\hat{G} \setminus \{v_1\}$ resulting by combining F_1 and F_2. By inductive hypothesis, $\hat{G} \setminus \{v_1\}$ must have a feasible face F. If F is different from F', then F will still be a feasible face in $\hat{G}$. On the other hand, if F' is the only feasible face in $\hat{G} \setminus \{v_1\}$, then v_r and v_s must be its only port vertices. In that case, both F_1 and F_2 are feasible faces in $\hat{G}$.

Case 2. (v_r, v_s) is not an edge of G. Since v_1 comes before both its neighbors in the perfect elimination order P, the triangulation T must contain the edge (v_r, v_s), and the two vertices must have different colors. Therefore, we remove (v_1, v_r) and (v_1, v_s) in $\hat{G}$ and introduce the new edge (v_r, v_s) incident to the same faces. The graph $\hat{G} \setminus \{v_1\} \cup \{(v_r, v_s)\}$ is c-triangulatable with fewer than j vertices, and hence by hypothesis, is a cycle or has a feasible face F. If the smaller graph is a cycle, then G is also a cycle. If F is different from F_1 or F_2, then F is feasible in G. Otherwise, we extend the feasible face (F_1 or F_2) by removing the edge (v_r, v_s) and introducing two edges (v_1, v_r) and (v_1, v_s) in its place. The resulting face is still feasible.

Since G was an arbitrary graph with j vertices, all biconnected, 3-triangulatable graphs with j vertices either are cycles or contain a feasible cycle. The theorem follows by induction. □

LEMMA 2.2. *If G is not a cycle and F is a feasible face in some plane representation $\hat{G}$ of G with port vertices a and b, then G is c-triangulatable if and only if $G \cup \{(a, b)\}$ is.*

Proof. The proof follows from [6] and the fact that a feasible face is also a feasible cycle in a biconnected plane graph. □

Now we are in a position to present our algorithm for finding feasible cycles efficiently. We find a plane representation for G and call it $\hat{G}$. We construct for each vertex a clockwise circular doubly linked list of adjacent vertices. For each edge, we store as v_1, v_2 its end vertices and f_1, f_2 its surrounding faces in $\hat{G}$ with the interpretation that f_1 is the face to the left of the edge if we direct the edge from v_1 to v_2. For each face F, we indicate whether it is internal or external and store as $F(e)$ an edge belonging to the face. With this information we can get all the edges of a face in time linear in the number of edges of the face.

Once a feasible cycle of G is triangulated, all its nonport vertices become simplicial and can be deleted at step 2.

The following information must be precomputed at the beginning of the algorithm of §1.

```
Find a planar embedding $\hat{G}$ as described above.
For each face $F$, compute $Count(F) := |\{v \mid v \in F \text{ and } degree(v) \geq 3\}|$.
Build two disjoint lists of faces
    $L := \{F \mid Count(F) \leq 2\}$,
    $\bar{L} := \{F \mid Count(F) > 2\}$.
```

The faces in L are feasible; the faces in $\bar{L}$ are not feasible because they contain too many high-degree vertices. The following code must be executed at every request to find a feasible cycle:

```
if $L \neq \emptyset$
  1. Remove face $F$ from $L$ with port vertices $a$ and $b$.
```

```
    2. **return** $F$ as the feasible cycle.
  **else** report the failure to find a feasible cycle.
```

If the cycle F is successfully 3-triangulated, we update $\hat{G}$ at step (2) as follows:

```
3. Remove all edges (along with vertices disconnected from $\hat{G}$) of $F$, and
     add edge $(a, b)$. Modify $degree(a)$ and $degree(b)$.
4. For faces $F_1$ and $F_2$ adjacent to $F$, set $F_1(e) = F_2(e) = (a, b)$.
     For $(a, b)$ set $f_1, f_2 = F_1, F_2$ accordingly.
5. If $degree(a) \leq 2$, then for every face $H$ containing $a$,
     $Count(H) = Count(H) - 1$.
     If $Count(H) \leq 2$, remove $H$ from $\bar{L}$ and add it to $L$.
6. Do step 5 for $b$.
```

THEOREM 2.3. *The modified algorithm to recognize* 3*-triangulatable graphs and triangulate them is correct and uses* $O(n)$ *time and* $O(n)$ *space.*

Proof. The correctness of the algorithm follows from [6] and the fact that our algorithm differs from theirs only in the way of finding feasible cycles and in the omission of deleting simplicial vertices at step 1. From Theorem 2.1 it follows that our method for finding feasible cycles is correct and that it is not necessary to delete simplicial vertices to find a feasible cycle. It is necessary to delete some simplicial vertices eventually, so that the algorithm makes progress; in our version this is done at step 2, as described above. Only a constant amount of work is done per edge throughout the entire algorithm. This implies a linear running time for our algorithm. From the data structures used in the algorithm, it is evident that our algorithm needs only linear space. Finally, we note that there are several linear-time, linear-space algorithms for embedding planar graphs [8]. □

REFERENCES

[1] A. V. AHO, J. E. HOPCROFT, AND J. D. ULLMAN, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

[2] H. BODLAENDER, M. FELLOWS, AND T. WARNOW, *Two strikes against the perfect phylogeny problem*, Technical Report RUU-CS-92-08, Department of Computer Science, Utrecht University, the Netherlands, 1992; in Proc. ICALP 92, to appear.

[3] H. BODLAENDER AND T. KLOKS, *A simple linear-time algorithm for triangulating three-colored graphs*, Proc. STACS 92, Lecture Notes in Comput. Sci. 577, 1992, Springer-Verlag, Berlin, pp. 415–423.

[4] P. BUNEMAN, *A characterization of rigid circuit graphs*, Discrete Math., 9 (1974), pp. 205–212.

[5] M. C. GOLUMBIC, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[6] S. KANNAN AND T. WARNOW, *Triangulating three-colored graphs*, Proc. 2nd Annual ACM-SIAM Symposium on Discrete Algorithms, 1991, pp. 337–343; SIAM J. Discrete Math., 5 (1992), pp. 249–258.

[7] F. R. MCMORRIS, T. J. WARNOW, AND T. WIMER, *Triangulating vertex colored graphs*, in Proc. 4th Annual ACM-SIAM Symposium on Discrete Algorithms, 1993.

[8] T. NISHIZEKI AND N. CHIBA, *Planar Graphs: Theory and Algorithms*, Annals of Discrete Math. 32, North-Holland, Amsterdam, 1988.

SIAM J. DISC. MATH.
Vol. 6, No. 2, pp. 294–308, May 1993

011

BARKER ARRAYS I: EVEN NUMBER OF ELEMENTS*

JONATHAN JEDWAB†

Abstract. A Barker array is a two-dimensional array with elements ±1 such that all out-of-phase aperiodic autocorrelation coefficients are 0, 1, or -1. No $s \times t$ Barker array with $s, t > 1$ and $(s, t) \neq (2, 2)$ is known, and it is conjectured that none exists. A class of arrays that includes Barker arrays is defined. Nonexistence results for this class of arrays in the case st even, providing support for the Barker array conjecture, are proved. Several connections, in the case st even, between this class of arrays and perfect, quasi-perfect, and doubly quasi-perfect binary arrays are demonstrated.

Key words. Barker array, aperiodic autocorrelation, binary array, nonexistence, perfect, quasi-perfect, doubly quasi-perfect

AMS(MOS) subject classifications. primary 05B20; secondary 05B10

1. Introduction.

An $s \times t$ *binary array* is a two-dimensional array (a_{ij}) for which

$$a_{ij} = \begin{cases} 1 \text{ or } -1 & \text{for all } 0 \le i < s, 0 \le j < t, \\ 0 & \text{otherwise.} \end{cases}$$

Define the *aperiodic autocorrelation function* of a binary array (a_{ij}) by

$$C(u, v) = \sum_i \sum_j a_{ij} a_{i+u,j+v},$$

where u and v are integers. In this paper, summations will be over all integers unless otherwise stated. We write $C_A(u, v)$ to distinguish the aperiodic autocorrelation function of A from that of any other binary array. A binary array is called *Barker* if $|C(u, v)| \le 1$ for all $(u, v) \neq (0, 0)$. The array $\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$ is Barker, but no Barker array with $s, t > 1$ and $(s, t) \neq (2, 2)$ is known. Alquaddoomi and Scholtz [1] conjectured that no such array exists and proved the necessary conditions that neither s nor t is an odd prime, that st is a square when s or t is even, and that $2st - 1$ is a square when $st \equiv 1 \pmod 4$. Jedwab [6] proved that, if s, t are even, then $s = t$.

In this paper, we define a property of binary arrays that we call *Barker structure*, which any $s \times t$ Barker array with $st > 2$ possesses. For an $s \times t$ binary array with Barker structure, we prove restrictions on the possible values of (s, t), as well as the array elements (a_{ij}), in the cases s, t even and s even, t odd. We also show that any such array is simultaneously perfect and quasi-perfect and that its existence implies the existence of larger arrays with restrictive autocorrelation properties. (For background material on perfect, quasi-perfect, and doubly quasi-perfect, arrays, the reader is referred to Jedwab et al. [9].)

In a further paper [8], we prove nonexistence results for binary arrays with Barker structure when s, t are odd.

2. Barker structure.

Define the *rowwise* and *columnwise semiperiodic autocorrelation function* of an $s \times t$ binary array by

(1) $P^R(u, v) = C(u, v) + C(u, v - t)$, defined on $-s < u < s, 0 \le v < t$,

*Received by the editors July 25, 1991; accepted for publication (in revised form) April 2, 1992.

†Hewlett-Packard Laboratories, Filton Road, Stoke Gifford, Bristol BS12 6QZ, United Kingdom. This work was carried out while the author was studying for a Mathematics Ph.D. at Royal Holloway and Bedford New College, University of London.

$$(2)\quad P^C(u,v) \;=\; C(u,v)+C(u-s,v), \quad \text{defined on } 0 \le u < s, -t < v < t,$$

respectively. Any expression involving $P^R(u,v)$ or $P^C(u,v)$ (or any other autocorrelation function referred to later in this paper) will implicitly refer only to values of (u,v) for which the function is defined. Given a binary array $A = (a_{ij})$, we call the values

$$x_i = \sum_j a_{ij}, \qquad y_j = \sum_i a_{ij}$$

the *row sums* and *column sums* of A, respectively. From Lemma 2 of [6], we have the following lemma.

LEMMA 2.1. *Let A be an $s \times t$ binary array and let (x_i) and (y_j) be, respectively, the row sums and column sums of A. Then*

$$\begin{aligned} \sum_{v=0}^{t-1} P^R(u,v) &= \sum_i x_i x_{i+u} \quad \textit{for all } u, \\ \sum_{u=0}^{s-1} P^C(u,v) &= \sum_j y_j y_{j+v} \quad \textit{for all } v. \end{aligned}$$

We now define the Barker structure property.

DEFINITION 1. Let A be an $s \times t$ binary array. A is said to have *Barker structure* if, for all $(u,v) \neq (0,0)$,

(i) For s,t even,

$$P^R(u,v) = 0, \qquad P^C(u,v) = 0.$$

(ii) For s even and t odd,

$$P^R(u,v) = \begin{cases} 0 & \text{for } u \text{ even}, \\ k(u) & \text{for } u \text{ odd}, \end{cases}$$

where $k(u) = 1$ or -1 for all $-s < u < s$, and $k(u) + k(u-s) = 0$ for all $0 < u < s$,

$$P^C(u,v) = 0.$$

(iii) For s,t odd,

$$P^R(u,v) = \begin{cases} k & \text{for } u \text{ even}, \\ 0 & \text{for } u \text{ odd}, \end{cases} \qquad P^C(u,v) = \begin{cases} k & \text{for } v \text{ even}, \\ 0 & \text{for } v \text{ odd}, \end{cases}$$

where $k = 1$ or -1 and $k \equiv st \pmod 4$.

THEOREM 2.2 (see [1]). *Let A be an $s \times t$ Barker array with $st > 2$. Then A has Barker structure.*

Theorem 2.2 is implied by equations (21)–(23) of [1]. However, we deliberately state the result in weaker form. In fact, we derive all our results for arrays possessing only Barker structure.

We note some preliminary restrictions on the values of (s, t) for an $s \times t$ binary array with Barker structure.

THEOREM 2.3 (see [1]). *Let A be an $s \times t$ binary array with Barker structure. Then there exists a (v, k, λ)-difference set in $\mathbb{Z}_s \times \mathbb{Z}_t$ with parameters as follows*:

(i) *For s or t even, $st = 4N^2$ for some integer N and $(v, k, \lambda) = (4N^2, 2N^2 - N, N^2 - N)$*,

(ii) *For $st \equiv 1 \pmod 4$, $2st - 1 = (2N+1)^2$ for some integer N and $(v, k, \lambda) = (2N^2 + 2N + 1, N^2, N(N-1)/2)$*,

(iii) *For $st \equiv 3 \pmod 4$, $st = 4N - 1$ for some integer N and $(v, k, \lambda) = (4N - 1, 2N - 1, N - 1)$.*

Although Theorem 2.3 was obtained in [1] only for Barker arrays, the method clearly applies to arrays with Barker structure. The parameters in Theorem 2.3 (i) and (iii) are those of Menon and Hadamard difference sets, respectively. (For a general treatment of difference sets, see [3] or [5].)

We obtain further restrictions on the dimensions of an $s \times t$ binary array with Barker structure by applying Lemma 2.1. This leads to equations in the row and column sums that are necessarily satisfied by such an array. In the following sections, we examine the following cases:

(i) s, t even—the equations are straightforward to solve,

(ii) s even and t odd—the equations reduce to a familiar unsolved problem.

We investigate the case s, t odd in a further paper [8] in which we do not solve the equations but obtain conditions on s and t that are necessary for the equations to have a solution.

3. The case where s, t even.

3.1. Row and column sum equations. We first examine some consequences of the equations in the row and column sums that are necessarily satisfied by an $s \times t$ binary array with Barker structure, where s, t are even. Call an $s \times t$ binary array *positive* if $\sum_i \sum_j a_{ij} \geq 0$. Without loss of generality, we may take a binary array (a_{ij}) with Barker structure to be positive, since $(-a_{ij})$ also has Barker structure.

From Lemma 2.1 and Definition 1(i), the row sums (x_i) satisfy

$$(3) \qquad \sum_i x_i x_{i+u} = \begin{cases} 0 & \text{for all } u \neq 0, \\ st & \text{for } u = 0. \end{cases}$$

Using these equations and the corresponding equations in the column sums, Jedwab [6] used Lemma 3.1 to prove Theorem 3.2.

LEMMA 3.1. *Let (x_i) be the row sums of an $s \times t$ binary array such that* (3) *is satisfied. Then $s \leq t$ and, for some $0 \leq I < s$,*

$$x_i = \begin{cases} 0 & \textit{for all } i \neq I, \\ \pm\sqrt{st} & \textit{for } i = I. \end{cases}$$

THEOREM 3.2. *Let A be an $s \times t$ binary array with Barker structure where s, t are even. Let (x_i) and (y_j) be the row and column sums of A. Then $s = t$ and, for some $0 \leq I < s$, $0 \leq J < t$,*

$$x_i = \begin{cases} 0 & \text{for all } i \neq I, \\ kt & \text{for } i = I, \end{cases} \qquad y_j = \begin{cases} 0 & \text{for all } j \neq J, \\ kt & \text{for } j = J, \end{cases}$$

where $k = 1$ if A is positive and $k = -1$ otherwise.

We now obtain further conditions on t and (a_{ij}) with the help of the following lemma, whose proof is straightforward. This describes the transformation of the aperiodic and semiperiodic autocorrelation functions under change of sign of alternate rows or columns of a binary array.

LEMMA 3.3. *Let $A = (a_{ij}), B = (b_{ij}), C = (c_{ij})$ be $s \times t$ binary arrays related by $b_{ij} = (-1)^j a_{ij}, c_{ij} = (-1)^i a_{ij}$ for all (i, j). Then, for all (u, v),*

(i) *It holds that*

$$\begin{aligned} C_B(u,v) &= (-1)^v C_A(u,v), \\ P_B^R(u,v) &= \begin{cases} (-1)^v P_A^R(u,v) & \text{for } t \text{ even}, \\ (-1)^v (C_A(u,v) - C_A(u, v-t)) & \text{for } t \text{ odd}, \end{cases} \\ P_B^C(u,v) &= (-1)^v P_A^C(u,v); \end{aligned}$$

(ii) *It holds that*

$$\begin{aligned} C_C(u,v) &= (-1)^u C_A(u,v), \\ P_C^R(u,v) &= (-1)^u P_A^R(u,v), \\ P_C^C(u,v) &= \begin{cases} (-1)^u P_A^C(u,v) & \text{for } s \text{ even}, \\ (-1)^u (C_A(u,v) - C_A(u-s, v)) & \text{for } s \text{ odd}. \end{cases} \end{aligned}$$

We can now establish further conditions on t and (a_{ij}).

DEFINITION 2. Let $A = (a_{ij})$ be an $s \times t$ binary array. Let (I, I', J, J') be a parameter set such that A has the following properties:

(i) $0 \leq I < s, 0 \leq I' < s, 0 \leq J < t, 0 \leq J' < t$,

(ii) $I + I' \equiv J + J' \pmod 2$,

(iii) $a_{Ij} = 1$ for all $0 \leq j < t$,

(iv) $a_{I'j} = (-1)^{j+J}$ for all $0 \leq j < t$,

(v) $a_{iJ} = 1$ for all $0 \leq i < s$,

(vi) $a_{iJ'} = (-1)^{i+I}$ for all $0 \leq i < s$,

(vii) $\sum_j a_{i,2j} = \sum_j a_{i,2j+1} = 0$ for all $i \neq I, I'$,

(viii) $\sum_i a_{2i,j} = \sum_i a_{2i+1,j} = 0$ for all $j \neq J, J'$.

A is called *balanced* with parameters (I, I', J, J').

THEOREM 3.4. *Let A be a positive $s \times t$ binary array with Barker structure where s, t are even. Then $s = t$ and A is balanced for some parameters (I, I', J, J'). If $t > 2$, then $t \equiv 0 \pmod 4$.*

Proof. From Theorem 3.2, we have $s = t$, and, for some $0 \leq I < s$, $0 \leq J < t$,

$$\sum_j a_{ij} = 0 \quad \text{for all } i \neq I, \tag{4}$$

$$\sum_i a_{ij} = 0 \quad \text{for all } j \neq J. \tag{5}$$

Since A is a positive array, Theorem 3.2 also gives

$$\begin{aligned} a_{Ij} &= 1 \quad \text{for all } 0 \leq j < t, \\ a_{iJ} &= 1 \quad \text{for all } 0 \leq i < s. \end{aligned} \tag{6}$$

These are balance properties (iii) and (v).

Now define $B = (b_{ij})$ by $b_{ij} = (-1)^j a_{ij}$. From Lemma 3.3(i) and Definition 1(i), B is also an $s \times t$ binary array with Barker structure where s, t are even. Hence, by Theorem 3.2, for some $0 \leq I' < s$, $0 \leq X < t$,

$$\begin{aligned} \sum_j b_{ij} &= 0 \quad \text{for all } i \neq I', \\ \sum_i b_{ij} &= 0 \quad \text{for all } j \neq X, \end{aligned} \tag{7}$$

$$b_{I'j} = k \quad \text{for all } 0 \leq j < t, \tag{8}$$

$$b_{iX} = k \quad \text{for all } 0 \leq i < s, \tag{9}$$

where $k = \pm 1$. We next determine X and k. Rewrite (8) and (9) in terms of (a_{ij}),

$$a_{I'j} = (-1)^j k \quad \text{for all } 0 \leq j < t, \tag{10}$$

$$a_{iX} = (-1)^X k \quad \text{for all } 0 \leq i < s. \tag{11}$$

By comparing (11) with (5) and (6), we deduce that $X = J$ and $(-1)^X k = 1$, so that $k = (-1)^J$. Substitution into (10) gives

$$a_{I'j} = (-1)^{j+J} \quad \text{for all } 0 \leq j < t, \tag{12}$$

which is balance property (iv).

Similarly, applying Lemma 3.3(ii) to $C = (c_{ij})$, where $c_{ij} = (-1)^i a_{ij}$, establishes that, for some $0 \leq J' < t$,

$$a_{iJ'} = (-1)^{i+I} \quad \text{for all } 0 \leq i < s, \tag{13}$$

which is balance property (vi). The ranges for I, I', J, J' given by Theorem 3.2 are those of balance property (i). Substitution of $j = J'$ into (12) and $i = I'$ into (13) gives two alternative expressions for $a_{I'J'}$,

$$a_{I'J'} = (-1)^{J'+J} = (-1)^{I'+I},$$

so that, for consistency,

$$I' + I \equiv J' + J \pmod 2,$$

which is balance property (ii).

Finally, suppose that $t > 2$, so that there exists some $0 \le i < t$ for which $i \ne I, I'$. For any such i, from (4) and (7),

$$\sum_j a_{ij} = 0, \qquad \sum_j (-1)^j a_{ij} = 0.$$

Therefore

$$\sum_j a_{i,2j} = \sum_j a_{i,2j+1} = 0 \quad \text{for all } i \ne I, I', \tag{14}$$

which is balance property (vii). However, $\sum_j a_{i,2j}$ is the sum of exactly $t/2$ nonzero terms, each of which is 1 or -1, so (14) implies that $t/2 \equiv 0 \pmod 2$, or, equivalently,

$$t \equiv 0 \pmod 4.$$

Balance property (viii) is proved in a similar manner to property (vii). □

3.2. Perfect, quasi-perfect, and doubly quasi-perfect binary arrays. We next show that the existence of an $s \times t$ binary array with Barker structure where s, t are even implies the existence of infinite families of binary arrays with restrictive autocorrelation properties.

We define the *periodic, periodic rowwise quasi-, periodic columnwise quasi-*, and *periodic doubly quasi-*autocorrelation function of an $s \times t$ binary array on $0 \le u < s$, $0 \le v < t$, respectively,

$$\begin{aligned}
R(u,v) &= C(u,v) + C(u,v-t) + C(u-s,v) + C(u-s,v-t),\\
Q^R(u,v) &= C(u,v) + C(u,v-t) - C(u-s,v) - C(u-s,v-t),\\
Q^C(u,v) &= C(u,v) - C(u,v-t) + C(u-s,v) - C(u-s,v-t),\\
D(u,v) &= C(u,v) - C(u,v-t) - C(u-s,v) + C(u-s,v-t).
\end{aligned}$$

An $s \times t$ binary array for which the autocorrelation function is 0 for all $(u,v) \ne (0,0)$ is called, respectively, *perfect, rowwise quasi-perfect, columnwise quasi-perfect*, and *doubly quasi-perfect*, written, respectively, PBA(s,t), RQPBA(s,t), CQPBA(s,t) and DQPBA(s,t). For further details, see Jedwab et al. [9] (Wild [17] showed the above definitions to be equivalent to those in [9]).

LEMMA 3.5. *Let A be an $s \times t$ binary array. Then*

$$P^R(u,v) = 0 \quad \textit{for all } (u,v) \ne (0,0),$$
$$(\textit{respectively}, P^C(u,v) = 0 \quad \textit{for all } (u,v) \ne (0,0))$$

if and only if

(i) *A is a* PBA(s,t), *and*

(ii) *A is a* RQPBA(s,t) (*respectively*, CQPBA(s,t)).

Proof. Using (1), we may write, for all (u, v),

$$R(u,v) = \begin{cases} P^R(u,v) + P^R(u-s,v) & \text{for } u \neq 0, \\ P^R(u,v) & \text{for } u = 0, \end{cases}$$

$$Q^R(u,v) = \begin{cases} P^R(u,v) - P^R(u-s,v) & \text{for } u \neq 0, \\ P^R(u,v) & \text{for } u = 0. \end{cases}$$

Then, for $(u, v) \neq (0,0)$,

$$P^R(u,v) = 0 \quad \text{for all } -s < u < s,\ 0 \leq v < t$$

if and only if

$$R(u,v) = Q^R(u,v) = 0 \quad \text{for all } 0 \leq u < s,\ 0 \leq v < t.$$

The second equivalence follows similarly from (2). □

We note that arrays for which $P^C(u,v) = 0$ for all $(u,v) \neq (0,0)$ (i.e., which are simultaneously perfect and columnwise quasi-perfect) were previously studied under the name *aperiodic perfect* arrays by Lüke, Bömer, and Antweiler [11] and, allowing array elements 0 as well as ±1, by Antweiler, Bömer, and Lüke [2].

THEOREM 3.6. *Let A be an $s \times t$ binary array where s, t are even. Then A has Barker structure if and only if $s = t$ and A is simultaneously a* PBA(s,t), *a* RQPBA(s,t) *and a* CQPBA(s,t).

Proof. The proof is immediate from Theorem 3.2, Lemma 3.5, and Definition 1(i). □

The simultaneous autocorrelation properties of A given in Theorem 3.6 allow the construction of infinite families of perfect, quasi-perfect, and doubly quasi-perfect binary arrays. We note from Corollary 4 of [9] that the existence of a DQPBA(s,t) is equivalent to the existence of a RQPBA(s,t) if $t/\gcd(s,t)$ is odd and to the existence of a CQPBA(s,t) if $s/\gcd(s,t)$ is odd.

THEOREM 3.7. *Let A be an $s \times t$ binary array with Barker structure where s, t are even. Then there exists each of the following types of array, for each $y \geq 0$:*

$$\text{PBA}(2^y t, 2^y t), \qquad \text{PBA}(2^{y+2} t, 2^y t), \qquad \text{DQPBA}(2^y t, 2^y t),$$
$$\text{RQPBA}(2^y t, 2^{y+2} t), \quad \text{DQPBA}(2^{y+1} t, 2^y t), \quad \text{RQPBA}(2^{y+1} t, 2^{y+2} t),$$
$$\text{RQPBA}(2^{y+1} t, 2^{y+4} t).$$

Proof. From Theorem 3.6, A is simultaneously a PBA(t,t), a RQPBA(t,t), and a CQPBA(t,t). The existence of the first four families follows from Corollary 5 of [9]. The existence of the remaining families follows from Theorem 7 of [9], provided that there exists a RQPBA$(2t,t)$. To complete the proof, we now show that, if A is simultaneously a PBA(s,t) and a RQPBA(s,t), then $B = \left[\begin{smallmatrix} A \\ A \end{smallmatrix}\right]$ is a RQPBA$(2s,t)$.

From Lemma 3.5,

$$P_A^R(u,v) = 0 \quad \text{for all } (u,v) \neq (0,0).$$

For all $0 \le u < 2s$, $0 \le v < t$,

$$\begin{aligned} P_B^R(u,v) &= \sum_{i=0}^{2s-1}\sum_{j} b_{ij}b_{i+u,(j+v)\operatorname{mod}t} \\ &= \sum_{i=0}^{s-1}\sum_{j} b_{ij}b_{i+u,(j+v)\operatorname{mod}t} + \sum_{i=0}^{s-1}\sum_{j} b_{i+s,j}b_{i+s+u,(j+v)\operatorname{mod}t}. \end{aligned} \tag{15}$$

If $u \ge s$, then the second term of (15) is 0, and so

$$\begin{aligned} P_B^R(u,v) &= \sum_{i=0}^{s-1}\sum_{j} a_{ij}a_{i+u-s,(j+v)\operatorname{mod}t} \\ &= P_A^R(u-s,v), \end{aligned}$$

whereas, if $u < s$, then from (15)

$$\begin{aligned} P_B^R(u,v) &= 2\sum_{i=0}^{s-u-1}\sum_{j} a_{ij}a_{i+u,(j+v)\operatorname{mod}t} + \sum_{i=s-u}^{s-1}\sum_{j} a_{ij}a_{i+u-s,(j+v)\operatorname{mod}t} \\ &= \begin{cases} 2P_A^R(u,v) + P_A^R(u-s,v) & \text{if } u \ne 0, \\ 2P_A^R(u,v) & \text{if } u = 0. \end{cases} \end{aligned}$$

Therefore, for $(u,v) \ne (0,0)$ or $(s,0)$, $P_B^R(u,v) = 0$ and hence $Q_B^R(u,v) = 0$. Also, $P_B^R(s,0) = st$, and so $Q_B^R(s,0) = P_B^R(s,0) - P_B^R(0,0) = 0$.

Hence B is rowwise quasi-perfect. □

Since the 2×2 array $\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$ has Barker structure, we deduce that for $t = 2$ there exists each of the types of arrays listed in Theorem 3.7 for each $y \ge 0$, as previously constructed in [9] and [10].

3.3. Nonexistence results for small t. We now pursue the combinatorial constraints given by the balance properties for an $s \times t$ binary array with Barker structure where s, t are even. We show how these constraints can be combined with the simultaneous autocorrelation properties to establish the nonexistence of such arrays for $t = 4$ and $t = 8$ and, subject to additional constraints on the structure of A, for $t = 12$ and $t = 16$.

Suppose that $A = (a_{ij})$ is a positive $s \times t$ binary array with Barker structure where s, t are even and $t > 2$. Then, by Theorem 3.4, $s = t = 4r$ for some r, and A is balanced for some parameters (I, I', J, J'). From Theorem 3.6, A is simultaneously a PBA(t,t), a RQPBA(t,t), and a CQPBA(t,t). Define $B = (b_{ij})$ by $b_{ij} = a_{i,(j+J)\operatorname{mod}t}$ for all (i,j). Then it is straightforward to show from Definition 2 that B is balanced with parameters $(I, I', 0, J'')$, where $J'' = (J' - J) \bmod t$, and simple arguments show that B is simultaneously a PBA(t,t) and a RQPBA(t,t). Without loss of generality, we may take $0 < J'' \le t/2$ since $J \ne J'$, and, by Lemma 3.3(i), we may if necessary first transform A via $a'_{ij} = (-1)^{i+I}a_{ij}$ for all (i,j) (so that the values of J, J' are interchanged) while preserving the Barker structure. Next, define $C = (c_{ij})$ by $c_{ij} = b_{(i+I)\operatorname{mod}t,j}$ for all (i,j). Then C is balanced with parameters $(0, I'', 0, J'')$, where $I'' = (I' - I) \bmod t$, and C is a PBA(t,t). We may similarly take $0 < I'' \le t/2$. From balance property (ii), $I'' \equiv J''$ (mod 2).

We therefore use the following algorithm to search for a positive $s \times t$ binary array (a_{ij}) with Barker structure $(s = t = 4r)$.

Algorithm 1.

(A) For each pair (I'', J'') satisfying $0 < I'' \leq t/2$, $0 < J'' \leq t/2$, $I'' \equiv J''$ (mod 2), generate all possible $t \times t$ binary arrays (c_{ij}) that are balanced with parameters $(0, I'', 0, J'')$.

(B) Retain only those arrays (c_{ij}) that are perfect.

(C) For each $0 \leq I < t$ and each array (c_{ij}) remaining from Step (B), let $b_{ij} = c_{(i-I) \bmod t, j}$ for all (i, j) and retain only those arrays (b_{ij}) that are rowwise quasi-perfect.

(D) For each $0 \leq J < t$ and each array (b_{ij}) remaining from Step (C), let $a_{ij} = b_{i, (j-J) \bmod t}$ for all (i, j) and retain only those arrays (a_{ij}) that are columnwise quasi-perfect.

For each pair (I'', J''), Step (A) is implemented as the following branching algorithm, which fixes successive elements of the array so that at each stage no balance property is violated.

Algorithm 2.

(A) Set

$$a_{0j} = 1, a_{I''j} = (-1)^j \quad \text{for all } 0 \leq j < t,$$

$$a_{i0} = 1, a_{iJ''} = (-1)^i \quad \text{for all } 0 \leq i < t.$$

(B) If there exists an (i, j) for which a_{ij} is not yet set then branch, setting $a_{ij} = 1$ for one branch and $a_{ij} = -1$ for the other branch. Otherwise, output (a_{ij}) and terminate this branch.

(C) If either of the balance properties (vii) and (viii) determines consistently the value of one or more unset array elements, set these elements accordingly and go to Step (C). If, however, balance properties (vii) and (viii) lead to an inconsistent assignment of unset array elements, discard (a_{ij}) and terminate this branch. If no unset array elements are determined, go to Step (B).

For the case where $t = 4$, Algorithm 1 was implemented by hand, whereas, for the case where $t = 8$, computer search was used. In both cases, all arrays remaining after Step (B) had $I'' = J'' = t/2$, and no array remained after Step (C). Therefore, for $t = 4, 8$, there is no perfect and rowwise quasi-perfect balanced $t \times t$ binary array. This implies the following proposition.

PROPOSITION 3.8. *There is no* 4×4 *or* 8×8 *binary array with Barker structure.*

(Although for $t = 4, 8$ there does not exist a perfect and rowwise quasi-perfect balanced $t \times t$ binary array, we note that, for $t = 2^r$ and for each $r \geq 1$, there exists a perfect $t \times t$ binary array that is balanced with parameters $(0, t/2, 0, t/2)$. Such a family of arrays can be obtained using the recursive construction of Theorem 8 of [9].)

The cases where $t = 12, 16$ contain too many possibilities to allow exhaustive search using Algorithm 1, but we can prove nonexistence subject to additional constraints on the elements (a_{ij}).

Given $s \times t$ binary arrays $A = (a_{ij})$, $B = (b_{ij})$, define the *columnwise interleaving* of A with B to be the $2s \times t$ binary array $C = (c_{ij}) = \mathrm{ic}(A, B)$ given by

$$c_{i,2j} = a_{ij}, \quad c_{i,2j+1} = b_{ij} \qquad \text{for all } (i, j).$$

We observe that in the cases where $t = 4, 8$ each array remaining after Step (B) of Algorithm 1 is of *interleaved form*, namely, $\mathrm{ic}(\begin{bmatrix} X \\ X \end{bmatrix}, \begin{bmatrix} Y \\ -Y \end{bmatrix})$, for some component arrays

X, Y. If we assume A to have interleaved form, we can derive necessary conditions on the component arrays X, Y from the balance and autocorrelation properties of A.

DEFINITION 3. Let $A = (a_{ij})$ be an $s \times t$ binary array. Let (I, J, J') be a parameter set such that A has the following properties:

(i) $0 \leq I < s, 0 \leq J < t, 0 \leq J' < t$,

(ii) $a_{Ij} = 1$ for all $0 \leq j < t$,

(iii) $a_{iJ} = 1$ for all $0 \leq i < s$,

(iv) $a_{iJ'} = (-1)^{i+I}$ for all $0 \leq i < s$,

(v) $\sum_j a_{ij} = 0$ for all $i \neq I$,

(vi) $\sum_i a_{2i,j} = \sum_i a_{2i+1,j} = 0$ for all $j \neq J, J'$.

A is called *partially balanced* with parameters (I, J, J').

Note that an $s \times t$ binary array that is balanced for some parameters (I, I', J, J') is partially balanced with parameters (I, J, J').

THEOREM 3.9. *Let A be a positive $s \times t$ binary array with Barker structure where s, t are even and $t > 4$. Let A be of interleaved form with component arrays X and $Y = (y_{ij})$. Then $s = t = 8r$ for some r, X is partially balanced for some parameters (L, K, K'), and X is simultaneously a* PBA$(4r, 4r)$ *and a* CQPBA$(4r, 4r)$. *Also,*

$$y_{Lj} \quad = \quad k \quad \textit{for all } 0 \leq j < 4r,$$

where $k = 1$ or -1,

$$\sum_j y_{ij} \quad = \quad 0 \quad \textit{for all } i \neq L,$$

and Y is simultaneously a RQPBA$(4r, 4r)$ *and a* DQPBA$(4r, 4r)$.

Proof (outline). By Theorem 3.4, $s = t = 4r'$ for some r' and A is balanced for some parameters (I, I', J, J'). The partial balance properties of X and the constraints on Y are derived directly from the balance properties of A. The constraint $r' \equiv 0 \pmod 2$ follows from partial balance property (vi) of X, using an argument similar to that at the end of the proof of Theorem 3.4. By Theorem 3.6, A is simultaneously a PBA(t, t) and a CQPBA(t, t). The autocorrelation properties of X and Y are then given by the following partial converse to Theorems 2 and 4 of [9], which is straightforward to verify. Assuming that A has interleaved form with component arrays X and Y, if A is perfect, then X is perfect and Y is rowwise quasi-perfect, and, if A is columnwise quasi-perfect, then X is columnwise quasi-perfect and Y is doubly quasi-perfect. □

Assume that A has interleaved form. By Proposition 3.8 and Theorem 3.9, the smallest case is $t = 16$, for which the component array X has size 8×8. We see from Theorem 3.9 that the partial balance and autocorrelation properties required of X are weaker than those previously required of A. Nevertheless, a search procedure similar to that of Algorithms 1 and 2 shows that there is no perfect and columnwise quasi-perfect 8×8 partially balanced binary array. (In fact, the set of 8×8 perfect binary arrays that are partially balanced with parameters $(0, 0, K'')$, for each $0 < K'' \leq 4$, is no larger than that remaining after Step (B) of Algorithm 1, despite the relaxation in balance conditions.) We therefore have the following result.

PROPOSITION 3.10. *There is no* 16×16 *binary array of interleaved form with Barker structure.*

Finally, we drop the assumption that A has interleaved form. Consideration of the balance and autocorrelation properties that are required with respect to both the rows and the columns of A suggests that restriction of the search to symmetric arrays might be helpful. Indeed, in the cases where $t = 4, 8$ the set of arrays remaining after Step (B) of Algorithm 1 contains a large subset of symmetric arrays. It is straightforward to modify Algorithms 1 and 2 to search for a symmetric positive $4r \times 4r$ binary array with Barker structure. Computer search for the case where $t = 12$ shows that there is no symmetric perfect balanced 12×12 binary array. This implies the following result.

PROPOSITION 3.11. *There is no symmetric* 12×12 *binary array with Barker structure.*

We conclude this section by summarizing the main results for the case where s, t are even.

THEOREM 3.12. *Let A be an $s \times t$ binary array with Barker structure where s, t are even. Then $s = t$, A is simultaneously a* $\mathrm{PBA}(t,t)$, *a* $\mathrm{RQPBA}(t,t)$, *and a* $\mathrm{CQPBA}(t,t)$, *and there exists each of the following types of array, for each $y \geq 0$:*

$$\mathrm{PBA}(2^y t, 2^y t), \qquad \mathrm{PBA}(2^{y+2} t, 2^y t), \qquad \mathrm{DQPBA}(2^y t, 2^y t),$$
$$\mathrm{RQPBA}(2^y t, 2^{y+2} t), \quad \mathrm{DQPBA}(2^{y+1} t, 2^y t), \quad \mathrm{RQPBA}(2^{y+1} t, 2^{y+2} t),$$
$$\mathrm{RQPBA}(2^{y+1} t, 2^{y+4} t).$$

If $t > 2$, then $t \equiv 0 \pmod 4$ and $t \geq 12$. If A is a positive array, then A is balanced for some parameters (I, I', J, J'). If A is symmetric, then $t \geq 16$. If A is of interleaved form, then $t \equiv 0 \pmod 8$ and $t \geq 24$.

The nonexistence of a $\mathrm{PBA}(t,t)$ with $t \equiv 0 \pmod 4$ in the range $t \leq 100$ has been shown by McFarland for $t = 28, 44, 76, 92$ [12] and for $t = 84$ [13]. Therefore there does not exist a $t \times t$ binary array with Barker structure for these values of t.

4. The case where s even, t odd. In this section, we use methods similar to those of §3 to deduce restrictions on an $s \times t$ binary array with Barker structure, where s is even and t is odd.

From Lemma 2.1 and Definition 1(ii), the row sums (x_i) and column sums (y_j) satisfy

$$\sum_i x_i x_{i+u} = \begin{cases} 0 & \text{for all } u \text{ even and } u \neq 0, \\ k(u)\,t & \text{for all } u \text{ odd}, \\ st & \text{for } u = 0, \end{cases} \tag{16}$$

where $k(u) = 1$ or -1 for all $-s < u < s$, and $k(u) + k(u - s) = 0$ for all $0 < u < s$,

$$\sum_j y_j y_{j+v} = \begin{cases} 0 & \text{for all } v \neq 0, \\ st & \text{for } v = 0. \end{cases} \tag{17}$$

The solution of (17) is given by Lemma 3.1. We now show that, if (16) has a solution, then there exists a Barker sequence of length s. The reader is referred to [7] for a summary of results on Barker sequences. We note, in particular, that the only known even lengths for a Barker sequence are 2 and 4 and that any length $s > 13$ must satisfy $s = 4S^2$ for some odd S, where S is not a prime power and $S \geq 689$ [4], [7], [15], [16]. We also note that Ryser's conjecture [14] on cyclic difference sets, if true, would imply that there is no even length Barker sequence of length $s > 4$.

LEMMA 4.1. *Let $s \geq 2$ and $(x_i : 0 \leq i < s)$ be integers and let p be a prime. Let*

$$p \mid \sum_i x_i x_{i+u} \quad \textit{for all } 0 < u < s. \tag{18}$$

Then $p \nmid x_i$ *for at most one* $0 \le i < s$.

Proof. We use induction on s. The case where $s = 2$ is equivalent to

$$p \mid x_0 x_1 \quad \Rightarrow \quad p \mid x_0 \text{ or } p \mid x_1,$$

which is true because p is prime. Assume now that the result is true for the case $s - 1$. Taking $u = s - 1$ in (18), we have $p \mid x_0 x_{s-1}$. Since p is prime, without loss of generality,

$$p \mid x_{s-1}. \tag{19}$$

Then, from (18),

$$p \mid \sum_{i=0}^{s-u-2} x_i x_{i+u} \quad \text{for all } 0 < u < s - 1.$$

By the inductive hypothesis, $p \nmid x_i$ for at most one $0 \le i < s - 1$. Together with (19), this establishes the result for the case s, and the induction is complete. □

THEOREM 4.2. *Let* (x_i) *be the row sums of an* $s \times t$ *binary array where* s *is even and* t *is odd. Suppose that* (x_i) *satisfy* (16), *where* $k(u) = 1$ *or* -1 *for all* $-s < u < s$. *Then* $t = T^2$ *for some odd* T, *and there exists a Barker sequence* (z_i) *of length* s *satisfying* $x_i = Tz_i$ *for all* i.

Proof. Let p be a prime dividing t. From (16), we see that

$$p \mid \sum_i x_i x_{i+u} \quad \text{for all } 0 \le u < s. \tag{20}$$

Therefore, by Lemma 4.1, $p \nmid x_i$ for at most one $0 \le i < s$. Taking $u = 0$ in (20) then shows that $p \mid x_i$ for all $0 \le i < s$. Write $x_i = px'_i$ for all i and $t = p^2 t'$, so that (16) becomes

$$\sum_i x'_i x'_{i+u} = \begin{cases} 0 & \text{for all } u \text{ even and } u \ne 0, \\ k(u)\, t' & \text{for all } u \text{ odd}, \\ st' & \text{for } u = 0. \end{cases}$$

The equations for (x'_i) have the same form as (16), so we may apply the above argument repeatedly to each prime factor of t. This leads to

$$t = T^2 \quad \text{for some odd } T, \qquad x_i = Tz_i \quad \text{for all } i, \tag{21}$$

where (z_i) satisfies

$$\sum_i z_i z_{i+u} = \begin{cases} 0 & \text{for all } u \text{ even and } u \ne 0, \\ k(u) & \text{for all } u \text{ odd}, \\ s & \text{for } u = 0, \end{cases} \tag{22}$$

and where $k(u) = 1$ or -1 for all $-s < u < s$. Taking $u = 0$ in (22),

$$\sum_{i=0}^{s-1} z_i^2 = s. \tag{23}$$

Write the array as (a_{ij}). Now t is odd, and so, from (21),

$$z_i = x_i/T = \sum_{j=0}^{t-1} a_{ij}/T \neq 0 \quad \text{for all } 0 \leq i < s.$$

Therefore (23) implies that $z_i = 1$ or -1 for all $0 \leq i < s$. Hence, (z_i) is a binary sequence of length s satisfying (22), which are the defining equations for a Barker sequence of even length. □

COROLLARY 4.3. *Let A be an $s \times t$ binary array with Barker structure where s is even and t is odd. Let (x_i) and (y_j) be the row and column sums of A. Then $s = 4S^2$ and $t = T^2$ for some odd S, T where S is not a prime power, $2S > T$, and, if $S > 1$, then $S \geq 689$. Furthermore, there exists a Barker sequence (z_i) of length s satisfying*

$$x_i = Tz_i \quad \textit{for all } i.$$

For some $0 \leq J < t$,

$$y_j = \begin{cases} 0 & \textit{for all } j \neq J, \\ 2kST & \textit{for } j = J, \end{cases}$$

where $k = 1$ if A is positive and $k = -1$ otherwise.

Proof. (x_i) and (y_j) satisfy (16) and (17), respectively. Applying Lemma 3.1 to (17), $s \geq t$, and, for some $0 \leq J < t$,

$$y_j = \begin{cases} 0 & \text{for all } j \neq J, \\ \pm\sqrt{st} & \text{for } j = J. \end{cases} \tag{24}$$

Since s is even and t is odd, $s \geq t$ becomes

$$s > t. \tag{25}$$

Applying Theorem 4.2 to (16),

$$t = T^2 \tag{26}$$

for some odd T, and there exists a Barker sequence (z_i) of length s satisfying

$$x_i = Tz_i \quad \text{for all } i.$$

Using the quoted results on Barker sequences, either $s = 2$ (but then $t = 1$ from (25) and, trivially, no array A with the required properties exists), or else

$$s = 4S^2 \tag{27}$$

for some odd S, where S is not a prime power, and, if $S > 1$, then $S \geq 689$. Substitution of (26) and (27) into (24) and (25) gives the result. □

Taking the value $S = 1$ in Corollary 4.3 gives the parameter values for a Barker sequence of length 4. The existence of an array of the desired type with $S > 1$ implies the existence of an unknown Barker sequence.

Using a similar method to the proof of Theorem 3.2, we can obtain the following additional restrictions on (a_{ij}).

LEMMA 4.4. *Let $A = (a_{ij})$ be an $s \times t$ binary array with Barker structure where $s = 4S^2$ is even and $t = T^2$ is odd. Then, for some $0 \leq J < t$,*

$$\sum_i a_{2i,j} = \sum_i a_{2i+1,j} = 0 \quad \textit{for all } j \neq J,$$

$$\left\{ \sum_i a_{2i,J}, \sum_i a_{2i+1,J} \right\} = \{0, 2kST\},$$

$$\sum_j a_{ij} = Tz_i \quad \textit{for all } i, \tag{28}$$

where $k = 1$ if A is positive and $k = -1$ otherwise, and (z_i) is a Barker sequence of length s.

Let $B = (b_{ij})$ be the $s \times t$ binary array related to A by $b_{ij} = (-1)^j a_{ij}$ for all (i, j). If B has Barker structure, then constraints (28) *strengthen to*

$$\left\{ \sum_j a_{i,2j}, \sum_j a_{i,2j+1} \right\} \ = \ \{0, Tz_i\} \quad \textit{for all } i. \tag{29}$$

(The reason that (29) depends on B having Barker structure is that the value of $P^R(u, v)$ does not change in a simple way under the transformation $b_{ij} = (-1)^j a_{ij}$ when t is odd. If A is an $s \times t$ Barker array with $st > 2$, then the condition on B certainly holds.)

We finally show that the existence of an $s \times t$ binary array with Barker structure where s is even and t is odd implies the existence of certain perfect and quasi-perfect binary arrays.

THEOREM 4.5. *Let A be an $s \times t$ binary array with Barker structure where s is even and t is odd. Then A is simultaneously a* PBA(s, t) *and a* CQPBA(s, t), *and there exist a* PBA$(2s, 2t)$, *a* PBA$(s, 4t)$, *and a* CQPBA$(s, 2t)$.

Proof. By Lemma 3.5 and Definition 1(ii), A is simultaneously a PBA(s, t) and a CQPBA(s, t). Then, from Theorem 2 of [9], there exist a PBA$(2s, 2t)$ and a PBA$(s, 4t)$. Following the proof of Theorem 3.7, $[\ A \quad A\ ]$ is a CQPBA$(s, 2t)$. □

We note from Corollary 4.3 that $s = 4S^2$ for some odd S, so we cannot deduce the existence of a doubly quasi-perfect binary array from the existence of a CQPBA(s, t) or a CQPBA$(s, 2t)$ using Corollary 4 of [9].

We conclude this section by summarizing the main results for the case where s is even and t is odd.

THEOREM 4.6. *Let A be an $s \times t$ binary array with Barker structure, where s is even and t is odd. Then $s = 4S^2$ and $t = T^2$ for some odd S, T, where S is not a prime power, $2S > T$, and, if $S > 1$, then $S \geq 689$. A is simultaneously a* PBA(s, t) *and a* CQPBA(s, t), *and there exist a* PBA$(2s, 2t)$, *a* PBA$(s, 4t)$, *and a* CQPBA$(s, 2t)$. *There exists a Barker sequence of length s.*

We remark that in the case where s is even and t is odd, Alquaddoomi and Scholtz's conjecture on the nonexistence of Barker arrays with $s, t > 1$ and $(s, t) \neq (2, 2)$ would be implied by Ryser's conjecture applied to Barker sequences if the latter were true.

5. Comments. If A is an $s \times t$ Barker array with $st > 2$, then A has Barker structure. The results of Theorems 3.12 and 4.6 seem to provide good reason to doubt the existence of an $s \times t$ binary array with Barker structure where $st > 4$ is even. In the case where s, t are even, the simultaneous autocorrelation properties required appear highly restrictive. In the case where s is even, and t is odd, the existence of such an array would disprove Ryser's long-standing conjecture on cyclic difference sets.

The smallest even value of $st > 4$ for which $t > 1$ and the nonexistence of an $s \times t$ binary array with Barker structure has not been determined occurs for s, t even at $(s, t) = (12, 12)$ and for s even, t odd at $(s, t) = (4.689^2, 9)$.

We consider the case where s, t are odd in a further paper [8].

Acknowledgments. I am very grateful to Fred Piper, Chris Mitchell, and Peter Wild, of RHBNC, and to Sheelagh Lloyd and Miranda Mowbray, of HP Labs Bristol for their invaluable assistance and encouragement throughout my research and for contributing many helpful ideas and comments to this paper.

REFERENCES

[1] S. Alquaddoomi and R. A. Scholtz, *On the nonexistence of Barker arrays and related matters*, IEEE Trans. Inform. Theory, 35 (1989), pp. 1048–1057.

[2] M. F. M. Antweiler, L. Bömer, and H.-D. Lüke, *Perfect ternary arrays*, IEEE Trans. Inform. Theory, 36 (1990), pp. 696–705.

[3] T. Beth, D. Jungnickel, and H. Lenz, *Design Theory*, Cambridge University Press, Cambridge, UK, 1986.

[4] S. Eliahou, M. Kervaire, and B. Saffari, *A new restriction on the lengths of Golay complementary sequences*, J. Combin. Theory Ser. A, 55 (1990), pp. 49–59.

[5] D. R. Hughes and F. C. Piper, *Design Theory*, Cambridge University Press, Cambridge, UK, 1985.

[6] J. Jedwab, *Nonexistence results for Barker arrays*, in The Institute of Mathematics and Its Applications Conference Series (New Series) No. 33: Cryptography and Coding II, Oxford University Press, New York, 1992, pp. 121–126.

[7] J. Jedwab and S. Lloyd, *A note on the nonexistence of Barker sequences*, Designs, Codes Cryptography, 2 (1992), pp. 93–97.

[8] J. Jedwab, S. Lloyd, and M. Mowbray, *Barker arrays* II: *Odd number of elements*, SIAM J. Discrete Math., 6 (1993), this issue.

[9] J. Jedwab, C. Mitchell, F. Piper, and P. Wild, *Perfect binary arrays and difference sets*, Discrete Math., to appear.

[10] J. Jedwab and C. J. Mitchell, *Infinite families of quasiperfect and doubly quasiperfect binary arrays*, Electron. Lett., 26 (1990), pp. 294–295.

[11] H. D. Lüke, L. Bömer, and M. Antweiler, *Perfect binary arrays*, Signal Process., 17 (1989), pp. 69–80.

[12] R. L. McFarland, *Necessary conditions for Hadamard difference sets*, in The IMA Volumes in Mathematics and Its Applications, Vol. 21, Coding Theory and Design Theory, D. Ray-Chaudhuri, ed., Springer-Verlag, New York, 1990, pp. 257–272.

[13] ———, *Sub-difference sets of Hadamard difference sets*, J. Combin. Theory Ser. A, 54 (1990), pp. 112–122.

[14] H. J. Ryser, *Combinatorial Mathematics*, Carus Mathematical Monographs No. 14, Mathematical Association of America, Washington, DC, 1963.

[15] R. J. Turyn, *Character sums and difference sets*, Pacific J. Math., 15 (1965), pp. 319–346.

[16] ———, *Sequences with small correlation*, in Error Correcting Codes, H. B. Mann, ed., John Wiley, New York, 1968, pp. 195–228.

[17] P. Wild, *Infinite families of perfect binary arrays*, Electron. Lett., 24 (1988), pp. 845–847.

SIAM J. DISC. MATH.
Vol. 6, No. 2, pp. 309–328, May 1993

BARKER ARRAYS II: ODD NUMBER OF ELEMENTS*

JONATHAN JEDWAB†‡, SHEELAGH LLOYD†, AND MIRANDA MOWBRAY†

Abstract. A Barker array is a two-dimensional array with elements ±1 such that all out-of-phase aperiodic autocorrelation coefficients are 0, 1, or -1. No $s \times t$ Barker array with $s, t > 1$ and $(s, t) \neq (2, 2)$ is known, and it is conjectured that none exists. Nonexistence results for a class of arrays that includes Barker arrays have been previously given, in the case where st is even. We prove nonexistence results for this class of arrays in the case where st is odd, providing further support for the Barker array conjecture.

Key words. Barker array, aperiodic autocorrelation, binary array, nonexistence

AMS(MOS) subject classifications. primary 05B20, secondary 05B10

1. Introduction. In a previous paper [2], we defined binary arrays with *Barker structure*, a class that contains all $s \times t$ Barker arrays with $st > 2$, and proved restrictions on s, t for the case where st is even. In this paper, we present nonexistence results for the case where st is odd, providing further support for Alquaddoomi and Scholtz's conjecture [1]. We use the notation of [2].

2. Row and column sum equations. We first obtain equations in the row and column sums of an $s \times t$ binary array with Barker structure, where s, t are odd. Using Lemma 2.1 and Definition 1 (iii) of [2], we obtain the following lemma.

LEMMA 2.1. *Let A be an $s \times t$ binary array with Barker structure where s, t are odd. Let (x_i) and (y_j) be the row and column sums of A. Then each x_i and y_j is an odd integer, and*

$$(1) \qquad \sum_i x_i x_{i+u} = \begin{cases} kt & \text{for all } u \text{ even and } u \neq 0, \\ 0 & \text{for all } u \text{ odd}, \\ st + k(t-1) & \text{for } u = 0, \end{cases}$$

$$(2) \qquad \sum_j y_j y_{j+v} = \begin{cases} ks & \text{for all } v \text{ even and } v \neq 0, \\ 0 & \text{for all } v \text{ odd}, \\ st + k(s-1) & \text{for } v = 0, \end{cases}$$

where $k = 1$ or -1 and $k \equiv st \pmod 4$.

We derive all our results from an analysis of equations (1) and (2), although we do not find a general solution. Throughout, we consider solutions only to (1), combining conditions on s and t obtained from both equations at the end.

We can deduce from Lemma 2.1 an expression for the *imbalance* $\sum_i \sum_j a_{ij} \equiv \sum_i x_i$ of the array A.

LEMMA 2.2. *Let $s, t, (x_i : 0 \leq i < s)$ be integers satisfying* (1), *where $k = 1$ or -1 and $k \equiv st \pmod 4$. Then*

$$\left(\sum_i x_i\right)^2 = \begin{cases} 2st - 1 & \text{for } st \equiv 1 \pmod 4, \\ 1 & \text{for } st \equiv 3 \pmod 4. \end{cases}$$

*Received by the editors July 25, 1991; accepted for publication (in revised form) April 2, 1992.

†Hewlett-Packard Laboratories, Filton Road, Stoke Gifford, Bristol BS12 6QZ, United Kingdom.

‡This work was carried out while this author was studying for a Mathematics Ph.D. at Royal Holloway and Bedford New College, University of London.

Proof. We have

$$\left(\sum_i x_i\right)^2 = \sum_i x_i^2 + 2\sum_i \sum_{j>i} x_i x_j$$
$$= \sum_i x_i^2 + 2\sum_i \sum_{u>0} x_i x_{i+u},$$

putting $j = i + u$. Therefore

$$\left(\sum_i x_i\right)^2 = \sum_i x_i^2 + 2\sum_{u=1}^{s-1} \left(\sum_i x_i x_{i+u}\right)$$
$$= st + k(t-1) + 2kt(s-1)/2,$$

on substitution from (1). Hence

$$\left(\sum_i x_i\right)^2 = (k+1)st - k$$
$$= \begin{cases} 2st - 1 & \text{for } st \equiv 1 \pmod 4, \\ 1 & \text{for } st \equiv 3 \pmod 4, \end{cases}$$

using the given value for k. □

A consequence of Lemma 2.2 is that $2st - 1$ is a square when $st \equiv 1 \pmod 4$, as noted in Theorem 2.3 (ii) of [2].

In the case where $t = 1$, the possible values of s are determined by known results on Barker sequences.

THEOREM 2.3. *Let $s > 1$ be an odd integer and let $t = 1$. Then there exists an $s \times t$ binary array with Barker structure if and only if $s =$ 3, 5, 7, 11, or 13.*

Proof. Let A be an $s \times t$ binary array with Barker structure. Let (x_i) be the row sums of A. Since $t = 1$, (x_i) is a binary sequence, and, from (1),

$$\sum_i x_i x_{i+u} = \begin{cases} k & \text{for all } u \text{ even and } u \neq 0, \\ 0 & \text{for all } u \text{ odd}, \\ s & \text{for } u = 0, \end{cases} \tag{3}$$

where $k = 1$ or -1. Therefore (x_i) is a Barker sequence of odd length $s > 1$, and so (see [3]) $s = 3, 5, 7, 11,$ or 13.

The converse is implied by the existence of a Barker sequence with each of these lengths. □

We henceforth consider $s, t > 1$. Our results are all based on the observation that any prime dividing t divides exactly $s - 1$ of the (x_i).

LEMMA 2.4. *Let $s, t, (x_i : 0 \leq i < s)$ be integers satisfying (1), where $s \geq 2$ and $k = 1$ or -1. Let p be a prime dividing t. Then there exists a unique integer $0 \leq I < s$ such that*

(i) *$p \mid x_i$ if and only if $i \neq I$,*

(ii) *$x_I^2 \equiv -k \pmod p$.*

Proof. Let $s, t, (x_i : 0 \leq i < s)$ be integers satisfying (1). Since

$$p \mid t, \tag{4}$$

(1) shows that

$$p \mid \sum_i x_i x_{i+u} \quad \text{for all } 0 < u < s.$$

By Lemma 4.1 of [2], for some $0 \leq I < s$,

$$p \mid x_i \quad \text{for all } i \neq I. \tag{5}$$

Put $u = 0$ in (1),

$$\begin{aligned} \sum_i x_i^2 &= st + k(t-1) \\ &\equiv -k \pmod{p}, \end{aligned}$$

from (4). Then, from (5),

$$x_I^2 \equiv -k \pmod{p}.$$

This shows that $p \not| \, x_I$, because $k = 1$ or -1. Combining with (5),

$$p \mid x_i \quad \text{if and only if } i \neq I.$$

Given p and the (x_i), it is clear that I is unique. □

COROLLARY 2.5. *Let A be an $s \times t$ binary array with Barker structure where s, t are odd, $s > 1$, and $st \equiv 1 \pmod{4}$. Then $s \equiv t \equiv 1 \pmod{4}$, and each prime p dividing t satisfies $p \equiv 1 \pmod{4}$.*

Proof. Let (x_i) be the row sums of A. From Lemma 2.1, the (x_i) satisfy (1), where $k = 1$. Let p be a prime dividing t. Then, from Lemma 2.4 (ii), $x_I^2 \equiv -1 \pmod{p}$ for some $0 \leq I < s$. Now p is odd, since $p \mid t$, and so

$$p \equiv 1 \pmod{4}. \tag{6}$$

Since (6) holds for any prime p dividing t, we have $t \equiv 1 \pmod{4}$. Then, from $st \equiv 1 \pmod{4}$, we also have $s \equiv 1 \pmod{4}$. □

For a given prime p dividing t, the value of I is uniquely determined by the (x_i). In some cases, the values of only p, s, and t are sufficient to determine or restrict the value of I. This leads to restrictions on s and t and is the objective of our analysis.

We first show that $I \neq 0, s-1$ for any prime p.

LEMMA 2.6. *Let $s, t, (x_i : 0 \leq i < s)$ be integers satisfying (1), where $s > 1$ is odd, $x_i \neq 0$ for at least one odd i, and $k = 1$ or -1. Let p be a prime dividing t and let $0 \leq I < s$ be the unique integer such that $p \mid x_i$ if and only if $i \neq I$. Then $I \neq 0$, $s - 1$.*

Proof. The existence of I is given by Lemma 2.4 (i). Suppose, if possible, that $I = 0$ or $s - 1$. By symmetry, we may relabel the (x_i), if necessary, so that $I = s - 1$ and

$$p \mid x_i \quad \text{if and only if } i \neq s - 1. \tag{7}$$

Since $x_i \neq 0$ for at least one odd i, we may define r to be the largest integer for which

$$p^r \mid x_{2j-1} \quad \text{for all } 1 \leq j \leq (s-1)/2. \tag{8}$$

From (7), $r \geq 1$. Now, for any $1 \leq j \leq (s-1)/2$, put $u = s - 2j$ in (1) to obtain

$$\sum_{i=0}^{2j-2} x_i x_{i+s-2j} + x_{2j-1} x_{s-1} = 0. \tag{9}$$

Since s is odd, exactly one of $i, i+s-2j$ is even and the other is odd, for all i. Furthermore, from (7),

$$p \mid x_i \quad \text{for all even } i \neq s-1,$$

while, from (8),

$$p^r \mid x_i \quad \text{for all odd } i.$$

Therefore $p^{r+1} \mid \sum_{i=0}^{2j-2} x_i x_{i+s-2j}$, and then, from (9),

$$p^{r+1} \mid x_{2j-1} x_{s-1}.$$

Now p is prime and, by (7), $p \nmid x_{s-1}$, so we conclude that

$$p^{r+1} \mid x_{2j-1} \quad \text{for all } 1 \leq j \leq (s-1)/2.$$

This contradicts the maximality of r. □

We next fix the parity of I.

LEMMA 2.7. *Let $s, t, (x_i : 0 \leq i < s)$ be integers satisfying* (1), *where $s > 1$ is odd, x_i is odd for all i, and $k = 1$ or -1. Let p be a prime dividing t and let $0 \leq I < s$ be the unique integer such that $p \mid x_i$ if and only if $i \neq I$. Then $I \equiv (s-1)/2 \pmod 2$.*

Proof. Summing (1) over all odd values of u,

$$\sum_{v \geq 0} \sum_i x_i x_{i+2v+1} = 0.$$

Straightforward manipulation leads to

$$\sum_i x_{2i} \sum_j x_{2j+1} = 0.$$

Therefore either $\sum_i x_{2i} = 0$ or $\sum_j x_{2j+1} = 0$.

Suppose first that $\sum_i x_{2i} = 0$. Then I is odd, since $p \mid x_{2i}$ for all $2i \neq I$. Also, $\sum_i x_{2i}$ is the sum of exactly $(s+1)/2$ nonzero terms, each of which by hypothesis is odd, and so $(s+1)/2 \equiv 0 \pmod 2$. Therefore

$$I \text{ is odd and } (s+1)/2 \equiv 0 \pmod 2. \tag{10}$$

If instead we suppose that $\sum_j x_{2j+1} = 0$, then, by similar reasoning,

$$I \text{ is even and } (s-1)/2 \equiv 0 \pmod 2. \tag{11}$$

We combine (10) and (11) as

$$I \equiv (s-1)/2 \pmod 2.$$ □

We now prove two lemmas constraining the (x_i), given the value of I.

LEMMA 2.8. *Let $s, (x_i : 0 \leq i < s)$ be integers and let p be a prime such that $p^2 \mid \sum_i x_i x_{i+u}$ for all $0 < u < s$. Let $0 \leq I < s/2$ be an integer such that $p \mid x_i$ if and only if $i \neq I$. Then $p^2 \mid x_j$ for all $2I < j < s$.*

Proof. Let j satisfy

$$2I < j < s. \tag{12}$$

Put $u = j - I$ so that

$$p^2 \mid \sum_i x_i x_{i+j-I}. \tag{13}$$

Now

$$p \mid x_i \quad \text{for all } i \neq I, \tag{14}$$

and so p^2 divides each product $x_i x_{i+j-I}$ in (13) unless $i = I$ or $i + j - I = I$. From (12), however, $i + j - I > I$, and so p^2 divides each product $x_i x_{i+j-I}$ in (13) except $x_I x_j$. Therefore

$$p^2 \mid x_I x_j.$$

However, $p \not\mid x_I$ by (14), and so $p^2 \mid x_j$. □

DEFINITION. Let p be a prime and x, y be integers where $x \geq 0$. Let $p^x \mid y$ and $p^{x+1} \not\mid y$. Then p^x is said to *strictly divide* y, written $p^x \mid\mid y$.

LEMMA 2.9. *Let $s, (x_i : 0 \leq i < s)$ be integers and let p be a prime such that $p^2 \mid \sum_i x_i x_{i+u}$ for all $0 < u < s$. Let $0 \leq I < s$ be an integer such that $p \mid x_i$ if and only if $i \neq I$.*

(i) *Suppose that $p \mid\mid x_j$ for some $0 \leq j < s$. Then $0 \leq 2I - j < s$ and $p \mid\mid x_{2I-j}$.*

(ii) *Let j satisfy $0 \leq j < s$ and $0 \leq 2I - j < s$. Then $p^2 \mid x_j$ if and only if $p^2 \mid x_{2I-j}$.*

Proof. (i) Let $p \mid\mid x_j$ for some $0 \leq j < s$. By a similar argument to that used in the proof of Lemma 2.8, to avoid the false conclusion $p^2 \mid x_j$, we require that $i + j - I = I$ has a solution for some $0 \leq i < s$. Consequently, $0 \leq 2I - j < s$ and

$$p^2 \mid x_j + x_{2I-j}.$$

Then $p \mid\mid x_j$ if and only if $p \mid\mid x_{2I-j}$.

(ii) Let j satisfy $0 \leq j < s$ and $0 \leq 2I - j < s$. Then similar reasoning shows that

$$p^2 \mid x_j + x_{2I-j},$$

from which $p^2 \mid x_j$ if and only if $p^2 \mid x_{2I-j}$. □

The equation $x_0 x_{s-1} = \pm t$, obtained by putting $u = s - 1$ in (1), is of particular importance. Given a prime p dividing t, we are often able to obtain information about the (x_i) from the distribution of powers of p between x_0 and x_{s-1}.

LEMMA 2.10. *Let $s, t, (x_i : 0 \leq i < s)$ be integers satisfying (1), where $s > 1$ is odd, $x_i \neq 0$ for at least one odd i, and $k = 1$ or -1. Let p be a prime such that $p^\alpha \mid\mid t$ for some integer $\alpha \geq 1$. Then $\alpha \geq 2$ and $p^\gamma \mid\mid x_0$, $p^{\alpha-\gamma} \mid\mid x_{s-1}$ for some $0 < \gamma < \alpha$.*

Proof. Put $u = s - 1$ in (1),

$$x_0 x_{s-1} = \pm t. \tag{15}$$

Since $p^\alpha \mid\mid t$, we then have $p^\gamma \mid\mid x_0$, $p^{\alpha-\gamma} \mid\mid x_{s-1}$ for some $0 \leq \gamma \leq \alpha$. By Lemma 2.6, $p \mid x_0, x_{s-1}$. Therefore $0 < \gamma < \alpha$, and, from (15), $p^2 \mid t$. □

COROLLARY 2.11. *Let A be an $s \times t$ binary array with Barker structure where s, t are odd and $s > 1$. Then each prime p dividing t satisfies $p^2 \mid t$.*

Proof. Let (x_i) be the row sums of A. From Lemma 2.1, the $(x_i : 0 \leq i < s)$ are odd integers satisfying (1), where $k = 1$ or -1. Let p be a prime dividing t. Then $p^2 \mid t$ by Lemma 2.10. □

3. The case where $\gamma = 1$. In this section, we consider solutions to (1) for which $p \,||\, x_0$ and $p^{\alpha-1} \,||\, x_{s-1}$, where p is a prime. The value of I is then determined by s and α, which, in turn, gives restrictions on s in terms of α.

LEMMA 3.1. *Let $\alpha \geq 2$ and $s, (x_i : 0 \leq i < s)$ be integers and let p be a prime such that*

$$p^\alpha | \sum_i x_i x_{i+u} \quad \textit{for all } 0 < u < s, \tag{16}$$

$$p||x_0, \tag{17}$$

$$p^{\alpha-1}||x_{s-1}. \tag{18}$$

Let $0 \leq I < s$ be an integer such that

$$p \mid x_i \quad \textit{if and only if } i \neq I. \tag{19}$$

If $\alpha = 2$, then $I = (s-1)/2$. If $\alpha > 2$, then, for all $1 \leq \beta \leq \alpha - 2$,

$$(\beta + 1)I < s - 1, \tag{20}$$

$$p^{\alpha-\beta}|x_{s-1-j} \quad \textit{for all } 0 \leq j < \beta I, \tag{21}$$

$$p^{\alpha-\beta-1}||x_{s-1-\beta I}. \tag{22}$$

Proof. Since $\alpha \geq 2$, apply Lemma 2.9 (i) with $j = 0$ to give

$$2I < s, \tag{23}$$

$$p||x_{2I}. \tag{24}$$

We show, by induction on j, that

$$p^{\alpha-1} \mid x_{s-1-j} \quad \text{for all } 0 \leq j < I. \tag{25}$$

The case where $j = 0$ is given by (18). Assume that for some

$$1 \leq j < I, \tag{26}$$

$$p^{\alpha-1} \mid x_{s-1-k} \quad \text{for all } 0 \leq k < j. \tag{27}$$

Put $u = s - 1 - j$ in (16),

$$p^\alpha \mid \left(x_0 x_{s-1-j} + \sum_{i=1}^{j} x_i x_{i+s-1-j} \right). \tag{28}$$

Now, by (26), $j < I$ and so by (19), $p \mid x_i$ for all $1 \leq i \leq j$. Furthermore, by (27), $p^{\alpha-1} \mid x_{i+s-1-j}$ for all $1 \leq i \leq j$. Therefore $p^\alpha \mid \sum_{i=1}^{j} x_i x_{i+s-1-j}$, and so, by (28),

$$p^\alpha \mid x_0 x_{s-1-j}.$$

Using (17), we conclude that $p^{\alpha-1} \mid x_{s-1-j}$, completing the induction on j and proving (25).

Put $u = s - 1 - I$ in (16),

$$p^\alpha \mid \left(x_0 x_{s-1-I} + \sum_{i=1}^{I-1} x_i x_{i+s-1-I} + x_I x_{s-1} \right). \tag{29}$$

From (19) and (25), $p^\alpha \mid \sum_{i=1}^{I-1} x_i x_{i+s-1-I}$. Therefore, from (29),

$$p^\alpha \mid (x_0 x_{s-1-I} + x_I x_{s-1}). \tag{30}$$

From (19), $p \nmid x_I$, and so, by (18), $p^{\alpha-1} \parallel x_I x_{s-1}$. Therefore, from (30),

$$p^{\alpha-1} \parallel x_0 x_{s-1-I}. \tag{31}$$

In the case where $\alpha = 2$, we conclude from (17) and (31) that $p \nmid x_{s-1-I}$ and then from (19), $s - 1 - I = I$ or, equivalently, $I = (s-1)/2$, as required. For the rest of the proof, take $\alpha > 2$. Then (17) and (31) imply that

$$p^{\alpha-2} \parallel x_{s-1-I}, \tag{32}$$

and, since $\alpha > 2$ and $p \nmid x_I$, we deduce $s - 1 - I \neq I$. Combine this with (23) to give

$$2I < s - 1. \tag{33}$$

We now prove (20)–(22) for all $1 \leq \beta \leq \alpha - 2$ by induction on β. The case where $\beta = 1$ is given by (33), (25), and (32), respectively. Assume that, for some

$$2 \leq \beta \leq \alpha - 2, \tag{34}$$

(20)–(22) hold for $\beta - 1$, so that

$$\beta I < s - 1, \tag{35}$$

$$p^{\alpha-\beta+1} | x_{s-1-j} \quad \text{for all } 0 \leq j < (\beta - 1)I, \tag{36}$$

$$p^{\alpha-\beta} \| x_{s-1-(\beta-1)I}. \tag{37}$$

Then, to complete the induction on β, we must prove the following:

$$(\beta + 1)I < s - 1, \tag{38}$$

$$p^{\alpha-\beta} | x_{s-1-j} \quad \text{for all } 0 \leq j < \beta I, \tag{39}$$

$$p^{\alpha-\beta-1} \| x_{s-1-\beta I}. \tag{40}$$

We first prove (38). From (36) and (37),

$$p^{\alpha-\beta} \mid x_{s-1-j} \quad \text{for all } 0 \leq j \leq (\beta - 1)I. \tag{41}$$

By (34), $\alpha - \beta \geq 2$, and so, from (41),

$$p^2 \mid x_{s-1-j} \quad \text{for all } 0 \leq j \leq (\beta - 1)I.$$

Comparison with (24) shows that

$$2I < s - 1 - (\beta - 1)I,$$

which is equivalent to (38).

We next prove (39). From (36), it is sufficient to establish that

$$(42) \qquad p^{\alpha-\beta} \mid x_{s-1-j} \quad \text{for all } (\beta - 1)I \le j < \beta I,$$

which we prove by induction on j. The case where $j = (\beta - 1)I$ is given by (37). Assume that for some

$$(43) \qquad (\beta - 1)I + 1 \le j < \beta I,$$

$$(44) \qquad p^{\alpha-\beta} \mid x_{s-1-k} \quad \text{for all } (\beta - 1)I \le k < j.$$

Put $u = s - 1 - j$ in (16),

$$(45) \qquad p^{\alpha} \mid \sum_i x_i x_{i+s-1-j}.$$

By (34), $\beta \ge 2$, and so

$$(46) \qquad \alpha - \beta + 1 \le \alpha - 1.$$

Therefore, from (45),

$$(47) \qquad p^{\alpha-\beta+1} \mid \sum_i x_i x_{i+s-1-j}.$$

Now, by (43), $j \ge (\beta - 1)I + 1$, and, by (34), $\beta \ge 2$, so

$$(48) \qquad j \ge I + 1.$$

We can therefore write (47) in the form

$$(49) \qquad p^{\alpha-\beta+1} \mid \left(x_0 x_{s-1-j} + \sum_{1 \le i < I, I < i \le j} x_i x_{i+s-1-j} + x_I x_{I+s-1-j} \right).$$

By (41) and (44),

$$p^{\alpha-\beta} \mid x_{i+s-1-j} \quad \text{for all } 1 \le i \le j.$$

Together with (19), this implies that

$$p^{\alpha-\beta+1} \mid \sum_{1 \le i < I, I < i \le j} x_i x_{i+s-1-j},$$

and so, from (49),

$$(50) \qquad p^{\alpha-\beta+1} \mid (x_0 x_{s-1-j} + x_I x_{I+s-1-j}).$$

By (48), $j \ge I + 1$, and, by (43), $j < \beta I$, and so, by (36), $p^{\alpha-\beta+1} \mid x_{I+s-1-j}$. Therefore, from (50),

$$p^{\alpha-\beta+1} \mid x_0 x_{s-1-j}.$$

From (17), we conclude that

$$p^{\alpha-\beta} \mid x_{s-1-j},$$

completing the induction on j and proving (42) and hence (39).

We lastly prove (40). Put $u = s - 1 - \beta I$ in (16) and use (46) to show that

$$(51) \qquad p^{\alpha-\beta+1} \mid \left(x_0 x_{s-1-\beta I} + \sum_{1 \le i < I, I < i \le \beta I} x_i x_{i+s-1-\beta I} + x_I x_{s-1-(\beta-1)I} \right).$$

By (39), $p^{\alpha-\beta} \mid x_{i+s-1-\beta I}$ for all $1 \le i \le \beta I$. Together with (19), this implies that

$$p^{\alpha-\beta+1} \mid \sum_{1 \le i < I, I < i \le \beta I} x_i x_{i+s-1-\beta I},$$

and so, from (51),

$$(52) \qquad p^{\alpha-\beta+1} \mid (x_0 x_{s-1-\beta I} + x_I x_{s-1-(\beta-1)I}).$$

From (19), $p \nmid x_I$, and so, by (37), $p^{\alpha-\beta} \mid\mid x_I x_{s-1-(\beta-1)I}$. Therefore, from (52),

$$p^{\alpha-\beta} \mid\mid x_0 x_{s-1-\beta I}.$$

We conclude from (17) that

$$p^{\alpha-\beta-1} \mid\mid x_{s-1-\beta I},$$

which is (40).

This completes the induction on β, proving (20)–(22) for all $1 \le \beta \le \alpha - 2$. □

We now use Lemma 3.1 to prove the intended result of this section.

THEOREM 3.2. *Let $s, t, (x_i : 0 \le i < s)$ be integers satisfying* (1), *where $s > 1$ is odd and $k = 1$ or -1. Let p be a prime such that $p^\alpha \mid\mid t$ for some integer $\alpha \ge 2$, and $p \mid\mid x_0$. Then*

(i) $s \equiv 1 \pmod{\alpha}$,

(ii) *If x_i is odd for all i, then* $(s-1)(\alpha-2) \equiv 0 \pmod{4\alpha}$,

(iii) *$I = (s-1)/\alpha$ is the unique integer such that $p \mid x_i$ if and only if $i \ne I$,*

(iv) *For all* $2 \le r \le \alpha$,

$$(53) \qquad p^r \mid x_j \quad \textit{for all } j > rI,$$

$$(54) \qquad p^{r-1} \mid\mid x_{rI}.$$

Proof. By Lemma 2.4 (i), let I be the unique integer such that $p \mid x_i$ if and only if $i \ne I$. Take $u = s - 1$ in (1) to give $x_0 x_{s-1} = \pm t$. Then $p^\alpha \mid\mid t$ and $p \mid\mid x_0$ imply that

$$(55) \qquad p^{\alpha-1} \mid\mid x_{s-1},$$

and we may apply Lemma 3.1.

We first prove that

$$(56) \qquad I = (s-1)/\alpha.$$

If $\alpha = 2$, then (56) is given directly by Lemma 3.1. Suppose that $\alpha > 2$. Apply Lemma 3.1, taking $\beta = 1$ in (20) to give

$$(57) \qquad 2I < s - 1$$

and taking $\beta = \alpha - 2$ in (21) and (22) to give

$$p^2 | x_{s-1-j} \quad \text{for all } 0 \leq j < (\alpha - 2)I, \tag{58}$$

$$p || x_{s-1-(\alpha-2)I}. \tag{59}$$

From (57) and Lemma 2.8,

$$p^2 \,|\, x_j \quad \text{for all } 2I < j < s. \tag{60}$$

Put $j = 0$ in Lemma 2.9 (i) to show that

$$p \,||\, x_{2I}. \tag{61}$$

Comparing (58) and (59) with (60) and (61), we conclude that

$$2I = s - 1 - (\alpha - 2)I,$$

which is equivalent to $I = (s-1)/\alpha$. We have therefore proved (56) for $\alpha \geq 2$.

Now I is an integer and so from (56), $s \equiv 1 \pmod{\alpha}$. If x_i is odd for all i, then substitution of (56) into Lemma 2.7 gives

$$(s-1)/\alpha \equiv (s-1)/2 \pmod{2},$$

or, equivalently, $(s-1)(\alpha - 2) \equiv 0 \pmod{4\alpha}$.

Finally, apply Lemma 3.1 to show that (21) and (22) hold for $\alpha > 2$ and for all $1 \leq \beta \leq \alpha - 2$. Equations (21) and (22) also hold for $\beta = 0$, since then (21) is vacuous and (22) is given by (55). Combining ranges, (21) and (22) hold for

$$\alpha \geq 2 \quad \text{and for all } 0 \leq \beta \leq \alpha - 2.$$

The substitution $r = \alpha - \beta$, together with (56), then shows that (53) and (54) hold for $\alpha \geq 2$ and for all $2 \leq r \leq \alpha$. □

4. Nonexistence results for small α. In this section, we use the results of §§2 and 3 to obtain nonexistence results for small values of α_j, where $t = \prod_j p_j^{\alpha_j}$ for distinct primes p_j. We express the nonexistence results in the form of restrictions on s and t.

In each case, we state a theorem in terms of integers (x_i) and then a corollary in terms of an $s \times t$ binary array with Barker structure. Each corollary follows directly from the preceding theorem by letting (x_i) be the row sums of the array and using Lemma 2.1, as in the proof of Corollary 2.11.

We already know from Corollary 2.11 that $\alpha_j \geq 2$ for each j. The next case of interest is $\alpha_j = 2$ for all j. We first explore the case where $\alpha = 2$ for some prime p.

LEMMA 4.1. *Let $s, t, (x_i : 0 \leq i < s)$ be integers satisfying* (1), *where $s > 1$ is odd, $x_i \neq 0$ for at least one odd i, and $k = 1$ or -1. Let p be a prime such that*

$$p^2 \,||\, t. \tag{62}$$

Then $p \,||\, x_0, x_{s-1}$ and

$$p | x_i \quad \textit{if and only if } i \neq (s-1)/2,$$

$$p^2 | (x_j + x_{s-1-j}) \quad \textit{for all } 0 \leq j < (s-1)/2.$$

Proof. By Lemma 2.10, $p \,||\, x_0, x_{s-1}$. Then, by Theorem 3.2 (iii),

$$p \mid x_i \quad \text{if and only if } i \neq (s-1)/2. \tag{63}$$

We now show that

$$p^2 \mid (x_j + x_{s-1-j}) \quad \text{for all } 0 \leq j < (s-1)/2. \tag{64}$$

For any $0 \leq j < (s-1)/2$, put $u = (s-1)/2 - j$ in (1) and use (62) to show that

$$p^2 \mid \sum_i x_i x_{i+(s-1)/2-j}. \tag{65}$$

From (63), $p^2 \mid x_i x_{i+(s-1)/2-j}$, unless either $i = (s-1)/2$ or $i+(s-1)/2-j = (s-1)/2$, so from (65), $p^2 \mid x_{(s-1)/2}(x_j + x_{s-1-j})$. By (63), $p \nmid x_{(s-1)/2}$, and so $p^2 \mid (x_j + x_{s-1-j})$, proving (64). □

Subject to the condition $s > 3$, we now show that $\alpha_j > 2$ for some j and use Theorem 3.2 to restrict s when $\alpha_j = 3$ for some j. If $s = 3$, (1) has a solution in odd integers (x_i) with $k = -1$, namely, $t = r^2$ for some odd r and $(x_0, x_1, x_2) = (r, \pm 1, -r)$.

THEOREM 4.2. *Let $s, t, (x_i : 0 \leq i < s)$ be integers satisfying* (1), *where $s > 3$ and $t > 1$ are odd, $x_i \neq 0$ for all i, and $k = 1$ or -1. Then*

(i) *There exists a prime p such that $p^3 \mid t$,*

(ii) *If $q^3 \,||\, t$ for some prime q and x_i is odd for all i, then $s \equiv 1 \pmod{12}$.*

Proof. Since $t > 1$, we may write $t = \prod_j p_j^{\alpha_j}$, where the (p_j) are distinct primes and $\alpha_j \geq 1$ for all j. By Lemma 2.10, $\alpha_j \geq 2$ for all j. We seek a contradiction by supposing that $\alpha_j = 2$ for all j, so that

$$t = \prod_j p_j^2. \tag{66}$$

Applying Lemma 4.1,

$$p_j || x_0, x_{s-1} \quad \text{for all } j, \tag{67}$$

$$p_j | x_i \quad \text{if and only if } i \neq (s-1)/2, \text{ for all } j, \tag{68}$$

$$p_j^2 | (x_i + x_{s-1-i}) \quad \text{for all } 0 \leq i < (s-1)/2, \text{ for all } j. \tag{69}$$

Using (66), we deduce from (68) and (69) that

$$\sqrt{t} | x_i \quad \text{for all } i \neq (s-1)/2, \tag{70}$$

$$t | (x_i + x_{s-1-i}) \quad \text{for all } 0 \leq i < (s-1)/2. \tag{71}$$

Put $u = s - 1$ in (1) to obtain

$$x_0 x_{s-1} = \pm t. \tag{72}$$

Take $i = 0,\ s-1$ in (70) and compare with (72) to show that

$$x_0 = \pm x_{s-1}. \tag{73}$$

For any j, take $i = 0$ in (69),

$$p_j^2 \mid (x_0 + x_{s-1}). \tag{74}$$

Suppose, if possible, that $x_0 = x_{s-1}$. Then, from (74), $p_j^2 \mid 2x_0$, and so, since p_j is odd, $p_j^2 \mid x_0$. This contradicts (67), and so $x_0 \neq x_{s-1}$. From (73),

$$x_0 = -x_{s-1}. \tag{75}$$

Put $u = s-2$ in (1) and substitute from (75), $x_0(x_{s-2} - x_1) = 0$. By hypothesis, $x_0 \neq 0$, and so

$$x_1 = x_{s-2}. \tag{76}$$

Take $i = 1$ in (71) and substitute from (76) to give $t \mid 2x_1$. Then, since t is odd, $t \mid x_1$, and so from (76),

$$t \mid x_1, x_{s-2}. \tag{77}$$

We now force a contradiction by bounding $\sum_i x_i^2$ from below. By hypothesis, $1 < s-2$, and so x_1, x_{s-2} are not the same variable. Therefore we may write

$$\sum_i x_i^2 = x_1^2 + x_{s-2}^2 + x_{(s-1)/2}^2 + \sum_{i \neq 1, s-2, (s-1)/2} x_i^2.$$

Since $x_i \neq 0$ for all i, from (70) and (77), we then have

$$\sum_i x_i^2 \geq t^2 + t^2 + 1 + (s-3)t.$$

Comparing this bound with the value for the left side obtained by putting $u = 0$ in (1)

$$st + t - 1 \geq 2t^2 + 1 + (s-3)t,$$

which is equivalent to $(t-1)^2 \leq 0$. This contradicts $t > 1$ and so proves (i).

Suppose now that $q^3 \,||\, t$ for some prime q and x_i is odd for all i. From Lemma 2.10, either $q \,||\, x_0$ or $q \,||\, x_{s-1}$. We may therefore apply Theorem 3.2 (ii), reversing the order of the (x_i) if necessary, to show that $s - 1 \equiv 0 \pmod{12}$, proving (ii). □

COROLLARY 4.3. *Let A be an $s \times t$ binary array with Barker structure where $s > 3$ and $t > 1$ are odd. Then there exists a prime p such that $p^3 \mid t$. If $q^3 \,||\, t$ for some prime q, then $s \equiv 1 \pmod{12}$.*

Given that $\alpha_j \geq 2$ for all j and $\alpha_k > 2$ for some k, we next consider the case where $\alpha_k = 3$ for exactly one k and $\alpha_j = 2$ for all $j \neq k$.

THEOREM 4.4. *Let $s, t, (x_i : 0 \leq i < s)$ be integers satisfying (1), where $s > 3$ and $t > 1$ are odd, x_i is odd for all i, and $k = 1$ or -1. Let $t = q^3 \prod_j p_j^{\alpha_j}$, where $q, (p_j)$ are distinct primes and $\alpha_j \geq 1$ for all j. Then $\alpha_j > 2$ for some j.*

Proof. By Lemma 2.10, $\alpha_j \geq 2$ for all j. Suppose, for a contradiction, that $\alpha_j = 2$ for all j, so that

$$t = q^3 \prod_j p_j^2. \tag{78}$$

By Lemma 4.1,

$$p_j \mid x_i \quad \text{for all } i \neq (s-1)/2, \text{ for all } j. \tag{79}$$

By Lemma 2.10, either $q \,||\, x_0$ or $q \,||\, x_{s-1}$. We may assume, by reversing the order of the x_i if necessary, that $q \,||\, x_0$. Then, by Theorem 3.2 (iii) and (iv),

$$q^2 | x_i \quad \text{for all } 2(s-1)/3 < i \le s-1,$$

$$q | x_i \quad \text{for all } 0 \le i \le 2(s-1)/3,\ i \ne (s-1)/3.$$

Together with (79), this implies that

$$q^2 \prod_j p_j | x_i \quad \text{for all } 2(s-1)/3 < i \le s-1,$$

$$q \prod_j p_j | x_i \quad \text{for all } 0 \le i \le 2(s-1)/3,\ i \ne (s-1)/3, (s-1)/2,$$

$$\prod_j p_j | x_{(s-1)/3}.$$

Since $x_i \ne 0$ for all i, we can therefore bound $\sum_i x_i^2$ from below,

$$\sum_i x_i^2 \ge \frac{(s-1)q^4}{3} \prod_j p_j^2 + \left(\frac{2(s-1)}{3} - 2 \right) q^2 \prod_j p_j^2 + \prod_j p_j^2.$$

Comparing this bound with the value for the left-hand side obtained by putting $u = 0$ in (1) and making the substitution $\prod_j p_j^2 = t/q^3$ from (78),

$$s + 1 \ge \frac{(s-1)q}{3} + \frac{2s-8}{3q} + \frac{1}{q^3}.$$

Rearrangement gives

$$s \le \frac{q^4 + 3q^3 + 8q^2 - 3}{q^2(q-1)(q-2)},$$

which can be written as

$$s \le 1 + 3f(q), \tag{80}$$

where

$$f(q) = \frac{2q^3 + 2q^2 - 1}{q^2(q-1)(q-2)}.$$

It is easy to check that

$$\begin{aligned} f(q) - f(q+1) &= \frac{2q^4 + 12q^3 + 18q^2 + 4q - 1}{(q+1)^2 q^2 (q-1)(q-2)} \\ &> 0 \quad \text{for all } q \ge 3. \end{aligned} \tag{81}$$

Now q is an odd prime, and so $q \ge 3$. Therefore, from (80) and (81),

$$s \le 1 + 3f(3) = 77/6 < 13. \tag{82}$$

By Theorem 4.2 (ii), however, $s \equiv 1 \pmod{12}$, and, by hypothesis, $s > 3$. This contradicts (82), completing the proof. □

COROLLARY 4.5. *Let A be an $s \times t$ binary array with Barker structure, where $s > 3$ and $t > 1$ are odd. Let $t = q^3 \prod_j p_j^{\alpha_j}$, where q, (p_j) are distinct primes and $\alpha_j \geq 1$ for all j. Then $\alpha_j > 2$ for some j.*

The final case we consider is $\alpha_j = 2$ or 4 for all j. We first explore the case where $\alpha = 4$ for some prime p. By Lemma 2.10, $p^\gamma \,||\, x_0$, where $\gamma = 1, 2$, or 3. The values $\gamma = 1$ or 3 are covered by Theorem 3.2, leaving only the value $\gamma = 2$ to deal with.

LEMMA 4.6. *Let s, $(x_i : 0 \leq i < s)$ be integers and let p be an odd prime such that*

$$p^4 | \sum_i x_i x_{i+u} \quad \textit{for all } 0 < u < s, \tag{83}$$

$$p^2 || x_0, \tag{84}$$

$$p^2 || x_{s-1}. \tag{85}$$

Let $0 \leq I < s$ be an integer such that

$$p \,|\, x_i \quad \textit{if and only if } i \neq I. \tag{86}$$

Then

$$I = (s-1)/2, \tag{87}$$

$$p^2 | x_j, x_{s-1-j} \quad \textit{for all } 0 \leq j \leq \lfloor (s-3)/4 \rfloor. \tag{88}$$

If also

$$x_0 = -x_{s-1}, \tag{89}$$

then

$$p^2 \,|\, x_j \quad \textit{for all } j \neq (s-1)/2. \tag{90}$$

Proof. We may assume, by reversing the order of the (x_i) if necessary, that

$$I \leq (s-1)/2. \tag{91}$$

We show, by induction on j, that

$$p^2 \,|\, x_j, x_{s-1-j} \quad \text{for all } 0 \leq j \leq \lfloor (I-1)/2 \rfloor. \tag{92}$$

The case where $j = 0$ is given by (84) and (85). Assume that for some

$$1 \leq j \leq \lfloor (I-1)/2 \rfloor, \tag{93}$$

$$p^2 \,|\, x_k, x_{s-1-k} \quad \text{for all } 0 \leq k < j. \tag{94}$$

Put $u = s - 1 - 2j$ in (83), showing that

$$p^3 \,|\, \left(\sum_{i=0}^{j-1} x_i x_{i+s-1-2j} + x_j x_{s-1-j} + \sum_{i=j+1}^{2j} x_i x_{i+s-1-2j} \right). \tag{95}$$

By (94), $p^2 \mid x_i$ for all $0 \le i \le j-1$. By (91) and (93), $s-1-2j > I$ and so by (86), $p \mid x_{i+s-1-2j}$ for all $0 \le i \le j-1$. Therefore $p^3 \mid \sum_{i=0}^{j-1} x_i x_{i+s-1-2j}$. Similarly, $p^3 \mid \sum_{i=j+1}^{2j} x_i x_{i+s-1-2j}$. Then, from (95),

$$p^3 \mid x_j x_{s-1-j},$$

and so

$$\text{either } p^2 \mid x_j \quad \text{or} \quad p^2 \mid x_{s-1-j}. \tag{96}$$

Now take $u = s-1-j$ in (83),

$$p^4 \mid \left(x_0 x_{s-1-j} + \sum_{i=1}^{j-1} x_i x_{i+s-1-j} + x_j x_{s-1} \right). \tag{97}$$

By (94), $p^2 \mid x_i, x_{i+s-1-j}$ for all $1 \le i \le j-1$, and so $p^4 \mid \sum_{i=1}^{j-1} x_i x_{i+s-1-j}$. Therefore, from (97),

$$p^4 \mid (x_0 x_{s-1-j} + x_j x_{s-1}).$$

Then, from (84) and (85),

$$p^2 \mid x_j \quad \text{if and only if } p^2 \mid x_{s-1-j}.$$

Therefore, using (96),

$$p^2 \mid x_j, x_{s-1-j},$$

completing the induction on j and proving (92).

Put $u = s-1-I$ in (83) to show that

$$p^3 \mid \left(\sum_{i=0}^{I-1} x_i x_{i+s-1-I} + x_I x_{s-1} \right). \tag{98}$$

We next prove (87), considering separately the cases I even and I odd.

Suppose first that I is odd, so that (92) and (98) become

$$p^2 | x_j, x_{s-1-j} \quad \text{for all } 0 \le j \le (I-1)/2, \tag{99}$$

$$p^3 | \left(\sum_{i=0}^{(I-1)/2} x_i x_{i+s-1-I} + \sum_{i=(I+1)/2}^{I-1} x_i x_{i+s-1-I} + x_I x_{s-1} \right). \tag{100}$$

From (99), $p^2 \mid x_{i+s-1-I}$ for all $(I+1)/2 \le i \le I-1$, and so, by (86), $p^3 \mid \sum_{i=(I+1)/2}^{I-1} x_i x_{i+s-1-I}$. Therefore, from (100),

$$p^3 \mid \left(\sum_{i=0}^{(I-1)/2} x_i x_{i+s-1-I} + x_I x_{s-1} \right). \tag{101}$$

From (86), $p \nmid x_I$, and so, by (85), $p^2 \parallel x_I x_{s-1}$. Therefore, from (101),

$$p^2 \parallel \sum_{i=0}^{(I-1)/2} x_i x_{i+s-1-I}. \tag{102}$$

Now, from (99), $p^2 \mid x_i$ for all $0 \le i \le (I-1)/2$. Suppose, if possible, that $s-1-I > I$. Then, by (86), $p \mid x_{i+s-1-I}$ for all $0 \le i \le (I-1)/2$, and so $p^3 \mid \sum_{i=0}^{(I-1)/2} x_i x_{i+s-1-I}$, contradicting (102). Therefore $s-1-I \le I$, which combines with (91) to give $I = (s-1)/2$.

Suppose instead that I is even, so that (92) and (98) become

$$(103) \qquad p^2 | x_j, x_{s-1-j} \quad \text{for all } 0 \le j \le I/2 - 1,$$

$$(104) \quad p^3 \left| \left(\sum_{i=0}^{I/2-1} x_i x_{i+s-1-I} + x_{I/2} x_{s-1-I/2} + \sum_{i=I/2+1}^{I-1} x_i x_{i+s-1-I} + x_I x_{s-1} \right) \right. .$$

Suppose, if possible, that

$$(105) \qquad s - 1 - I > I.$$

From (103), $p^2 \mid x_i$ for all $0 \le i \le I/2 - 1$ and $p^2 \mid x_{i+s-1-I}$ for all $I/2+1 \le i \le I-1$. Hence, by (86) and (105), $p^3 \mid (\sum_{i=0}^{I/2-1} x_i x_{i+s-1-I} + \sum_{i=I/2+1}^{I-1} x_i x_{i+s-1-I})$, and so, from (104),

$$p^3 \mid (x_{I/2} x_{s-1-I/2} + x_I x_{s-1}).$$

As before, $p^2 \parallel x_I x_{s-1}$, and therefore

$$p^2 \parallel x_{I/2} x_{s-1-I/2}.$$

It follows from (86) and (91) that

$$(106) \qquad p \| x_{I/2},$$

$$(107) \qquad p \| x_{s-1-I/2}.$$

Apply Lemma 2.9 (ii) for all $0 \le j < I/2$ so that, from (103), $p^2 \mid x_j$ for all $3I/2 < j \le 2I$. Apply Lemma 2.8 to show that $p^2 \mid x_j$ for all $2I < j < s$. Combine to give

$$(108) \qquad p^2 \mid x_j \quad \text{for all } 3I/2 < j < s.$$

Apply Lemma 2.9 (i) with $j = I/2$ so that, from (106),

$$(109) \qquad p \parallel x_{3I/2}.$$

Comparing (103) and (107) with (108) and (109), we conclude that

$$s - 1 - I/2 = 3I/2,$$

contradicting (105). Therefore $s-1-I \le I$, which combines with (91) to give (87).

We therefore have $I = (s-1)/2$, regardless of whether I is even or odd, and the form (88) is obtained by substituting for I into (92).

Suppose finally that (89) holds. By (87), the form (90) is equivalent to

$$(110) \qquad p^2 \mid x_j, x_{s-1-j} \quad \text{for all } 0 \le j < I,$$

which we prove by induction on j. The case where $j = 0$ is given by (88). Assume that, for some

$$0 < j < I, \tag{111}$$

$$p^2 \mid x_k, x_{s-1-k} \quad \text{for all } 0 \leq k < j. \tag{112}$$

Put $u = s - 1 - j$ in (83),

$$p^4 \mid \left(x_0 x_{s-1-j} + \sum_{i=1}^{j-1} x_i x_{i+s-1-j} + x_j x_{s-1} \right). \tag{113}$$

By (112), $p^4 \mid \sum_{i=1}^{j-1} x_i x_{i+s-1-j}$, and then substitution from (89) into (113) gives

$$p^4 \mid x_{s-1}(x_j - x_{s-1-j}).$$

Then, from (85),

$$p^2 \mid (x_j - x_{s-1-j}). \tag{114}$$

By (111), $I - j > 0$ so we may take $u = I - j$ in (83) and use (87) to show that

$$p^2 \mid \left(\sum_{i \neq j, I} x_i x_{i+I-j} + x_j x_I + x_I x_{s-1-j} \right).$$

From (86), $p^2 \mid \sum_{i \neq j, I} x_i x_{i+I-j}$, and so

$$p^2 \mid x_I (x_j + x_{s-1-j}).$$

However, $p \not| \, x_I$ by (86), and therefore

$$p^2 \mid (x_j + x_{s-1-j}). \tag{115}$$

Summing (114) and (115), $p^2 \mid 2x_j$ and, since p is odd, $p^2 \mid x_j$. Therefore, from (115),

$$p^2 \mid x_j, x_{s-1-j},$$

completing the induction on j and proving (110) and therefore (90). □

We can now treat the case where $\alpha_j = 2$ or 4 for all j.

THEOREM 4.7. *Let* $s, t, (x_i : 0 \leq i < s)$ *be integers satisfying* (1), *where* $s > 3$ *and* $t > 1$ *are odd*, $x_i \neq 0$ *for all* i, *and* $k = 1$ *or* -1. *Let*

$$t = \left(\prod_j p_j^2 \right) \left(\prod_k q_k^4 \right), \tag{116}$$

where the (p_j, q_k) *are distinct primes. Then* $s \equiv 1 \pmod 4$.

Proof. Suppose, for a contradiction, that

$$s \equiv 3 \pmod 4. \tag{117}$$

Applying Lemma 4.1,

$$p_j \| x_0, x_{s-1} \quad \text{for all } j, \tag{118}$$

$$p_j | x_i \quad \text{if and only if } i \neq (s-1)/2, \text{ for all } j. \tag{119}$$

By Lemma 2.10,

$$q_k^{\gamma_k} \| x_0, \; q_k^{4-\gamma_k} \| x_{s-1} \quad \text{for all } k, \tag{120}$$

where each $\gamma_k = 1$, 2, or 3. By Theorem 3.2 (i), if $\gamma_k = 1$ or 3 for any k, then $s \equiv 1 \pmod 4$, contradicting (117), and so, from (120),

$$q_k^2 \| x_0, x_{s-1} \quad \text{for all } k. \tag{121}$$

Then, by Lemmas 2.4 and 4.6,

$$q_k \,|\, x_i \quad \text{if and only if } i \neq (s-1)/2, \text{ for all } k. \tag{122}$$

Using (116), we deduce from (118) and (121) that

$$\sqrt{t} \,|\, x_0, x_{s-1}. \tag{123}$$

Put $u = s-1$ in (1), giving $x_0 x_{s-1} = \pm t$. Then (123) implies that

$$x_0 = \pm x_{s-1}. \tag{124}$$

Now from (116) and Theorem 4.2 (i), there exists some k such that $q_k^4 \,|\, t$. For any such k, take $u = (s-1)/2$ in (1) and use (117) to write

$$q_k^3 \,|\, (x_0 x_{(s-1)/2} + \sum_{i=1}^{(s-3)/4} x_i x_{i+(s-1)/2} + \sum_{i=(s+1)/4}^{(s-3)/2} x_i x_{i+(s-1)/2} + x_{(s-1)/2} x_{s-1}). \tag{125}$$

Applying Lemma 4.6,

$$q_k^2 \,|\, x_i, x_{s-1-i} \quad \text{for all } 0 \leq i \leq (s-3)/4,$$

which, together with (122), implies that $q_k^3 \,|\, (\sum_{i=1}^{(s-3)/4} x_i x_{i+(s-1)/2} + \sum_{i=(s+1)/4}^{(s-3)/2} x_i x_{i+(s-1)/2})$. Therefore, from (125),

$$q_k^3 \,|\, x_{(s-1)/2}(x_0 + x_{s-1}),$$

and since, by (122), $q_k \nmid x_{(s-1)/2}$,

$$q_k^3 \,|\, (x_0 + x_{s-1}). \tag{126}$$

Suppose, if possible, that $x_0 = x_{s-1}$. Then, from (126), $q_k^3 \,|\, 2x_0$, and so, since q_k is odd, $q_k^3 \,|\, x_0$. This contradicts (121), and so $x_0 \neq x_{s-1}$. From (124),

$$x_0 = -x_{s-1}. \tag{127}$$

Now we can apply Lemma 4.6 to obtain

$$q_k^2 \,|\, x_i \quad \text{for all } i \neq (s-1)/2, \text{ for all } k. \tag{128}$$

Together with (116) and (119), this gives

$$\sqrt{t}\,|\,x_i \quad \text{for all } i \neq (s-1)/2. \tag{129}$$

Take $u = s-2$ in (1) and substitute from (127),

$$x_0(x_{s-2} - x_1) = 0.$$

Since $x_0 \neq 0$,

$$x_1 = x_{s-2}. \tag{130}$$

Next, take $u = (s-3)/2$ in (1),

$$t \,\Bigg|\, \left(x_1 x_{(s-1)/2} + \sum_{i \neq 1,(s-1)/2} x_i x_{i+(s-3)/2} + x_{(s-1)/2} x_{s-2} \right). \tag{131}$$

Now, from (116), $p_j^2 \,|\, t$ for all j, and so (119) and (131) imply that $p_j^2 \,|\, (x_1 + x_{s-2})$ for all j. Then (130) gives $p_j^2 \,|\, x_1, x_{s-2}$ for all j. Similarly, $q_k^4 \,|\, t$ for all k, and so (128), (130), and (131) imply that $q_k^4 \,|\, x_1, x_{s-2}$. Combining and using (116),

$$t \,|\, x_1, x_{s-2}. \tag{132}$$

We now proceed as in the proof of Theorem 4.2 (i), using (129) and (132) to show that $(t-1)^2 \leq 0$, contradicting $t > 1$. Therefore we conclude that (117) is false, and hence $s \equiv 1 \pmod 4$. □

COROLLARY 4.8. *Let A be an $s \times t$ binary array with Barker structure where $s > 3$ and $t > 1$ are odd. Let $t = \prod_j p_j^{\alpha_j}$, where the (p_j) are distinct primes and $\alpha_j = 2$ or 4 for all j. Then $st \equiv 1 \pmod 4$.*

Proof. By Theorem 4.7, $s \equiv 1 \pmod 4$. Since t is the product of even powers of primes, $t \equiv 1 \pmod 4$. Therefore $st \equiv 1 \pmod 4$. □

This completes our analysis for small values of α_j.

The nonexistence results in this paper, for $s \times t$ binary arrays with Barker structure where s, t are odd, are all based on (1). Using (2) as well as (1), we may interchange s and t in each of our results. In particular, we can exclude the case where $s = 3$ and $t > 1$ by Corollary 2.11. We conclude this section by summarizing the nonexistence results arising from both (1) and (2), although for clarity we mostly do not repeat results with s and t interchanged.

THEOREM 4.9. *Let $A = (a_{ij})$ be an $s \times t$ binary array with Barker structure where s, t are odd and $s > 1$. If $st \equiv 1 \pmod 4$, then $2st - 1 = (\sum_i \sum_j a_{ij})^2$, $s \equiv t \equiv 1 \pmod 4$ and $p \equiv 1 \pmod 4$ for each prime p dividing s or t. If $t = 1$, then $s = 3, 5, 7, 11$, or 13. Otherwise, if $t > 1$, write $t = \prod_j p_j^{\alpha_j}$, where the (p_j) are distinct primes and $\alpha_j \geq 1$ for all j. Then*

(i) *$\alpha_j \geq 2$ for all j,*
(ii) *$\alpha_k > 2$ for some k,*
(iii) *If $\alpha_k = 3$ for some k, then $s \equiv 1 \pmod{12}$,*
(iv) *If $\alpha_k = 3$ for some k, then $\alpha_j > 2$ for some $j \neq k$,*
(v) *If $\alpha_j = 2$ or 4 for all j, then $st \equiv 1 \pmod 4$.*

5. Comments. The smallest odd value of $st > 13$ for which the nonexistence of an $s \times t$ binary array with Barker structure is not determined by Theorem 4.9 occurs at $\{s, t\} = \{3^5, 3^6\}$. The existence of such an array implies the existence of a (177147, 88573, 44286)-difference set in $\mathbb{Z}_{243} \times \mathbb{Z}_{729}$ [2].

In our opinion, the apparent scarcity of solutions to the necessary equations, both in the row and column sums, provides good reason to doubt the existence of an $s \times t$ binary array with Barker structure where $st > 13$ is odd.

Acknowledgments. The first author is very grateful to Fred Piper, Chris Mitchell, and Peter Wild of RHBNC for their invaluable assistance and encouragement throughout his research.

REFERENCES

[1] S. Alquaddoomi and R. A. Scholtz, *On the nonexistence of Barker arrays and related matters*, IEEE Trans. Inform. Theory, 35 (1989), pp. 1048–1057.

[2] J. Jedwab, *Barker arrays* I: *Even number of elements*, SIAM J. Discrete Math., 6 (1993), in this issue.

[3] R. Turyn and J. Storer, *On binary sequences*, Proc. Amer. Math. Soc., 12 (1961), pp. 394–399.

SIAM J. DISC. MATH.
Vol. 6, No. 2, pp. 329–333, May 1993

ON THE ENUMERATION OF STEINER-TREE TOPOLOGIES FOR THE POINTS ON A CIRCLE*

KENNETH R. VOLLMAR† AND YANJUN ZHANG†

Abstract. The problem of counting the number $F(n, s)$ of Steiner-tree topologies with s Steiner points for n points on a circle is considered. The paper shows that $F(n, s)$ is closely related to the number $F^*(n, s)$ of Steiner-tree topologies with s Steiner points for n arbitrary points, which was studied by E. N. Gilbert and H. O. Pollack in their seminal paper on Steiner minimal trees [*SIAM J. Appl. Math.*, 16 (1968), pp. 1–29]. Specifically, it is shown that $F(n, s) = F^*(n, s)/R(n)$, where $R(n) = (n-1)!/2^{n-2}$, independent of s.

Key words. Steiner-tree topology, enumeration, convex polygon

1. Introduction. In their seminal paper [GP68], Gilbert and Pollack studied the problem of finding a shortest network, called the *Steiner minimal tree* (SMT), that connects a set of given points on the Euclidean plane. An SMT for a set P of given points may contain vertices not in P. These vertices are called *Steiner points*. The points in P are called *regular points*. We can show that an SMT must satisfy the conditions that (i) any two edges meet at an angle of at least 120°, and (ii) every Steiner point has degree exactly 3. These conditions imply that all leaves of an SMT are regular points and that every Steiner point is incident to exactly three edges, any two of which must meet at an angle of 120°. It is possible, though only by rarest accident, that a regular point of an SMT is incident to three edges. Computing an SMT for a set of points at general positions was shown a decade later to be NP-hard computationally [GGJ77]. An expository account on SMT can be found in a recent article [BG89].

In an attempt to explore enumerative methods for computing SMT, Gilbert and Pollack examined, in the same paper, the problem of counting the number of possible topologies corresponding to the underlying graph of an SMT. They considered the connected graphs, each of which we call a *Steiner-tree topology*, on n *labeled* regular vertices and s *unlabeled* Steiner vertices in which the degree of each regular vertex is at most 2 and the degree of each Steiner vertex is exactly 3. They ruled out regular vertices of degree 3 by citing the rare occurrence of such regular points in an SMT. Two Steiner-tree topologies of n regular vertices and s Steiner vertices are different if their graph adjacency matrices are not identical under any permutation of the Steiner vertices, as the Steiner vertices are unlabeled. We can easily show by induction that $n \geq s + 2$. Let $F^*(n, s)$ be the number of different Steiner-tree topologies with n labeled regular vertices and s unlabeled Steiner vertices. Gilbert and Pollack showed that, for $n \geq s+2$,

$$F^*(n, s) = \binom{n}{s+2} \frac{(n+s-2)!}{2^s s!}. \tag{1}$$

In this paper, we report an interesting numerical relation between $F^*(n, s)$ and the number of Steiner-tree topologies of a restricted type. We consider the problem of counting the number of different Steiner-tree topologies corresponding to the restricted SMT problem, where the regular points are the vertices of a convex polygon, or points on a circle as the topologies are concerned. Note that the degree of a regular point of a SMT in this case must be 1 or 2, not 3. To reflect the restraints on the resulting Steiner-tree topologies, we require that the corresponding Steiner-tree topology be a connected

*Received by the editors November 11, 1991; accepted for publication (in revised form) May 5, 1992.

†Department of Computer Science and Engineering, Southern Methodist University, Dallas, Texas 75275.

planar graph *inside* the circle of its regular points when its labeled regular vertices are arranged in a circle in increasing order. Let $F(n, s)$ be the number of such Steiner-tree topologies on n labeled regular vertices and s unlabeled Steiner vertices. The main result of this paper is the following theorem, which we prove in the next section.

THEOREM 1. *For* $n \geq s + 2$,

$$F(n, s) = \frac{F^*(n, s)}{R(n)}, \tag{2}$$

where

$$R(n) = \frac{(n-1)!}{2^{n-2}}, \tag{3}$$

independent of s.

COROLLARY 1. *For* $n \geq s + 2$,

$$F(n, s) = \binom{n}{s+2} \frac{2^{n-s-2}(n+s-2)!}{s!(n-1)!}.$$

There is a geometric view of $F^*(n, s)$ and $F(n, s)$. Arrange the labeled regular vertices of an arbitrary Steiner-tree topology on a circle in the increasing order and draw the topology inside the circle of the regular vertices. This may result in "edge crossing" in the layout of the topology. The Steiner-tree topologies we considered are precisely those that result in no edge crossing. Theorem 1 states that the restriction of no edge crossing in the layout of the topology when the regular vertices are arranged along a circle reduces the number of Steiner-tree topologies of n regular vertices and s Steiner vertices exactly by a factor of $R(n) = 2^{-(n-2)}(n-1)!$, independent of s. Our proof of Theorem 1, however, provides no intuitive explanations for this geometric view.

It is an open question whether the problem of computing an SMT for the vertex set of a convex polygon can be solved exactly in polynomial time. The best known result is a fully-polynomial approximation scheme given by Provan [Pro88]. We may attempt to use an enumerative method of some sort to find the exact solution. Then the quantity $F(n, s)$ gives the possible number of Steiner-tree topologies, exponential in n, that may have to be considered. In fact, the relation between $F(n, s)$ and $F^*(n, s)$ was first observed when the first author used a computer to generate all possible Steiner-tree topologies of the SMT for the vertices of a convex polygon with a small number of vertices.

Table 1 shows all $F(n, s)$'s for $n \leq 10$, computed by Corollary 1, in which $F(n, s)$ exhibits a rapid growth as n increases.

2. The proof. In this section, we prove Theorem 1. We prove a sequence of lemmas, of which the first two are central to the proof of the theorem. For brevity, we call a Steiner-tree topology simply a *topology*.

Let $L(n, s)$ be the set of topologies with n labeled regular vertices and s Steiner vertices that are connected planar graphs when the n labeled regular vertices are arranged on a circle in the increasing order. Thus, $F(n, s) = |L(n, s)|$. Let v be a fixed regular vertex and let $L_v(n, s)$ be the set of topologies in $L(n, s)$ in which v is a leaf. Let $\overline{L}_v(n, s) = L(n, s) \setminus L_v(n, s)$, the set of topologies in $L(n, s)$ in which v is not a leaf. Let $F_v(n, s) = |L_v(n, s)|$ and $\overline{F}_v(n, s) = |\overline{L}_v(n, s)| = F(n, s) - F_v(n, s)$. By symmetry, $F_v(n, s) = F_u(n, s)$ for two regular vertices v and u.

LEMMA 1. *For* $n \geq s + 2$,

$$F_v(n, s) = \frac{s+2}{n} F(n, s). \tag{4}$$

TABLE 1
$F(n, s)$ *for* $n \leq 10$.

	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$	$n=9$	$n=10$
$s=0$	1	3	8	20	48	112	256	576	1,280
$s=1$		1	8	40	160	560	1,792	5,376	15,360
$s=2$			2	25	180	980	4,480	18,144	67,200
$s=3$				5	84	784	5,376	30,240	147,840
$s=4$					14	294	3,360	27,720	184,800
$s=5$						42	1,056	14,256	137,280
$s=6$							132	3,861	60,060
$s=7$								429	14,300
$s=8$									1,430
Total	1	4	18	90	486	2,772	16,452	100,602	629,550

Proof. A topology with s Steiner vertices has $s+2$ leaves. Hence, $\sum_{v \in V} F_v(n, s) = (s+2)F(n, s)$ as each topology appears $s+2$ times in the summation. However, $\sum_{v \in V} F_v(n, s) = nF_v(n, s)$ by symmetry. The lemma follows. □

LEMMA 2. *For* $n > s+2$ *and* $s > 0$,

$$F_v(n, s) = 2F_v(n-1, s) + \bar{F}_v(n, s-1). \tag{5}$$

Proof. Let T be a topology in $L_v(n, s)$ in which v is a leaf. Let u be the vertex adjacent to v in T. Consider two cases.

Case 1. u is a regular vertex. In this case, by the planarity of the topology in the circle, u must be one of two neighboring vertices of v on the circle. The reduced topology $T \backslash \{v\}$ corresponds to a topology in $L_u(n-1, s)$, and this correspondence is unique. Hence, there are $2F_v(n-1, s)$ topologies in this case.

Case 2. u is a Steiner vertex. In this case, u has degree 3 and is adjacent to v and two other vertices w and x. Let T' be the topology resulting from contracting u to v. Now v is of degree 2 in T', adjacent to w and x, and T' has $s-1$ Steiner vertices and n regular vertices. Furthermore, T' is planar. Thus, T' corresponds a topology in $\bar{L}_v(n, s-1)$, and this correspondence is unique. Hence, there are $\bar{F}_v(n, s-1)$ topologies in this case. The lemma follows. □

LEMMA 3. *For* $n \geq 2$,

$$F(n, 0) = n2^{n-3} = \frac{F^*(n, 0)}{R(n)}.$$

Proof. A topology in $L(n, 0)$ is simply a nonintersecting path connecting all the n vertices on the circle. We traverse the path from one endpoint a of the path. Let v be the latest traversed point of the path. Let $u(w)$ be the closest vertex to v on the circle clockwise (counterclockwise) that has not been traversed. At each step, the path can only be extended from v to either u or w, giving two different paths. Hence, the number of such paths with endpoint a is 2^{n-2}. The number of choices for a is n, and each path has two endpoints, and thus is counted twice. Hence, $F(n,0) = n2^{n-3}$. To verify the second equality, note that, by (1), $F^*(n,0) = \binom{n}{2}(n-2)! = \frac{1}{2}n!$. Then, by (3), $F^*(n,0)/R(n) = \frac{1}{2}n! \times 2^{n-2}/(n-1)! = n2^{n-3}$. □

LEMMA 4. *For* $s \geq 0$,

$$F(s+2,s) = \binom{2s}{s}\frac{1}{s+1} = \frac{F^*(s+2,s)}{R(s+2)}.$$

Proof. When $n = s+2$, the Steiner-tree topology is a *full* topology. The fact that $F(s+2,s) = \binom{2s}{s}(1/(s+1))$ is well known [Coc69], [Niv65, Chap. 11], where the integer $\binom{2s}{s}(1/(s+1))$ is, in fact, the sth *Catalan number*. To verify the second equality, note that, by (1) and (3),

$$\frac{F^*(s+2,s)}{R(s+2)} = \frac{(2s)!}{2^s s!} \times \frac{2^s}{(s+1)!} = \binom{2s}{s}\frac{1}{s+1}. \qquad □$$

We now return to prove our main result.

Proof of Theorem 1. By Lemma 3, we must only verify the theorem for $s > 0$. We use induction on $d = n - s$.

Basis. $d = 2$. By Lemma 4, Theorem 1 holds for $d = 2$.

Inductive step. Assume that the theorem holds for n and s such that $n - s < d$

Consider the n and $s > 1$ such that $d = n - s > 2$. Substituting (4) into (5) and noting that $\bar{F}_v(n,s) = F(n,s) - F_v(n,s)$, we obtain

$$\frac{s+2}{n}F(n,s) = \frac{2(s+2)}{n-1}F(n-1,s) + F(n,s-1) - \frac{s+1}{n}F(n,s-1)$$

or

$$(6) \qquad F(n,s) = \frac{2n}{n-1}F(n-1,s) + \frac{n-s-1}{s+2}F(n,s-1).$$

Applying the induction hypothesis to $F(n-1,s)$ and $F(n,s-1)$ in (6),

$$(7) \qquad F(n,s) = \frac{2n}{n-1} \times \frac{F^*(n-1,s)}{R(n-1)} + \frac{n-s-1}{s+2} \times \frac{F^*(n,s-1)}{R(n)}.$$

Substituting (1) into (7) and noting that $R(n) = (n-1)!/2^{n-2} = ((n-1)/2)R(n-1)$,

$$\begin{aligned}
F(n,s) &= \frac{1}{R(n)}\left[\frac{n(n+s-3)!}{2^s s!}\binom{n-1}{s+2} + \frac{n-s-1}{s+2} \times \frac{(n+s-3)!}{2^{s-1}(s-1)!}\binom{n}{s+1}\right] \\
&= \frac{(n+s-3)!}{R(n)2^s s!}\left[n\binom{n-1}{s+2} + \frac{2s(n-s-1)}{s+2}\binom{n}{s+1}\right] \\
&= \frac{(n+s-3)!}{R(n)2^s s!}\left[(n-s-2)\binom{n}{s+2} + 2s\binom{n}{s+2}\right] \\
&= \frac{(n+s-2)!}{R(n)2^s s!}\binom{n}{s+2} \\
&= \frac{F^*(n,s)}{R(n)},
\end{aligned}$$

which is (2). The induction is complete. □

Acknowledgment. The authors wish to thank a referee for pointing out reference [Coc69].

REFERENCES

[BG89] M. W. BERN AND and R. L. GRAHAM, *The shortest-network problem*, Sci. American, 260 (1989), pp. 84–89.

[Coc69] E. J. COCKAYNE, *Computation of minimal length full Steiner trees on the vertices of a convex polygon*, Math. Comp., 23 (1969), pp. 521-531.

[GGJ77] M. R. GAREY, R. L. GRAHAM, AND D. S. JOHNSON, *The complexity of computing Steiner minimal trees*, SIAM J. Appl. Math., 32 (1977), pp. 835–859.

[GP68] E. N. GILBERT AND H. O. POLLACK, *Steiner minimal trees*, SIAM J. Appl. Math., 16 (1968), pp. 1–29.

[Niv65] I. NIVEN, *Mathematics of Choice, or How to Count Without Counting*, Random House, New York, 1965.

[Pro88] J. S. PROVAN, *Convexity and the Steiner tree problem*, Networks, 18 (1988), pp. 55–72.

SIAM J. DISC. MATH.
Vol. 6, No. 3, pp. 335-352, August 1993

001

COUNTING EMBEDDINGS OF PLANAR GRAPHS USING DFS TREES*

JIAZHEN CAI†

Abstract. Previously counting embeddings of planar graphs used P-Q trees and was restricted to biconnected graphs. Although the P-Q tree approach is conceptually simple, its implementation is complicated. In this paper, the author solves this problem using DFS trees, which are easy to implement. The author also gives formulas that count the number of embeddings of general planar graphs (not necessarily connected or biconnected) in $O(n)$ arithmetic steps, where n is the number of vertices of the input graph. Finally, the algorithm can be extended to generate all embeddings of a planar graph in linear time with respect to the output.

Key words. graph, depth-first search, embedding, planar graph, articulation point, connected component

AMS subject classifications. 68R10, 68Q35, 94C15

1. Introduction. In [14] Wu stated the following four basic planar graph problems:

1. Decide whether a connected graph G is planar;

2. Find a minimal set of edges the removal of which will render the remaining part of G planar;

3. Give a method of embedding G in the plane in the case where G is planar;

4. Enumerate and count all possible planar embeddings of G in the plane in the case where G is planar.

Wu solved all these problems using systems of algebraic equations. His solutions are elegant, but his implementations are not so efficient. Other solutions to these problems basically follow two different approaches. One uses DFS trees [4], [8]; the other uses P-Q trees [3], [5], [9]-[11].

The P-Q tree approach is considered to be conceptually simpler, but its implementation is much more complicated. Efficient P-Q tree solutions have been discovered for all the four problems. Lempel, Even, and Cederbaun [10] solved problem 1. Chiba et al. solved problems 3 and 4 [5]. These solutions are all linear-time. Recently, Di Battista and Tamassia [6] have claimed an $O(\log n)$-time-per-operation solution to the problem of maintaining a planar graph under edge additions, which implies an $O(m \log n)$-time solution to problem 2. Here m is the number of edges, and n is the number of vertices of the input graph. On the other hand, the DFS tree approach was used only for problems 1 and 2: a linear-time DFS tree algorithm (the HT algorithm) for problem 1 was given by Hopcroft and Tarjan [8] in 1974, and an $O(m \log n)$-time algorithm for problem 2 was given by Cai, Han, and Tarjan [4] recently. The HT algorithm can also be extended to solve problem 3, but the modification is complicated.

The previous solutions for the four planar graph problems all consider biconnected graphs only. The extension from biconnected graphs to general graphs is straightforward for problems 1-3, but not for problem 4. For connected graphs, Stallmann [12] solved the enumeration version of problem 4 in time linear to the size of the output, but his solution for the counting problem is complicated and cannot be accomplished in polynomial time. For unconnected graphs, we know no published solution for problem 4.

In this paper, we give an $O(n)$-time DFS tree solution for the counting version of problem 4. While the P-Q tree solution in [5] only counts the embeddings of biconnected

* Received by the editors January 2, 1991; accepted for publication (in revised form) August 20, 1992. Part of this work was completed while the author was visiting the University of Wisconsin-Madison. The research of this author was partially supported by National Science Foundation grant CCR-9002428.

† Courant Institute, New York University, New York, New York 10012.

graphs, we also solve the interesting combinatorial problem of counting embeddings of general graphs. Our algorithms extend easily to generate one embedding or all embeddings of a planar graph in time linear to the input and output, and hence solve problems 3 and 4. Thus, we complete the DFS tree solutions for the four planar graph problems.

The rest of the paper is organized as follows. Section 2 is preliminaries. We solve the counting problem for biconnected graphs in § 3 and then show how to count embeddings for more general planar graphs in §§ 4 and 5.

2. Preliminaries. Consider an undirected graph $G = (V, E)$ with vertex set V and edge set E. Denote $|V|$ by n and $|E|$ by m. We assume that G has no self loops and has no multiple edges. We can draw a picture H on a surface, which can be either a plane or a sphere, as follows: For each vertex $v \in V$, we draw a distinct node v'; for each edge $(v, w) \in E$, we draw a simple arc connecting the two nodes v' and w'. We call this arc an *embedding* of the edge (v, w). If arcs of H do not cross each other, we say that H is an *embedding* of G. An embedding on the plane is called a *planar embedding*, and an embedding on the sphere is called a *sphere embedding*. It is easy to see that G has a planar embedding if and only if it has a sphere embedding. If G has an embedding, then we say that G is *planar*. Since we are interested only in graphs with no isolated vertices, we will frequently identify graphs with their edge sets.

One easy transformation between planar embeddings and sphere embeddings is the sphere projection shown in Fig. 1. Under the sphere projection, each point on the sphere, except the projection center o, has a distinct image on the plane, and each point on the plane is the image of some point on the sphere. Let H be a sphere embedding of a graph G with f faces. According to Euler's formula [2], if G has m edges, n vertices, and c connected components, then $f = m - n + c + 1$. Using the sphere projection, we can get f topologically different planar embeddings of G from a given sphere embedding of G by selecting the center of projection in different faces. Thus, if G has N sphere embeddings, then it has Nf planar embeddings.

We will represent embeddings by their *planar maps* and *adjacency relations*. A *planar map* M for a given embedding H of G is a mapping from V to lists of E such that, for each $v \in V$, $M(v)$ gives the clockwise circular ordering of the edges around v in H. In this case, we say that H and M *match* each other. For connected graphs, sphere embeddings with the same planar map are topologically equivalent. Therefore we need

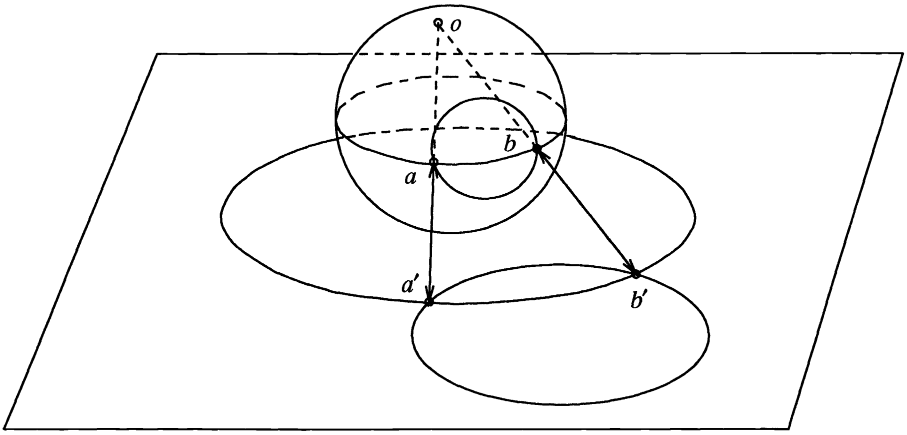


FIG. 1. *Sphere projection.*

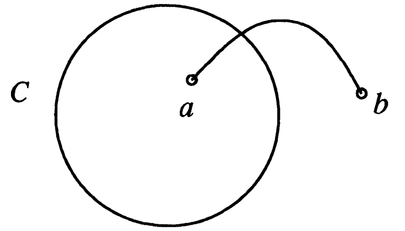


FIG. 2

only count planar maps in this case. However, for graphs with more than one connected component, planar maps do not specify the relative positions of the embeddings of different connected components.

Let H be a sphere embedding of G. We define an *adjacency relation* R on the set of faces of the embeddings of different components in H as follows. Let $C_1, \ldots, C_k$ be the connected components of G, and let $H_1, \ldots, H_k$ be the embeddings of $C_1, \ldots, C_k$ in H, respectively. We say that two embeddings H_i and H_j are *neighbors* of each other in H if there is a face in H whose boundary contains edges from both C_i and C_j. If C_i and C_j are neighbors in H, then there is a face F_i of H_i that contains H_j, and a face F_j of H_j that contains H_i. In this case, we say the two faces F_i and F_j are *adjacent* to each other, and the unordered pair (F_i, F_j) is in R. Thus, in general, a sphere embedding can be specified by a planar map plus an adjacency relation.

The following facts are important to our discussion.

Observation 1. Let C be a simple closed curve on the plane as in Fig. 2; let a be a point inside C and b be a point outside C. Then any curve that joins a and b will cross C.

Observation 2. Let G_1 be the undirected graph represented by Fig. 3, where P is a path joining the two vertices a and b on cycle C. Then in any embedding of G_1, all the edges of path P are on the same side of the cycle C.

Observation 3. Let G_2 be the undirected graph represented by Fig. 4, where a_1, a_2, b_1, and b_2 are four distinct vertices that appear in order on C. Then, in any embedding of G_2, the two paths P_1 and P_2 are on opposite sides of the cycle C.

Observation 4. Let G_3 be the undirected graph represented by Fig. 5, where a, c_1, c_2, and b are vertices that appear in order on C, and c_1 and c_2 may be the same. Then, in any embedding of G_3, the two subgraphs P_1 (containing paths from o_1 to a, b, and c_1) and P_2 (containing paths from o_2 to a, b, and c_2) are on opposite sides of the cycle C.

All four of the above observations are intuitively obvious and can be proved by the Jordan Curve Theorem [7], [13].

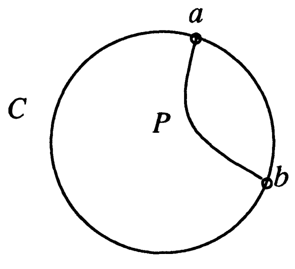


FIG. 3

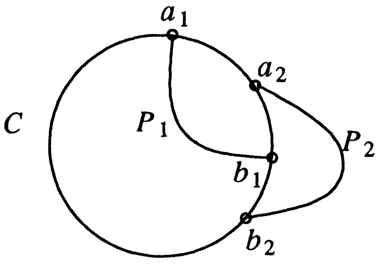


FIG. 4

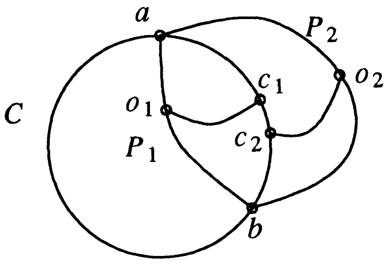


FIG. 5

3. Number of embeddings for biconnected graphs. We first discuss how to count planar maps of biconnected graphs. We will reduce this problem into a sequence of successively simpler problems before we eventually solve it.

In this section, we assume that $G = (V, E)$ is given in its DFS representations [1], where $V = \{1, \dots, n\}$ is the set of DFS numbers of the vertices in G, and E is partitioned into a set of tree edges T and a set of back edges B. If $[v, w]$ is a tree edge, the $v < w$. If $[v, w]$ is a back edge, then $w < v$, and there is a tree path in T from w to v. In either case, we say that $[v, w]$ *leaves* v and *enters* w and is *connected* to v and w.

We define *successors* for both vertices and edges. If $[v, w]$ is a tree edge, then w is a *successor* of v. If $[v, w]$ is a tree edge and $[w, x]$ is any edge, then $[w, x]$ is a *successor* of $[v, w]$. Back edges have no successors. We also define descendants and ancestors for both vertices and edges. A *descendant* of vertex (respectively, edge) x is defined recursively as either x itself or a successor of a descendant of x. If y is a descendant of x, then x is an *ancestor* of y. If y is a successor of x, then x is a *predecessor* of y.

In this section, we also assume that G is a biconnected graph with at least two edges. Then each tree edge has at least one successor, and T forms a tree with only one edge leaving the root.

Let $e = [v, w] \in E$. Let Y be the set of vertices y such that there exists a back edge $[x, y]$ that is a descendant of e. Then Y is not empty. We define $low_1(e)$ to be the smallest integer in Y and $low_2(e)$ to be the second smallest integer in $Y \cup \{n + 1\}$. The two mappings low_1 and low_2 can be computed in $O(m)$ time during the depth-first search on G [8]. Since G is biconnected, it has no articulation points. Thus, if v is not the root of T, then $low_1(e) < v$ [1].

As in [8], we define the function ϕ on E as follows:

$$\phi(e) = \begin{cases} 2\, low_1(e) & \text{if } low_2(e) \geqq v, \text{ where } e = [v, w], \\ 2\, low_1(e) + 1 & \text{otherwise.} \end{cases}$$

For each vertex $v \in V$, we arrange all the edges leaving v into a list $\Phi(v)$ in increasing order by their ϕ values. The ordering Φ can be computed in $O(m)$ time using a bucket sort. The first edge in $\Phi(v)$ is called the *reference edge* of v, denoted by $e_{v,\mathrm{ref}}$. We use E_0 to represent the set of all nonreference edges in E.

For $e = [v, w] \in E$, we define $S(e)$, the *segment* of e, to be the subgraph of G that consists of all the descendants of e. We use $ATT(e)$ to denote the set of back edges $[x, y]$ in $S(e)$ such that y is an ancestor of v. Each back edge in $ATT(e)$ is called an *attachment* of e. Thus, if $[x, y]$ is an attachment of e, then $low_1(e) \leqq y \leqq v$. If $low_1(e) < y < v$, then we say that $[x, y]$ is *normal*. Otherwise, we say that $[x, y]$ is *special*.

For each edge $e = [v, w] \in E$, we define $cycle(e)$ as follows: If e is a back edge, then $cycle(e) = \{e\} \cup \{e'\colon e'$ belongs to the tree path from w to $v\}$; if e is a tree edge, then $cycle(e) = cycle(e_{w,ref})$. Since we assume that G is a biconnected graph with more than one edge, then, for any edge $e = [v, w] \in E$, $cycle(e)$ is defined. The only edge on $cycle(e)$ that enters v is denoted by $e_{v,in}$. If v is not the root, then $e_{v,in}$ is the only tree edge entering v. Each embedding C_e of $cycle(e)$ is a simple closed curve, which divides the plane (or sphere) into two regions. When we travel on C_e along the direction of its edges, we see one region on the left-hand side and the other region on the right-hand side. We use $sub(e)$ to denote the subgraph $S(e) \cup cycle(e)$. It is easy to see that the vertex $low_1(e)$ is always on $cycle(e)$, and $sub(e) - S(e) = \{e'\colon e'$ belongs to the tree path from $low_1(e)$ to $v\}$. If e is the only tree edge leaving the root, then $sub(e)$ is the whole graph.

Figure 6 illustrates some of these definitions, where $e = [4, 5]$; $low_1(e) = 1$; $low_2(e) = 2$; $cycle(e) = \{[4, 5], [5, 6], [6, 7], [7, 8], [8, 1], [1, 2], [2, 3], [3, 4]\}$; $S(e)$ contains all the edges in the graph except $[1, 2]$, $[2, 3]$, $[3, 4]$; $sub(e)$ is the whole graph; $ATT(e) = \{[8, 1], [9, 3], [12, 1], [14, 2], [13, 4]\}$.

3.1. Partial maps. Let H be an embedding of G. Let M be the planar map of H. For each $v \in V$, we assume that the list $M(v)$ starts from the edge $e_{v,\mathrm{in}}$. For any vertex v in V and any two edges e_i and e_j connected to v, if e_i appears before e_j in $M(v)$, then we say that e_i is *embedded on the left of* e_j and e_j is *embedded on the right of* e_i in H.

A mapping M' from V to lists of edges in E is called a *partial map* of G if there is a planar map M of G such that, for each $v \in V$, $M'(v)$ can be obtained from $M(v)$ by

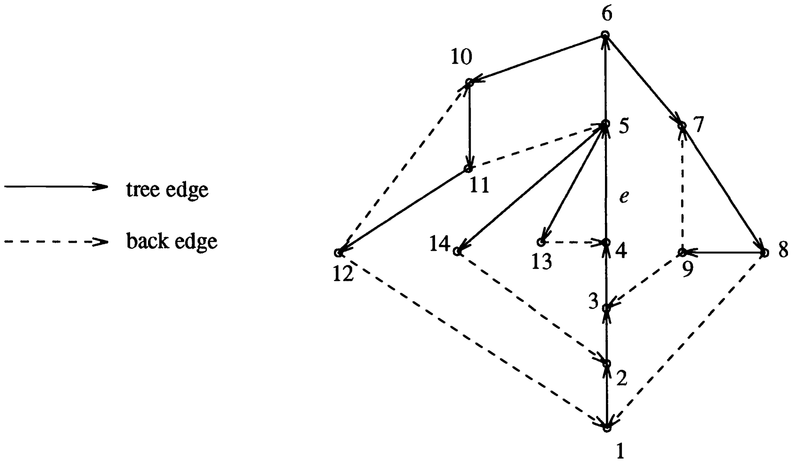


FIG. 6

deleting all the edges entering v. In this case, we say that M is an *extension* of M'. If H is an embedding that matches M, we also say that H and M' match each other. The following lemma establishes the one-to-one correspondence between planar maps and partial maps.

LEMMA 1. *If M' is a partial map of G, then there is a unique planar map M of G that is an extension of M'.*

Proof. Let H be an embedding of G that matches M'. Let M be the planar map of H. We show that M is uniquely determined by M'.

Let *label* be a numbering of back edges from 1 to $|B|$ such that, for any $v \in V$, for any two edges e_i and e_j leaving v, and for any two back edges $t_i \in S(e_i)$ and $t_j \in S(e_j)$, if $M'(v) = [\dots, e_i, \dots, e_j, \dots]$, then $label(t_i) < label(t_j)$. It is clear that *label* is uniquely determined by M'.

Let $v \in V$. Let e be an edge leaving v. Let $in(e)$ be the set of back edges in $S(e)$ entering v, not including $e_{v,in}$. Let $back(e)$ be the unique back edge on $cycle(e)$. Consider an edge $t \in in(e)$. By the definition of *label*, we know that t is embedded on the left of e in H if and only if $label(t) < label(back(e))$. Thus, the position of t in $M(v)$ relative to e is uniquely determined by M'.

Then consider two edges t_1 and t_2 in $in(e)$ such that either $label(t_1) < label(t_2) < label(back(e))$ or $label(back(e)) < label(t_1) < label(t_2)$. Again, by the definition of *label*, we know that t_1 is embedded on the right of t_2. Thus, for any two edges in $in(e) \cup \{e\}$, their relative positions in $M(v)$ are uniquely determined by the mapping label.

Now consider any two edges e_i and e_j in $M'(v)$ such that e_i appears before e_j in $M'(v)$. Since G is biconnected, then all edges in $in(e_i) \cup \{e_i\}$ are embedded on the left of all the edges in $in(e_j) \cup \{e_j\}$ in H. Thus, $M(v)$ is uniquely determined by *label.* □

Therefore, to count planar maps, we need only to count partial maps.

The above proof also suggests a simple linear-time algorithm that builds a planar map M from a partial map M'. First, we compute the mappings *label*, *back*, and *in* in a depth-first search on G, which takes $O(n)$ time (recall that, for a planar graph, $m = O(n)$.) Then, for each edge $e = [v, w] \in E$, we split $in(e)$ into two lists $L_e = [l_1, \dots, l_i]$ and $R_e = [r_1, \dots, r_j]$ such that $label(r_1) > \cdots > label(r_j) > label(back(e)) > label(l_1) > \cdots > label(l_i)$. This can be done again in $O(n)$ time using a bucket sort. For each v in V, let $M'(v) = [e_1, \dots, e_k]$. Then $M(v) = [e_{v,in}] + L_{e_1} + [e_1] + R_{e_1} + \cdots + L_{e_k} + [e_k] + R_{e_k}$, where $+$ is the list concatenation.

3.2. Singular edges. We call an edge $e = [v, w]$ in E_0 *singular* if $low_2(e) \geqq v$. A set of all singular edges leaving the same vertex and having the same low_1 value is called a *singular set*. We have the following lemma.

LEMMA 2. *Let M' be a partial map of G. Let $e_i = [v, w_i]$ and $e_j = [v, w_j]$ be two edges on the same side of $e_{v,ref}$ in $M'(v)$. If $\phi(e_i) = \phi(e_j)$, then both e_i and e_j are singular.*

Proof. We prove this lemma by contradiction. Suppose that one of e_i and e_j, say e_i, is not singular. Then $low_2(e_i) < v$. Since $\phi(e_i) = \phi(e_j)$, then $low_2(e_j) < v$ also. By Observation 4, $S(e_i)$ and $S(e_j)$ cannot be embedded on the same side of $cycle(e_{v,ref})$. Therefore, e_i and e_j cannot be embedded on the same side of $e_{v,ref}$, a contradiction. □

LEMMA 3. *Let $e_i = [v, w_i]$ and $e_j = [v, w_j]$ be two edges in a singular set. Let M' be any partial map of G. Let M'_1 be a mapping obtained from M' by switching the positions of the two edges e_i and e_j in $M'(v)$. Then M'_1 is also a partial map of G.*

Proof. Let H be an embedding of G that matches M'. Since e_i and e_j are in the same singular set, then $low_1(e_i) = low_1(e_j)$. Also, v and $low_1(e_i)$ are the only two vertices that are shared by $S(e_i)$, $S(e_j)$, and the rest of G. Therefore, either one of $S(e_i)$ and

$S(e_j)$ can be reembedded into any face in H whose boundary contains the two vertices v and $low_1(e_i)$. In particular, we can obtain another embedding H' of G from H by switching the positions of the embeddings of $S(e_i)$ and $S(e_j)$. Then M'_1 is the partial map that matches H'. □

3.3. Feasible maps and valid partitions. If U is a set and X, Y are two disjoint sets such that $X \cup Y = U$, then we call $[X, Y]$ an *ordered partition* of U. Let $Q = [LL, RR]$ be an ordered partition of E_0. We say that Q is a *valid partition* of E_0 if there exists an embedding H of G such that in H, each edge $[v, w] \in LL$ is embedded on the left of $e_{v,ref}$, and each edge $[v, w] \in RR$ is embedded on the right of $e_{v,ref}$. In this case, we say that Q is *derived* from H. If M is a planar map or partial map of G that matches H, we also say that Q is derived from M.

Let M' be a mapping from V to lists of edges in E such that, for each $v \in V$, $M'(v)$ is a permutation of the edges leaving v. We call M' a *feasible map* of G if there exists a valid partition $Q = [LL, RR]$ of E_0 so that, for all $v \in V$, if $M'(v) = [l_1, \dots, l_s, e_{v,ref}, r_1, \dots, r_t]$, then (1) $l_1, \dots, l_s \in LL$, and $r_1, \dots, r_t \in RR$, and (2) $\phi(l_1) \geqq \cdots \geqq \phi(l_s)$ and $\phi(r_1) \leqq \cdots \leqq \phi(r_t)$.

LEMMA 4. *A mapping M' from V to lists of edges in E is a partial map of G if and only if M' is a feasible map of G.*

Proof. ⇒ Suppose that M' is a partial map. Let H be an embedding of G that matches M', and let $Q = [LL, RR]$ be the unique valid partition derived from H. Let $v \in V$. Let $M'(v) = [l_1, \dots, l_s, e_{v,ref}, r_1, \dots, r_t]$. Then condition (1) in the definition of feasible map is trivially true. To see that condition (2) is also true, consider two edges e_i and e_j in $M'(v)$ with $\phi(e_i) > \phi(e_j)$. We need to show that (i) if both e_i and e_j belong to LL, then e_i appears before e_j in $M'(v)$, and (ii) if both of them belong to RR, then e_i appears after e_j in $M'(v)$. Assume that both e_i and e_j are in LL. Then e_i, therefore the whole $S(e_i)$, is embedded on the left of $cycle(e_{v,ref})$. The condition $\phi(e_i) > \phi(e_j)$ implies that there is a back edge $[x, y]$ in $S(e_i)$ such that $low_1(e_j) < y < v$. Since the tree path from $low_1(e_j)$ to v is shared by $cycle(e_{v,ref})$ and $cycle(e_j)$, then $[x, y]$; therefore, $S(e_i)$ is embedded on the left of $cycle(e_j)$. Thus, e_i appears before e_j in the list $M'(v)$. The discussion for the situation (ii) is similar.

⇐ Suppose that M' is a feasible map. Then there exists a valid partition $Q = [LL, RR]$ such that, for all $v \in V$, if $M'(v) = [l_1, \dots, l_s, e_{v,ref}, r_1, \dots, r_t]$, then conditions (1) and (2) are satisfied. Let M be the partial map of G from which Q is derived. By the *only if* part of Lemma 4, M is also a feasible map of G with respect to Q. The conditions (1) and (2) in the definition of feasible map implies that, for each $v \in V$, $M'(v)$ can be obtained from $M(v)$ by permuting edges with the same ϕ values within $\{l_1, \dots, l_s\}$ and $\{r_1, \dots, r_t\}$. By Lemmas 2 and 3, M' is also a partial map. □

By Lemma 4, we need only to count feasible maps, which can be constructed easily from valid partitions.

3.4. SAME and DIFF. Let H be an embedding of G. For convenience, we say that an edge $e = [v, w] \in E_0$ is *red* in H if e is embedded on the left of $e_{v,ref}$, and *blue* otherwise. We partition E_0 into equivalence classes called *groups*. Two edges in E_0 are in the same group if and only if they have the same color in each embedding of G. We call the set of such groups SAME. We further organize these groups into *pairs*. Two groups W and Z in SAME are put into one (unordered) pair (W, Z) if and only if the color of the edges in W is always different than the color of the edges in Z. We call the set of such pairs DIFF. We will show in § 3.6 that the two sets SAME and DIFF can be computed in $O(n)$ time during planarity testing.

Let $Q = [LL, RR]$ be an ordered partition of E_0. We say that Q is *consistent with* SAME if each group in SAME is totally contained in either LL or RR. We say that Q is *consistent with* DIFF if, for each pair $(W, Z) \in$ DIFF, one of the two groups W and Z is contained in LL and the other is contained in RR.

By the definition of DIFF and SAME, any valid partition of E_0 is consistent with SAME and DIFF. We will further prove that any ordered partition of E_0 consistent with SAME and DIFF is valid. For this, we need some more definitions and lemmas.

Let $e = [v, w]$ be a tree edge. Let $\Phi(w) = [e_1, \ldots, e_k]$. Let $Q = [LL, RR]$ be an ordered partition of E_0. For $i = 1, \ldots, k$, let $G_i = sub(e_1) \cup \cdots \cup sub(e_i)$. Let H_i be an embedding of G_i. We say that H_i is *conformable* to Q (with respect to e) if, around each vertex $u \geqq w$ in H_i, all the edges embedded on the left of $e_{u,ref}$ belong to LL and all the edges embedded on the right of $e_{u,ref}$ belong to RR. By convention, any embedding of $sub(e)$ is conformable to Q (with respect to e) if e is a back edge.

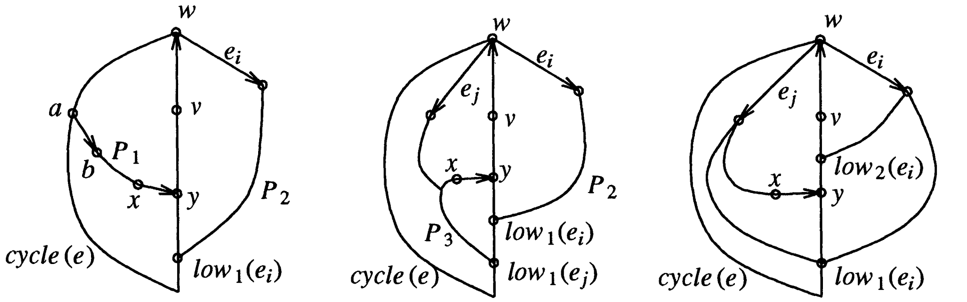
Let $[x, y]$ be an attachment of e not on $cycle(e)$. Let $[a, b]$ be the nearest ancestor of $[x, y]$ such that a is on $cycle(e)$. We call $[a, b]$ the *root* of $[x, y]$ (with respect to e), denoted by $root([x, y])$. We prove the following lemma.

LEMMA 5. *If* $[x, y]$ *is an attachment of* e *in* G_{i-1} *not on* $cycle(e)$, *and* $low_1(e_i) < y$, *then there is a pair* (W, Z) *in* DIFF *such that* $e_i \in W$ *and* $root([x, y]) \in Z$, *where* $1 < i \leqq k$.

Proof. Let $[a, b] = root([x, y])$. Let W be the group in SAME containing e_i, and let Z be the group in SAME containing $[a, b]$. Let P_1 be the simple directed path in $sub(e)$ whose first edge is $[a, b]$ and whose last edge is $[x, y]$. Let P_2 be a simple directed path in $sub(e)$ whose first edge is e_i and whose last vertex is $low_1(e_i)$. Consider two cases.

Case 1. $a > w$. By Observation 3, P_1 and P_2 cannot be embedded on the same side of $cycle(e)$ in any embedding of G (see Fig. 7). Therefore, $(W, Z) \in$ DIFF.

Case 2. $a = w$. In this case, $[a, b] = e_j$ for some $1 < j < i$, and $low_1(e_j) \leqq low_1(e_i) < y$. Then there must be an undirected simple path P_3 in $sub(e_j)$ between $low_1(e_j)$ and y that contains x. If $low_1(e_j) < low_1(e_i) < y$, then P_3 and P_2 cannot be embedded on the same side of $cycle(e)$ by Observation 3. If $low_1(e_j) = low_1(e_i)$, then $low_2(e_j) \leqq y < w$. Therefore, $low_2(e_i) < w$ (recall that $\phi(e_i) \geqq \phi(e_j)$). Thus, $S(e_i)$ and $S(e_j)$ cannot be embedded on the same side of $cycle(e)$ by Observation 4. In either case, e_i and e_j cannot be embedded on the same side of $cycle(e)$, and therefore $(W, Z) \in$ DIFF. □



$a > w$

$a = w$ and $low_1(e_i) > low_1(e_j)$

$a = w$ and $low_1(e_i) = low_1(e_j)$

FIG. 7

LEMMA 6. *Let $Q = [LL, RR]$ be an ordered partition of E_0 consistent with* SAME. *Let H_e be an embedding of sub(e) conformable to Q. If $e \in LL$ (RR), then all the normal attachments of e are embedded on the left- (right-)hand side of cycle(e) in H_e.*

Proof. Assume without loss of generality that $e \in LL$. Let $[x, y]$ be a normal attachment of e. Let $[a, b] = root([x, y])$. Let P_1 be the simple directed path whose first edge is $[a, b]$ and whose last edge is $[x, y]$. Let e' be the predecessor of e. Note that the tree path from $low_1(e)$ to v is shared by *cycle*(e) and *cycle*(e'). By Observation 2, P_1 and e are always on the same side of *cycle*(e') in any embedding of G. Thus, if e is embedded on the left (right) of *cycle*(e'), then $[a, b]$ must be embedded on the left (right) of *cycle*(e). This means that e and $[a, b]$ are in the same group of SAME. Since Q is consistent with SAME, and $e \in LL$, then $[a, b] \in LL$. Since H_e is conformable to Q, then $[a, b]$, and therefore $[x, y]$, are embedded on the left-hand side of *cycle*(e). □

Now we prove the main lemma of this section.

LEMMA 7. *An ordered partition $Q = [LL, RR]$ of E_0 is valid if it is consistent with* SAME *and* DIFF.

Proof. Assume that Q is consistent with SAME and DIFF. To see that Q is valid, we show that there exists a planar embedding of G from which Q can be derived. For this purpose, we show by induction that, for all $e = [v, w] \in E$, we can construct an embedding H_e of *sub*(e) that is conformable to Q.

If e is a back edge, then any embedding of *sub*(e) is conformable to Q by convention.

Next, we assume that $e = [v, w]$ is a tree edge with $\Phi(w) = [e_1, \dots, e_k]$, and, for each $i = 1, \dots, k$, there is a planar embedding H_{e_i} of *sub*(e_i) that is conformable to Q (with respect to e_i).

To construct H_e, we first let $H_1 = H_{e_1}$. Then, for $i = 2, \dots, k$, we add H_{e_i} into H_{i-1} to get H_i. As a result, we will have $H_e = H_k$.

Consider adding H_{e_i} to H_{i-1}, where $1 < i \leqq k$. Assume inductively that H_{i-1} is conformable to Q (with respect to e). Also assume without loss of generality that $e_i \in LL$. By Lemma 6, all the normal attachments of e_i are embedded on the left of *cycle*(e_i) in H_{e_i}. Thus, with the sphere projection, we can transform H_{e_i} into a planar embedding of *sub*(e_i) in which the tree path from $low_1(e_i)$ to w borders the outer face.

If there is no attachment of e embedded on the left of *cycle*(e) in H_{i-1}, we can embed H_{e_i} to the left of *cycle*(e) in the face whose boundary contains the tree path from $low_1(e)$ to w. Otherwise, let $[x, y]$ be one of the highest attachments of e embedded on the left of *cycle*(e) in H_{i-1}. (We say an attachment $[x, y]$ is *higher* than another attachment $[x', y']$ if $y > y'$.) Let $[a, b] = root([x, y])$. By induction hypothesis, H_{i-1} is conformable to Q. Therefore $[a, b] \in LL$. Since $e_i \in LL$ also, there can be no pair (W, Z) in DIFF such that $e_i \in W$ and $[a, b] \in Z$. By Lemma 5, $low_1(e_i) \geqq y$. Then we can embed H_{e_i} into H_{i-1} on the left side of *cycle*(e) in the face whose boundary contains the tree path from y to w. In this way, e_i is embedded on the left of $e_1, \dots, e_{i-1}$, and H_i is conformable to Q. □

According to Lemmas 1, 4, and 7, all planar maps of G can be easily generated from the function ϕ and the two sets SAME and DIFF as follows:

1. Generate valid partitions using Lemma 7;
2. For each valid partition generated in 1, generate partial maps using Lemma 4;
3. For each partial map generated in 2, construct a planar map using the method described at the end of § 3.1.

3.5. Counting planar maps. To count the number of planar maps, we further simplify the problem as follows. We arbitrarily select a representative from each singular set. If

M' is a feasible map and M'' is obtained from M' by deleting all nonrepresentative singular edges, then we say M'' is a *reduced map* from M' and that M' is *generated* from M''. Similarly, if Q is a valid partition and Q' is obtained from Q by deleting all nonrepresentative singular edges, then Q' is called a *reduced partition*. If M'' is a reduced map from M', Q' is a reduced partition from Q, and Q is derived from M', then we also say that Q' is *derived* from M'' and that M'' is *constructed* from Q'. It is not difficult to see that, from each reduced map, we can derive a unique reduced partition, and, from each reduced partition, we can construct a unique reduced map. Thus, to count feasible maps, we can first count reduced partitions, then count the feasible maps that can be generated from each reduced map.

To count reduced partitions, let SAME$'$ and DIFF$'$ be obtained from SAME and DIFF, respectively, by deleting all the nonrepresentative singular edges. A pair $[W, Z]$ in DIFF$'$ is *trivial* if either W or Z is empty. By Lemma 7, it is easy to see that, if $[L, R]$ is an ordered partition of SAME$'$ such that neither L nor R contains groups from the same nontrivial pair in DIFF$'$, then $[\bigcup_{W\in L} W, \bigcup_{W\in R} W]$ is a reduced partition. Let d be the number of nontrivial pairs in DIFF$'$, and let s be the number of nonempty sets in SAME$'$ that are not contained in any of the nontrivial pairs in DIFF$'$. Then there are 2^{d+s} reduced partitions and therefore 2^{d+s} reduced maps.

Next, we consider the number of feasible maps that can be generated from each reduced map. Let *singular*(e) be the singular set containing e, and let *same*(e) be the group in SAME containing e. Immediately from Lemma 3 and its proof we have the following lemma.

LEMMA 8. (i) *Let e be a singular edge. If $|same(e)| > 1$, then $singular(e) \subseteq same(e)$;*

(ii) *Let e_1 and e_2 be two edges in the same singular set. Then the unordered pair $(same(e_1), same(e_2))$ is not in* DIFF.

We say that a singular edge e is *bound* if $singular(e) \subseteq same(e)$, and *free* otherwise. We can construct a feasible map M' from a reduced map M'' by inserting nonrepresentative singular edges as follows. Let $e = [v, w]$ be a representative singular edge, and let $g(e) = |singular(e)|$. If e is bound, then all the edges in *singular*(e) must be inserted consecutively in the same side of $e_{v,ref}$ in $M'(v)$. Therefore, we replace e in $M''(v)$ by any of the $g(e)!$ permutations of *singular*(e). If e is free, then the edges in *singular*(e) can appear in different sides of $e_{v,ref}$ in $M'(v)$ by Lemma 8. Therefore, we divide *singular*(e) into two parts S_1 and S_2, assuming that S_1 contains e. Then we replace e by a permutation of S_1 and insert a permutation of S_2 into the other side of $e_{v,ref}$ in $M''(v)$ in the position determined by the condition (2) in the definition of feasible maps. In this case, we have $(g(e) + 1)!/2$ different choices.

Now, let RS be the set of representative singular edges. For all $x \in RS$, define $h(x) = g(x)!$ if x is bound, and $(g(x) + 1)!/2$ otherwise. Then, from each reduced map, we can generate $\prod_{x\in RS} h(x)$ different partial maps. By Lemma 1, we have the following result.

THEOREM 1. *The total number of planar maps of G is*

$$2^{d+s} \prod_{x\in RS} h(x).$$

The remaining question is how to compute the two sets SAME and DIFF efficiently.

3.6. Compute the sets SAME and DIFF. Now, we show how to compute the two sets SAME and DIFF in linear time during planarity testing. The planarity testing algorithm we will use in this section is a variant of the HT algorithm reported in [4] and is summarized in the next section for convenience.

3.6.1. Planarity testing. As before, we assume that G is a biconnected graph with more than one edge. Then the tree edges in T form a single tree with only one tree edge leaving the root. Denote this tree edge by e_0. Since $sub(e_0)$ is the whole graph, then we can determine the planarity of G with a procedure that can determine the planarity of $sub(e)$ for all $e \in E$.

We say that an edge e is *planar* if $sub(e)$ is planar. To determine the planarity of an edge e, we consider two cases. If e is a back edge, then $sub(e) = cycle(e)$, which is always planar. Otherwise, e is a tree edge having at least one successor. In this case, we first determine the planarity of each of its successors. If all these successors are planar, then we determine the planarity of e based on the structure of its attachments. Following are the details.

Structure of attachments. The planarity of an edge $e = [v, w]$ directly depends on the structure of its attachments. If e is planar, we partition the edges of $ATT(e)$ into *blocks* as follows. We put two back edges of $ATT(e)$ in the same block if they are on the same side of $cycle(e)$ in every embedding of $sub(e)$. Two blocks *interlace* each other if they are on opposite sides of $cycle(e)$ in every embedding of $sub(e)$. By this definition, each block of $ATT(e)$ can interlace at most one other block.

The back edge on $cycle(e)$ is the only attachment of e that will not be embedded on either side of $cycle(e)$. By convention, this back edge forms a block by itself, called the *neutral block* of e, which does not interlace other blocks of $ATT(e)$.

In Fig. 6, $ATT(e)$ can be divided into the following four blocks: $B_1 = \{[8, 1]\}$, $B_2 = \{[12, 1], [14, 2]\}$, $B_3 = \{[9, 3]\}$, and $B_4 = \{[13, 4]\}$. B_1 is neutral. B_2 and B_3 are interlacing.

A block of attachments of e is *normal* if it contains some normal attachment of e. Otherwise, we say that it is *special*. We say that $sub(e)$ is *strongly planar* with respect to e if e is planar and if all the normal blocks of $ATT(e)$ can be embedded on the same side of $cycle(e)$. If $sub(e)$ is strongly planar (with respect to e), then we say that e is strongly planar. We have the following lemma.

LEMMA 9. *Let $e = [v, w] \in T$, and let e_i be a successor of e such that $e_i \neq e_{w,ref}$. Then e_i is strongly planar if and only if the subgraph $S(e_i) \cup cycle(e)$ is planar.*

Note that, in an embedding of $S(e_i) \cup cycle(e)$, the special blocks of e_i do not have to be on the same side of $cycle(e_i)$; see Fig. 8.

We represent a block of back edges $K = \{[v_1, w_1], [v_2, w_2], \dots, [v_t, w_t]\}$ by a list $L = [w_1, w_2, \dots, w_t]$, where $w_1 \leqq w_2 \leqq \cdots \leqq w_t$. Frequently, we will identify

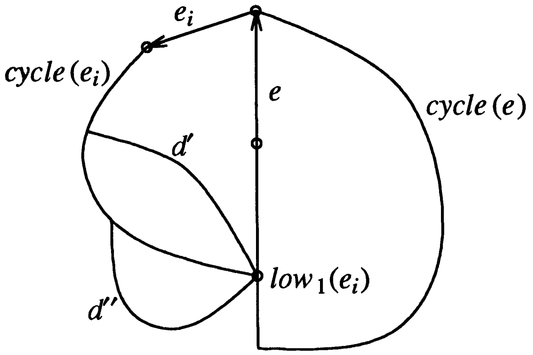


FIG. 8. *The two special attachments d' and d'' of e_i can be on different sides of $cycle(e_i)$, although they are on the same side of $cycle(e)$.*

blocks with their list representations. Define *first*(K) = *first*$(L) = w_1$ and *last*(K) = *last*$(L) = w_t$. If L is empty, we define *first*(K) = *first*$(L) = n + 1$ and *last*(K) = *last*$(L) = 0$. We can further organize the blocks of $ATT(e)$ as follows: If two blocks X and Y interlace, we put them into a pair $[X, Y]$, assuming that *last*$(X) \geqq$ *last*(Y); if a nonempty block X does not interlace any other block, we form a pair $[X, [\ \ ]]$.

Let $[X_1, Y_1]$ and $[X_2, Y_2]$ be two pairs of interlacing blocks. We say that $[X_1, Y_1] \leqq [X_2, Y_2]$ if and only if *last*$(X_1) \leqq \min($*first*(X_2), *first*$(Y_2))$. We say that a list of interlacing pairs $[q_1, \ldots, q_s]$ is *well ordered* if $q_1 \leqq \cdots \leqq q_s$. Empty lists or lists of one pair are well ordered by convention. In [4] we proved that all the interlacing pairs of $ATT(e)$ can be organized into a well-ordered list $[p_1, \ldots, p_t]$. We call this list *att*(e).

In Fig. 6, *att*$(e) = [p_1, p_2, p_3]$, where $p_1 = [[1], [\ \ ]]$, $p_2 = [[3], [1, 2]]$, and $p_3 = [[4], [\ \ ]]$.

Compute att(e). Now we are ready to compute *att*(e). The planarity of e will be decided at the same time.

Consider an edge $e = [v, w] \in E$. If e is a back edge, then its only attachment is e itself. Therefore, *att*$(e) = [[[w], [\ \ ]]]$. Otherwise, let $\Phi(w) = [e_1, \ldots, e_k]$. We first recursively compute *att*(e_i) for each e_i in $\Phi(w)$, then we compute *att*(e) in four steps, shown in the following algorithm.

Algorithm A

Step 1. For $i = 1, \ldots, k$, delete all occurrences of w appearing in blocks within *att*(e_i). Because these occurrences appear together at the end of the blocks that are contained in the last pairs of *att*(e_i) only, a simple list traversal suffices to delete all these occurrences in time $O(1 +$ *number of deletions*$)$. After this, initialize *att*(e) to be *att*(e_1).

Step 2. For $i = 2, \ldots, k$, merge all the blocks of *att*(e_i) into one intermediate block B_i. See Fig. 9.

According to Lemma 9, this step can be done only if the normal blocks of *att*(e_i) do not interlace. (If they interlace, the graph is not planar, and the computation fails.) To merge a series of blocks, simply concatenate their ordered list representations (such concatenation is order-preserving).

Step 3. Merge blocks in *att*(e). See Fig. 10.

By Observation 3, all blocks D in *att*(e) with *last*$(D) >$ *low*$_1(e_2)$ must be merged into one block B_1. (If any two of these blocks interlace, the graph is not planar, and the computation fails.) This is achieved by merging from the high end of *att*(e). This step turns *att*(e) into a list of pairs $p_1 \leqq \cdots \leqq p_h$ with only p_h possibly having a block D with *last*$(D) >$ *low*$_1(e_2)$.

Step 4. For $i = 2, \ldots, k$, add blocks B_i into *att*(e).

To process B_i, consider the last pair P: $[X, Y]$ of *att*(e). Consider three cases: (i) if B_i cannot be embedded on either side of *cycle*(e), then G is not planar, and the computation of *att*(e) fails; (ii) if B_i interlaces X only, then merge B_i into Y. Next, switch X and Y if *last*$(X) <$ *last*(Y); (iii) if B_i interlaces neither X nor Y, then add $[B_i, [\ \ ]]$ to the high end of *att*(e); $P := [B_i, [\ \ ]]$.

By the following lemma, testing whether B_i interlaces X or Y takes $O(1)$ time. Also by that lemma, it is not possible that B_i interlaces Y only, since *last*$(X) \geqq$ *last*(Y) (see Fig. 11).

LEMMA 10. *B_i and D can be embedded on the same side of cycle(e) if and only if low$_1(e_i) \geqq$ last(D), where $D = X$ or $D = Y$.*

In [4] we proved the following theorem.

THEOREM 2. (1) *Algorithm* A *computes att*(e) *successfully if and only if e is planar*; (2) *If e is planar, then Algorithm* A *computes att*(e) *correctly.*

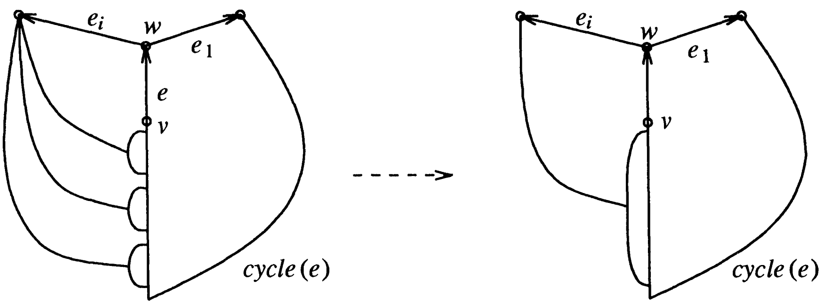


FIG. 9

3.6.2. Compute the sets SAME and DIFF. Next, we augment Algorithm A so as to compute the two sets SAME and DIFF during the planarity testing.

Let $e \in E$ be an edge of G. Let e_a an attachment of e not on *cycle*(e). Then *root*(e_a) and e_a are embedded on the same side of *cycle*(e) in any embedding of G. Thus, for each nonneutral block X of e, there is a unique group in SAME that contains the roots of the attachments in X. We call this group *buddy*(X). It is easy to see that, if $[X, Y]$ is a pair of nonempty interlacing blocks of $ATT(e)$, then (*buddy*(X), *buddy*(Y)) is a pair in DIFF. Furthermore, in the proof of Lemma 6, we note that,

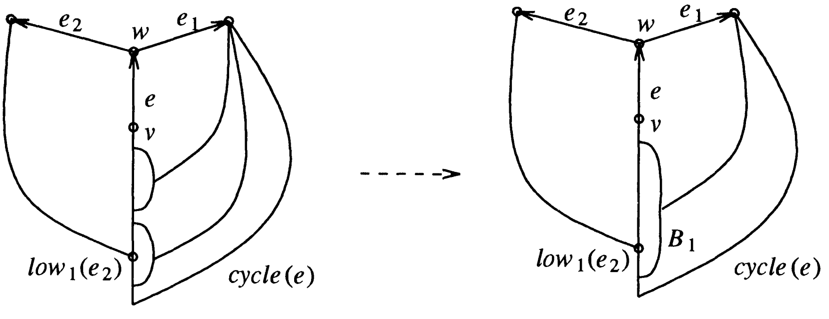
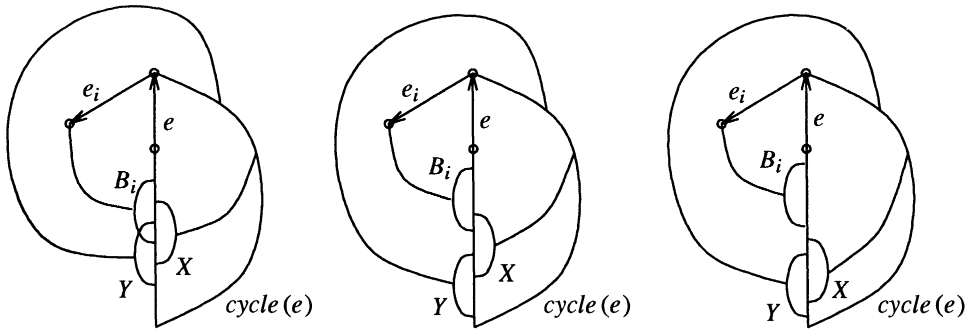


FIG. 10



B_i cannot be embedded in either side of *cycle* (e)

B_i interlaces X only

B_i interlaces neither X nor Y

FIG. 11

if e_a is normal and if $e \in E_0$, then $root(e_a)$ and e belong to the same group in SAME. Thus, if X is a normal block of e and $e \in E_0$, then $buddy(X)$ also contains e. For convenience, we further extend the definition of *buddy* as follows. If $[X, Y]$ is a pair in $ATT(e)$ such that $Y = [\quad]$ and $(buddy(X), U) \in$ DIFF, then define $buddy(Y) = U$. According to these observations, we can compute the two sets SAME and DIFF with the following enhancement to Algorithm A.

Enhancement B

1. Initialization. For all $e \in B$, let $buddy([e]) = \emptyset$. Let SAME $= \{\{e\}: e \in E_0\}$ and DIFF $= \{(\{e\}, \emptyset): e \in E_0\}$;

2. In step 2 of Algorithm A, for $i = 2, \ldots, k$, before we merge $att(e_i)$, we initialize $buddy(B_i)$ to be $\{e_i\}$. For each pair $[X, Y]$ or $[Y, X]$ in $att(e_i)$ such that X is normal with respect to e_i, let U be the set such that $(buddy(B_i), U) \in$ DIFF; in SAME, merge $buddy(X)$ into $buddy(B_i)$ and merge $buddy(Y)$ into U; in DIFF, merge the two pairs $(buddy(B_i), U)$ and $(buddy(X), buddy(Y))$ into one pair $(buddy(B_i) \cup buddy(X), U \cup buddy(Y))$.

3. In step 3, let $[X, Y]$ be the last pair in the list $att(e_1)$ before merging. For each pair $[X_1, Y_1]$ in $att(e_1)$ merged into $[X, Y]$, do the following: In SAME, merge $buddy(X_1)$ into $buddy(X)$ and merge $buddy(Y_1)$ into $buddy(Y)$; in DIFF, merge the two pairs $(buddy(X), buddy(Y))$ and $(buddy(X_1), buddy(Y_1))$ into one pair $(buddy(X) \cup buddy(X_1), buddy(Y) \cup buddy(Y_1))$.

4. In step 4, for $i = 2, \ldots, k$, let U be the set such that $(buddy(B_i), U) \in$ DIFF. If $[B_i, Z]$ becomes the top pair of $att(e)$, where $Z = [\quad]$, then let $buddy(Z) = U$. If B_i is merged into Y, then in SAME, merge $buddy(B_i)$ into $buddy(Y)$ and merge U into $buddy(X)$; in DIFF, merge the two pairs $(buddy(B_i), U)$ and $(buddy(X), buddy(Y))$ into one pair $(U \cup buddy(X), buddy(B_i) \cup buddy(Y))$.

One way to prove the correctness of Enhancement B is to prove that (i) if an ordered partition $P = [LL, RR]$ of E_0 is valid, then it is consistent with the two sets SAME and DIFF computed by Enhancement B; and (ii) if an ordered partition $P = [LL, RR]$ of E_0 is consistent with the two sets SAME and DIFF computed by Enhancement B, then it is valid.

We see that (i) is true because, in the enhancement code, two edges are put in the same group of SAME only if they have the same color in each embedding of G, and two groups form a pair in DIFF only if they always have different colors. Assertion (ii) is basically the same as Lemma 7, except that the two sets SAME and DIFF here are computed by Enhancement B, not given by their definitions. Since the proof of Lemma 7 is based on Lemmas 5 and 6, we then need only to prove these two lemmas under the new condition.

LEMMA 11. *Lemma* 5 *remains true if the two sets* SAME *and* DIFF *are computed by Enhancement* B.

Proof. Consider the attachment $[x, y]$ given in Lemma 5. Let $[a, b] = root([x, y])$. Let $[X, Y]$ be the top pair of blocks in $att(e)$ in step 4 of the planarity testing. Since $att(e)$ is well ordered, then $[x, y]$ is contained in either X or Y. If $[x, y] \in Y$, then $low_1(e_i) < last(Y)$, and G is not planar. Thus $[x, y] \in X$, and $low_1(e_i) < last(X)$. Therefore B_i is merged into Y in step 4. Then $root([x, y]) \in buddy(X)$, $e_i \in buddy(Y)$, and $(buddy(X), buddy(Y)) \in$ DIFF. □

LEMMA 12. *Lemma* 6 *remains true if the two sets* SAME *and* DIFF *are computed by Enhancement* B.

Proof. Consider the edge e, the embedding H_e, and the partition Q given in Lemma 6. Assume without loss of generality that $e \in LL$. Let $[x, y]$ be a normal attachment of

e. We need to show that $[x, y]$ is embedded on the left-hand side of *cycle*(e) in H_e. Let $[a, b] = root([x, y])$. Let e' be the predecessor of e. Let X be the block of attachment in *ATT*(e') that contains $[x, y]$. Then Enhancement B will put both e and $[a, b]$ into *buddy*(X). This means that e and $[a, b]$ are in the same group of SAME. Since we assume that $e \in LL$, then $[a, b] \in LL$. Since H_e is conformable to Q, then $[a, b]$, and therefore $[x, y]$, are embedded on the left-hand side of cycle (e) in H_e. □

As a result of Lemmas 11 and 12, Lemma 7 remains true for the two sets SAME and DIFF computed by our Enhancement B. Therefore, we have the following result.

THEOREM 3. *If G is planar, then Algorithm* A *with Enhancement* B *compute the sets* SAME *and* DIFF *correctly.*

4. Number of embeddings for connected components.

Next, consider a connected graph G with several biconnected components. Suppose that we know the number of embeddings of each biconnected component. We discuss how to find the total number of embeddings of G. This problem was previously considered by Stallmann [12], but his solution is complicated and not efficient. In this section, we give a simple closed formula for this problem that is computable in $O(n)$ arithmetic steps.

We start with the simple situation that G has two biconnected components G_1 and G_2 sharing an articulation point a. Suppose that there are m_1 edges connected to a in G_1 and m_2 such edges in G_2. Let H_1 be an embedding of G_1 on a sphere S_1, and let H_2 be an embedding of G_2 on another sphere S_2. Imagine that S_1 and S_2 are balloons. To combine H_1 and H_2 into a sphere embedding of G, we choose a face F_1 of H_1 and a face F_2 of H_2 such that their boundaries contain a. Make a hole on F_1 so that a is the only point shared by the boundaries of the hole and F_1. Do the same thing with F_2. Glue these two holes on their boundaries, making sure that the two embeddings of a are put together. Blowing the combined balloon into a sphere gives an embedding of G. There are m_1 faces in H_1 whose boundaries contain a, and there are m_2 such faces in H_2. Thus, we can obtain $m_1 m_2$ different sphere embeddings of G by combining H_1 and H_2.

The above method of combining sphere embeddings can be generalized to get sphere embeddings of graphs with more biconnected components and more articulation points. However, counting the number of embeddings becomes more complicated in the general case. For graphs with one articulation point, we have the following result.

LEMMA 13. *Let G be a planar graph consisting of j biconnected components G_1, $\dots$, G_j sharing an articulation point a. For each $i = 1, \dots, j$, let m_i be the number of edges connected to a in G_i, and let k_i be the number of different sphere embeddings of G_i. Then, for $j > 2$, the total number of different sphere embeddings of G is*

$$k_1 k_2 \cdots k_j m_1 m_2 \cdots m_j (A - 1)(A - 2) \cdots (A - j + 2),$$

where $A = m_1 + \cdots + m_j$.

Proof. We need only to prove the following assertion: For a fixed group of embeddings $H_1, \dots, H_j$ of $G_1, \dots, G_j$, we can obtain $m_1 m_2 \cdots m_j (A - 1)(A - 2) \cdots (A - j + 2)$ different embeddings of G by gluing balloons. We call this set of embeddings of G an $E_{m_1, \dots, m_j}$ set.

We prove the assertion by induction on A. The basis is trivial, when $m_1 = \cdots = m_j = 1$. Now we assume that the assertion is true for any $A < k$, where $k > j$. Consider the case when $A = k$. Then there exists some $i = 1, \dots, j$ such that $m_i > 1$. We assume without loss of generality that $m_1 > 1$. For each $i = 1, \dots, j$, let $e_{i,1}, \dots, e_{i,m_j}$ be the clockwise sequence of edges around a in H_i. We divide an $E_{m_1, \dots, m_j}$ set into j groups, as follows:

Group 1 contains all the embeddings such that $e_{1,1}$ is followed by $e_{1,2}$;

Group 2 contains all the embeddings such that $e_{1,1}$ is followed by $e_{2,l}$, where $l = 1, \ldots, m_2$;

$\ldots$

Group j contains all the embeddings such that $e_{1,1}$ is followed by $e_{j,l}$, where $l = 1, \ldots, m_j$.

In Group 1, if we glue the two edges $e_{1,1}$ and $e_{1,2}$ together in each embedding, we get an $E_{m_1-1,m_2,\ldots,m_j}$ set. By the induction hypothesis, the size of Group 1 is

$$(m_1 - 1)m_2 \cdots m_j(A - 2) \cdots (A - j + 1).$$

For each $i = 2, \ldots, j$, we divide Group i into m_i subgroups, so that in every embedding of the lth subgroup, $e_{1,1}$ is followed by $e_{i,l}$. By gluing the two edges $e_{1,1}$ and $e_{i,l}$ together in each of the embedding in the lth subgroup, we get an $E_{m_1+m_i-1,m_2,\ldots,m_{i-1},m_{i+1},\ldots,m_j}$ set, which has the size

$$(m_1 + m_i - 1)m_2 \cdots m_{i-1}m_{i+1} \cdots m_j(A - 2) \cdots (A - j + 2).$$

Therefore, the size of Group i is

$$m_i[(m_1 + m_i - 1)m_2 \cdots m_{i-1}m_{i+1} \cdots m_j(A - 2) \cdots (A - j + 2)]$$
$$= (m_1 + m_i - 1)m_2 \cdots m_j(A - 2) \cdots (A - j + 2).$$

Adding the sizes of Group 1, ..., Group j, we see that the size of $E_{m_1,m_2,\ldots,m_j}$ is

$$m_1m_2 \cdots m_j(A - 1) \cdots (A - j + 2). \qquad \square$$

Now consider a connected graph G with more than one articulation point. To count the number of embeddings, we first choose one articulation point a. Let G_a be the subgraph of G that consists of all the biconnected components sharing a. Using Lemma 13, we can count the number of embeddings of the subgraph G_a. Then we treat G_a as one biconnected component and solve the remaining problem recursively. The result is summarized in the following theorem.

THEOREM 4. *Let G be a planar graph. Let Γ be the set of biconnected components of G, and let Θ be the set of articulation points of G. For each biconnected component C in Γ, let k_C be the number of sphere embeddings of C. For each $a \in \Theta$, let Γ_a be the set of biconnected components of G sharing a, and let A_a be the number of edges connected to a. For each $a \in \Theta$ and each component $C \in \Gamma_a$, let $m_{C,a}$ be the number of edges in C connected to a. Then the total number of sphere embeddings of G is*

$$\prod_{C\in\Gamma} k_C \prod_{a\in\Theta} \left(\prod_{C\in\Gamma_a} m_{C,a} \prod_{i=1}^{|\Gamma_a|-2} (A_a - i) \right).$$

The analysis in this section also suggests a recursive procedure that generates all planar maps of G without repetition from the planar maps of the biconnected components of G.

5. Counting embeddings for unconnected graphs. Finally, we consider how to count the embeddings of graphs having several connected components, given the number of embedding of each of the connected components.

THEOREM 5. *Let G be a planar graph consisting of c connected components $C_1, \ldots, C_c$, where $c > 1$. If, for $i = 1, \ldots, c$, C_i has n_i sphere embeddings each having f_i faces, then the number of sphere embeddings of G is*

$$\left(1 + \sum_{i=1}^{c} (f_i - 1)\right)^{c-2} \prod_{i=1}^{c} n_i f_i.$$

Proof. For each $i = 1, \dots, c$, we choose a fixed embedding H_i of C_i. We denote the set of these embeddings by Δ. We call the embeddings in Δ *subembeddings* to distinguish them from the embeddings of G. Very similarly to the description in § 4, we can combine the subembeddings in Δ into an embedding of G by gluing balloons. The main difference is that, in this case, the holes made should not touch the boundary of any face. Let Ψ be the set of all embeddings of G that can be obtained from Δ this way. We need to prove that

$$|\Psi| = \left(1 + \sum_{i=1}^{c} (f_i - 1)\right)^{c-2} \prod_{i=1}^{c} f_i. \tag{*}$$

We prove the claim by induction on c, the total number of connected components of G. For $c = 2$, the claim is obviously true. Then we assume that the claim is true for all $c < k$, where $k > 2$. We want to show that the claim is also true for $c = k$. We partition Ψ into $c - 1$ groups $\Psi_1, \dots, \Psi_{c-1}$, such that, for $i = 1, \dots, c - 1$, group Ψ_i contains the embeddings H of G in which H_1 is the neighbor of exactly i other subembeddings in Δ (recall that two subembeddings H_s and H_t are neighbors of each other in H if there is a face in H whose boundary contains edges from both H_s and H_t.) We further divide Ψ_i into $\binom{c-1}{i}$ subgroups such that in all embeddings of each subgroup, H_1 has the same set of neighbors. Consider one such subgroup $\Psi_{i,P}$ in which H_1 has the set of neighbors $P = \{H_{t_1}, \dots, H_{t_i}\}$. Let $Q = \{H_2, \dots, H_c\} - P$. An embedding in $\Psi_{i,P}$ can be obtained in two stages. First, we combine H_1 and all the subembeddings in P into one embedding X. Since, for each $j = t_1, \dots, t_i$, each of the f_j faces of H_j can be adjacent to each of the f_1 faces of H_1, then we have the number c_1 of different choices in the first stage is $(f_1 f_{t_1}) \cdots (f_1 f_{t_i})$. Next, we combine X and the subembeddings in Q into an embedding Y in $\Psi_{i,P}$. Since subembeddings in Q are not neighbors of H_1, then we can treat X as a component with $\sum_{H_s \in P} (f_s - 1)$ faces. Applying (*) inductively, we find that the number c_2 of different choices in the second stage is

$$\left(1 + \left(\sum_{H_s \in P} (f_s - 1) - 1\right) + \sum_{H_s \in Q} (f_s - 1)\right)^{c-i-2} \sum_{H_s \in P} (f_s - 1) \prod_{H_s \in Q} f_s$$

$$= \left(\sum_{j=2}^{c} (f_j - 1)\right)^{c-i-2} \sum_{H_s \in P} (f_s - 1) \prod_{H_s \in Q} f_s.$$

Thus, the size of subgroup $\Psi_{i,P}$ is

$$c_1 c_2 = \left(\sum_{j=2}^{c} (f_j - 1)\right)^{c-i-2} \sum_{H_s \in P} (f_s - 1) \prod_{H_s \in Q} f_s \prod_{H_s \in P} (f_1 f_s)$$

$$= \sum_{H_s \in P} (f_s - 1) \left(\sum_{j=2}^{c} (f_j - 1)\right)^{c-i-2} f_1^{i-1} \prod_{j=1}^{c} f_j.$$

Therefore the size of Ψ_i is

$$\sum_{\substack{P \subseteq \{H_2, \dots, H_c\} \\ |P| = i}} |\Psi_{i,P}| = \sum_{\substack{P \subseteq \{H_2, \dots, H_c\} \\ |P| = i}} \left(\sum_{H_s \in P} (f_s - 1) \left(\sum_{j=2}^{c} (f_j - 1)\right)^{c-i-2} f_1^{i-1} \prod_{j=1}^{c} f_j\right)$$

$$= \left(\sum_{\substack{P \subseteq \{H_2, \dots, H_c\} \\ |P| = i}} \sum_{H_s \in P} (f_s - 1)\right) \left(\sum_{j=2}^{c} (f_j - 1)\right)^{c-i-2} f_1^{i-1} \prod_{j=1}^{c} f_j$$

$$= \binom{c-2}{i-1} \sum_{j=2}^{c} (f_j - 1) \left(\sum_{j=2}^{c} (f_j - 1) \right)^{c-i-2} f_1^{i-1} \prod_{j=1}^{c} f_j$$

$$= \binom{c-2}{i-1} \left(\sum_{j=2}^{c} (f_j - 1) \right)^{c-i-1} f_1^{i-1} \prod_{j=1}^{c} f_j.$$

Finally, the size of C is

$$\sum_{i=1}^{c-1} |\Psi_i| = \sum_{i=1}^{c-1} \left(\binom{c-2}{i-1} \left(\sum_{j=2}^{c} (f_j - 1) \right)^{c-i-1} f_1^{i-1} \prod_{j=1}^{c} f_j \right)$$

$$= \sum_{i=0}^{c-2} \left(\binom{c-2}{i} \left(\sum_{j=2}^{c} (f_j - 1) \right)^{c-i-2} f_1^{i} \right) \prod_{j=1}^{c} f_j$$

$$= \left(1 + \sum_{j=1}^{c} (f_j - 1) \right)^{c-2} \prod_{j=1}^{c} f_j. \qquad \square$$

From the above discussion, it is not difficult to give a recursive procedure that generates all the adjacency relations on the set of faces of the subembeddings in Δ.

Acknowledgments. I thank R. E. Tarjan for introducing me to the counting problems solved in this paper. I also thank Matthias Stallmann for pointing out that my previous formula for counting embeddings of connected graphs with two biconnected components does not easily extend to the general case. That comment inspired my discovery of Lemma 13.

REFERENCES

[1] A. Aho, J. Hopcroft, and J. Ullman, *Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

[2] C. Berge, *The Theory of Graphs and Its Applications*, Alision Doig, trans., Methuen, London, 1964.

[3] K. S. Booth and G. S. Lueker, *Testing for the consecutive ones property, interval graphs, and graph planarity using* PQ-*tree algorithms*, J. Comput. System Sci., 13 (1976), pp. 335–379.

[4] J. Cai, X. Han, and R. E. Tarjan, *An m* log *n Algorithm for the Maximal Planar Subgraph Problem*, Tech. Report, Dept. of Computer Science, Princeton University, Princeton, NJ, 1991.

[5] N. Chiba, T. Nishizeki, S. Abe, and T. Ozawa, *A linear algorithm for embedding planar graphs using* PQ-*trees*, J. Comput. System Sci., 30 (1985), pp. 54–76.

[6] G. Di Battista and R. Tamassia, *Incremental planarity testing* (*extended abstract*), in Proc. 30th Annual IEEE Sympos. on Foundations of Computer Science, Research Triangle Park, NC, 1989, pp. 436–441.

[7] D. Hall and G. Spencer, *Elementary Topology*, John Wiley, New York, 1955.

[8] J. Hopcroft and R. Tarjan, *Efficient planarity testing*, J. Assoc. Comput. Mach., 21 (1974), pp. 549–568.

[9] R. Jayakumar, K. Thulasiraman, and M. N. S. Swamy, $O(n^2)$ *algorithms for graph planarization*, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, 8 (1989), pp. 257–267.

[10] A. Lempel, S. Even, and I. Cederbaun, *An algorithm for planarity testing of graphs*, in Theory of Graphs, Internat. Sympos., July 1966, Rome, pp. 215–232.

[11] M. Stallmann, *Using* PQ-*Trees for Planar Embedding Problems*, Tech. Report, North Carolina State University, Raleigh, NC, December 1985.

[12] ———, *Enumerating the Embeddings of a Planar Graph*, preliminary draft, North Carolina State University, Raleigh, NC, March 1989.

[13] W. T. Thron, *Introduction to the Theory of Functions of a Complex Variable*, John Wiley, New York, 1953.

[14] W. Wu, *On the planar imbedding of linear graphs*, J. Systems Sci. Math. Sci., 5 (1985), pp. 290–302.

SIAM J. DISC. MATH.
Vol. 6, No. 3, pp. 353-362, August 1993

INDUCED CYCLE STRUCTURES OF THE HYPEROCTAHEDRAL GROUP*

WILLIAM Y. C. CHEN†

Abstract. In this paper, the n-dimensional hypercube Q_n is treated as a graph whose vertex set consists of sequences of 0's and 1's of length n, and the hyperoctahedral group B_n is the automorphism group of Q_n. It is well known that B_n can be represented by the group of signed permutations, namely, any signed permutation induces a permutation on the vertices of Q_n, which preserves adjacency. Moreover, the set of signed permutations on n elements also induces a permutation group on the edges of Q_n, denoted H_n. The author studies the cycle structures of both B_n and H_n. The method proposed here is to determine the induced cycle structure by computing the number of fixed vertices or fixed edges of a signed permutation in the cyclic group generated by a signed permutation of given type. Here we define the type of a signed permutation by a double partition based on its signed cycle decomposition. In this way, one can compute the cycle indices of both B_n and H_n by counting fixed vertices and fixed edges of a signed permutation. The formula for the cycle index of B_n is much more natural and considerably simpler than that of Harrison and High [*J. Combin. Theory*, 4 (1968), pp. 277-299]. Meanwhile, the cycle structure of H_n seems not to have been studied before, although it is well motivated by nonisomorphic edge colorings of Q_n, as well as by the recent interest in edge symmetries of computer networks.

Key words. hypercube, hyperoctahedral group, induced cycle structure, Pólya theory

AMS subject classifications. 05A15, 05C25

1. Introduction.

The hyperoctahedral group considered in this paper will be understood as the automorphism group of the n-dimensional hypercube, or simply the n-cube. As in [1], we choose to treat the n-cube as a graph, usually denoted Q_n. To be more specific, the vertex set of Q_n consists of all the sequences of 0's and 1's of length n, and two such sequences are adjacent whenever they differ at exactly one position. Nevertheless, this standpoint is by no means substantially different from that of treating the hypercube as a regular solid in n-dimensional Euclidean space. The recent surge of interest in symmetry properties of computer networks has led to the investigation of automorphism groups, as well as the induced edge automorphism groups of currently studied network models, including the hypercube. Throughout, we use B_n to denote the automorphism group of Q_n, and H_n to denote the induced permutation group of B_n on the edges of Q_n. Sometimes the term *line-group* of a graph G is used for the permutation group on the edges of G induced by the automorphism group of G. In this sense, H_n is the line-group of Q_n.

In view of Pólya theory on enumeration under group action, an important feature of an automorphism group is its cycle structure [11], [13], [14]. In fact, the study of the cycle structure of B_n has an interesting history. From the signed permutation representation of B_n, namely, the fact that B_n can be represented by the wreath product of S_n and S_2, Pólya [12] noted that the number of types of Boolean functions in n variables equals the number of nonisomorphic vertex colorings of the n-cube by using two colors [3]-[7]. This led to the question of computing the cycle structure of B_n. More information about the origin of this problem can be found in a recent paper [18]. Although B_n is isomorphic to the wreath product $S_n[S_2]$, which is a permutation group on $2n$ elements whose cycle index can be obtained by those of S_n and S_2 in terms of the operation called *plethysm* or Pólya's composition, B_n itself is a more sophisticated permutation group on 2^n elements,

* Received by the editors March 17, 1992; accepted for publication (in revised form) August 12, 1992.

† C-3, Mail Stop B265, Los Alamos National Laboratory, Los Alamos, New Mexico 87545. chen@monsoon.c3.LANL.GOV

and it does not seem to possess relatively simple cycle structure. In fact, Pólya [12] computed the cycle index of B_n up to $n = 4$. The problem of counting types of Boolean functions received more attention with the advent of switching circuit theory. The first complete solution was obtained by Slepian [16] based on Young's results on irreducible representations of B_n. Later, Harrison and High [8] succeeded in obtaining the cycle index of B_n, which also leads to a solution to the problem of counting types of Boolean functions. However, the formula of Harrison and High is rather involved. Our method proves to be more natural and considerably simpler than that of Harrison and High; moreover, our approach is more effective regarding its applicability to more general situations such as the cycle structure of the line-group H_n of Q_n, a permutation group on $n2^{n-1}$ edges. It appears that the cycle structure of H_n has been untouched before, although it is well motivated by the enumeration of nonisomorphic edge colorings of Q_n, as well as by the recent interest in edge symmetries of computer networks.

Our first objective is to obtain the cycle polynomials of both B_n and H_n. As we know in many circumstances, such as counting types of Boolean functions, we do not really need all the information contained in the cycle index of B_n. Instead, for a permutation group G, sometimes it suffices to use the following polynomial:

$$K(G; x) = \frac{1}{|G|} \sum_k w_k x^k,$$

where w_k is the number of permutations in G with k cycles. Clearly, $K(G; x)$ can be obtained from the cycle index $Z(G; x_1, x_2, \cdots)$ of G by substituting each x_i with x. We call $K(G; x)$ the *cycle polynomial* of G. As expected, cycle polynomials are easier to compute than cycle indices. Keeping in mind that the signed permutation representation of B_n is considerably easier than B_n itself, we naturally expect that the cycle structure of B_n should follow somehow from the cycle structure of signed permutations. First, we observe a simple connection between the cycle structure of a permutation and the Burnside Lemma so that counting cycles reduces to counting fixed points. Second, by using a result of Chen and Stanley [1] concerning the number of fixed vertices of a symmetry of Q_n and our Cycle Splitting Lemma, we can compute the cycle structure of any permutation in B_n. The notion of balanced signed cycles defined in [1] proved to be crucial in our approach. We note that our method is not only effective for B_n and H_n, but also for other permutation groups induced by the wreath product of two permutation groups. It is interesting that the notion of double partitions used in the representation theory of B_n arises naturally in the present context, and that we can explicitly give the induced cycle structure of any signed permutation in terms of its type (in the form of a double partition). Furthermore, we can compute the cycle indices of both B_n and H_n by counting fixed vertices and fixed edges of a signed permutation of given type—the second objective of this paper.

2. The Cycle Counting Lemma. Let G be a group and S be a finite set. Let Π be a permutation group on S. Given a homomorphism ρ from G to Π,

$$\rho\colon g \rightarrow \pi_g, \qquad g \in G, \quad \pi_g \in \Pi,$$

we usually say that G is a group acting on S in the sense that an element of G acts on S through its image under the homomorphism ρ. With the homomorphism ρ being understood, we simply call Π an induced group of G. As far as we are concerned in this paper, G will be the wreath product $S_n[S_2]$, or the group of signed permutations on n elements, and ρ will be the isomorphism from G to the automorphism group B_n or the edge automorphism group H_n of Q_n. Specifically, the hyperoctahedral group B_n is an induced group of $S_n[S_2]$ acting on the vertices of Q_n. Given a signed permutation π, the acting rule (i.e., the isomorphism ρ as above) of π on Q_n is explained as permuting the sequence of 0's and 1's, and then taking complements at certain positions, the detailed definition will be given in the next section. Another induced group of $S_n[S_2]$ is the edge automorphism group H_n of Q_n. By definition, an automorphism of a graph induces a permutation on the edges. Thus, H_n is an induced group of $S_n[S_2]$ acting on the edges of Q_n. The objective of this paper is to study the cycle structures of B_n and H_n as permutation groups on the vertices and edges of Q_n.

Given an element g in a group G, suppose that it induces a permutation π_g on S. By the induced cycle structure of g, we mean the cycle structure of the induced permutation on S. We will use the Burnside Lemma to compute the number of cycles of an induced permutation of g. To this end, let us recall some basic terminology related to the Burnside Lemma. Given two elements s_1 and s_2 in S, we say s_1 is *equivalent* to s_2, denoted $s_1 \sim s_2$, if there exists an element $g \in G$ such that

$$\pi_g s_1 = s_2.$$

Then it is easy to verify that $\sim$ is an equivalence relation on S. For any $g \in G$, we denote by $\psi(g)$ the number of elements $s \in S$ such that $\pi_g s = s$, namely, the number of elements fixed by g. Then the Burnside Lemma states that the number of equivalence classes of S under $\sim$ equals

$$\frac{1}{|G|} \sum_{g \in G} \psi(g).$$

Using the Burnside Lemma, we may compute the number of cycles of an induced permutation in terms of the number of its fixed points.

LEMMA 2.1 (Cycle Counting Lemma). *Let G be a group acting on S, and $g \in G$. Then the number of cycles of the induced permutation π_g equals*

$$\frac{1}{o(g)} \sum_{\sigma \in (g)} \psi(\sigma),$$

where $o(g)$ is the order of g in G and $\psi(\sigma)$ is the number of elements in S fixed by σ.

Proof. We simply write π for π_g. To use the Burnside Lemma, we observe that two elements $s_1, s_2 \in S$ are in the same cycle in the cycle decomposition of π if and only if there exists a permutation $\sigma = \pi^i$ for some i such that $\sigma(s_1) = s_2$. Therefore, the number of cycles of π is the same as the number of equivalence classes of S under the action of the permutation group $(\pi) = \{e, \pi, \pi^2, \cdots\}$, which is clearly finite. □

3. The cycle polynomial of B_n.

We first recall some definitions from [1]. For any positive integer n, we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. We may represent an element $w \in B_n$ by a *signed permutation* on $[n]$, i.e., a permutation on $[n]$ with a + or − sign attached to each element $1, 2, \ldots, n$. For simplicity, we may omit the + signs. Thus $(\overset{+}{2}\ \overset{+}{4}\ \overset{-}{5})(\overset{-}{3})(\overset{+}{1}\ \overset{-}{6})$ or $(2\ 4\ \overset{-}{5})(3)(1\ \overset{-}{6})$ represents an element of B_6 with underlying permutation $(2\ 4\ 5)(3)(1\ 6)$ (written in cycle notation). We call such a representation of a signed permutation the *signed cycle decomposition*. Let w be a signed permutation with underlying permutation π. Then w acts on a vertex $u_1 u_2 \cdots u_n$ of Q_n by the following rule:

$$w(u_1 u_2 \cdots u_n) = v_1 v_2 \cdots v_n,$$

where

$$(3.1)\qquad v_j = \begin{cases} u_{\pi(j)}, & \text{if } j \text{ has the sign } +, \\ 1 - u_{\pi(j)}, & \text{if } j \text{ has the sign } -. \end{cases}$$

Thus the action of π on $u = u_1u_2\cdots u_n$ can be understood as the action of permuting u into $u_{\pi(1)}u_{\pi(2)}\cdots u_{\pi(n)}$, and then taking complements at positions where π has minus signs. If we define the *sign vector* $(s_1, s_2, \ldots, s_n)$ of a signed permutation w as

$$s_j = \begin{cases} 0, & \text{if } j \text{ has the sign } +, \\ 1, & \text{if } j \text{ has the sign } -, \end{cases}$$

then (3.1) can be rewritten as

$$(3.2)\qquad v_j \equiv u_{\pi(j)} + s_j \pmod 2.$$

In this way, a symmetry or an automorphism of Q_n can be represented by a pair (π, s), where π is a permutation on $[n]$ and s is a sign vector. For two symmetries π and σ of Q_n, we define their product by

$$(\pi\sigma)(u_1u_2\cdots u_n) = \sigma(\pi(u_1u_2\cdots u_n)),$$

where $u_1u_2\cdots u_n$ is any vertex of Q_n. Note that the above convention is consistent with the usual definition of product of two ordinary permutations; i.e., for two permutations π and σ on $[n]$, $\pi\sigma$ is defined by $(\pi\sigma)(i) = \sigma(\pi(i))$ for any i. If no confusion arises, we identify a signed permutation π with its underlying permutation when applied to an element in $[n]$ rather than a vertex of Q_n. It is straightforward to prove the following proposition.

PROPOSITION 3.1. *Suppose that* $\alpha = (\pi, s)$ *and* $\beta = (\sigma, t)$ *are two symmetries of* Q_n, *where* π *and* σ *are permutations on* $[n]$, *and* $s = (s_1, s_2, \ldots, s_n)$, $t = (t_1, t_2, \ldots, t_n)$ *are sign vectors. Then the symmetry* $\alpha\beta$ *has underlying permutation* $\sigma\pi$ *and sign vector*

$$\sigma(s) + t = (s_{\sigma(1)} + t_1, s_{\sigma(2)} + t_2, \ldots, s_{\sigma(n)+t_n}).$$

As noted by the referee, we can directly define B_n on the set of signed permutations by the following multiplication rule:

$$(\pi, s)(\sigma, t) = (\sigma\pi, \sigma(s) + t).$$

From this point of view, the proof of Proposition 3.1 becomes a verification of the fact that such a definition of the signed permutation group is a representation of the automorphism group of Q_n. The following corollary of Proposition 3.1 will be used later.

COROLLARY 3.2. *Let* $\alpha = (\pi, s)$ *be a signed permutation on* $[n]$, *with sign vector* $s = (s_1, s_2, \ldots, s_n)$. *Then* α^k *has underlying permutation* π^k *and sign vector*

$$(3.3)\qquad (s_1 + s_{\pi(1)} + \cdots + s_{\pi^{k-1}(1)}, \ldots, s_n + s_{\pi(n)} + \cdots + s_{\pi^{k-1}(n)}) \pmod 2.$$

A *double partition* (λ, μ) of an integer n, denoted $(\lambda, \mu) \vdash n$, is an ordered pair (λ, μ) of partitions such that $|\lambda| + |\mu| = n$, where $|\lambda|$ denotes the sum of all parts of λ [2], [10], [15], [17]. A double partition (λ, μ) can also be denoted by $(\lambda, \mu) \vdash (p, q)$, if $|\lambda| = p$ and $|\mu| = q$. The number of parts of λ will be denoted by $l(\lambda)$. Given two partitions λ and μ, we define $\lambda \cup \mu$ to be the partition obtained by joining the parts of λ and μ together. For example, $2\ 2\ 1 \cup 3\ 2\ 1 = 3\ 2\ 2\ 2\ 1\ 1$. The notion of a double partition is closely related to that of balanced cycles introduced in [1]. A signed cycle is said to be *balanced* if it contains an even number of minus signs; otherwise, it is called *unbalanced*. Moreover, a signed permutation is said to be *balanced* if every cycle in its signed cycle decomposition is balanced, and it is said to be *totally unbalanced* if every cycle in its signed cycle decomposition is unbalanced. Given a signed permutation π, the cycle type of π is defined by a double partition (λ, μ) such that λ is the cycle type of balanced cycles of π, and μ is the cycle type of unbalanced cycles of π. For example, the type of the signed permutation $(3\ \bar{7}\ 4)(1\ 5\ \bar{6}\ \bar{2})(8\ 10)(\bar{9})$ is $(2\ 4,\ 1\ 3)$. It is not difficult to see that two signed permutations belong to the same conjugacy class of B_n if and only if they have the same type.

For a partition $\lambda = 1^{\lambda_1} 2^{\lambda_2} \cdots n^{\lambda_n}$ of n, i.e., the number of i occurs λ_i times in λ for any i, we use $[{n \atop \lambda}]$ to denote the number of permutations on $[n]$ of type λ. It is well known that

$$\begin{bmatrix} n \\ \lambda \end{bmatrix} = \frac{n!}{1^{\lambda_1}\lambda_1! 2^{\lambda_2}\lambda_2! \cdots}.$$

Given a double partition $(\lambda, \mu) \vdash (p, q)$ of n, it is not difficult to show that the number of signed permutations of type (λ, μ) equals

$$(3.4) \qquad \binom{n}{p} \begin{bmatrix} p \\ \lambda \end{bmatrix} \begin{bmatrix} q \\ \mu \end{bmatrix} 2^{n - l(\lambda) - l(\mu)}.$$

Suppose that $S \cup T$ is a disjoint union of $[n]$ such that $|S| = p$ and $|T| = q$. Consider all balanced permutations π of type λ on the set S. Given an underlying cycle of length m, there are 2^{m-1} ways to form a balanced cycle by attaching a sign to each element in the underlying cycle. Thus, given an underlying permutation of type λ on S, we can form $2^{p - l(\lambda)}$ balanced permutations. A similar argument shows that given any underlying permutation of type μ on T, we may form $2^{q - l(\mu)}$ totally unbalanced permutations. Combining these two arguments, we obtain (3.4).

The following lemma gives the parity of the number of minus signs in each cycle of the signed permutation π^k, where the underlying permutation of π is a cycle. We follow the usual notation (i, j) for the greatest common divisor of i and j.

LEMMA 3.3 (Cycle Splitting Lemma). *Suppose that π is a signed permutation whose underlying permutation is a cycle of length n, and suppose that π has Δ minus signs. Then π^k can be decomposed into (k, n) signed cycles with each of length $n/(k, n)$. Moreover, the number of minus signs in each signed cycle of π^k is congruent to $k/(k, n)\Delta$ modulo* 2.

Proof. Without loss of generality, we may assume that π has underlying permutation $C = (1\ 2 \cdots n)$. Let $\delta = (\delta_1, \delta_2, \ldots, \delta_n)$ be the sign vector of π; it is known that C^k can be decomposed into (k, n) cycles with each of length $n/(k, n)$. Thus, the underlying cycle decomposition of π^k also has (k, n) cycles with each of length $n/(k, n)$. Let $d = (k, n)$. In general, a cycle of C^k containing the element i has the following form:

$$\begin{gathered} i \to i + k, \\ i + k \to i + 2k, \\ \vdots \\ i + (n/d - 1)k \to i, \end{gathered}$$

where all the numbers in the above diagram are taken modulo n. Let $(\theta_1, \theta_2, \ldots, \theta_n)$ be the sign vector of π^k. Since $C(j) \equiv j + 1 \pmod{n}$, we have $C^k(j) \equiv j + k \pmod{n}$. Applying Corollary 3.2, it follows that

$$\theta_i \equiv \delta_i + \delta_{i+1} + \cdots + \delta_{i+k-1} \qquad (\text{mod } 2).$$

The number of minus signs contained in the above cycle is congruent to $\theta_i + \theta_{k+i} + \cdots + \theta_{(n/d-1)k+i}$ modulo 2. Then we have

$$\begin{aligned}\sum_{j=0}^{n/d-1} \theta_{jk+i} &\equiv \sum_{l=0}^{k-1} \sum_{j=0}^{n/d-1} \delta_{jk+l} \\ &\equiv (\delta_i + \delta_{i+1} + \cdots + \delta_{i+k-1}) \\ &\quad + (\delta_{i+k} + \delta_{i+k+1} + \cdots + \delta_{i+2k-1}) + \cdots \\ &\quad + (\delta_{i+(n/d-1)k} + \delta_{i+(n/d-1)k+1} + \cdots + \delta_{i+(n/d)k-1}) \pmod 2.\end{aligned}$$

Note that $(n/d)k \equiv 0 \pmod n$. Thus, $i + (n/d)k - 1$ and i can be regarded as consecutive numbers (mod n) so that all the above summands can be arranged on a circle of length $(n/d)k$. Since all the indices of δ in the above summation are taken modulo n, the above sum can be further simplified to

$$\begin{aligned}&\delta_1 + \delta_2 + \cdots + \delta_{(n/d)k} \\ &= \delta_1 + \delta_2 + \cdots + \delta_{(k/d)n} \\ &= (k/d)(\delta_1 + \delta_2 + \cdots + \delta_n) \\ &= (k/d)\Delta.\end{aligned}$$

Hence the number of minus signs in each cycle of π^k is congruent to $(k/d)\Delta$ modulo 2. □

By the above lemma, it can be seen that, if π is a balanced cycle, then π^k is balanced for any k and that, if π is an unbalanced cycle of length n, then π^k is balanced or totally unbalanced according to whether $k/(k, n)$ is even or odd. Furthermore, the Cycle Splitting Lemma can be used to determine the cycle structure of π^k based on the cycle structure of π.

LEMMA 3.4. *Let π be an unbalanced cycle of length n, and let k be a positive integer. If we write n and k as $n = 2^i s$ and $k = 2^j t$, where both s and t are odd, then π^k is balanced if and only if $j > i$.*

Proof. Since s and t are odd, we have

$$\frac{k}{(k, n)} = \frac{2^j t}{(2^i s, 2^j t)} = \frac{2^j}{2^{\min(i,j)}} \cdot \frac{t}{(s, t)}.$$

Then it is easy to see that $k/(k, n)$ is even if and only if $j > i$. By the Cycle Splitting Lemma, it follows that π^k is balanced if and only if $k/(k, n)$ is even. This completes the proof. □

We now recall a result from [1] concerning the number of fixed vertices of a symmetry of Q_n. This result, together with Lemmas 3.3 and 3.4, will be sufficient to yield the cycle polynomial of B_n.

PROPOSITION 3.5. *Let π be a symmetry of Q_n represented by a signed permutation. If π is balanced, then it has 2^k fixed vertices, where k is the number of balanced cycles of π; otherwise, π has no fixed vertex.*

To describe the main result of this section, we need the following notation. Let λ be a partition of n, and let π be a permutation on $[n]$ of type λ. We use $C_\lambda(x)$ to denote the cycle polynomial of the cyclic group (π), and we call it the *cyclic polynomial* of λ. For a permutation π of type λ, it is easy to see that the order of the π equals $[\lambda]$, where $[\lambda]$ stands for the least common multiple of the components of λ. Let $\lambda = 1^{\lambda_1} 2^{\lambda_2} \cdots n^{\lambda_n}$. Since for any cycle C of length i, C^k decomposes into (i, k) cycles with each of length $i/(i, k)$, the cycle structure of π^k, denoted λ^k, is given by

$$\lambda^k = \prod_i [i/(i, k)]^{(i,k)\lambda_i}. \tag{3.5}$$

It follows from (3.5) that the number of cycles in π^k equals

$$l(\lambda^k) = \sum_i (i, k)\lambda_i. \tag{3.6}$$

Thus the cyclic polynomial of λ is given by

$$C_\lambda(x) = \frac{1}{[\lambda]} \sum_{k=0}^{[\lambda]} x^{\sum_{i=1}^{n}(i,k)\lambda_i}. \tag{3.7}$$

We are now are ready to state the main result of this section.

THEOREM 3.6. *Let (λ, μ) be a double partition of n, and let i be the maximum number such that 2^i is a factor of some part of μ. Set $r = 2^{i+1}$ if $\mu \neq 0$; otherwise set $r = 1$. Suppose that π is a signed permutation on $[n]$ of type (λ, μ). Then the number of induced cycles of π acting on the vertices of Q_n equals*

$$\frac{1}{r} C_{\lambda^r \cup \mu^r}(2) = \frac{1}{r[\lambda^r, \mu^r]} \sum_{k=1}^{[\lambda^r,\mu^r]} 2^{l(\lambda^{rk} \cup \mu^{rk})}. \tag{3.8}$$

Proof. By Lemma 2.1, the number of reduced cycles of π on the vertices of Q_n is determined by the number of fixed vertices of the signed permutations π^k. If $\mu = 0$, then by definition we have $r = 1$, and (3.8) follows from Lemma 2.1 and Proposition 3.5. We now assume that $\mu \neq 0$. By Proposition 3.5, π^k does not have any fixed vertex if π^k is not balanced. To make π^k balanced, by Lemma 3.4, k must contain the factor r; otherwise, there exists an unbalanced cycle θ of π such that θ^k is totally unbalanced. In other words, π^k has no fixed vertex unless $\pi^k \in (\pi^r)$. Clearly, π^r is a balanced permutation of type $\lambda^r \cup \mu^r$. Suppose that π is of order m. Since the identity permutation is balanced, it follows that m must contain the factor r. Since π^r is balanced, the order of (π^r) is just the order of an ordinary permutation of type $\lambda^r \cup \mu^r$, which is $[\lambda^r, \mu^r]$. Therefore, the order of π equals $m = r[\lambda^r, \mu^r]$. By Lemma 2.1, it follows that the number of induced cycles of π on the vertices of Q_n equals

$$\frac{1}{r[\lambda^r, \mu^r]} \sum_k 2^{(\text{the number of cycles of } \pi^{rk})} = \frac{1}{r} C_{\lambda^r \cup \mu^r}(2). \qquad \square$$

COROLLARY 3.7. *The cycle polynomial of B_n is given by*

$$\frac{1}{2^n n!} \sum_{p+q=n} \binom{n}{p} \sum_{(\lambda,\mu) \vdash (p,q)} \begin{bmatrix} p \\ \lambda \end{bmatrix} \begin{bmatrix} q \\ \mu \end{bmatrix} 2^{n-l(\lambda)-l(\mu)} x^{C_{\lambda^r \cup \mu^r}(2)/r},$$

where r is given as in Theorem 3.6.

By Pólya's theorem, the number of nonisomorphic vertex colorings of Q_n by using m colors equals the cycle polynomial of B_n evaluated at $x = m$. In particular, for $m = 2$, it yields the number of types of Boolean functions in n variables.

4. The cycle polynomial of H_n. In this section, we restrict ourselves to induced permutations of signed permutations on the edges of Q_n. In the same vein of the preceding section, we expect that the number of cycles in an induced permutation on the edges of Q_n depends only on the type of the original signed permutation. The aim of this section is to compute the number of induced cycles on the edges of Q_n of a signed permutation of given type. To this end, we first consider the number of fixed edges of a signed permutation of given type. Again, a signed permutation is considered to act on edges of Q_n through its induced permutation. Now we need the following result from [1]: Let π be a signed permutation acting on the edges of Q_n; then π has a fixed edge if and only if π is balanced and contains a 1-cycle, or π contains an unbalanced 1-cycle and all the other cycles are balanced (namely, π is of type $(\lambda, 1)$, where λ is a partition of $n-1$). Using this result, we may derive the number of fixed edges of a signed permutation of given type.

PROPOSITION 4.1. *Let π be a signed permutation acting on the edges of Q_n. If π is balanced and of type λ, then it has $\lambda_1 2^{l(\lambda)-1}$ fixed edges. If π is of type $(\lambda, 1)$, then it has $2^{l(\lambda)}$ fixed edges.*

Proof. We first consider the case when π is balanced. If $\lambda_1 = 0$, i.e., π has no 1-cycle, then it has no fixed edge either. So we may assume that $\lambda_1 \geqq 1$. As in [1], an edge of Q_n is represented by a sequence of $n-1$ 0's or 1's with one occurrence of the symbol $*$. For example, $00101*10$ denotes the edge joining the vertices 00101010 and 00101110. Treating π as a symmetry on the vertices of Q_n, it then fixes an edge, $a_1 \cdots a_{i-1} * a_{i+1} \cdots a_n$, if and only if π contains the 1-cycle (i) (by the separation argument in [1]). In such a case, $a_1 \cdots a_{i-1} a_{i+1} \cdots a_n$ becomes a fixed vertex for the signed permutation π' obtained from π by removing the cycle (i). Thus, by Proposition 3.5, there are $2^{l(\pi')} = 2^{l(\pi)-1}$ choices for the subsequence $a_1 \cdots a_{i-1} a_{i+1} \cdots a_n$. Moreover, considering all the 1-cycles of π, there are λ_1 choices for the position of $*$, so that the total number of fixed edges of π equals $\lambda_1 2^{l(\pi)-1}$.

Let us now consider the case when π is of type $(\lambda, 1)$, that is, π contains only one unbalanced 1-cycle, say, $(\bar{i})$, and all other cycles of π are balanced. Then the separation argument of [1] shows that the symbol $*$ must appear at the ith position in the above representation of fixed edges of π. Thus, a fixed edge of π is of the form $a_1 \cdots a_{i-1} * a_{i+1} \cdots a_n$, and the number of choices for the subsequence $a_1 \cdots a_{i-1} a_{i+1} \cdots a_n$ equals $2^{l(\lambda)}$, which makes the number of fixed edges of π. □

Analogous to the strategy of computing the cycle polynomial of B_n, to compute the cycle polynomial of H_n we need to count the number of induced cycles on the edges of Q_n of a signed permutation of given type. Because of the two cases in Proposition 4.1, we proceed accordingly. For a partition α, we use $\beta_j(\alpha)$ to denote the number of occurrences of j in α. Let $\lambda = 1^{\lambda_1} 2^{\lambda_2} \cdots n^{\lambda_n}$. From (3.5), it follows that

$$\beta_1(\lambda^k) = \sum_{i \mid k} i\lambda_i. \tag{4.1}$$

We now give the main result of this section, which leads to the cycle polynomial of H_n.

THEOREM 4.2. *Suppose that π is a signed permutation of type $(\lambda, 1)$; then the number of induced cycles of π equals*

$$\frac{1}{2[\lambda^2]} \left(\sum_{k=1}^{2[\lambda^2]} 2^{l(\lambda^k)} + \sum_{k=1}^{[\lambda^2]} \beta_1(\lambda^{2k}) 2^{l(\lambda^{2k})} \right). \tag{4.2}$$

If π is a signed permutation of type (λ, μ), where $\mu \neq 1$, then the number of induced cycles of π is given by

$$\frac{1}{r[\gamma]} \sum_{k=1}^{[\gamma]} \beta_1(\gamma^k) 2^{l(\lambda^k)-1}, \tag{4.3}$$

where r is defined as in Theorem 3.6 and $\gamma = \lambda^r \cup \mu^r$.

Proof. We first prove (4.2). Suppose that π is of type $(\lambda, 1)$. Recall that, for such a type, the number r equals 2, and the order of π equals $2[\lambda^2]$. If k is odd, then π^k is of type $(\lambda^k, 1)$. By Proposition 4.1, the number of edges fixed by π^k equals $2^{l(\lambda^k)}$. If k is even, then π^k is balanced and of type $\lambda^k \cup 1$, and from Proposition 4.1 it follows that the number of fixed edges of π^k equals

$$\beta_1(\lambda^k \cup 1)2^{l(\lambda^k \cup 1)-1} = (\beta_1(\lambda^k) + 1)2^{l(\lambda^k)} = 2^{l(\lambda^k)} + \beta_1(\lambda^k)2^{l(\lambda^k)}.$$

Hence by Lemma 2.1, the number of induced cycles of π adds up to (4.2).

Next, we prove (4.3). Suppose that π is of type (λ, μ), where $\mu \neq 1$. We claim that π^k does not have any fixed edge unless π^k is balanced. We may assume that $\mu \neq 0$; otherwise, the claim becomes obvious. Suppose that π^k is not balanced. Then there exists an unbalanced cycle θ of π such that θ^k contains an unbalanced cycle. By the Cycle Splitting Lemma, every cycle of θ^k must be unbalanced. Let i be the length of the cycle θ; then θ^k contains (i, k) cycles with each of length $i/(i, k)$. If $i > 1$, then either $(i, k) > 1$ or $i/(i, k) > 1$; that is, θ^k contains either an unbalanced cycle of length at least 2 or at least two unbalanced 1-cycles. By Proposition 4.1, π^k cannot have any fixed edge. We now consider the case when θ^k is balanced for every unbalanced cycle θ of π with length at least two. If such a cycle θ exists, then k must be even (by Lemma 3.4). Thus π must be balanced because for any signed 1-cycle σ, σ^k is balanced whenever k is even. Finally, we are left with the case when π does not have any unbalanced cycles of length at least two. Since $\mu \neq 1$, π has at least two unbalanced 1-cycles. Therefore, for any odd number k, π^k has the same number of unbalanced 1-cycles as π, which implies that π^k has no fixed edge and for any even number k, π^k becomes balanced. Thus, we have arrived at the conclusion that π^k does not have any fixed edge unless π^k is balanced. As we have shown in the proof of Theorem 3.6, π^k is balanced if and only if $\pi^k \in \langle \pi^r \rangle$. By Proposition 4.1, π^{rk} has $\beta_1(\gamma^k)2^{l(\gamma^k)-1}$ fixed edges. Since π is of order $r[\gamma]$, by Lemma 2.1, we obtain (4.3). □

Similar to Corollary 3.7, Theorem 4.2 gives the cycle polynomial of H_n by summing over all double partitions of n. Let $K(H_n; x)$ be the cycle polynomial of H_n; then, by Pólya's theorem, $K(H_n; m)$ gives the number of nonisomorphic edge-colorings of Q_n by using m colors.

5. Cycle indices of B_n and H_n. In view of Pólya's theorem, the cycle index of a permutation group gives the generating function of nonisomorphic coloring patterns, which contains more information than just the number of nonisomorphic colorings. For this reason, sometimes it is necessary to know the cycle index of a permutation group. In this section, we achieve this goal for both B_n and H_n. The cycle index of B_n has been computed by Harrison and High [8] in a rather complicated way, but our formula is much more natural and clearer. Moreover, our formula for H_n is believed to be new.

Again, we resort to the number of fixed vertices and the number of fixed edges of a signed permutation of given type. Given any permutation π of type λ on a set S, by (4.1) and the number theoretic Möbius inversion formula, it immediately follows that the cycle structure of π is, in fact, determined by the number of fixed points of π^k for $1 \leqq k \leqq o(\pi)$, where $o(\pi)$ is the order of π. For clarity, we state this fact as follows.

PROPOSITION 5.1. *Let π be a permutation on S; then the number of k-cycles of π is given by*

$$\frac{1}{k}\sum_{i|k} \mu(k/i)\psi(\pi^i), \tag{5.1}$$

where μ is the classical Möbius function, and $\psi(\pi^i)$ is the number of fixed points of π^i.

As expected, the purpose of the remainder of this paper is to obtain the induced cycle structures on the vertices and edges of Q_n of a signed permutation of given type. In accordance with the above proposition, this problem reduces to computing the number of fixed vertices and fixed edges of a signed permutation π^k based on the type of π. At this point, we have already encountered these numbers in computing the cycle polynomials of B_n and H_n. In the proofs of Theorem 3.6 and Theorem 4.2, we have actually shown the following two propositions. Recall that for a double partition (λ, μ), the number r is determined by μ as in Theorem 3.6.

PROPOSITION 5.2. *Let π be a signed permutation of type (λ, μ); then π^k has $2^{l(\lambda^k \cup \mu^k)}$ fixed vertices if $r | k$; otherwise, π^k has no fixed vertex.*

PROPOSITION 5.3. *Suppose that π is a signed permutation of type $(\lambda, 1)$; then π^k has $2^{l(\lambda^k)}$ fixed edges if k is odd; otherwise, π^k has $(\beta_1(\lambda^k) + 1)2^{l(\lambda^k)}$ fixed edges. If π is a signed permutation of type (λ, μ), where $\mu \neq 1$, then the number of fixed edges of π^k is given by $\beta_1(\lambda^k \cup \mu^k)2^{l(\lambda^k \cup \mu^k)-1}$ if $r | k$; otherwise, π^k has no fixed edge.*

Finally, we note that the maximum length of an induced cycle of a signed permutation π of type (λ, μ) is bounded by the order of π, which has been shown to be $r[\lambda^r \cup \mu^r]$. Since the number of signed permutations of given type is determined in (3.4), like Corollary 3.7, the cycle indices of B_n and H_n can be obtained by summing the cycle structures of signed permutations π of type (λ, μ) over all double partitions (λ, μ).

Acknowledgment. I thank the referee for many helpful comments on an early version of this paper.

REFERENCES

[1] W. Y. C. CHEN AND R. P. STANLEY, *Derangements on the n-cube*, Discrete Math., to appear.

[2] L. GEISSINGER AND D. KINCH, *Representations of the hyperoctahedral groups*, J. Algebra, 53 (1978), pp. 1-20.

[3] N. GRAHAM, *An Investigation of Hypercube Invariants*, Ph.D. thesis, New Mexico State Univ., Las Cruces, NM, 1989.

[4] F. HARARY AND E. PALMER, *The power group enumeration theorem*, J. Combin. Theory, 1 (1966), pp. 157-173.

[5] ———, *Graphical Enumeration*, Academic Press, New York, 1973.

[6] M. A. HARRISON, *Introduction to Switching and Automata Theory*, McGraw-Hill, New York, 1965.

[7] ———, *Counting theorems and their applications to classification of switching functions*, in Recent Developments in Switching Theory, A. Mukhopadhyay, ed., Academic Press, New York, 1971, pp. 85-120.

[8] M. A. HARRISON AND R. G. HIGH, *On the cycle index of a product of permutation groups*, J. Combin. Theory, 4 (1968), pp. 277-299.

[9] HARVARD COMPUTATION LABORATORY STAFF, *Synthesis of electronic computing and control circuits*, Cambridge, MA, 1951.

[10] G. JAMES AND A. KERBER, *The Representation Theory of the Symmetric Group*, Addison-Wesley, Reading, MA, 1981.

[11] G. PÓLYA, *Kombinatorische Anzahlbestimmungen für Gruppen, Graphen, und chemische Verbindungen*, Acta Math., 68 (1937), pp. 145-253.

[12] ———, *Sur les types des propositions composées*, J. Symbolic Logic, 5 (1940), pp. 98-103.

[13] G. PÓLYA AND R. C. READ, *Combinatorial Enumeration of Groups, Graphs, and Chemical Compounds*, Springer-Verlag, New York, 1987.

[14] J. H. REDFIELD, *The theory of group reduced distributions*, Amer. J. Math., 49 (1927), pp. 433-455.

[15] V. REINER, *Signed permutation statistics and cycle type*, European J. Combin., to appear.

[16] D. SLEPIAN, *On the number of symmetry types of Boolean functions of n variables*, Canad. J. Math., 5 (1953), pp. 185-193.

[17] R. P. STANLEY, *Some aspects of groups acting on finite posets*, J. Combin. Theory Ser. A, 32 (1982), pp. 132-161.

[18] E. M. PALMER, R. C. READ, AND R. W. ROBINSON, *Balancing the n-cube: A census of colorings*, J. Algebraic Combinatorics, 1 (1992), pp. 257-273.

SIAM J. DISC. MATH.
Vol. 6, No. 3, pp. 363–374, August 1993

COLLISIONS AMONG RANDOM WALKS ON A GRAPH*

DON COPPERSMITH†, PRASAD TETALI‡, AND PETER WINKLER§

Abstract. A token located at some vertex v of a connected, undirected graph G on n vertices is said to be taking a "random walk" on G if, whenever it is instructed to move, it moves with equal probability to any of the neighbors of v. The authors consider the following problem: Suppose that *two* tokens are placed on G, and at each tick of the clock a certain demon decides which of them is to make the next move. The demon is trying to keep the tokens apart as long as possible. What is the expected time M before they meet?

The problem arises in the study of self-stabilizing systems, a topic of recent interest in distributed computing. Since previous upper bounds for M were exponential in n, the issue was to obtain a polynomial bound. The authors use a novel potential function argument to show that in the worst case $M = (\frac{4}{27} + o(1))n^3$.

Key words. random walk, graph, Markov chain, collision, token management

AMS subject classifications. 60J15, 68E10, 05C35

1. Introduction. Let G be a connected graph on n vertices and let v be a fixed vertex of G. A *random walk* on G, beginning at v, is a stochastic process whose state at any time t is given by a vertex of G; at time 0, it is at vertex v, and, if at time t it is at vertex u, then at time $t + 1$ it will be at one of the neighbors of u, each neighbor having been chosen with equal probability.

The random walk thus constitutes a Markov chain, with state transition probability $p_{x,y} = 0$ if y is not adjacent to x, and $p_{x,y} = 1/d(x)$ if y is adjacent to x, and x has degree $d(x)$. The Markov chain will be irreducible (unless G is bipartite), and it is easily verified that its stationary distribution π satisfies $\pi_x = d(x)/2m$, where m is the number of edges of G.

Thus, we have that, in the limit, the probability of being at any particular vertex is proportional to its degree—regardless of the structure of G. This remarkable fact is the key to numerous applications.

In Aleliunas et al. [4], random walks are used to establish the existence of short universal sequences for traversing graphs; in Doyle and Snell [12], they are elegantly associated with electrical networks; in Borre and Meissl [5], they are employed to estimate measurements given by approximate differences. Broder [7] and Jerrum and Sinclair [15] made use of random walks on graphs to obtain the first randomized polynomial-time algorithm for approximating the value of the permanent of a matrix; see also Dagum et al. [10]. Random walks were used by Dyer, Frieze, and Kannan [13] to estimate the volume of a convex body and by Karzanov and Khachiyan [16] to sort partial orders when comparisons are expensive. Recently, Coppersmith et al. [8] have found an application of random walks to on-line algorithms.

Aldous [1] gives many other contexts in which random walks on graphs arise, and a valuable bibliography [2] compiled by the same author lists numerous additional references on the subject.

In this work, we are motivated by the work of Israeli and Jalfon [14] on self-stabilizing token management schemes. A protocol for a distributed computing network is said to be *self-stabilizing* if, no matter what state it is begun in (or perturbed to), it eventually enters a "legal" state and resumes normal operation.

* Received by the editors January 21, 1992; accepted for publication (in revised form) August 20, 1992.

† T. J. Watson Research Center, IBM, Yorktown Heights, New York 10598.

‡ AT&T Bell Laboratories, Murray Hill, New Jersey 07974.

§ Bellcore, Morristown, New Jersey 07962.

In a token management scheme, only one processor at a time is supposed to be "enabled" to change state or perform some particular task. This processor is said to possess the token, the token being an abstract object that is passed from processor to processor like the baton in a relay race.

The obvious problems in designing a self-stabilizing token management scheme are in recovering from (a) a situation in which no token is present and (b) a situation in which several tokens are present. The former can be neatly avoided by allowing a processor to use information from its neighbors to determine whether it has a token. For example, in Israeli and Jalfon's scheme, each processor has a special token-management register, and a processor deems itself to possess the token just when the value in its register is at least as great as the value in any of its neighbor's.

Then, of course, if x is the highest value in any processor's token management register, all processors holding x possess a token—as do other processors that hold "local" maxima.

To reduce the number of tokens to 1, the tokens are passed randomly to neighbors, and, whenever two or more collide, they merge to become a single token. The tokens do not, however, move simultaneously; the processors are not synchronized but are fired at the will of a "demon" who, in the worst case, is trying to delay stabilization. Naturally, we must assume that the demon must activate token-possessing processors from time to time, so the question becomes the following: What is the *expected* number of such activations before the tokens have collapsed to 1?

If said time is polynomial in n, then it follows from Markov's inequality that, in polynomial time, we can make the probability of stabilization as close to 1 as desired; furthermore, it clearly suffices to consider the case where the system begins with just two tokens. Accordingly, we define the *meeting time* $M_G(u, v)$ to be the expected number of moves before tokens placed initially at vertices u and v of G collide, given optimal (delaying) play by the demon in deciding at each step which token moves.

Israeli and Jalfon noted that, if G is an n-cycle, that is, if the processors are placed in a ring, then it makes no difference which of the two tokens is moved; hence $M_G(u, v)$ is bounded by the "hitting time" (definitions below) across the cycle, which is about $n^2/4$.

However, in a general connected, undirected, n-vertex graph G, they were able only to get the exponential upper bound

$$M_G(u, v) = O((\Delta - 1)^{D-1}),$$

where Δ is the maximum degree of G and D is its diameter.

Our main contribution has been to obtain a polynomial upper bound in the general case, namely,

$$M_G(u, v) \leqq \tfrac{4}{27}n^3 \quad \text{plus lower-order terms.}$$

Our primary technique is the use of a somewhat odd-looking potential function to bound the meeting time; the function is designed to drop by 1 in expected value no matter which token is moved. Using distance and electrical resistance arguments, we are then able to reduce the constant to $\frac{4}{27}$, which is the best possible.

Since the demon is an adversary, our results apply also to the case where the decision as to which token moves next is random and to the case where the tokens move simultaneously. A recent result of Aldous [3] implies that, in the latter case, meeting time is bounded by a (large) constant times the maximum hitting time. Thus our results generalize Aldous's to cover the adversarial case, and it happens that we also lower Aldous's constant to 1. However, it should be noted that we do *not* determine the adversary's optimal strategy in the general case.

2. Notation and preliminaries. The *hitting time* $H_G(x, y)$ from x to y is defined to be the expected number of steps for a random walk on G beginning at vertex x to reach vertex y for the first time. Thus, for example, if x and y are at opposite ends of a path on n vertices, then we have a "standard" random walk with reflecting barrier, and any of a number of arguments shows that the hitting time from x to y is precisely $(n-1)^2$.

We might at first be tempted to guess that this represents the largest possible hitting time in an n-vertex graph, but, in fact, it has been known for many years that there are n-vertex graphs G (barbells, for example) with vertices x, y such that $H_G(x, y)$ is $\Omega(n^3)$. The precise upper bound for $H_G(x, y)$ was found by Brightwell and Winkler [6]; the unique extremal graph is a "lollipop" consisting of a clique on $m = \lfloor(2n+1)/3\rfloor$ vertices with a path on the remaining $n - m$ vertices attached at one end. The start vertex x is in the clique, and the end vertex y is, of course, at the far end of the path. The hitting time proves to be precisely

$$\tfrac{4}{27}n^3 - \tfrac{1}{9}n^2 + \tfrac{2}{3}n - 1 + c,$$

where $c = 0, -\frac{2}{27}$, or $-\frac{2}{9}n + \frac{14}{27}$, according as $n \equiv 0, 1$, or $2 \bmod 3$, respectively.

This same value is thus a *lower* bound for the worst-case value of $M_G(x, y)$, obtainable if the tokens begin as described above and the demon moves only the token that started at x (this is, in fact, the demon's best strategy on the lollipop). However, as we see, it is *not* generally the case that, on a fixed graph G, the maximum meeting time is bounded by the maximum hitting time.

Let us now fix an n-vertex graph G. A (possibly randomized) *strategy* S for the demon on G consists of instructions that tell the demon, for each possible starting position and progress of the game so far and each possible value of some random variable, which token he should move. We let $M^S(x, y)$ be the expected number of moves until the tokens meet, when the game is begun with tokens at x and y, and the demon follows strategy S.

An *optimal* strategy is one for which $M^S(x, y)$ is maximal, that is, equal to $M_G(x, y)$, for every x and y; a *pure* strategy is one in which the demon's choice at each turn depends only on the current locations of the tokens. Note that a pure strategy is equivalent to a *tournament* on the vertices of G; from each pair u, v of vertices, the "winner" is the vertex from which the demon will move the token when the tokens are located on u and v.

For convenience, we adopt the following convention: If f is any real-valued function on the vertices of a graph, then $f(\bar{v})$ is defined to be the average of $f(u)$ over all neighbors u of v. Thus, for example, hitting time satisfies $H_G(x, y) = 1 + H_G(\bar{x}, y)$ for all distinct x and y.

3. Lemmas.

LEMMA 1. *On any graph G, the demon has a pure optimal strategy.*

Proof. For any two distinct vertices x and y, let $S(x, y)$ be a strategy maximizing $M^{S(x,y)}(x, y)$. Define a tournament T by letting x be the winner over y when, given a starting position with tokens at x and y, $S(x, y)$ moves the token at x, similarly for y. If either token may be moved, we assign a winner arbitrarily. We claim that the pure strategy S corresponding to the tournament T is optimal.

If not, let $\alpha > 0$ be the maximum value of $M_G(x, y) - M^S(x, y)$, and, of all pairs x, y attaining this discrepancy, choose one of minimum distance. Assume that x beats y in T and that, with tokens starting at x and y, $S(x, y)$ moves x with probability $p > 0$. Then

$$M^{S(x,y)}(x, y) = 1 + pM_G(\bar{x}, y) + (1-p)M_G(x, \bar{y}),$$

since $S(x, y)$ is supposed to be optimal at x, y; furthermore, we must have

$$M^{S(x,y)}(x, y) = 1 + M_G(\bar{x}, y),$$

else moving the token at y would be a superior strategy. Then, however,

$$\begin{aligned} M^{S(x,y)}(x, y) &= 1 + M_G(\bar{x}, y) \\ &\leqq 1 + M^S(\bar{x}, y) + \alpha \\ &= M^S(x, y) + \alpha \\ &= M_G(x, y) \\ &= M^{S(x,y)}(x, y). \end{aligned}$$

The catch is that the inequality in the middle is strict since one of the neighbors z of x must be at smaller distance to y than x was; thus $M_G(z, y) - M^S(z, y) < \alpha$ or $z = y$, so that $M_G(z, y) = 0$. Either way, the contradiction proves the lemma. □

We remark that Lemma 1 can also be proved by formulating it in terms of Markov decision processes and applying the general theory, as found, e.g., in [11].

The next lemma establishes a critical relation among hitting times.

LEMMA 2. *Let x, y, and z be vertices of a connected, undirected graph G. Then*

$$H_G(x, y) + H_G(y, z) + H_G(z, x) = H_G(x, z) + H_G(z, y) + H_G(y, x).$$

Proof. Essentially, this equality is a consequence of the reversibility of the Markov chain for random walks on an undirected graph. Note that the left-hand side of the equation in the lemma is the expected time for a random walk to go from x to y, then to z and back to x, and similarly for the right.

Now fix a number r and begin a random walk at x, ending when x is reached again for the rth time. Let $x, v_1, v_2, \ldots, v_k, x$ be the outcome of the walk; its probability is

$$\frac{1}{d(x)} \prod_{i=1}^{k} \frac{1}{d(v_i)},$$

where $d(u)$ is the degree of the vertex u, i.e., the number of edges incident to u. However, this value is, of course, the same as the probability of the reverse walk $x, v_k, v_{k-1}, \ldots, v_1, x$. Now, we claim that the number of x-to-y-to-z-to-x tours in one of these walks is the same as the number of x-to-z-to-y-to-x tours in its reverse; to see this, note that the greedy algorithm for finding such tours starting from the left is optimal and thus yields at least as many such tours as we can find by listing x-to-z-to-y-to-x tours from the right; the symmetric argument establishes equality. It follows that the expected lengths of the two types of tours from x to x are the same, proving the lemma. □

Remark. For those readers accustomed to thinking of random walks in terms of electrical circuits, we note that the lemma follows also from the following formula, which appears in Tetali [17]:

$$H_G(x, y) = mR_G(x, y) + \frac{1}{2} \sum_z d(z)[R_G(y, z) - R_G(x, z)]$$

in which m is the number of edges of G, and $R_G(u, v)$ is the effective resistance between u and v when G is regarded as an electrical network with a unit resistor on each edge.

A strategy S for the demon will be called a *hitting time strategy* if, whenever the tokens are on x and y with $H_G(x, y) > H_G(y, x)$, S requires that the demon move the token from x. It looks reasonable to guess that the demon always has a hitting time strategy that is optimal, but this proves not to be the case. However, from the next lemma, we can deduce the fact that there is a pure hitting time strategy whose tournament is transitive.

LEMMA 3. *On any graph G, the vertex-relation given by*

$$u \leqq v \quad \text{if and only if } H_G(u, v) \leqq H_G(v, u)$$

is transitive, i.e., constitutes a preorder on the vertices of G.

Proof. The proof is immediate from the equation of Lemma 2. For us, the important consequence of Lemma 3 is that there is always a vertex t that is *minimal* in this preorder and thus satisfies $H_G(v, t) \geqq H_G(t, v)$ for every other vertex v of G. Such a vertex t will be called *hidden*. (As an example, the reader may verify that a vertex of a *tree* is *hidden* just if its average distance to other vertices of the tree is maximum.)

4. Examples. In tuning our intuition with regard to hitting times and meeting times, it is helpful to look at some examples. In the following cases, hitting times can be verified by using Tetali's formula or by solving simple equations.

Let us begin with a simple but already counterintuitive case: What should the demon's strategy be on a path? It seems perhaps that he should always move the token nearest the center, but, in fact, it makes no difference what he does, so long as he never moves a token from an endpoint unnecessarily.

A strategy for the demon is termed a *degree strategy* if, whenever the tokens are on vertices of different degrees, he moves the token from the vertex of larger degree. Thus, on the path, a degree strategy never moves a token off an endpoint unless it must.

THEOREM 1. *On a path, a strategy for the demon is optimal if and only if it is a degree strategy.*

Proof. Let G be the path on vertices $v_0, \ldots, v_{n-1}$. As a pair of tokens moves within G, let us imagine a single "distance token" moving around a second copy G' of G according to the following rule: If the tokens on G are at vertices v_i and v_j, then the location v'_k of the distance token is given by $k = |i - j|$.

If the demon follows a degree strategy, then the distance token takes a completely normal random walk on G', with absorbing state v'_0 and reflecting state v'_{n-1}. The expected duration of the game is precisely the absorbing time from the distance token's initial position. If, however, at any pair v_0, v_i or v_i, v_{n-1}, the demon has positive probability of moving the token at the endpoint, then the probabilities along the edges leading from the corresponding vertex on G' are skewed toward v'_0, decreasing the expected time to finish. □

For our next example, we make only a small departure from the path; let G be the graph depicted in Fig. 1, consisting of a path of nine vertices with a pendant edge attached to the middle vertex v_5.

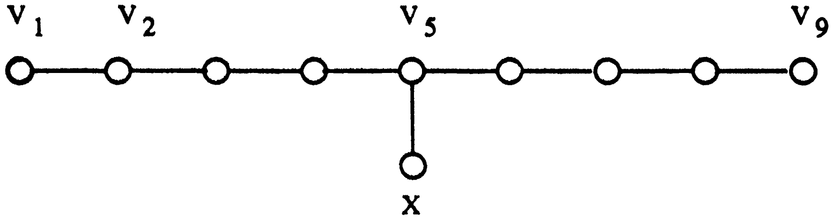


FIG. 1. *Hitting time strategy not optimal.*

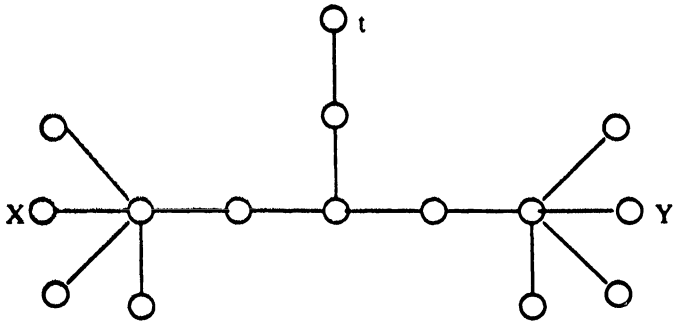


FIG. 2. *Meeting time more than maximum hitting time.*

In this graph, $H_G(v_2, x) = 32$ and $H_G(x, v_2) = 40$, so, if the demon follows a hitting time strategy, he will move from x when the tokens sit at x and v_2. This strategy will net him an expected meeting time of $M(x, v_2) = 46\frac{6}{7}$ steps from that position.

However, it turns out that the demon's optimal strategy never moves a token at x unless the other token is on v_1 or v_9. If the demon follows this rule and otherwise moves the token that is closer to v_5, he achieves $M(x, v_2) = 52$.

For our last example, we consider the tree T pictured in Fig. 2. Here we have the following hitting times: $H_T(x, y) = 84$, $H_T(y, t) = 73$, and $H_T(t, y) = 67$. Note that t is the unique hidden vertex here. Although the maximum hitting time in T is indeed 84, we can easily verify that the maximum *meeting* time from x to y is more than 87.

To see this, let the demon proceed as follows: Move the token that begins at x and continue moving it until it hits y or t. If it hits t first—which happens with probability > 50%—the demon switches horses by moving the token at y, thus gaining the difference $73 - 67 = 6$ between $H_T(y, t)$ and $H_T(t, y)$.

We see, however, that the maximum meeting time in a graph can never be more than *twice* the maximum hitting time.

5. The main theorem.

THEOREM 2. *Let G be any connected, undirected graph and let t be a hidden vertex of G. Then, for every pair x, y of vertices of G,*

$$M_G(x, y) \leqq H_G(x, y) + H_G(y, t) - H_G(t, y).$$

Proof. We define a *potential function* Φ in accordance with the right-hand side of the above inequality; thus

$$\begin{aligned}\Phi(x, y) &= H_G(x, y) + H_G(y, t) - H_G(t, y)\\ &= H_G(y, x) + H_G(x, t) - H_G(t, x)\end{aligned}$$

because of Lemma 2. Thus Φ is symmetric, and, no matter which token the demon decides to move, its expected value after the move will decline by 1, as follows:

$$\Phi(x, y) = 1 + \Phi(\bar{x}, y) = 1 + \Phi(x, \bar{y}).$$

Since Φ is nonnegative (on account of t being a hidden vertex), the statement of the theorem is already plausible, but, to make it rigorous, an argument similar to that used in Lemma 1 seems to be the most elementary route.

Assume, accordingly, that the theorem is false and let β be the maximum value of $M_G(x, y) - \Phi(x, y)$. Among all pairs x, y realizing β, choose one of minimum distance, which cannot, of course, be 0, since $\Phi(x, x) \geqq 0 = M_G(x, x)$. We may assume the demon's strategy with tokens on x and y is to move x. Then

$$\begin{aligned} M_G(x, y) &= \Phi(x, y) + \beta \\ &= 1 + \Phi(\bar{x}, y) + \beta \\ &\geqq 1 + M_G(\bar{x}, y) = M_G(x, y); \end{aligned}$$

again, at least one neighbor of x must be closer to y than x was, so the inequality is strict, and the contradiction proves the theorem. □

In view of the Brightwell-Winkler bound on hitting time [6], we can now readily obtain a cubic bound of $(\frac{8}{27})n^3$ for $M_G(x, y)$. However, more careful analysis (below) will obtain the right constant. Before proceeding, however, let us rescue some of the virtue of the hitting time strategies, by showing that their behavior with respect to hidden vertices is correct.

COROLLARY 1. *For any hidden vertex t of G, the demon has a (pure) optimal strategy that never moves a token off t.*

Proof. For any other vertex x, we have

$$M_G(x, t) \leqq \Phi(x, t) = H_G(x, t),$$

so the demon cannot achieve a higher expected time to finish than he gets by merely waiting for the token from x to hit t. □

The *commute time* $C_G(x, y)$ between vertices x and y of a graph G is defined simply by

$$C_G(x, y) = H_G(x, y) + H_G(y, x).$$

LEMMA 4. *In a graph G on n vertices, $n \geqq 13$, for any three distinct vertices (x, y, z), we have*

$$C_G(x, y) + C_G(y, z) + C_G(z, x) \leqq \tfrac{8}{27}n^3 + \tfrac{8}{3}n^2 + \tfrac{4}{9}n - \tfrac{592}{27}.$$

Proof. By Chandra et al. [9], we know that

$$C_G(x, y) = 2mR_G(x, y),$$

where m is the number of edges of G and where $R_G(x, y)$ is the effective resistance between x and y (see the remark above).

Let $\rho_G(x, y)$ denote the distance between x and y in G, that is, the number of edges in a shortest path between x and y. Then $R_G(x, y) \leqq \rho_G(x, y)$; so

$$C_G(x, y) + C_G(y, z) + C_G(z, x) \leqq 2m[\rho_G(x, y) + \rho_G(y, z) + \rho_G(z, x)].$$

Case I. Let $\rho_G(x, y) + \rho_G(y, z) + \rho_G(z, x) = 2k$ be even.

Then there are nonnegative integers a, b, c such that

$$\rho_G(x, y) = a + b, \quad \rho_G(y, z) = b + c, \quad \rho_G(z, x) = c + a, \quad \text{and} \quad k = a + b + c$$

(note that a, b, c are nonnegative, owing to the triangle inequality for distances). Partition the n vertices into the following $a + b + c + 1$ nonempty subsets:

$$\begin{gathered} D(x, 0), D(x, 1), \ldots, D(x, a-1), \\ D(y, 0), D(y, 1), \ldots, D(y, b-1), \\ D(z, 0), D(z, 1), \ldots, D(z, c-1), \\ \text{and Residue} = \text{everything else,} \end{gathered}$$

where $D(x, i)$ is the set of vertices t with $\rho_G(x, t) = i$. Disjointness follows from the distances: If $D(x, i) \cap D(y, j)$ is nonempty for $i < a, j < b$, then $\rho_G(x, y) \leqq i + j < a + b$. Furthermore, $D(x, i)$ and $D(y, j)$ must be nonadjacent, since otherwise $\rho_G(x, y) \leqq i + j + 1 < a + b$. Since G is connected, this implies that Residue is nonempty.

The only edges that can exist are between $D(x, i)$ and $D(x, i + 1)$, between $D(x, a - 1)$ and Residue, similarly for y and z, and within the subsets themselves.

Let F be composed of one vertex from each of the $k + 1 = a + b + c + 1$ subsets and let $H = G - F$ be the remaining $n - k - 1$ vertices. The number of edges among F is at most k. The number of edges among H is at most $(n - k - 1)(n - k - 2)/2$. The number of edges between H and F is at most $4(n - k - 1)$. (Each member of H can be adjacent to at most four members of F, and can achieve 4 only if it is in Residue. If it is adjacent to five members of F, it is adjacent to 4 among $D(x, i)$, $D(y, j)$, and Residue, and thus gives a shortcut between x and y.)

Thus the number of edges is bounded by

$$k + (n - k - 1)(n - k - 2)/2 + 4(n - k - 1),$$

and by [9], the sum of the round-trip times is bounded by

$$2 \times 2k \times [k + (n - k - 1)(n - k - 2)/2 + 4(n - k - 1)].$$

If n is at least 13, this expression is maximized at $k = \lceil n + 3/3 \rceil$.

Thus, depending on n, we get the following bounds:

$$n \equiv 0 \pmod 3, k = (n + 3)/3, \quad \text{bound} = (8/27)n^3 + (8/3)n^2 - 16,$$

$$n \equiv 1 \pmod 3, k = (n + 5)/3, \quad \text{bound} = (8/27)n^3 + (8/3)n^2 + 4n/9 - 740/27,$$

$$n \equiv 2 \pmod 3, k = (n + 4)/3, \quad \text{bound} = (8/27)n^3 + (8/3)n^2 + 4n/9 - 592/27.$$

Case II. Suppose that $\rho_G(x, y) + \rho_G(y, z) + \rho_G(z, x) = 2k + 1$ is odd. Then we can find nonnegative integers a, b, c such that

$$\rho_G(x, y) = a + b + 1, \quad \rho_G(y, z) = b + c + 1,$$

$$\rho_G(z, x) = c + a + 1, \quad \text{and} \quad k = a + b + c + 1.$$

Now construct $k + 2$ disjoint nonempty subsets, along with a Residue that may or may not be empty, below:

$$D(x, 0), D(x, 1), \ldots, D(x, a),$$

$$D(y, 0), D(y, 1), \ldots, D(y, b),$$

$$D(z, 0), D(z, 1), \ldots, D(z, c),$$

and Residue = everything else (may be empty).

Let F be composed of one vertex from each $D(x, i)$, $D(y, j)$, $D(z, k)$ and let H be everything else (including Residue). The only edges in F are between $D(x, i)$ and $D(x, i + 1)$, between $D(x, a)$ and $D(y, b)$, and others by symmetry, totalling $a + b + c + 1 + 1 + 1 = a + b + c + 3 = k + 2$. There are at most $(n - k - 2)(n - k - 3)/2$ edges in H. A vertex in H can be adjacent to at most four vertices in F. (Otherwise, it is adjacent to at least four subsets among $D(x, i)$ and $D(y, j)$ and thus forms a shortcut between x and y, or similarly between y and z, or z and x.) So at most $4(n - k - 2)$ edges exist between H and F.

The total number of edges is at most

$$k+2+(n-k-2)(n-k-3)/2+4(n-k-2).$$

The sum of the round trips is bounded by

$$2\times(2k+1)\times[k+2+(n-k-2)(n-k-3)/2+4(n-k-2)].$$

This is maximized when $k=\lfloor(n+3)/3\rfloor$, if n is at least 9. Once again, depending on the value of n, we get the following bounds:

$$n\equiv 0\ (\mathrm{mod}\ 3),\ k=(n+3)/3,\quad \text{bound}=(8/27)n^3+(20/9)n^2-18,$$

$$n\equiv 1\ (\mathrm{mod}\ 3),\ k=(n+2)/3,\quad \text{bound}=(8/27)n^3+(20/9)n^2-392/27,$$

$$n\equiv 2\ (\mathrm{mod}\ 3),\ k=(n+1)/3,\quad \text{bound}=(8/27)n^3+(20/9)n^2-4n/9-280/27.$$

It follows that, irrespective of whether k is even or odd, when $n\geqq 13$ we obtain an upper bound of at most

$$\tfrac{8}{27}n^3+\tfrac{8}{3}n^2+\tfrac{4}{9}n-\tfrac{592}{27}.$$

THEOREM 3. *Maximum meeting time (for $n\geqq 13$) is bounded by*

$$(4/27)n^3+(4/3)n^2+(2/9)n-296/27.$$

Proof. It follows that

$$C_G(x,y)=H_G(x,y)+H_G(y,x),$$

$$M_G(x,y)\leqq H_G(x,y)+H_G(y,z)-H_G(z,y)=H_G(y,x)+H_G(x,z)-H_G(z,x).$$

A fortiori,

$$M_G(x,y)\leqq H_G(x,y)+H_G(y,z)+H_G(z,y),$$

$$M_G(x,y)\leqq H_G(y,x)+H_G(x,z)+H_G(z,x).$$

Thus

$$2M_G(x,y)\leqq H_G(x,y)+H_G(y,z)+H_G(z,y)+H_G(y,x)+H_G(x,z)+H_G(z,x)$$

$$=C_G(x,y)+C_G(y,z)+C_G(z,x).$$

If z is distinct from x and y, then the lemma gives the theorem. If $z=x$, then

$$M_G(x,y)\leqq H_G(x,y)+H_G(y,x)-H_G(x,y)=H_G(y,x)<C_G(x,y).$$

Select z' different from x and y and note that $C_G(y,z')+C_G(z',x)\geqq C_G(x,y)$. So $2M_G(x,y)\leqq C_G(x,y)+C_G(y,z')+C_G(z',x)$. Then the lemma again suffices. □

What is the precise maximum meeting time on an n-vertex graph? Although we have seen that, for a given G, maximum meeting time may exceed maximum hitting time, we strongly suspect that the extremal case for meeting time is the Brightwell–Winkler lollipop of [6], where maximum meeting time and maximum hitting time are the same (the vertex at the end of the "stick" is hidden).

6. Additional remarks. Our methods may be applied also to meeting times for two tokens when the choice of which token to move is not made by an adversary. For example, the choice may be random; it may alternate; it might be made by an "angel" who is trying to *minimize* the expected time to collision. In all these cases, our upper bound still applies; moreover, it is still tight to within a multiplicative constant, since the meeting times for points on opposite sides of the barbell graph depicted in Fig. 3 are always cubic. Throughout this paper, we assumed that a token moved from a vertex to any of the neighboring vertices with *equal probability*. We must remark that our main theorem (Theorem 2), in fact, holds true in a more general case—a token, when instructed to move, takes one step of a "general random walk" prescribed by the transition probabilities of a reversible Markov chain. Thus the meeting time in this case is still upper bounded by twice the hitting time.

Suppose that we reinstate the demon but give him more than two tokens to work with; as in the description of Israeli and Jalfon's self-stabilizing token management scheme above, the rule is that, whenever two tokens meet, one is eliminated. However, for our purposes, it is more convenient to think of tokens not being eliminated but simply "glued together" when they meet; then when there are k tokens at the beginning, we can employ throughout the multivariable potential

$$\Phi_k(x_1, \dots, x_k) = \frac{1}{k-1} \sum_{i<j} M_G(x_i, x_j),$$

where the x_i's will cease to be all distinct once collisions begin. Suppose, for example, that tokens corresponding to indices $i \in I$ are currently on vertex v and are designated for movement by the demon; then the expected value of $M_G(x_i, x_j)$ drops by at least 1 when $|\{i, j\} \cap I| = 1$, which occurs for $|I|(k - |I|) \geqq k - 1$ of the terms, while the other terms remain constant. Hence the expected value of Φ_k diminishes by at least 1 at every step, as desired.

Note that $M_G(x_i, x_j)$ remains at zero when i and j are both in I, since tokens i and j continue to travel to the same vertex. It is for this reason that we use M_G rather than our original Φ in the definition of Φ_k; the expected value of $\Phi(v, v)$ can in some circumstances jump as v moves, e.g., when v is a hidden vertex.

Now the proof of Theorem 2, suitably amended, shows that the expected time before reducing to a single token is at most the maximum value of Φ_k on G, which is, in turn, bounded by k times the maximum two-token meeting time. Thus we have the following result.

THEOREM 4. *The expected number of steps before tokens on k of the n vertices of a graph reduce to one is at most* $(4/27)kn^3$.

In the worst case for the Israeli–Jalfon protocol, we may as well have a token on every vertex, in which case our bound is of order n^4. Indeed, the value of Φ_n can reach order n^4, as in the barbell graph; nevertheless, the "total meeting time" in that case is still only order n^3. We do not currently know of any sequence of graphs for which the total meeting time is asymptotically more than the lollipop's $(4/27)n^3$. Hence, we have the following conjecture.

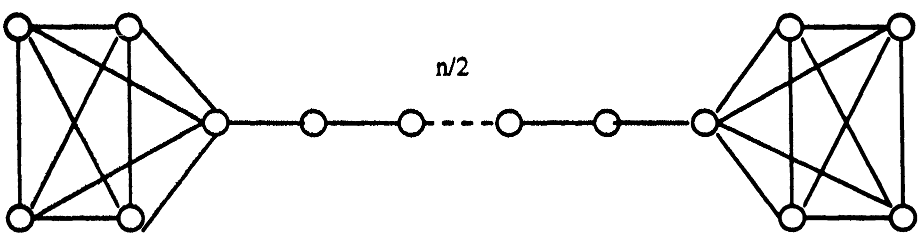


FIG. 3. *Cubic meeting times.*

CONJECTURE 1. *The total expected time for tokens at every vertex of an n-vertex graph to reduce to one is $O(n^3)$.*

Note that it may in some graphs take more time to reduce three tokens (say) to two, in the worst case, than two to one. Define the *collision time* for a collection of tokens on a graph to be the expected number of steps before any two tokens collide, with the schedule demon trying to keep them apart as usual. As an example, let G consist of several copies of a long path with their right-hand endpoints identified; starting with tokens at three left-hand endpoints, the demon moves one until it becomes adjacent to a second, then moves the third exclusively. Asymptotically, this takes 50% longer than the largest two-token meeting time, obtained by hitting one endpoint with a token from another.

However, addition of more tokens, in this case, reduces collision time. We cannot find any case where this is not so; hence the following conjecture holds.

CONJECTURE 2. *The maximum collision time on any graph is achieved either with two or three tokens.*

There is one further consideration, which leads perhaps to the most intriguing conjecture of all. Let us put two tokens on a graph and let them take random walks, as before, but now suppose the schedule demon is *clairvoyant*—that is, he can see where each token will go, infinitely far into the future. The question is, with this advantage, can he now keep the tokens apart *forever*? Of course, we must ask the question in a probabilistic context, since the tokens may, e.g., be headed toward each other along the same path, at the outset. In some graphs (e.g., a tree or a cycle), it is easy to see that the demon will come to grief with probability 1. However, we think that on sufficiently large complete graphs the demon will win with probability greater than 0 (in fact, K_4 may be big enough). Hence, we have Conjecture 3.

CONJECTURE 3. *Let two tokens begin random walks on a large complete graph. Then with probability* > 0, *the clairvoyant schedule demon can keep them apart forever.*

Acknowledgments. The authors have enjoyed the benefit of useful conversations on this topic with Stuart Haber, Shlomo Halfin, Marc Jalfon, Michal Karonski, and Milena Mihail.

REFERENCES

[1] D. J. ALDOUS, *Applications of random walks on graphs*, 1989, preprint.

[2] ———, *Bibliography: Random walks on graphs*, 1989, preprint.

[3] ———, *Meeting times for independent Markov chains*, Stochastic Processes Appl., 3 (1991), pp. 185–193.

[4] 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 Proc. 20th Annual Sympos. on Foundations of Computer Science, San Juan, Puerto Rico, October 1979, pp. 218–223.

[5] K. BORRE AND P. MEISSL, *Strength Analysis of Leveling-Type Networks*, Geodaetisk Institut, Vol. 50, Copenhagen, 1974.

[6] G. BRIGHTWELL AND P. WINKLER, *Maximum hitting time for random walks on graphs*, Random Structures Algorithms, 3 (1990), pp. 263–276.

[7] A. Z. BRODER, *How hard is it to marry at random? (On the approximation of the permanent)*, in Proc. 18th Annual ACM Symposium on Theory of Computing, 1986, Berkeley, CA, pp. 50–58.

[8] D. COPPERSMITH, P. DOYLE, P. RAGHAVAN, AND M. SNIR, *Random walks on weighted graphs, and applications to on-line algorithms*, in Proc. 22nd Annual ACM Symposium on Theory of Computing, 1990, Baltimore, MD, pp. 369–378.

[9] A. K. CHANDRA, P. RAGHAVAN, W. L. RUZZO, R. SMOLENSKY, AND P. TIWARI, *The electrical resistance of a graph captures its commute and cover times*, in Proc. 21st Annual ACM Symposium on Theory of Computing, May 1989, Seattle, WA, pp. 574–586.

[10] P. Dagum, M. Luby, M. Mihail, and U. Vazirani, *Polytopes, permanents and graphs with large factors*, in Proc. 29th Annual IEEE Symposium on Foundations of Computer Science, 1989, Research Triangle Park, NC, pp. 412–421.

[11] C. Derman, *Finite State Markovian Decision Processes*, Academic Press, New York, 1970.

[12] P. G. Doyle and J. L. Snell, *Random Walks and Electric Networks*, Mathematical Association of America, Washington, DC, 1984.

[13] M. Dyer, A. Frieze, and R. Kannan, *A random polynomial time algorithm for estimating volumes of convex bodies*, in Proc. 21st Annual ACM Symposium on the Theory of Computing, May 1989, Seattle, WA, pp. 375–381.

[14] A. Israeli and M. Jalfon, *Token management schemes and random walks yield self stabilizing mutual exclusion*, in Proc. 9th Annual ACM Symposium on Principles of Distributed Computing, Quebec City, Canada, 1990, pp. 119–131.

[15] M. Jerrum and A. Sinclair, *Conductance and the rapid mixing property for Markov chains: The approximation of the permanent resolved*, in Proc. 20th Annual ACM Symposium on Theory of Computing, 1988, Chicago, IL, pp. 235–243.

[16] A. Karzanov and L. Khachiyan, *On the conductance of order Markov chains*, Tech. Report DCS TR 268, Rutgers University, New Brunswick, NJ, June 1990.

[17] P. Tetali, *Random walks and effective resistance of networks*, J. Theoret. Probab., 1 (1991), pp. 101–109.

SIAM J. DISC. MATH.
Vol. 6, No. 3, pp. 375–387, August 1993

004

MINIMUM EDGE DOMINATING SETS*

J. D. HORTON† AND K. KILAKOS‡

Abstract. Let $G = (V, E)$ be a finite undirected graph with n vertices and m edges. A minimum edge dominating set of G is a set of edges D, of smallest cardinality $\gamma'(G)$, such that each edge of $E - D$ is adjacent to some edge of D. Let $S(G)$ be the subdivision graph of G and let $T(G)$ be the total graph of G. Let $\alpha(G)$ be the stability number of G (cardinality of a largest stable set) and let $\alpha_2(G)$ be the 2-stability number of G (cardinality of a largest set of vertices in G, no two of which are joined by a path of length 2 or less). The following results are obtained. For any G, $\gamma'(S(G)) + \alpha_2(G) = n$ and $2\gamma'(T(G)) + \alpha(T(G)) = n + m$ or $n + m + 1$. Also, for any depth-first search tree S of G, $\gamma'(S)/2 \leqq \gamma'(G) \leqq 2\gamma'(S)$, and these bounds are tight.

The edge domination problem is NP-complete for planar bipartite graphs, their subdivision, line, and total graphs, perfect claw-free graphs, and planar cubic graphs. The stable set problem and the edge domination problem are NP-complete for iterated total graphs.

The edge domination problem is solvable in $O(n^3)$ time for claw-free chordal graphs, locally connected claw-free graphs, the line graphs of total graphs, the line graphs of chordal graphs, the line graph of any graph in which each nonbridge edge is in a triangle, and the total graphs of any of the preceding graphs.

Key words. graph theory, complexity, line graphs, total graphs, subdivision graphs, dominating set, stable set, 2-stable set

AMS subject classifications. primary 68R10; secondary 05C35, 05C70

1. Introduction.

Despite being closely linked with several important graph problems, edge domination has not been extensively studied. Most of the known results appear in Yannakakis and Gavril [20]. Two obvious connections with well-known problems relate edge dominating sets to matchings and vertex dominating sets. An edge dominating set of any graph G is a vertex dominating set in the line graph of G, and an independent edge dominating set of G is a maximal matching of G. Many of the results in this paper first appeared in [13] and come primarily from associating the edge domination problem with the stable set problem. In total graphs, line graphs, and claw-free graphs, these two concepts are closely related and allow us to answer complexity questions in these classes of graphs.

All graphs considered here are finite, undirected, without loops or multiple edges. The graph $G = (V, E)$ has vertex set V with n vertices, and edge set E with m edges. An edge $e = \{v, u\}$ has vertices v and u that are *saturated* by e. Given a set of edges D, denote the vertices saturated by edges of D by V_D.

An edge $\{v, u\}$ is said to *dominate* all edges that have v or u as a vertex, including itself. A set of edges M is said to be *independent* (a *matching*) if no two of its edges have a vertex in common. A set of edges D is said to be an *edge dominating set* if every edge is dominated by an edge in D. It is known that the size of a minimum independent edge dominating set is equal to the size of a minimum edge dominating set in a graph (see [2] or [20] for a proof). We call the cardinality of the smallest edge dominating set of G the *edge domination number* of G and denote it by $\gamma'(G)$. A set I of vertices of G is a *stable set* if no two of its vertices are adjacent. The set I is said to be a *2-stable set* if the distance of any two vertices in T is greater than 2. We denote the *stability number* (*2-stability number*) of G by $\alpha(G)(\alpha_2(G))$, the cardinality of a largest stable (2-stable) set of G. Also, we denote the cardinality of a largest matching in G by $m(G)$. A set of

* Received by the editors January 5, 1989; accepted for publication (in revised form) August 11, 1992.

† Faculty of Computer Science, University of New Brunswick, Fredericton, New Brunswick, Canada.

‡ Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada N2L 3G8.

vertices that is incident with all edges is called a *vertex cover*. A set of edges such that every vertex is incident with exactly k of the edges is called a *k-factor*. In particular, a one-factor is called a *perfect matching*.

The *subdivision graph* of G is $S(G) = (V \cup E, E')$, where $E' = \{\{e, v\} : e \in E$, and v is incident with $e\}$. In effect, each edge of G is replaced by a path of length 2. The *line graph* of G is $L(G) = (E, E^*)$, where $E^* = \{\{e, f\} \mid e$ and f are adjacent edges of $E\}$. The edges of G become vertices of $L(G)$, and two edges of G are adjacent vertices of $L(G)$ if and only if the edges are adjacent in G. The *total graph* of G is defined by $T(G) = (V \cup E, E \cup E' \cup E^*)$. Thus the total graph is the union of the graph, its line graph, and its subdivision graph. The total graph is also the square of the subdivision graph; that is, two vertices of $T(G)$ are joined if and only if the same two vertices are joined by a path of length 1 or 2 in $S(G)$. The vertex of $T(G)$ or $S(G)$ corresponding to edge $\{v, u\}$ of G is called an *e-vertex* and is denoted by $[v, u]$.

We conclude this section by stating some results that will be useful in subsequent sections. The first of these results (Proposition 1.1) is trivial. Theorems 1.2 and 1.3 are due to Yannakakis and Gavril [20].

PROPOSITION 1.1. *Let D be a set of edges of a graph G. Then D is an edge dominating set if and only if the vertices not saturated by D form a stable set.*

THEOREM 1.2 (see [20]). *Let I be a maximum stable set of $T(G)$. Then, for every maximal matching M of G containing all the e-vertices of I, the set M, together with all vertices of G not saturated by M, form a maximum stable set of $T(G)$, and M is a minimum independent edge dominating set of G.*

THEOREM 1.3 (see [20]). *If D is a minimum independent edge dominating set of G, then D, together with the vertices of G not saturated by D, form a maximum stable set of $T(G)$.*

2. Depth-first search trees.

The numerous applications involving depth-first search spanning trees has motivated a considerable amount of research towards them. For instance, Savage [18] has shown that $m(G) \leqq 2m(T)$, where T is a depth-first search spanning tree of G. A similar result holds true also for $\gamma'(G)$.

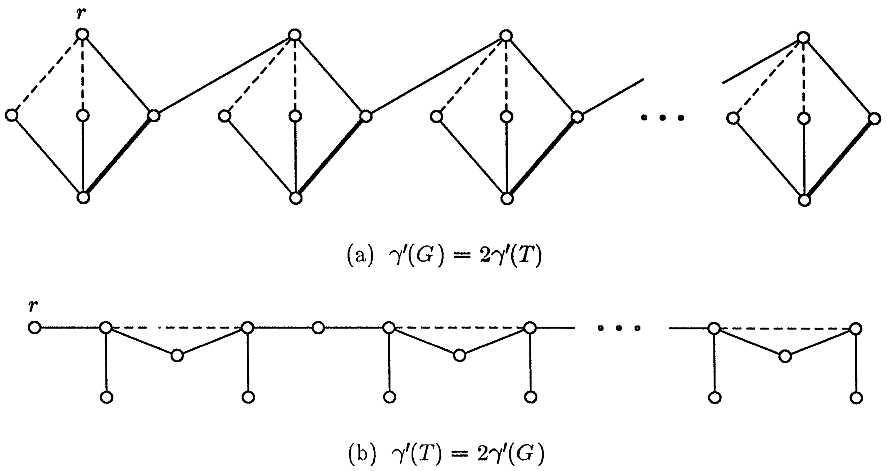
THEOREM 2.1. *If T is a depth-first search tree of a connected graph G, then $\gamma'(T)/2 \leqq \gamma'(G) \leqq 2\gamma'(T)$. Both bounds are tight.*

Proof. Clearly, no two leaves of T are joined by an edge in G. Thus any set of edges that saturates all the interior vertices of T (including the root) dominates all the edges of G. Consider a minimum edge dominating set D of T. For an interior vertex v of T, either v is saturated by D or all its children are saturated by D. Hence the number of interior vertices not saturated by D is at most $|D| = \gamma'(T)$. Adding edges to D to cover these unsaturated interior vertices gives an edge dominating set of G of cardinality at most $2\gamma'(T)$. Thus the upper bound is proved. The lower bound is actually valid for any spanning tree T. Consider

$$\gamma'(T) \leqq m(T) = n - \alpha(T) \leqq n - \alpha(G) \leqq 2\gamma'(G).$$

The first inequality says that a minimum maximal matching is no larger than a maximum matching. The next equality is a statement of König's [14] theorem for bipartite graphs. The next inequality follows because a stable set in G is also a stable set in T. Finally, the last inequality follows from Proposition 1.1.

It remains to show that both bounds are tight. For the upper bound, consider the graph G shown in Fig. 1(a) consisting of $5k$ vertices and $7k - 1$ edges, where k is an arbitrary positive integer. A depth-first search tree T with an edge dominating set of size k is also given. G contains k 4-gons, each of which requires two edges of an edge dominating



(a) $\gamma'(G) = 2\gamma'(T)$

(b) $\gamma'(T) = 2\gamma'(G)$

FIG. 1

set D. Since there is no edge joining any pair of the 4-gons, $\gamma'(G) = 2k$. For the lower bound, let T be a path on $4k$ vertices rooted at a vertex r of degree 1, augmented by attaching a child to each vertex at an odd distance from r. Referring to Fig. 1(b), we see that T is a depth-first search tree of the graph G, obtained from T by adding edges between the vertices at an odd distance from r. It is easy to see that $\gamma'(T) = 2k$, while $\gamma'(G) = k$. □

3. Subdivision graphs. Yannakakis and Gavril [20] initially studied the complexity of the edge domination problem for planar graphs and bipartite graphs and showed that it is NP-complete for both of these classes of graphs. However, the edge domination problem is NP-complete even for the intersection of these graph classes. First, we prove a theorem that relates 2-stable sets of a graph G with independent edge dominating sets in the subdivision graph of G, $S(G)$.

LEMMA 3.1. *If I is a nonempty set of vertices of a connected graph $G = (V, E)$, then there exists a maximal matching M of $S(G) - I$ such that $|M| = n - |I|$.*

Proof. Let $H = G - I$ and let H_i be a connected component of H. Construct a spanning tree T of H_i rooted at a vertex w adjacent to a vertex v of I in G. Include the edge $\{w, [w, v]\}$ in M. Observe that, for any e-vertex $[x, y]$ of $S(T)$, with x closer than y to the root w of T, the edge $\{[x, y], y\}$ in $S(G)$ can also be included in M. Extending this process to every connected component of H, we construct M such that $|M| = n - |I|$. Thus all vertices of $V - I$ are saturated, and no pair of e-vertices of $S(G)$ are joined by an edge, so that M is a maximal matching. □

LEMMA 3.2. *If M is a maximal matching of $S(G)$, then $I = V - V_M$ is a 2-stable set of G.*

Proof. Suppose that I is not a 2-stable set. Let v and u be any two vertices of I that are not distance 2 or more apart. If v and u are adjacent, then neither $\{v, [v, u]\}$ nor $\{u, [v, u]\}$ is dominated by M. Thus we can let w be a vertex adjacent to both v and u. Then at least one of $\{v, [v, w]\}$ and $\{u, [u, w]\}$ cannot be dominated by M. Hence I must be a 2-stable set of G. □

The preceding two lemmas lead immediately to the following result.

THEOREM 3.3. *For a graph G and its subdivision graph $S(G)$, an independent edge dominating set D is a minimum independent edge dominating set of $S(G)$ if and only if $V - V_D$ is a maximum 2-stable set of G.*

COROLLARY 3.4. *The edge domination problem is* NP-*complete when restricted to the subdivision graphs of planar bipartite graphs with no vertex degree exceeding* 3.

Proof. The stable set problem is NP-complete for planar cubic graphs (see the Appendix). Also, Chang and Nemhauser [3] have shown that the 2-stability problem is NP-complete by applying the following transformation from the stable set problem. Given a graph G, they construct G by replacing each edge $\{v, w\}$ by a tree consisting of the paths (v, u, w) and (u, u', u''). G has a stable set of size k if and only if G' has a 2-stable set of size $m + k$. The combination of these two results shows that the 2-stability problem is NP-complete for planar bipartite graphs with no vertex degree exceeding 3. By Theorem 3.3, this problem is polynomially equivalent to finding a minimum edge dominating set in $S(G)$. □

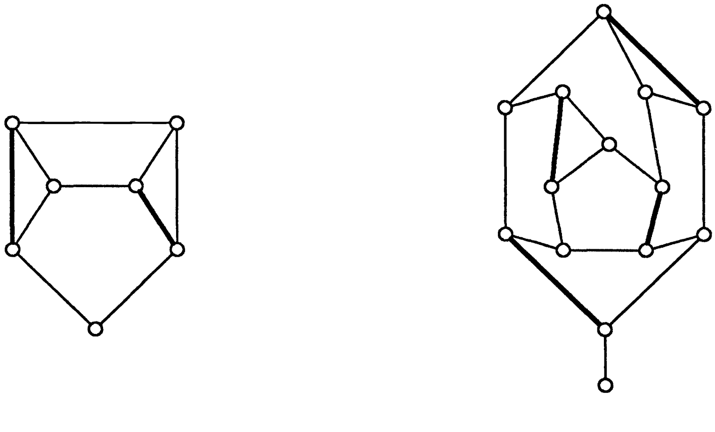
Since the subdivision graph of a planar bipartite graph is also planar bipartite, we have the following result.

COROLLARY 3.5. *The edge domination problem is* NP-*complete for planar bipartite graphs.*

The results obtained so far can be extended to the k-iterated subdivision graphs, $S_k(G)$. Define for a graph G and a positive integer k, $S_k(G) = S(S_{k-1}(G))$, where $S_0(G) = G$. Note that a 2-stable set of G is a stable set in the square of G. Since $T(G)$ is the square of $S(G)$, Theorems 1.2 and 1.3 show how to find in polynomial time a 2-stable set in $S_i(G)$ given an edge dominating set in $S_{i-1}(G)$, and vice versa. Theorem 3.3 shows how to find in polynomial time an independent edge dominating set in $S_i(G)$ given a 2-stable set in $S_{i-1}(G)$, and the opposite also holds. These two results, combined with the fact that the 2-stability and edge domination problems are both NP-complete in the class of planar bipartite graphs, gives us the following.

COROLLARY 3.6. *For any fixed positive integer k, the edge domination problem and the* 2-*stability problem are both* NP-*complete for k-iterated subdivision graphs of planar bipartite graphs.*

However, the problems of 2-stability and edge domination are not of equivalent complexity on all classes of graphs. It is shown in [3] that the 2-stability problem is NP-complete for split graphs. $G = (V, E)$ is a *split graph* if V can be partitioned into a stable set I and a clique C. On the other hand, Proposition 1.1 makes it easy to find in polynomial time a minimum edge dominating set for this class of graphs. We must choose a set of



Degree 1 attachment Degree 2 attachment

FIG. 2

edges that saturates all but possibly one vertex of C, the complete subgraph in G. If C contains a vertex v that is not adjacent to any vertex of I, then leading v unsaturated solves the problem, and $\gamma'(G) = \lceil(|C| - 1)/2\rceil$. Otherwise, $\gamma'(G) = \lceil |C|/2 \rceil$.

The edge domination problem is NP-complete for planar cubic graphs as well. Corollary 3.4 shows that it is NP-complete for planar bipartite graphs of maximum degree 3. It is easy to find components to attach to vertices of degree 1 or 2, that both make the graph cubic and that do not affect how a minimum edge dominating set occurs in the original graph. Two such components are shown in Fig. 2, with edge dominating sets. Each edge of the dominating set dominates five edges, no edge is dominated twice, and the number of edges in the component are exactly divisible by 5. Hence these edge dominating sets are minimum.

4. Line graphs.

4. Line graphs. Often, problems that are difficult for general graphs become solvable if restricted to line graphs. Edge domination is not such a problem.

THEOREM 4.1. *The edge domination problem is* NP-*complete for the line graphs of planar bipartite graphs.*

Proof. We transform vertex cover for 3-connected cubic planar graphs to this problem (see the Appendix). The transformation is similar to that used in [20] for proving Theorem 1. Given a 3-connected cubic planar graph G, each vertex v_i is replaced by the subgraph H_i shown in Fig. 3(a). The three edges formerly incident with v_i now become incident with the vertices x_i, y_i, and z_i. The replacement must not allow a vertex x_i to be joined to a vertex x_j. To prevent this from happening, first decompose the 3-connected cubic graph into a 1-factor and a 2-factor [17]. Orient each circuit of the 2-factor and always replace a vertex v_i so that x_i is joined to the next vertex in the circuit, y_i is joined to the previous vertex in the circuit, and z_i is joined to the neighbouring vertex in the 1-factor. Then an original edge of G, $\{v_i, v_j\}$, gets replaced by an edge $\{x_i, y_j\}$, $\{x_j, y_i\}$ or $\{z_i, z_j\}$. Call this new graph G^*.

We claim that G has a vertex cover of size k if and only if G^* has an edge dominating set of size $7n + k$, where n is the number of vertices in G. As a first step in the proof of this claim, note that H_i satisfies the following properties:

(1) H_i has an edge dominating set of size 7;

(2) H_i has an edge-dominating set of size 8 that saturates x_i, y_i, and z_i;

(3) There is no set of six edges of H_i that dominates all the edges of $H_i - \{x_i, y_i, z_i\}$;

(4) There is no set of seven edges of H_i that dominates all edges of $H_i - y_i$ and includes an edge incident to x_i or z_i;

(5) There is no set of seven edges of H_i that dominates all edges of $H_i - z_i$ and includes an edge incident to x_i or y_i;

(6) There is no set of seven edges of H_i that dominates all edges of $H_i - \{y_i, z_i\}$ and includes an edge incident with x_i.

The verification that H_i satisfies these six conditions can be made easier by shrinking three paths of length 3 and decreasing the number of edges in the edge dominating sets by three.

Let C be a vertex cover of G with k vertices. We now find an edge dominating set of G^*. For each vertex v_i not in C, by (1) we can dominate all the edges of H_i with seven edges. For each vertex v_j in C, by (2) we can dominate all the edges of H_i with eight edges, saturating x_j, y_j, and z_j. These $8k$ edges also dominate all edges derived from edges of G because C is a vertex cover. Thus all edges of G^* are dominated by $7n + k$ edges.

Conversely, let D be an edge dominating set of G^* of size d. Let d_i be the number of edges of D in the subgraph H_i. By (3), d_i is greater than 6; thus $d \geqq 7n$.

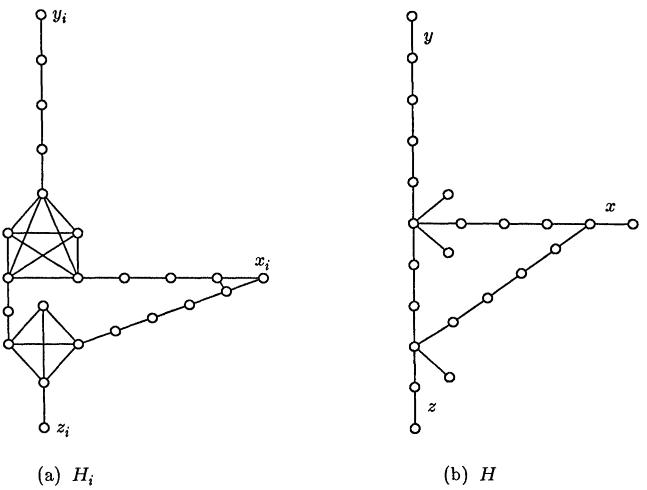


FIG. 3. $H_i = L(H)$.

Now form a set of, say k, vertices C of the original graph G. Put v_i into C if $d_i \geqq 8$ or if the edge $\{x_i, y_i\}$ is in D. If the edge $\{z_i, z_j\}$ is in D, put either v_i or v_j into C. To each vertex of C, there is an edge of D in excess of the $7n$ mentioned before. Thus $m \geqq 7n + k$.

It still remains to show that C is a covering set of G. Consider any edge of G, say $e = \{v_i, v_j\}$, and the corresponding edge e^* of G^*, which is of the form $\{x_i, y_j\}$ or $\{z_i, z_j\}$. If this edge is in D, then either v_i or v_j is in C, and so e is covered by C. Otherwise, e^* is dominated by an adjacent edge of H_i or H_j. In this latter situation, if either d_i or d_j is at least 8, then either v_i or v_j is in C and e is covered. The remaining case is that $d_i = 7 = d_j$. By (4) and (5), e^* can be dominated from H_i only if some edge of H_i is dominated by edges of M that are not in H_i. In particular, if $e^* = \{z_i, z_j\}$, then by (4) the seven edges of $M \cap H_i$ cannot include an edge incident with z_i, so the edge attached to x_i from outside H_i, say $\{x_i, y_t\}$ is in D. Then, however, v_i is in C. If instead $e^* = \{x_i, y_j\}$ and e^* is dominated by an edge from H_j, by (5), D includes an edge $\{x_j, y_t\}$ and v_j is in C. The one remaining case is that $e^* = \{x_i, y_j\}$ and that e^* is dominated by an edge from H_i. By (6) this is impossible. This completes the proof that the vertex cover problem in G is equivalent to the edge domination problem in G^*.

G^* is the line graph of a planar bipartite graph, since each H_i is isomorphic to the line graph of the graph H in Fig. 3(b). It is bipartite, and the distance between the other endpoints of e_x, e_y, and e_z are all even. Hence, if H replaces each vertex of G in the natural way, the remaining graph is planar, bipartite, and has G^* as its line graph. □

COROLLARY 4.2. *The edge domination problem is* NP-*complete for perfect claw-free graphs.*

Proof. The line graph of a bipartite graph is both perfect and claw-free. □

5. Total graphs. Theorems 1.2 and 1.3 associate edge dominating sets in G with stable sets in the total graph $T(G)$. However, a direct relationship exists between these two parameters in $T(G)$ alone. First, consider the related problem of finding a maximum cardinality matching in $T(G)$ minus a stable set.

THEOREM 5.1. *Let G be a connected graph and let I be any stable set of vertices of $T(G)$. Then either $H = T(G) - I$ has a perfect matching or, for any vertex v of I, $T(G) - (I - v)$ has a perfect matching.*

Proof. It is easy to see that H is connected. Let S be a spanning tree of G and let r be the root of S. Consider the following algorithm that visits all the vertices of S in postorder and adds edges to M.

The algorithm starts with M being the empty set. Let x be the vertex that the algorithm is currently visiting and let z be x's parent in S. If x is the root r, then z does not exist, and the steps must be modified appropriately. At this point in the algorithm, all vertices beneath x in the tree have been visited, and so all these vertices and all e-vertices adjacent to them, other than neighbours of x itself, have been paired and are saturated by M. No ancestor of x has yet been visited, so that the e-vertices of M corresponding to edges above x in the tree S have not yet been paired. Perform the following operations at x.

(1) Pair off all e-vertices of H adjacent to x, other than $[x, z]$, that are not yet saturated, and add the pairs to M. This operation may or may not leave any e-vertex of the form $[x, v]$ unsaturated.

(2) If $[x, v]$ is still not saturated, and if x is in H and is not yet saturated, then add $\{x, [x, v]\}$ to M.

(3) If $[x, v]$ is unsaturated and x is either not in H or has already been paired, put $\{[x, v], [x, z]\}$ into M. If x is the root so that z does not exist, $[x, v]$ is left unpaired, and the algorithm halts.

(4) If all e-vertices of H beneath x in S are saturated, and x is in H and is not yet saturated, add $\{x, [x, z]\}$ to M. If $[x, z]$ is in I, then $\{x, z\}$ is added to M instead. Again, if x is the root, x is left unpaired, and the algorithm halts.

(5) Proceed to the next vertex in the post-order of S.

It is necessary to verify that steps (3) and (4) can always be performed when required. In step (3), for a nonroot vertex x, $[x, z]$ must be available to be paired. It cannot yet have been paired, since the edge $\{x, z\}$ is above x in the tree S. The only problem that could occur is that $[x, z]$ is in the stable set I of $T(G)$ that has been removed to form H. However, $[x, z]$ cannot be in I if x is I. The only other way step (3) is required is if x has already been paired. This can only happen in step (4) on a previous iteration when x is playing the role of the parent z. Then, however, an e-vertex adjacent to x is in I, so that $[x, z]$ cannot be in I.

In step (4), we must check that z is available to be paired when needed. It is needed only if $[x, z]$ is in I, so that z is not in I. However, this can happen only for one child x of Z, since I is a stable set. Hence z is needed at most once during the course of the algorithm and will be in H and not be saturated when required to be paired.

To prove that the algorithm works, it is sufficient to note that, for any nonroot vertex x, the algorithm pairs all e-vertices beneath x as well as x itself, and that no e-vertex for the tree S is paired until an endpoint occurs in the post-order tree transversal. Thus the necessary conditions, required before processing a vertex x, are met after all descendents of x have been processed.

The algorithm stops when the root r is processed. If the number of nodes of H is even, then all vertices are paired, and M is a perfect matching. Otherwise, the algorithm leaves either root r or a neighbouring e-vertex unmatched. □

Theorem 5.1 and Proposition 1.1 establish the following relationships between stability and edge domination in total graphs.

COROLLARY 5.2. *Let $T(G)$ be the total graph of a connected graph G having n vertices and m edges, let D be a minimum edge dominating set of $T(G)$, and let I be a maximum stable set of $T(G)$. Then*

(1) $\gamma'(T(G)) = \lceil (n + m - \alpha(T(G)))/2 \rceil$,

(2) *The vertices not saturated by D form a stable set of cardinality $\alpha(T(G))$ or $\alpha(T(G)) - 1$,*

(3) *If M is a maximum matching of $T(G) - I$, then any independent edge dominating set of $T(G)$ having M as a subset is minimum.*

Theorem 5.1 also leads to more NP-completeness results.

COROLLARY 5.3. *The edge domination problem for total graphs of planar bipartite graphs is* NP-*complete.*

Proof. The proof follows by reduction from the stable set problem on the total graphs of planar bipartite graphs (see Theorems 1.2 and 1.3 and Corollary 3.4). Given a connected planar bipartite graph G with n vertices and m edges, find $\gamma'(T(G))$. By Corollary 5.2, $\alpha(G) = n + m - 2\gamma'(T(G))$ or $n + m - 2\gamma'(T(G)) + 1$. Define another graph G' from G by adding one edge, with a triangle attached to the other end, to any vertex of G. Clearly, $\alpha(T(G')) = \alpha(T(G)) + 2$. Hence, $\alpha(T(G')) = (n + 3) + (m + 4) - 2\gamma'(T(G')) + x_1 = \alpha(T(G)) + 2 = n + m - 2\gamma'(T(G)) + 2 + x_2$, where x_1 and x_2 are 0 or 1. Thus we have $2(\gamma'(T(G')) - \gamma'(T(G))) = 5 + x_1 - x_2$. Hence, $d(T(G')) - d(T(G)) = 2$ or 3. If this difference is 2, then $\alpha(T(G)) = n + m - 2\gamma'(T(G)) + 1$, whereas, if the difference is 3, then $\alpha(T(G)) = n + m - 2\gamma'(T(G))$. Thus a polynomial algorithm for the edge domination problem for the class of total graphs leads to a polynomial algorithm for the stable set problem in these graphs. The converse is also true by Corollary 5.2. □

Define the *iterated total graph* $T_k(G)$ by $T_k(G) = T(T_{k-1}(G))$ for $k = 1, 2, \ldots,$ and $T_0(G) = G$. Theorems 1.2 and 1.3 show that finding a maximum stable in $T_i(G)$ is polynomially equivalent to finding a minimum edge dominating set in $T_{i-1}(G)$. Also, Corollary 5.2 combined with the construction used in Corollary 5.3 shows how to find an independent edge dominating set from a stable set in $T_i(G)$, and vice versa.

COROLLARY 5.4. *For any fixed positive integer k, the edge domination problem, as well as the stable set problem, are* NP-*complete on k-iterated total graphs.*

6. Algorithms for edge domination. The previous sections have shown that it is hard to find a minimum edge dominating set in many types of graphs. Nevertheless, there are some classes of graphs for which the edge domination problem can be solved efficiently. Mitchell and Hedetniemi [16], Yannakakis and Gavril [20], and Farber [6] have all given efficient algorithms for finding minimum edge dominating sets in trees. A more general algorithm is included in the work of Corneil and Keil [4] on domination in k-trees.

A very simple algorithm, based on Proposition 1.1, produces an approximate solution, given a solution to the minimum stable set problem. This is the basic algorithm for all the algorithms in this paper. Given a graph $G = (V, E)$,

(1) Find a maximum stable set of vertices in G, I,

(2) Find a maximum matching in $G - I$, M,

(3) For each vertex v of $V - I - V_M$, choose an incident edge e_v. Then add these edges to M to form an edge dominating set $D = M \cup \{e_v | v \in V - I - V_M\}$.

The algorithm gives an edge dominating set D since $V - V_D = I$ is a stable set. Step (2) is polynomial, as shown by Edmonds [5]; step (3) is linear. Thus the algorithm is polynomial for a given class of graphs if there is a polynomial solution to the stable set problem for that class of graphs. This fact was used in the transformation of the stable set problem to the edge domination problem in § 5.

Of course, the algorithm does not necessarily give a minimum edge dominating set. In fact, the algorithm can be off by up to a factor of 2, even for trees if the wrong maximum stable set is chosen. Figure 4 exhibits such a family of graphs. Choosing the black vertices leaves $2k$ mutually disjoint vertices to be covered. On the other hand, the k edges that are the centre edges of the k vertical paths of length 3 form an edge dominating set. The approximation algorithm seems to be particularly poor for bipartite graphs.

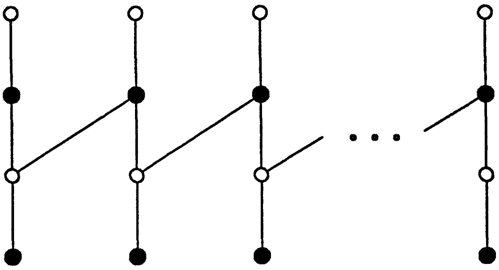


FIG. 4

The algorithm solves the edge domination problem for split graphs, as noted in § 4. The reason is that $G - I$ is a complete graph and hence has either a perfect matching or a near perfect matching. The algorithm finds a minimum edge dominating set in any graph G if $G - I$ has either a perfect matching or a near-perfect matching (matches all but one vertex). Several authors have noted that all connected claw-free graphs have this property. See Jünger, Reinelt, and Pulleyblank [12].

THEOREM 6.1. *If G is a connected claw-free graph, then either G, or for some vertex v, $G - v$ has a perfect matching, which can be found in $O(m)$ time.*

Proof. Let T be a breadth-first search tree of G. Then, except for the children of the root, siblings must be joined by an edge, since otherwise they together with their parent and grandparent form a claw. Using a post-order transversal of the tree, pair off as many of the unmatched children of a vertex as possible. If an unmatched child is left, pair it with the vertex itself. When the root is reached in the transversal, only the root and some of its children remain unmatched. Some of the root's children may not be connected to each other, but still all but two of the children can be matched with each other. If three of the children could not be paired, then they would form a claw with the root. As the final step in the algorithm, the root is matched with some unpaired child if there is one. This algorithm is linear in the number of edges. □

Thus, if G is a claw-free graph such that $G - I$ is connected, then there is a polynomial algorithm to find a minimum edge dominating set in G, since both Minty [15] and Shibi [19] have given polynomial algorithms to find a maximum stable set in claw-free graphs. The latter algorithm is $O(n^3)$. We have already seen one class of graphs that has this property and that is total graphs (see Theorem 5.1).

A second class of graphs for which the removal of a stable set cannot disconnect the graph is two-connected chordal graphs. A graph is said to be *chordal* if any simple circuit of length greater than 3 has a *chord*, that is, an edge of the graph joining two vertices of the circuit that are not adjacent in the circuit. A linear time, $O(m)$ algorithm to find a maximum stable set in a chordal graph G is given in [10].

A third class of graphs with the property that the removal of any stable set does not disconnect the graph is the class of locally connected graphs. A graph is said to be *locally connected* if the subgraph induced by the neighbourhood of any vertex is connected. Clearly, a path in such a graph can avoid a vertex by taking a detour around it in its neighbourhood, and hence a path in such a graph can be replaced by a path with the same endpoints but avoiding any given stable set of vertices. Two subclasses of the locally connected claw-free graphs are the line graphs of 2-connected chordal graphs, and the line graphs of the total graphs. Both 2-connected chordal graphs and total graphs have the property that any edge is in a triangle. The neighbourhood of an e-vertex in a line graph always consists of two cliques. The triangle guarantees that these two cliques are connected by an edge and hence that the neighbourhood of the e-vertex is connected.

Thus we have the following results.

COROLLARY 6.2. *There is a polynomial algorithm to find the maximum edge dominating set in the following classes of graphs:*

(a) *2-connected claw-free chordal graphs;*

(b) *locally connected claw-free graphs;*

(c) *line graphs of total graphs.*

The time-complexity of the algorithm is $O(n^3)$, which can be improved to $O(m)$ for case (b).

The class of graphs for which the edge domination problem can be solved in polynomial time can be made larger. For the following algorithm to work, we must deal with a slightly more general problem. Given a graph G and a set of vertices R that we call the *required* vertices, find the smallest edge dominating set D such that all the required vertices are saturated by the edge dominating set. Note that the basic algorithm solves this extended problem for the classes of graphs mentioned in Corollary 6.2, if one modification is made. Instead of finding a maximum stable set in G, find a maximum stable set I in $G - R$. Then M matches vertices of R as much as possible, and D saturates the vertices of R. That D is a minimum edge dominating set that saturates R follows because $G - I$ has a perfect or near perfect matching in these classes of claw-free graphs. The following theorem gives a polynomial algorithm for some graphs that are not claw-free.

THEOREM 6.3. *Consider the class of graphs in which for each graph G the 2-connected induced subgraphs of G all belong to classes of graphs for which the edge domination with required vertices problem can be solved in polynomial time. Then the edge domination with required vertices problem can be solved in polynomial time for this class of graphs.*

Proof. Let G be a member of this class of graphs and let R be a required set of vertices. The algorithmic technique used is divide-and-conquer. We may assume that G is connected, since otherwise each connected component can be handled separately. If G is 2-connected, then the problem can be solved in polynomial time by hypothesis. Hence we may assume that G has a cut vertex v.

Let A be one component of $G - v$ containing at most $(n - 1)/2$ vertices and let B be the remainder of the graph, $G - A - v$. If A consists of only one vertex, then $\{u, v\}$ must be the only edge incident to u. The problem can be solved by making v a required vertex and finding a minimum edge dominating set saturating the required vertices in the smaller graph $G - u$. We henceforth assume that A contains at least two vertices.

Note the following inequalities, where $d_R(H)$ denotes the cardinality of $D_R(H)$, which in turn denotes a minimum edge dominating set of the subgraph H that saturates all the vertices of R in H. Also, let $H + x$ denote the subgraph induced by the vertices of H together with the vertex x

$$d_R(A) \leqq d_R(A + v) \leqq d_{R \cup \{v\}}(A + v) \leqq d_R(A) + 1.$$

The first inequality is true, since an edge dominating set in $A + v$ gives an edge dominating set of A of the same size, any edge incident to v being replaced by any adjacent edge in A. The second inequality is true, since an edge dominating set that saturates $R \cup \{u\}$ also saturates just R. The last inequality is true because an edge dominating set of A, together with any edge incident with v, gives an edge dominating set of A saturating v. These inequalities are also valid with A replaced by B.

Clearly, exactly one of these three inequalities is strict. Thus a minimum edge dominating set of G that saturates R, $D_R(G)$, must be one (or more) of

(1) $D_R(A) \cup D_{R \cup \{v\}}(B + v)$,

(2) $D_R(A + v) \cup D_R(B + v)$,

(3) $D_{R \cup \{v\}}(A + v) \cup D_R(B)$.

In the first case, v is saturated by an edge from B; in the last case, v is saturated by an edge from A; in the other case, v may not be saturated at all. However, we do not need to find all of these sets. Calculating which of the above inequalities for A is strict is sufficient to determine one formula that can be used. Thus we need only calculate all three of $D_R(A)$, $D_R(A+v)$, $D_{R\cup\{v\}}(A+v)$ and one of $D_{R\cup\{v\}}(B+v)$, $D_R(B+v)$, and $D_R(B)$. These, of course, are calculated recursively, using the algorithm itself, unless the subgraph involved is 2-connected, in which case the algorithm from the hypothesis is used. In fact, v can always be chosen so that $A+v$ is 2-connected. (Use a post-order transversal of the block-cutpoint tree (see [7] or [11, p. 36]), which can be found using depth-first search in $O(m)$ time (see [1, p. 185]).)

Let the time complexity of the algorithms solve the edge domination problem with required vertices for a 2-connected subgraph with n vertices be $f(n)$. Let the time complexity of the above algorithm be $t(n)$ (assuming that $A+v$ is always 2-connected and that the block-cutpoint tree is already known). Then, if $f(n)=O(n^{1+\varepsilon})$, $t(n)=O(n^{1+\varepsilon})$, whereas, if $n^{1+\varepsilon}=O(f(n))$, then $t(n)=O(f(n))$, where ε is any positive real number. This follows by induction from the relationship (when G is not 2-connected)

$$t(n) \leqq t(a) + 2f(a+1) + t(n-a),$$

where a is the number of vertices of A, $1 \leqq a < n/2$; $t(a)$ is a bound on the time to find $D_R(A)$; $f(a+1)$ is a bound on the time to find $D_R(A+v)$ and $D_{R\cup\{v\}}(A+v)$; $t(n-a)$ is a bound on the time to find the rest of $D_R(G)$ in the rest of the graph. □

There are several classes of graphs for which the edge domination problem can be solved in this way.

COROLLARY 6.4. *The edge domination problem can be solved in $O(n^3)$ time for the following classes of graphs*:

(a) *trees and cacti*,

(b) *claw-free chordal graphs*,

(c) *line graph of chordal graphs*,

(d) *line graph of any graph in which each nonbridge edge is in a triangle*.

The time bound can be reduced to $O(n^2)$ for cases (a) *and* (b).

Proof. (a) The problem can be solved in $O(n)$ time for trees and simple circuits.

(b) The edge domination problem for 2-connected claw-free chordal graphs can be solved in $O(m)$ time (Corollary 6.2(b)).

(c) Any nonbridge edge of a chordal graph must be in a triangle, and hence this case reduces to case (d).

(d) The 2-connected components of a line graph $L(G)$ are the line graphs of the 2-edge-connected component of G. Every edge of such a component is a nonbridge edge and hence in a triangle. By the discussion before Corollary 6.2, the line graph of a graph in which every edge is in a triangle is a locally connected claw-free graph. Thus the basic algorithm solves the problem for these 2-connected components, and the algorithm of Theorem 6.3 is applicable. □

Theorems 1.2 and 1.3 show that the stable set problem in $T(G)$ is polynomially equivalent to the edge domination problem in G. As well, Theorem 5.1 shows how to get a minimum edge dominating set for $T(G)$ from a stable set in $T(G)$.

COROLLARY 6.5. *The stable set problem and the edge domination problem can be solved in polynomial time for the class of total graphs and iterated total graphs, of any graph mentioned in Corollaries* 6.3 *and* 6.4.

7. Conclusion.

There are still many open questions concerning the complexity of the minimum edge domination problem. We do not know whether the problem is polynomial or NP-complete for many classes of graphs such as chordal graphs.

We were somewhat surprised to find that edge domination is such a difficult problem. It is NP-complete for many classes of graphs for which the stable set problem is solvable in polynomial time, including subdivision graphs, planar bipartite graphs, line graphs, perfect graphs, claw-free graphs, and even the intersection of the last two classes. On the other hand, we know of no class of graphs for which the stable set problem is NP-complete, and the edge domination problem is solvable in polynomial time. Of course, this may be due to our basic algorithm that requires having a maximum stable set first, before finding a minimum edge dominating set.

Appendix. Proof that vertex cover is NP-complete for 3-connected planar cubic graphs. The proof requires several transformations, starting from vertex cover for general graphs. We transform a general connected graph G by adding two new vertices adjacent to all other vertices of G, including each other. This new graph G^* has a vertex cover of size $k + 2$ if and only if G has a vertex cover of size k. G^* is clearly 3-connected.

The next transformation takes a general 3-connected graph to a planar 3-connected graph. The standard technique embeds the graph in the plane with only a polynomial number of edge-crossings and replaces each edge-crossing with a cross-over subgraph as given by Garey, Johnson, and Stockmeyer [9]. The following crossover subgraph, Fig. 5, can be used to maintain 3-connectedness.

At the same time, one replacement by this crossover increases the vertex cover by exactly 11.

To make a 3-connected planar graph cubic, each vertex of degree $d > 3$ can be replaced by a subgraph with one vertex of degree $d - 1$ and 9 vertices of degree 3 (Fig. 6).

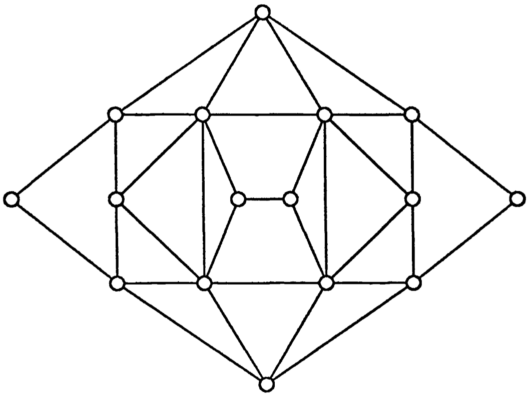


FIG. 5. *The crossover graph.*

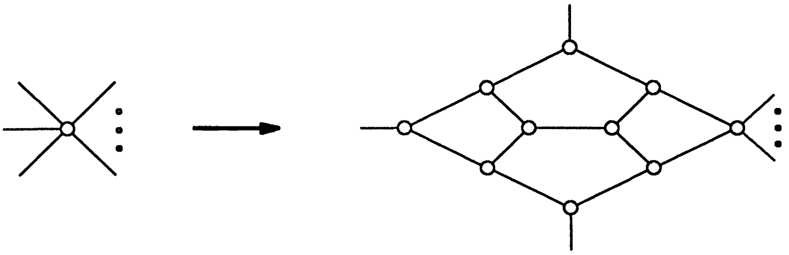


FIG. 6. *Degree reduction.*

This transformation is similar to that used in Garey and Johnson [8]. One replacement always increases the vertex cover by exactly 5. The number of replacements required to make the graph cubic equals the sum of the amounts by which the degrees of the vertices exceed 3, that is, $2m - 3n$. It is clear that the properties of 3-connectedness and planarity are maintained.

Acknowledgments. Thanks are due to the anonymous referees for improving Lemma 3.1, suggesting changes throughout the text, and drawing our attention to several literature references.

REFERENCES

[1] A. V. AHO, J. E. HOPCROFT, AND J. D. ULLMAN, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

[2] R. B. ALLAN AND R. LASKAR, *On domination and independent domination numbers of a graph*, Discrete Math., 23 (1978), pp. 73–76.

[3] G. J. CHANG AND G. L. NEMHAUSER, *The k-domination and k-stability problems in sum-free chordal graphs*, SIAM J. Algebraic Discrete Meth., 5 (1985), pp. 332–345.

[4] D. CORNEIL AND J. M. KEIL, *A dynamic programming approach the dominating set problem on k-trees*, SIAM J. Algebraic Discrete Meth., 8 (1987), pp. 534–543.

[5] J. EDMONDS, *Paths trees and flowers*, Canad. J. Math., 17 (1965), pp. 449–467.

[6] M. FARBER, *Applications of linear programming duality to problems involving independence and domination*, Tech. Report TR81-13, Dept. Comput. Sci., Simon Fraser Univ., Burnaby, British Columbia, Canada, 1981.

[7] T. GALLAI, *Elementare relationem bezüglich der glieder und trennenden punkte von graphen*, Magyar Tud. Akad. Mat. Kutato Int. Kozl., 9 (1964), pp. 235–236.

[8] M. R. GAREY AND D. S. JOHNSON, *The rectilinear Steiner tree problem is* NP-*complete*, SIAM J. Appl. Math., 32 (1977), pp. 826–834.

[9] M. R. GAREY, D. S. JOHNSON, AND L. STOCKMEYER, *Some simplified* NP-*complete graph problems*, Theoret. Comput. Sci., 1 (1976), pp. 237–267.

[10] F. GAVRIL, *Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph*, SIAM J. Comput., 1 (1972), pp. 180–187.

[11] F. HARARY, *Graph Theory*, Addison-Wesley, Reading, MA, 1969.

[12] M. JÜNGER, G. REINELT, AND W. R. PULLEYBLANK, *On partitioning the edges of graphs into connected subgraphs*, J. Graph Theory, 9 (1985), pp. 539–549.

[13] K. KILAKOS, *On the complexity of edge domination*, Master's thesis, University of New Brunswick, New Brunswick, Canada, 1988.

[14] D. KÖNIG, *Graphs and Matrices*, Mat. Fiz. Lapok, 38 (1931), pp. 116–119. (In Hungarian.)

[15] J. MINTY, *On maximal independent sets of vertices in claw-free graphs*, J. Combin. Theory Ser. B, 28 (1980), pp. 284–304.

[16] S. MITCHELL AND S. HEDETNIEMI, *Edge domination in trees*, in Proc. 8th Southeastern Conference on Combinatorics, Graph Theory and Computing, Utilitas Mathematica Publishing, Winnipeg, Canada, 1977, pp. 489–509.

[17] J. PETERSEN, *Die theorie der regulären graphs*, Acta Math., 15 (1891), pp. 193–220.

[18] C. SAVAGE, *Maximum matchings in trees*, Inform. Process. Lett., 10 (1980), pp. 202–205.

[19] N. SHIBI, *Algorithm de recherche d'un stable de cardinalité maximum dans un graphe sans étoile*, Discrete Math., 29 (1980), pp. 53–76.

[20] M. YANNAKAKIS AND F. GAVRIL, *Edge dominating sets in graphs*, SIAM J. Appl. Math., 38 (1980), pp. 364–372.

SIAM J. DISC. MATH.
Vol. 6, No. 3, pp. 388-399, August 1993

005

RECIPROCAL SUMS OVER PARTITIONS AND COMPOSITIONS*

A. KNOPFMACHER† AND J. N. RIDLEY‡

Abstract. The authors obtain precise asymptotic estimates for certain combinatorial sums over products of reciprocals of the summands in the partition or composition of a natural number n. These estimates are applied to determine the mean value of a certain arithmetical function over the polynomial ring $\mathbb{F}_q[X]$, where $\mathbb{F}_q$ is the finite field with q elements.

Key words. generating functions, infinite products, finite fields

AMS subject classifications. 05A15, 11T06

1. Introduction. In considering a problem concerning the factorization of polynomials over a large finite field, Greene and Knuth [3, p. 52] were led to consider the infinite product generating function

$$h(z) = \prod_{n=1}^{\infty}\left(1+\frac{z^n}{n}\right). \tag{1.1}$$

If we regard this generating function purely from a combinatorial point of view, the coefficient of z^n in $h(z)$ corresponds to the sum

$$h(n) = \sum_{\substack{i_1+i_2+\cdots+i_k=n\\ 1\leqq i_1<i_2<\cdots<i_k\leqq n\\ k\geqq 1}} \frac{1}{i_1 i_2\cdots i_k}, \tag{1.2}$$

summed over all *distinct partitions* of the positive integer n. It seems natural to investigate the asymptotic behaviour of the analogous combinatorial sums

$$f(n) = \sum_{\substack{i_1+i_2+\cdots+i_k=n\\ 1\leqq i_1\leqq i_2\leqq\cdots\leqq i_k\leqq n\\ k\geqq 1}} \frac{1}{i_1 i_2\cdots i_k}, \tag{1.3}$$

summed over all *partitions* of n, and

$$g(n) = \sum_{\substack{i_1+i_2+\cdots+i_k=n\\ i_j\geqq 1, 1\leqq j\leqq k,\\ k\geqq 1}} \frac{1}{i_1 i_2\cdots i_k}, \tag{1.4}$$

summed over all *compositions* (ordered partitions) of n.

Corresponding to the asymptotic estimate

$$h(n) = e^{-\gamma} + \frac{e^{-\gamma}}{n} + O\left(\frac{\log n}{n^2}\right), \qquad n \to \infty,$$

obtained by Greene and Knuth [3], we show the following theorem by a related but somewhat different approach.

* Received by the editors December 18, 1990; accepted for publication (in revised form) August 11, 1992.

† Department of Computational and Applied Mathematics, University of the Witwatersrand, Johannesburg, Private Bag 3, WITS, 2050 South Africa.

‡ Department of Mathematics, University of the Witwatersrand, Johannesburg, Private Bag 3, WITS, 2050 South Africa.

THEOREM 1. *Let $f(n)$ be defined by* (1.3); *then, as $n \to \infty$,*

$$f(n) = e^{-\gamma}(n - \log n + 1 + \log 2 - \gamma) + O\left(\frac{\log n}{n}\right).$$

Here and henceforth, γ denotes Euler's constant. In addition, by applying techniques of complex integration we show the following theorem.

THEOREM 2. *Let $g(n)$ be defined by* (1.4); *then, as $n \to \infty$,*

$$g(n) = \frac{1}{e-1}\left(\frac{e}{e-1}\right)^n + O(1).$$

As a consequence of Theorem 2, we can deduce an asymptotic estimate for the following sum over the unsigned Stirling numbers of the first kind.

COROLLARY 3. *It holds that*

$$\sum_{k=1}^{n} k!s(n,k) = n!\left\{\frac{1}{(e-1)}\left(\frac{e}{e-1}\right)^n + O(1)\right\}, \qquad n \to \infty.$$

In § 3 we apply Theorem 2 to obtain an estimate for the mean value of a certain arithmetical function over $\mathbb{F}_q[X]$, where $\mathbb{F}_q[X]$ denotes the polynomial ring in one indeterminate X, over a finite field $\mathbb{F}_q$. Finally, in § 4 we interpret Theorems 1 and 2 in terms of cycles of permutations.

2. Analysis. Our approach to Theorems 1 and 2 is based on an analysis of the respective (ordinary) generating functions whose coefficients are $f(n)$ and $g(n)$. In the first case, it is easily verified that $f(n)$ has generating function

$$f(z) = \prod_{n=1}^{\infty}\left(1 - \frac{z^n}{n}\right)^{-1} \tag{2.1}$$

(compare (1.1)). In the second case, we use the result (see, e.g., Jordan [4, p. 146]) that

$$g_k(n) = \sum_{\substack{i_1+i_2+\cdots+i_k=n \\ i_j \geq 1, 1 \leq j \leq k}} \frac{1}{i_1 i_2 \cdots i_k}$$

has generating function $(-\log(1-z))^k$ for fixed k. Since $g(n) = \sum_{k=1}^{n} g_k(n)$, it follows that $g(n)$ has generating function

$$\sum_{k=1}^{\infty} (-\log(1-z))^k = \frac{-\log(1-z)}{1+\log(1-z)}. \tag{2.2}$$

To prove Theorem 1, we require a number of preliminary results, see below:

$$\begin{aligned} &\text{Define } r(z) = (1-z)^2 f(z), \qquad z \neq 1 \\ &\text{and let } r(z) = \sum_{n=0}^{\infty} r_n z^n,\ r_0 = 1. \end{aligned} \tag{2.3}$$

Also, let $c(z)$ denote the formal power series

$$c(z) = \sum_{n=1}^{\infty} c_n z^n = z\frac{r'(z)}{r(z)}. \tag{2.4}$$

Formal integration and exponentiation of (2.4) leads to

$$r(z) = \exp\left(\sum_{n=1}^{\infty} \frac{c_n}{n} z^n\right). \tag{2.5}$$

PROPOSITION 4. *For $n \geqq 1$,*

$$\text{(i)} \quad nr_n = \sum_{k=1}^{n} c_k r_{n-k}, \tag{2.6}$$

$$\text{(ii)} \quad c_1 = -1 \quad \textit{and} \quad c_n = \sum_{\substack{d|n \\ 1<d<n}} \left(\frac{d}{n}\right)^{d-1} \quad \textit{for } n \geqq 2. \tag{2.7}$$

Proof. (i) By (2.4) and (2.3), we have the identity

$$\left(\sum_{n=1}^{\infty} c_n z^n\right)\left(\sum_{n=0}^{\infty} r_n z^n\right) = \sum_{n=1}^{\infty} nr_n z^n.$$

By equating coefficients on each side, (2.6) follows.

(ii) We have

$$\sum_{n=1}^{\infty} (2 + c_n) z^n = \frac{2z}{1-z} + \sum_{n=1}^{\infty} c_n z^n = z\frac{f'(z)}{f(z)} \quad \text{by (2.3).}$$

By dividing by z and then integrating, we obtain

$$\sum_{n=1}^{\infty} \frac{(2+c_n)z^n}{n} = \log f(z) = -\sum_{n=1}^{\infty} \log\left(1 - \frac{z^n}{n}\right)$$
$$= \sum_{n=1}^{\infty}\sum_{m=1}^{\infty} \frac{z^{nm}}{mn^m} = \sum_{n=1}^{\infty}\left(\sum_{d|n} d^{-1}\left(\frac{d}{n}\right)^d\right) z^n.$$

It follows that

$$2 + c_n = \sum_{d|n}\left(\frac{d}{n}\right)^{d-1} = 2 + \sum_{\substack{d|n \\ 2\leqq d<n}}\left(\frac{d}{n}\right)^{d-1},$$

as required.

PROPOSITION 5. *We have that*

$$\text{(i)} \quad c_n = \frac{1+(-1)^n}{n} + O\left(\frac{1}{n^2}\right); \tag{2.8}$$

$$\text{(ii)} \quad r_n = O\left(\frac{1}{n^2}\right); \tag{2.9}$$

$$\text{(iii)} \quad r(-1) = 4 \quad \textit{and} \quad r(1) = e^{-\gamma}. \tag{2.10}$$

Proof. (i) If n is even, then, by separating out the term for $d = 2$ from (2.7), we obtain

$$c_n = \frac{2}{n} + \sum_{\substack{d|n \\ 3\leqq d<n}}\left(\frac{d}{n}\right)^{d-1}$$
$$= \frac{2}{n} + \frac{1}{n^2}\sum_{\substack{d|n \\ 3\leqq d<n}} d^2\left(\frac{d}{n}\right)^{d-3}$$

$$= \frac{2}{n} + \frac{1}{n^2} O\left(\sum_{d=3}^{\infty} d^2 \left(\frac{1}{2}\right)^{d-3} \right)$$

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

If n is odd, then there is no term for $d = 2$, so $c_n = O(1/n^2)$, and (2.8) follows.

(ii) We use induction on n. By (2.6),

$$n r_n = \sum_{k=1}^{n-1} c_k r_{n-k} + c_n$$

$$= O\left(\sum_{k=1}^{n-1} \frac{1}{k} \frac{1}{(n-k)^2} \right) + O\left(\frac{1}{n}\right),$$

using (2.8) and the inductive hypothesis. This gives

$$n r_n = O\left(\frac{1}{n^2} \sum_{k=1}^{n-1} \left(\frac{1}{k} + \frac{1}{n-k} \right) \right) + O\left(\frac{1}{n} \sum_{k=1}^{n-1} \frac{1}{(n-k)^2} \right) + O\left(\frac{1}{n}\right)$$

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

which gives (2.9).

(iii) It follows from (2.9) that the power series (2.3) for $r(z)$ converges absolutely and uniformly on and inside the unit circle, so $r(z)$ is continuous for $|z| \leqq 1$. At $z = -1$, we may use the product expression (2.1) to give

$$r(-1) = (1 - (-1))^2 \prod_{n=1}^{\infty} \left(1 - \frac{(-1)^n}{n} \right)^{-1}$$

$$= 4 \cdot \left(\frac{2}{1} \cdot \frac{1}{2} \cdot \frac{4}{3} \cdot \frac{3}{4} \cdot \frac{6}{5} \cdot \frac{5}{6} \cdots \right)^{-1} = 4.$$

From (2.3) and (2.1), we have for $|z| < 1$ that

$$\log r(z) = 2 \log (1 - z) - \sum_{n=1}^{\infty} \log \left(1 - \frac{z^n}{n} \right)$$

$$= \log (1 - z) - \sum_{n=2}^{\infty} \log \left(1 - \frac{z^n}{n} \right)$$

$$= - \sum_{n=1}^{\infty} \frac{z^n}{n} + \sum_{n=2}^{\infty} \left\{ \frac{z^n}{n} + \sum_{m=2}^{\infty} \frac{z^{mn}}{n^m m} \right\}$$

$$= -1 + \sum_{n=2}^{\infty} \sum_{m=2}^{\infty} \frac{z^{mn}}{n^m m}$$

$$= -1 + \lim_{M \to \infty} \sum_{m=2}^{M} \sum_{n=2}^{\infty} \frac{z^{mn}}{n^m m}.$$

Thus, since $r(z)$ is continuous for $|z| \leqq 1$,

$$\log r(1) = -1 + \lim_{M \to \infty} \sum_{m=2}^{M} \left\{ -\frac{1}{m} - \log \left(1 - \frac{1}{m} \right) \right\}$$

$$= \lim_{M \to \infty} \left\{ - \sum_{m=1}^{M} \frac{1}{m} + \log \left(\frac{2}{1} \cdot \frac{3}{2} \cdot \frac{4}{3} \cdots \frac{M}{M-1} \right) \right\}$$

$$= -\gamma.$$

Remark. If $\zeta(s) = \sum_{n=1}^{\infty} (1/n^s)$ denotes the Riemann zeta function, we can deduce from the proof of Proposition 5 the identity

$$\sum_{s=2}^{\infty} \frac{\zeta(s) - 1}{s} = 1 - \gamma.$$

By contrast, it is easily seen that $\sum_{s=2}^{\infty} (\zeta(s) - 1) = 1$. These elementary identities do not appear in standard textbooks on the Riemann zeta function such as [10].

PROPOSITION 6. *As* $n \to \infty$,

$$r_n = \frac{(e^{-\gamma} + 4(-1)^n)}{n^2} + O\left(\frac{\log n}{n^3}\right). \tag{2.11}$$

Proof. From (2.6), (2.8), and (2.9),

$$\begin{aligned}
nr_n &= \sum_{k=1}^{n} c_k r_{n-k} = \sum_{m=0}^{n-1} c_{n-m} r_m \\
&= \sum_{m=0}^{n-1} \frac{(1 + (-1)^{n-m})}{n - m} r_m + \left\{O\left(\frac{1}{n^2}\right) + O\left(\sum_{m=1}^{n-1} \frac{1}{m^2} \frac{1}{(n-m)^2}\right)\right\} \\
&= \left\{\frac{1}{n} \sum_{m=0}^{n-1} (1 + (-1)^{n-m}) r_m + O\left(\sum_{m=1}^{n-1} \left(\frac{1}{n-m} - \frac{1}{n}\right) \frac{1}{m^2}\right)\right\} + O\left(\frac{1}{n^2}\right) \\
&= \left\{\frac{1}{n} (r(1) + (-1)^n r(-1)) + O\left(\frac{1}{n^2}\right)\right\} + O\left(\frac{1}{n^2} \sum_{m=1}^{n-1} \left(\frac{1}{n-m} + \frac{1}{m}\right)\right) + O\left(\frac{1}{n^2}\right) \\
&= \frac{1}{n} (e^{-\gamma} + 4(-1)^n) + O\left(\frac{\log n}{n^2}\right)
\end{aligned}$$

by (2.10).

Proof of Theorem 1. By equating coefficients in the defining identity $f(z) = (1 - z)^{-2} r(z)$ and using (2.10), we obtain

$$\begin{aligned}
f(n) &= \sum_{k=0}^{n} (n + 1 - k) r_k = (n + 1) \sum_{k=0}^{n} r_k - \sum_{k=0}^{n} k r_k \\
&= (n + 1) e^{-\gamma} - (n + 1) \sum_{k=n+1}^{\infty} r_k - \sum_{k=0}^{n} k r_k.
\end{aligned} \tag{2.12}$$

First, by (2.11),

$$\begin{aligned}
\sum_{k=n+1}^{\infty} r_k &= e^{-\gamma} \sum_{k=n+1}^{\infty} \frac{1}{k^2} + 4 \sum_{k=n+1}^{\infty} \frac{(-1)^k}{k^2} + O\left(\sum_{k=n+1}^{\infty} \frac{\log k}{k^3}\right) \\
&= \frac{e^{-\gamma}}{n} + O\left(\frac{1}{n^2}\right) + O\left(\frac{\log n}{n^2}\right).
\end{aligned} \tag{2.13}$$

Hence

$$(n + 1) \sum_{k=n+1}^{\infty} r_k = e^{-\gamma} + O\left(\frac{\log n}{n}\right). \tag{2.14}$$

Also by (2.11),

$$\sum_{k=1}^{n} kr_k = e^{-\gamma} \sum_{k=1}^{n} \frac{1}{k} + 4 \sum_{k=1}^{n} \frac{(-1)^k}{k} + \sum_{k=1}^{n} O\left(\frac{\log k}{k^2}\right)$$
$$= e^{-\gamma} \log n + O(1),$$

from which we can immediately deduce the estimate

$$f(n) = e^{-\gamma}(n - \log n) + O(1).$$

The sharper estimate of Theorem 1 requires a more detailed analysis, which is set forth in Lemma 7, below.

LEMMA 7. *As* $n \to \infty$,

$$\sum_{k=1}^{n} kr_k = e^{-\gamma}\{\log n + \gamma - 1 - \log 2\} + O\left(\frac{\log n}{n}\right). \tag{2.15}$$

Proof. By (2.6),

$$\sum_{k=1}^{n} kr_k = \sum_{k=1}^{n} \sum_{j=1}^{k} c_j r_{k-j} = \sum_{j=1}^{n} c_j \sum_{k=j}^{n} r_{k-j}$$
$$= \sum_{j=1}^{n} c_j \sum_{k=0}^{n-j} r_k = \sum_{j=1}^{n} c_j \left\{ \sum_{k=0}^{\infty} r_k - \sum_{k=n-j+1}^{\infty} r_k \right\} \tag{2.16}$$
$$= e^{-\gamma} \sum_{j=1}^{n} c_j - \sum_{j=1}^{n} c_j \sum_{k=n-j+1}^{\infty} r_k.$$

Now, by (2.7),

$$\sum_{j=1}^{n} c_j = -1 + \sum_{j=4}^{n} \sum_{\substack{d \mid j \\ 2 \leqq d \leqq j/2}} \left(\frac{d}{j}\right)^{d-1}$$
$$= -1 + \sum_{\substack{j=4 \\ 2 \mid j}}^{n} \frac{2}{j} + \sum_{j=6}^{n} \sum_{\substack{d \mid j \\ 3 \leqq d \leqq j/2}} \left(\frac{d}{j}\right)^{d-1}. \tag{2.17}$$

Now

$$\sum_{\substack{j=4 \\ 2 \mid j}}^{n} \frac{2}{j} = \sum_{k=2}^{[n/2]} \frac{1}{k} = \log\left(\frac{n}{2}\right) + \gamma - 1 + O\left(\frac{1}{n}\right). \tag{2.18}$$

Next, if $j = qd$, then

$$\sum_{j=6}^{n} \sum_{\substack{d \mid j \\ 3 \leqq d \leqq j/2}} \left(\frac{d}{j}\right)^{d-1} = \sum_{q=2}^{[n/3]} \sum_{d=3}^{[n/q]} \left(\frac{1}{q}\right)^{d-1}$$
$$= \sum_{q=2}^{[n/3]} \frac{1}{q^2} \frac{1 - (q^{-1})^{[n/q]-2}}{1 - q^{-1}}$$
$$= \sum_{q=2}^{[n/3]} \frac{1}{q(q-1)} - \sum_{q=2}^{[n/3]} \frac{q^{-[n/q]}}{1 - q^{-1}} \tag{2.19}$$
$$= 1 + O\left(\frac{1}{n}\right) + O\left(\sum_{q=2}^{[n/3]} q^{-[n/q]}\right).$$

To bound this sum, we use

$$\sum_{q=2}^{[n/3]} q^{-[n/q]} = O\left\{\left(\left[\frac{n}{3}\right]-1\right)\max_{2\leq q\leq [n/3]} q^{-[n/q]}\right\}$$
$$= O\left\{\left(\left[\frac{n}{3}\right]-1\right)\left[\frac{n}{3}\right]^{-3}\right\} = O\left(\frac{1}{n^2}\right),$$

since the function x^{nx} decreases for $0 < x < e^{-1}$. Substituting this into (2.19), and then substituting (2.19) and (2.18) into (2.17), yields

$$(2.20) \qquad \sum_{j=1}^{n} c_j = \log n - \log 2 + \gamma - 1 + O\left(\frac{1}{n}\right).$$

Next, consider the second sum in (2.16). Since the inner sum is $O(1/(n-j+1))$, by (2.13) and $c_j = O(1/j)$ by (2.8),

$$(2.21) \qquad \begin{aligned}\sum_{j=1}^{n} c_j \sum_{k=n-j+1}^{\infty} r_k &= O\left(\sum_{j=1}^{n}\frac{1}{j}\frac{1}{n-j+1}\right)\\ &= O\left(\frac{\log n}{n}\right).\end{aligned}$$

We substitute (2.21) and (2.20) into (2.16) to obtain (2.15), which completes the proof of Lemma 7.

Theorem 1 now follows from (2.12), (2.14), and (2.15).

Remarks. (i) Equation (2.1) is a special case of a general product expansion $\prod_{n=1}^{\infty}(1 - a_n z^n)^{-1}$, which can be developed for an arbitrary power series $A(z) = 1 + A_1 z + A_2 z^2 + \cdots$. These and other more general product representations for power series are investigated in [6]. In addition, the problem of finding the region of analyticity of such product expansions corresponding to a given function $A(z)$ is considered in [5].

(ii) We can further improve the error estimate in Theorem 1 from $O(\log n/n)$ to $e^{-\gamma}\log n/n + O(1/n)$, but we omit the details. This improved estimate, which we denote by $\hat{f}(n)$, can be used to obtain accurate numerical estimates for the coefficients $f(n)$, even for fairly small n, as Table 1 shows.

(iii) Since the generating function (2.1) has the unit circle as a natural boundary, we cannot apply standard asymptotic techniques such as the Darboux method (see, e.g., Bender [1]) or the transfer methods for singularity analysis of generating functions, developed recently by Flajolet and Odlyzko [2]. These methods are, however, applicable in the case of generating function (2.2) but lead to weaker error estimates than that of Theorem 2.

Proof of Theorem 2. By substituting $w = 1 - z$ into (2.2), we obtain

$$\sum_{n=1}^{\infty} g(n)(1-w)^n = \frac{-\log w}{1+\log w}.$$

We wish to expand this function as a power series about $w = 1$. It has a branch point at $w = 0$ because of $\log w$ and simple pole at $w = e^{-1}$ with residue e^{-1}. Therefore $-\log w/(1+\log w) - e^{-1}/(w - e^{-1})$ is analytic for $|w-1| < 1$. In addition, this function is bounded on the circle $|w-1| = 1$ because

$$\frac{-\log w}{1+\log w} = \frac{-1}{(\log w)^{-1}+1} \to -1 \quad \text{as } w \to 0.$$

TABLE 1
Estimates of $f(n)$ with errors of order $1/n$.

n	$f(n)$	$\hat{f}(n)$	$n(f(n)-\hat{f}(n))$
100	54.21582	54.21274	0.3077
200	109.95989	109.95853	0.2723
300	165.87349	165.87263	0.2581
400	221.85541	221.85479	0.2501
500	277.87451	277.87402	0.2449
600	333.91701	333.91661	0.2412
700	389.97562	389.97528	0.2384
800	446.04599	446.04569	0.2363
900	502.12532	502.12506	0.2345
1000	558.21172	558.21149	0.2331
1100	614.30383	614.30362	0.2319
1200	670.40065	670.40046	0.2308
1300	726.50142	726.50124	0.2299
1400	782.60556	782.60539	0.2292
1500	838.71259	838.71244	0.2285
1600	894.82214	894.82200	0.2278
1700	950.93391	950.93378	0.2273
1800	1007.04764	1007.04752	0.2268
1900	1063.16312	1063.16300	0.2263
2000	1119.28017	1119.28005	0.2259

Suppose therefore that

$$G(w) = \frac{-\log w}{1+\log w} - \frac{e^{-1}}{w-e^{-1}} = \sum_{n=0}^{\infty} a_n (w-1)^n. \tag{2.22}$$

By Taylor's theorem,

$$a_n = \frac{1}{2\pi i}\oint_C \frac{G(w)}{(w-1)^{n+1}}\,dw,$$

where, by the above, C can be taken to be the circle $|w-1| = 1$. Therefore

$$|a_n| \leqq \max\{|G(w)| : |w-1| = 1\}, \tag{2.23}$$

since C has circumference 2π. Now, for w on C,

$$\begin{aligned} |G(w)| &\leqq \left|\frac{\log w}{1+\log w}\right| + \left|\frac{1}{ew-1}\right| \\ &\leqq \frac{1}{\min\limits_{|w-1|=1}\{1+(\log w)^{-1}\}} + 1 = O(1). \end{aligned} \tag{2.24}$$

For $|1-w| < 1-e^{-1}$, however,

$$\begin{aligned} \sum_{n=1}^{\infty} g(n)(1-w)^n &= \sum_{n=0}^{\infty} a_n(w-1)^n + \frac{e^{-1}}{(w-1)-(e^{-1}-1)} \\ &= \sum_{n=0}^{\infty} a_n(w-1)^n + \frac{e^{-1}}{1-e^{-1}}\sum_{n=0}^{\infty}\left(\frac{1-w}{1-e^{-1}}\right)^n. \end{aligned}$$

It follows that

$$g(n) = \frac{e^{-1}}{(1-e^{-1})^{n+1}} + (-1)^n a_n$$
$$= \frac{1}{(e-1)}\left(\frac{e}{e-1}\right)^n + O(1),$$

using (2.24).

Remarks. (i) Using a computer, we obtain the numerical estimate

$$\min_{|w-1|=1}\left(1 + \frac{1}{\log w}\right) = 0.72159\cdots.$$

From this, we deduce that

$$\left|g(n) - \frac{1}{(e-1)}\left(\frac{e}{e-1}\right)^n\right| < \frac{1}{0.7216} + 1 = 2.3858\cdots.$$

(ii) The generating function $g(z)$ can be used to derive the following recurrence relation for $g(n)$, which is useful for computational purposes. Since

$$g(z)(1 + \log(1-z)) = -\log(1-z),$$

we can equate coefficients on each side of this identity to obtain

$$(2.25) \qquad g(n) = \sum_{k=0}^{n-1} \frac{g(k)}{n-k}, \qquad n \geqq 1,$$

where we define $g(0) = 1$.

Proof of Corollary 3. By Jordan [4, p. 146], the unsigned Stirling number of the first kind $s(n, k)$ is given by the formula

$$(2.26) \qquad s(n,k) = \frac{n!}{k!} \sum_{i_1+i_2+\cdots+i_k=n} \frac{1}{i_1 i_2 \cdots i_k},$$

summed over all compositions of n of length k. It follows that $(1/n!) \sum k! s(n,k) = g(n)$, from which we now deduce the asymptotic estimate using Theorem 2.

3. Ordered irreducible factorizations of polynomials. As is the case with Greene and Knuth's estimates for $h(n)$, our asymptotic estimates for $g(n)$ can be applied to investigate a problem involving polynomials over a large finite field.

Let $\mathbb{F}_q[X]$ denote a polynomial ring in one indeterminate X over a finite field $\mathbb{F}_q$ with exactly q elements. Let P_n denote the set of all monic polynomials of degree n in $\mathbb{F}_q[X]$, of cardinality q^n. Let $\pi(n) = \pi_q(n)$ denote the number of monic irreducible polynomials in P_n. It is well known (see [8, p. 93]) that

$$(3.1) \qquad \pi(n) = \frac{q^n}{n}(1 + O(q^{-n/2})).$$

For $a(X) \in P_n$, we use $\Omega(a)$ and $\omega(a)$ to denote the arithmetical functions that count the total number (respectively, the distinct number) of irreducible factors of $a(X)$.

These functions are investigated in [7] for general additive arithmetical semigroups that satisfy a certain axiom. For convenience, we confine our discussion below to the case of $\mathbb{F}_q[X]$. We introduce a new arithmetical function as follows: For any $a(X) \in P_n$, define $b(a)$ to be the number of *ordered factorizations* of $a(X)$ into irreducibles. The function b is prime independent, since, if $p(X)$ is any irreducible polynomial, then $b(p^m(X)) = 1$ for all m, which is independent of the choice of $p(X)$. However, the function b is not multiplicative, since $b(p_1p_2) = 2 \neq b(p_1)b(p_2) = 1$ for irreducible polynomials p_1, p_2. In general, if $a(X) = p_1^{\alpha_1}(X)p_2^{\alpha_2}(X)\cdots p_k^{\alpha_k}(X)$, where $k = \omega(a)$, $\alpha_1 + \alpha_2 + \cdots + \alpha_k = \Omega(a)$, we have $b(a) = \Omega(a)!/\alpha_1!\alpha_2!\cdots\alpha_k!$. We deduce that

$$\omega(a)! \leqq b(a) \leqq \Omega(a)! \tag{3.2}$$

with $b(a) = \omega(a)!$ in the case that $a(X)$ is square-free.

An arithmetical function $F : \mathbb{F}_q[X] \to \mathbb{R}$ is said to be of *normal order* $G(n)$ if, for almost all polynomials $a(X)$ in P_n (i.e., for all except $o(q^n)$ polynomials in P_n),

$$F(a) = G(n)(1 + o(1)) \quad \text{as } n \to \infty.$$

We now show that $\log b(a)$ has normal order $\log n(\log\log n - 1)$.

THEOREM 8. *For almost all $a(X) \in P_n$ and any $\delta > 0$,*

$$\log b(a) = \log n(\log\log n - 1)(1 + O(\log n)^{-1/2+\delta}) \quad \text{as } n \to \infty.$$

Proof. By Theorem 9.7 of Knopfmacher [7, p. 90] applied in the case of $\mathbb{F}_q[X]$, we have, for almost all $a \in P_n$ and for any $\delta > 0$, that

$$\omega(a) = \log n(1 + O((\log n)^{-1/2+\delta})), \qquad \Omega(a) = \log n(1 + O((\log n)^{-1/2+\delta})).$$

Thus, for almost every $a \in P_n$,

$$\log \omega(a) = \log\log n + \log(1 + O(\log n)^{-1/2+\delta}) = \log\log n + O(\log n)^{-1/2+\delta}.$$

Now, by Stirling's formula,

$$\log n! = n(\log n - 1)\left(1 + O\left(\frac{1}{n}\right)\right).$$

Hence, for almost every $a \in P_n$,

$$\begin{aligned}\log \omega(a)! &= \log n(1 + O(\log n)^{-1/2+\delta})(\log\log n - 1)\\ &\quad\times\left(1 + O\left(\frac{(\log n)^{-1/2+\delta}}{\log\log n}\right)\right)\left(1 + O\left(\frac{1}{\log n}\right)\right)\\ &= \log n(\log\log n - 1)(1 + O(\log n)^{-1/2+\delta}).\end{aligned}$$

Since exactly the same estimate holds for $\log \Omega(a)!$, we can deduce the result from (3.2).

In addition to the normal value, it is useful to investigate the *mean value* of $b(a)$, given by $(1/q^n)\sum_{a\in P_n} b(a)$. By letting $q \to \infty$, we can apply Theorem 2 to estimate the asymptotic mean value of $b(a)$ as $n \to \infty$.

THEOREM 9. *As $q \to \infty$, the asymptotic mean value of $b(a)$ tends to*

$$\frac{1}{q^n}\sum_{a\in P_n} b(a) = \frac{1}{e-1}\left(\frac{e}{e-1}\right)^n + O(1), \quad n \to \infty.$$

Proof. By letting $q \to \infty$ in (3.1), we may take the proportion of irreducible polynomials in P_n, namely $\pi(n)/q^n$, to be $1/n$. Then

$$\begin{aligned}\frac{1}{q^n}\sum_{a\in P_n} b(a) &= \frac{1}{q^n}\sum_{\substack{i_1+i_2+\cdots+i_k=n\\ i_j\geqq 1, 1\leqq i\leqq k\\ k\geqq 1}} \pi(i_1)\pi(i_2)\cdots\pi(i_k)\\ &= g(n).\end{aligned}$$

Now we apply Theorem 2 to obtain the asymptotic estimate.

Note. (i) From the logarithm of the asymptotic mean value, we obtain $\log g(n) = n \log (e/(e-1)) + O(1)$, in contrast to the normal order of $\log b(a)$ in Theorem 8.

(ii) The mean value of the related arithmetical function $b_k(a)$, which denotes the number of ordered factorizations of $a(X)$ into products of exactly k irreducibles, is estimated as $n \to \infty$ by Knopfmacher [7, Lem. 9.12, p. 94] for fixed k. These estimates are then applied to obtain the mean value of several related arithmetical functions when k is fixed.

(iii) By letting $q \to \infty$ and using (2.28), we can show in the same way as in the proof of Theorem 9 that $b_k(a)$ has mean value $(k!/n!)s(n, k)$. By applying known asymptotics for $s(n, k)$ (see, e.g., [9]), it is possible to extend the estimates for the mean value of $b_k(a)$ in [7] to the range $1 \leqq k \leqq n$.

As noted by the referee, our estimate for $f(n)$ can also be applied to determine the asymptotic mean value of an arithmetical function $d : \mathbb{F}_q[X] \to \mathbb{R}$. For any $a(X) \in P_n$, define $d(a)$ to be the number of ways of writing $a(X) = p_1(X)p_2(X)\cdots p_s(X)$, where p_i are irreducible and $\deg p_i \geqq \deg p_{i+1}$.

THEOREM 10. *As $q \to \infty$, the asymptotic mean value of $d(a)$ tends to*

$$e^{-\gamma}(n - \log n + 1 + \log 2 - \gamma) + O\left(\frac{\log n}{n}\right), \quad n \to \infty.$$

Proof. We have that

$$\frac{1}{q^n} \sum_{a \in P_n} d(a) = \frac{1}{q^n} \sum_{\substack{i_1 + i_2 + \cdots + i_k = n \\ i_1 \leqq i_2 \leqq \cdots \leqq i_k \\ 1 \leqq i_j; k \geqq 1}} \pi(i_1)\pi(i_2)\cdots\pi(i_k)$$

$$= f(n) \quad \text{as } q \to \infty.$$

4. Cycles of permutations. Let t_n be the probability that a permutation of n letters has cycles whose lengths are all different. Wilf [11, p. 97] shows that $\{t_n\}$ has ordinary generating function $\prod_{n=1}^{\infty} (1 + z^n/n)$. It follows from the estimate for $h(n)$ obtained by Greene and Knuth [3] that

$$t_n = e^{-\gamma} + \frac{e^{-\gamma}}{n} + O\left(\frac{\log n}{n^2}\right), \quad n \to \infty.$$

Similarly, we can interpret Theorems 1 and 2 in terms of cycles of permutations.

Given a permutation $\pi \in \prod_n$, the set of permutations on n letters, let $u(\pi)$ denote the number of *ordered decompositions* of π into cycles.

THEOREM 12. *The asymptotic mean value of $u(\pi)$ is*

$$\frac{1}{n!} \sum_{\pi \in \Pi_n} u(\pi) = \left(\frac{1}{e-1}\right)\left(\frac{e}{e-1}\right)^n + O(1), \quad n \to \infty.$$

Proof. The result follows immediately from Corollary 3, since

$$\frac{1}{n!} \sum_{\pi \in \Pi_n} u(\pi) = \frac{1}{n!} \sum_{k=1}^{n} k!s(n, k).$$

Finally, given $\pi \in \prod_n$, define $v(\pi)$ to be the number of decompositions of π into cycles $\pi_1\pi_2\cdots\pi_s$, where the cycles π_i are in nondecreasing order with respect to their lengths. Then $\sum_{\pi \in \Pi_n} v(\pi)$ has exponential generating function given by (2.1), which leads to the following theorem.

THEOREM 13. *The asymptotic mean value of $v(\pi)$ is*

$$\frac{1}{n!}\sum_{\pi\in\Pi_n} v(\pi) = e^{-\gamma}(n - \log n + 1 + \log 2 - \gamma) + O\left(\frac{\log n}{n}\right), \quad n \to \infty.$$

REFERENCES

[1] E. A. BENDER, *Asymptotic methods in enumeration*, SIAM Rev., 16 (1974), pp. 485–515.

[2] P. FLAJOLET AND A. ODLYZKO, *Singularity analysis of generating functions*, SIAM J. Discrete Math., 3 (1990), pp. 216–240.

[3] D. H. GREENE AND D. E. KNUTH, *Mathematics for the Analysis of Algorithms*, 2nd ed. Birkhäuser, Boston, 1982.

[4] C. JORDAN, *Calculus of Finite Differences*, 3rd ed., Chelsea, London, 1979.

[5] A. KNOPFMACHER, *Infinite product factorizations of analytic functions*, J. Math. Anal. Appl., 162 (1991), pp. 526–536.

[6] A. KNOPFMACHER, J. KNOPFMACHER, AND J. N. RIDLEY, *Unique factorizations of formal power series*, J. Math. Anal. Appl., 149 (1990), pp. 402–411.

[7] J. KNOPFMACHER, *Analytic Arithmetic of Algebraic Function Fields*. Marcel Dekker, New York, 1979.

[8] R. LIDL AND H. NIEDERREITER, *Finite Fields*, Addison-Wesley, Reading, MA, 1983.

[9] L. MOSER AND M. WYMAN, *Asymptotic development of the Stirling numbers of the first kind*, J. London Math. Soc., 33 (1958), pp. 133–146.

[10] E. L. TITCHMARSH AND D. R. HEATH-BROWN, *The Theory of the Riemann Zeta-Function*, 2nd ed., Oxford University Press, Oxford, UK, 1988.

[11] H. S. WILF, *generatingfunctionology*, Academic Press, New York, 1990.

SIAM J. DISC. MATH.
Vol. 6, No. 3, pp. 400-417, August 1993

DOMINATION ON COCOMPARABILITY GRAPHS*

DIETER KRATSCH† AND LORNA STEWART‡

Abstract. The authors determine the algorithmic complexity of domination and variants on cocomparability graphs, a class of perfect graphs containing both the interval and the permutation graphs. Minimum dominating, total dominating, connected dominating, and independent dominating sets can be constructed in polynomial time. On the other hand, DOMINATING CLIQUE and MINIMUM DOMINATING CLIQUE remain NP-complete on cocomparability graphs.

Key words. domination, cocomparability graphs, graph algorithms

AMS subject classifications. 05C85, 68Q25, 68R10

1. Introduction. Many NP-complete graph problems become tractable when restricted to special subclasses of perfect graphs. For a particular problem, we are interested in graph classes that are as close as possible to the borderline between P and NP; i.e., there is no larger class for which a polynomial time algorithm is known. This motivates the search for larger graph classes for which the problem is still tractable.

Two well-known graph classes that admit many polynomial time algorithms for NP-complete graph problems are the interval and the permutation graphs. However, if we generalize both in a natural way to chordal and comparability graphs, respectively, things change, and many problems become NP-complete. This is especially the case for the well-known domination problem ([GT 2] of [18]) and its variants, total and connected domination. However, this generalization may be in the "wrong direction." Both classes of graphs have the additional property of being cocomparability, i.e., the complement of comparability graphs (see [19]). We intend to show that this is a generalization in a "good direction."

In this paper, we study the algorithmic complexity of domination and four variants, where the input is restricted to cocomparability graphs. The hope is to generalize the known polynomial time algorithms for interval and permutation graphs. It turns out that this is indeed possible in four cases; only clique domination becomes NP-complete.

Another encouragement for this work came from Colbourn and Lubiw, who studied the CARDINALITY STEINER TREE problem ([ND 12] of [18]) for cocomparability graphs using exactly the labeling that is our main tool. They were able to give a polynomial time algorithm and suggested the study of the somewhat similar connected domination problem.

Finally, another source for our work can be seen in the study of vertex orderings of graphs in [10]. The paper contains a complete study of the graph classes that arise by forbidding one or more ordered subgraphs on three vertices. There are four different classes of graphs resulting if we forbid exactly one ordered subgraph on three vertices (which is not one of the trivial cases—empty or complete), namely chordal, cochordal, comparability, and cocomparability graphs. The first three classes are well studied for many NP-complete graph problems. We extend this to the last class, which naturally contains all graph classes resulting from forbidden ordered subgraphs if one of them is

* Received by the editors October 2, 1989; accepted for publication (in revised form) August 20, 1992. This work was supported by the Natural Sciences and Engineering Research Council of Canada. This work was done while both authors were visiting the University of Toronto.

† Fakultät Mathematik, Friedrich-Schiller-Universität, Universitätschochhaus, O-6900 Jena, Germany.

‡ Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada, T6G 2H1.

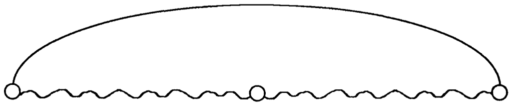


FIG. 1. *The forbidden ordered subgraph* (*in all figures, wavy lines indicate nonedges*).

the forbidden ordered subgraph for cocomparability graphs (see Fig. 1). Thus we consider this as a first attempt at a general approach to designing polynomial algorithms for NP-complete graph problems on graph classes that admit a certain labeling of the vertices.

The paper is organized as follows. Section 2 gives the necessary definitions and notation and summarizes known results. Then the following algorithms on cocomparability graphs resulting from a somewhat general dynamic programming approach are presented: $O(n^3)$ algorithm for constructing a minimum independent dominating set in § 3; $O(n^3)$ algorithm for constructing a minimum connected dominating set, which can always be assumed to be an induced path, in § 4; $O(n^6)$ algorithms for constructing a minimum dominating set and a minimum total dominating set, respectively, in §§ 5 and 6. Finally, in § 7 we show that finding a minimum dominating clique is NP-hard and that even the question of whether a cocomparability graph has a dominating clique is NP-complete.

2. Preliminaries. For the standard graph-theoretic notions not mentioned here, we refer the reader to [1], [19]. Throughout the paper, we consider finite, simple, undirected graphs. Let $G = (V, E)$ be a graph with vertex set V and edge set E; then n and m denote the cardinality of V and E, respectively. G_W, the subgraph of G induced by $W \subseteq V$, is the graph with vertex set W and exactly the edges from E that have both incident vertices in W. A path in a graph is a sequence $x_1 - x_2 - x_3 - \cdots - x_k$ of vertices such that x_i is adjacent to x_{i+1} for $i \in \{1, 2, \ldots, k-1\}$. A path on k vertices is called induced or simple if it has exactly $k-1$ edges, i.e., only the edges of the path itself. An induced path on k vertices is also called P_k.

A set $V' \subseteq V$ is called independent (a clique) if, for any pair of vertices $u, v \in V'$, it holds that $\{u, v\} \notin E (\{u, v\} \in E)$. A set $S \subseteq V$ is a dominating set if each vertex of $V - S$ is adjacent to some vertex of S. Usually, the following variants of dominating sets are also considered. In each case, an additional property of G_S is required. S is called an independent dominating set if S is both independent and a dominating set. A connected (respectively, total) dominating set S has the additional property that G_S is connected (respectively, has no isolated vertices). Finally, S is called a dominating clique if S is both a clique and a dominating set. For each of these kinds of dominating sets, the construction of a minimum (cardinality) set of this type is NP-hard, and the corresponding decision problem is NP-complete. Considerable work has been done in recent years to clarify the algorithmic complexity of these problems when restricted to special classes of graphs. For an overview, see [9]. We give the results important for our work and concentrate on perfect graphs.

The five variants are all solvable in polynomial time on interval [2], [4], [15], [16], [22], [24], [27] and permutation graphs [4], [5], [7], [9], [17]. Our aim is to extend the polynomial algorithms to a larger class of graphs containing both interval and permutation graphs, namely, the cocomparability graphs. On the other hand, the decision problems (i.e., given a graph and an integer k, decide whether G has a dominating set, or an independent, connected, or a total dominating set, of size at most k) are NP-complete for bipartite graphs [8], [14], [23] and hence for comparability graphs. The minimum dominating clique problem on comparability graphs is fairly easy, since every comparability graph having a dominating clique has one of size at most two [4]. All the problems, except independent domination [15], remain NP-complete for split graphs; this can be shown by using one reduction only [3], [8]. Thus all these problems remain NP-complete for chordal graphs; however, independent domination is tractable even for chordal graphs [15]. Naturally, the four basic problems CLIQUE, INDEPENDENT SET, CHROMATIC NUMBER, and PARTITION INTO CLIQUES are polynomial on perfect graphs. For cocomparability graphs, we can use the algorithms for comparability graphs [19] on the complement of the graph for the complementary problem; e.g., find a maximum clique for the cocomparability graph by constructing a maximum independent set on its complement.

We conclude with presenting the necessary knowledge on the graph classes considered. For any graph classes not defined here and for more details, refer to [19]. A graph is chordal if each cycle of length greater than three has a chord, i.e., an edge joining two nonconsecutive vertices of the cycle. A graph $G = (V, E)$ is a comparability graph if there is a transitive and antisymmetric orientation F of the edges of G; hence, for every $\{u, v\} \in E$, either $(u, v) \in F$ or $(v, u) \in F$, and

$$(u, v) \in F \wedge (v, w) \in F \rightarrow (u, w) \in F.$$

If $\mathcal{G}$ is a class of graphs, then we call the class of its complements co-$\mathcal{G} := \{\bar{G} : G \in \mathcal{G}\}$. We are interested in the classes of cochordal and cocomparability graphs. Interval graphs are the intersection graphs of intervals of the real line. They are exactly the graphs that are both chordal and cocomparability. Permutation graphs are defined as follows: Let π be a permutation from $\{1, 2, \ldots, n\}$. Then $G_\pi = (\{1, 2, \ldots, n\}, E_\pi)$ with $E_\pi = \{\{i, j\} : i, j \text{ in reversed order in } \pi\}$ is the inversion graph of π. G is a permutation graph if and only if there is a permutation π such that G is isomorphic to G_π. Permutation graphs are exactly those graphs that are both comparability and cocomparability graphs.

It is well known that chordal graphs have perfect elimination schemes of the vertices [19]. This labeling of the vertices is completely characterized by the fact that it does not contain the ordered subgraph $ch = (\{1, 2, 3\}, \{\{1, 2\}, \{1, 3\}\})$. For any comparability graph, we can get an ordering of the vertices by any topological sort with respect to the orientation F, corresponding exactly to a labeling not containing $cp = (\{1, 2, 3\}, \{\{1, 2\}, \{2, 3\}\})$. Such an ordering for a comparability graph G gives a labeling for $\bar{G}$ not containing $\overline{cp} = (\{1, 2, 3\}, \{\{1, 3\}\})$. Thus, cocomparability graphs can be characterized as those graphs having a labeling that does not contain $\overline{cp}$. An interesting direction of research is the characterization of other classes of graphs in terms of forbidden ordered subgraphs. This has been studied in [10].

The following technical definitions are used throughout the paper. First, we assume that the vertices of the cocomparability graph $G = (V, E)$ are ordered on a straight line according to a labeling without $\overline{cp}$ and numbered from left to right with $1, 2, \ldots, n$. Note that there is an $O(n^2)$ algorithm to construct this labeling for a given cocomparability graph G by constructing the transitive orientation for $\bar{G}$ [25] and then any topological sort gives a labeling of G with the desired property. We will often use the natural ordering on $\{1, 2, \ldots, n\}$ as forcing an ordering on V. Note that in our labeling the existence of an edge $\{i, j\} \in E$ implies for every vertex k with $i < k < j$ that $\{i, k\} \in E$ or $\{k, j\} \in E$, since $\overline{cp}$ is forbidden. (This proves to be the main reason that all our algorithms work.)

$[i, j] := \{k : i \leqq k \leqq j\}$ is a useful notation, and we will say that $\{i, j\} \in E$ covers the interval $[i, j]$, meaning that each vertex between i and j has at least one of i and j as a neighbour. This is generalized as follows: Let $S \subseteq V$, where G_S is connected. Then S

covers [min (S), max (S)]. We say that $S \subseteq V$ dominates $T \subseteq V$ if and only if each vertex $t \in T$ has at least one neighbour in S. Thus, for G_S connected, the interval [min (S), max (S)] is dominated by S.

3. Independent domination. We present an $O(n^3)$ algorithm that constructs a minimum independent dominating set of a given cocomparability graph. This is done, as for all other variants having polynomial time algorithms, by a dynamic programming approach using a linear scan through the labeling of the given graph. (As in all following sections, we assume that the vertices are labeled $1, 2, 3, \ldots, n$, according to the order in the labeling.)

The independent dominating set S is constructed by considering all possible cases of including in S and excluding from S, respectively, a certain vertex i. All the necessary information on the computation, for a certain $S \subseteq \{1, 2, 3, \ldots, i\}$ chosen up to the processing of i, is stored in states $Z \in \{0, 1, \ldots, n\}^3$. Thereby the components of $Z = [z_1, z_2, z_3]$ indicate

z_1 number of vertices in current S;

z_2 last vertex of the current S with respect to the labeling, i.e., max (S);

z_3 last processed vertex (current level of computation).

We describe the storage of the states in two arrays A and B, immediately after the algorithm.

```
ALGORITHM 1. Start with A containing only $[0, 0, 0]$, $s = |V|$.
For $i := 1$ to $n$ do
    For each state $Z = [z_1, z_2, i-1]$ in array $A$ do
        if $\{z_2, i\} \notin E$ and each $x$ with $z_2 < x < i$ is dominated by $z_2$ or $i$ then
            INCLUDE$(Z, i) = [z_1 + 1, i, i]$
            if $\{i, x\} \in E$ for each $x$ with $i < x$
            then $s := \min\{s, z_1 + 1\}$
            else Store INCLUDE$(Z, i)$ in the array $B$
        Store EXCLUDE$(Z, i) = [z_1, z_2, i]$ in $B$.
    $A := B$.
```

Storing in the array B is done in the following way. The index in the array is given by z_2. Then z_1 is stored if it is less than the present value of $B(z_2)$; otherwise, the value of $B(z_2)$ is not changed. It is not necessary to store the value of z_3; it is simply the number of the iteration.

THEOREM 3.1. INDEPENDENT DOMINATING SET *is solvable in time* $O(n^3)$ *for cocomparability graphs.*

Proof. In each iteration, at most n different states are stored, and each of these states gives at most two new states. All computation for a state is completed in $O(n)$ time. The $O(n^3)$ time bound follows.

It remains only to prove the correctness of the algorithm. Each state corresponds to at least one set $S \subseteq V$. We show that, for each state INCLUDE $(Z, i) = [z_1 + 1, i, i]$ with $\{i, x\} \in E$ for each x with $i < x$, all the corresponding sets S are independent dominating sets of size $z_1 + 1$.

Let S be any set corresponding to the state $Z = [z_1, z_2, i-1]$. If S is independent, then $S \cup \{i\}$ is independent, too, even though the algorithm checks only that $\{z_2, i\} \notin E$. This is because any vertex $a \in S$ with $a < z_2$ and $\{a, i\} \in E$ would create the forbidden ordered subgraph (see Fig. 1).

Furthermore, $S \cup \{i\}$ is a dominating set. For the first vertex f taken into S, the algorithm checks that all x with $x < f$ are dominated by f. In addition, the algorithm

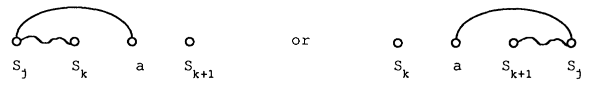


FIG. 2. *Situation in the labeling.*

checks that all vertices between two consecutive vertices of S are dominated by these two vertices. At the end, INCLUDE (Z, i) is under consideration for the minimum, since the last vertex i dominates all x with $x > i$.

Could there exist a smaller independent dominating set S^* in which vertices outside the set are not necessarily dominated by their "closest" neighbours in the set? The answer is "no." Assume that another vertex $s_j \in S^*$ not closest to $a \notin S^*$ is adjacent to a, where s_k and s_{k+1} are the elements of S^* that are closest to a (see Fig. 2). Then one of the vertices of S^* closest to a is also adjacent to a; i.e., it is enough to consider the region inside an interval $[s_k, s_{k+1}]$ and check whether the two vertices s_k and s_{k+1} dominate $[s_k, s_{k+1}]$.

Finally, the algorithm checks all possibilities avoiding only bad states; i.e., if $[z_1, z_2, z_3]$ and $[z'_1, z_2, z_3]$ are both states and $z'_1 > z_1$ then the algorithm need only extend the smaller set, i.e., only $[z_1, z_2, z_3]$ is stored.

Therefore the value of s after the termination of the algorithm is really the minimum size of an independent dominating set in the cocomparability graph G. □

Remark 3.2. Using the same states but storing all the states in a matrix of type $[0 \cdots n, 0 \cdots n]$ and again storing only the minimal z_1 at the entry $[z_2, z_3]$ and using pointers from a state back to the one that was its origin, we can in the same time bound construct a minimum independent dominating set of G. Furthermore, it is possible to solve the problem even for arbitrary real vertex weights with the same algorithm and within the same time bound, if z_1 stores now the sum of the weights for all vertices of S.

4. Connected domination. In this section, we present an algorithm for connected domination. We first describe the minimum cardinality Steiner tree algorithm of [6], which serves as an introduction to the connected dominating set algorithm. The two problems are closely related on many classes of perfect graphs (see [9], [21], [27]).

The minimum cardinality Steiner tree problem is as follows. Given a graph $G = (V, E)$ and a subset $T \subseteq V$ of target points, find a subset $U \subseteq V - T$ of minimum cardinality such that the subgraph of G induced by $U \cup T$ is connected. We consider this problem when the given graph G is a cocomparability graph with an associated cocomparability ordering.

Let l and r be the leftmost and rightmost vertices of T in the cocomparability ordering for G. Any connected subgraph S of G containing both l and r necessarily dominates all vertices between l and r, including all vertices of T. This is because every vertex x between l and r is either in S or is between two adjacent vertices of S. Thus, either $x \in S$ or the ordering implies that x is dominated by S.

On the other hand, any Steiner tree containing all vertices of T is a connected subgraph containing l and r. Thus, a minimum cardinality Steiner tree is exactly a connected subgraph of G containing both l and r, having the minimum number of vertices of $V - T$. This is nothing but a minimum weight path between l and r where vertices of T have weight zero and vertices of $V - T$ have weight one.

Therefore, to find a minimum cardinality Steiner tree in a cocomparability graph G for a particular set of target vertices T, we can produce a cocomparability ordering, place appropriate weights on the vertices, and then make use of a shortest path algorithm. Because of the small integer weights, such a shortest path algorithm could be implemented to run in $O(m + n)$ time, as mentioned in [26].

Initially, it appears that a similar approach cannot be used to solve the connected dominating set problem, since a connected dominating set is not necessarily a shortest path between a particular pair of vertices. For example, in the graph of Fig. 3, $\{2, 3, 4\}$ is a minimum connected dominating set that is not a shortest path. Note, however, that there is a shortest path between 2 and 5 that is a minimum connected dominating set. In a shortest path approach to this problem, all vertices would have weight one; any shortest path in this situation is an induced simple path in the graph. We now show that, in a cocomparability graph, there always exists a minimum connected dominating set that induces a simple path.

We make use of the following definition. For any graph $G = (V, E)$ and set $U \subseteq V$, a private neighbour of $u \in U$ with respect to U is a vertex $v \notin U$ such that $\{u, v\} \in E$ and for all $u' \in U - \{u\}$, $\{u', v\} \notin E$.

LEMMA 4.1 (Key Lemma). *For a connected cocomparability graph G, there exists a minimum cardinality connected dominating set that induces a simple path in G.*

Proof. Let G be a cocomparability graph that is ordered according to a cocomparability labeling. Let S be a minimum cardinality connected dominating set for G, and suppose that S does not induce a simple path.

If min $(S) = 1$ and max $(S) = n$, then let S' be the vertices of a shortest path between 1 and n in G. S' is a connected dominating set that induces a simple path in G. In addition, $|S'| \leqq |S|$, since S is connected.

Suppose that min $(S) \neq 1$ and max $(S) = n$. Let S' be the vertices of a shortest path between 1 and n in $S \cup \{1\}$. S' is clearly a connected dominating set that induces a path. Furthermore, $|S'| \leqq |S|$, since S does not induce a path. A similar result is obtained when min $(S) = 1$ and max $(S) \neq n$.

Suppose that min $(S) \neq 1$ and max $(S) \neq n$. Let S^* be a shortest path between 1 and n in $S \cup \{1, n\}$. If $|S - S^*| = 2$, then S^* is a minimum connected dominating set for G that induces a path. If $|S - S^*| = 0$, then S is a minimum connected dominating set for G that induces a path. If $|S - S^*| = 1$, then there is exactly one vertex $x \in S$ that is not on the shortest path. Therefore, $S - \{x\}$ must induce a path. The vertex x must have a private neighbour with respect to S, for, otherwise, $S - \{x\}$ would be a smaller connected dominating set, a contradiction. In the ordering, any such private neighbour y must be such that $y <$ min $(S - \{x\})$ or $y >$ max $(S - \{x\})$. Otherwise, y would be adjacent to some other vertex of S.

If x has two private neighbours, $y <$ min $(S - \{x\})$ and $z >$ max $(S - \{x\})$, then $S' = \{x, y, z\}$ is a minimum connected dominating set for G. It is clear that all vertices between min (S') and max (S') are dominated by S'. Consider a vertex w outside this range. w must be adjacent to x or to some vertex of $S - \{x\}$. If $\{w, x\} \notin E$, then any edge from w to a vertex of $S - \{x\}$ covers y or z, and, since neither y nor z can be adjacent to any vertex of $S - \{x\}$, this forces the edge $\{w, y\}$ or $\{w, z\}$. We know that $\{y, z\} \notin E$, since this would force each vertex of S to be adjacent to y or z, contradicting the fact that y and z are private neighbours of x. In addition, since S does not induce a

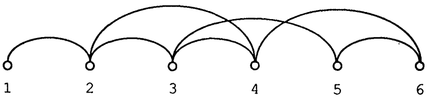


FIG. 3

simple path, it must contain a subgraph isomorphic to $K_{1,3}$ or a cycle, implying that $|S| \geqq 3$. Therefore, S' is a minimum connected dominating set for G that induces a path.

Finally, suppose that every private neighbour y of x has $y < \min(S - \{x\})$, or every private neighbour y of x has $y > \max(S - \{x\})$. Suppose that every private neighbour y of x is such that $y < \min(S - \{x\})$ and consider $S' = (S - \{x\}) \cup \{1\}$. Note that $S' = S^* - \{n\}$. S' induces a path that clearly dominates every vertex to the left of $\max(S') = \max(S)$. Any vertex to the right of $\max(S)$ is not a private neighbour of x; therefore all such vertices are dominated by S'. A similar argument holds when all private neighbours of x are to the right of $S - \{x\}$. □

From the lemma, we see that a minimum connected dominating set can be found using an approach similar to the Steiner tree algorithm. We must only compute all shortest paths between all pairs of vertices; a minimum cardinality such path that dominates all vertices is then a minimum connected dominating set. For each shortest path, we can check in polynomial time whether it dominates G, but unfortunately the number of shortest paths can be exponential in n. We can improve upon this by noting the following.

LEMMA 4.2. *Let $P = \{p_1, p_2, \dots, p_k\}$, $k \geqq 3$ be an induced path in a cocomparability graph G with $p_k > p_1$ in the cocomparability ordering. Then, for all $1 \leqq i \leqq k-2$, $p_{i+2} > p_i$.*

Proof. Let i be the smallest index for which $p_{i+2} < p_i$. Suppose that $i = 1$. Since $p_k > p_1$, there must be an edge $\{p_j, p_{j+1}\}$, $j \geqq 3$, which covers p_1, creating an edge $\{p_1, p_j\}$ or $\{p_1, p_{j+1}\}$, contradicting the fact that P is an induced path. Suppose that $i > 1$. If $p_1 < p_{i+2}$, then there must be some edge $\{p_j, p_{j+1}\}$, $j < i$, which covers p_{i+2}, forcing an edge $\{p_{i+2}, p_j\}$ or $\{p_{i+2}, p_{j+1}\}$. If $p_1 > p_{i+2}$, then there must be an edge $\{p_j, p_{j+1}\}$, $j \geqq i+2$, which covers p_1, forcing an edge $\{p_1, p_j\}$ or $\{p_1, p_{j+1}\}$. Both cases contradict the fact that P is an induced path. □

This lemma tells us that an induced path can always be traversed from left to right as follows. Each step is either a forward step, i.e., the next vertex is further to the right than any previous vertex, or a backward step, i.e., the next vertex is to the left of the previous vertex but to the right of all other vertices. There cannot be two consecutive backward steps.

LEMMA 4.3. *If $S \subseteq V$ is a minimum connected dominating set for a cocomparability graph $G = (V, E)$ with cocomparability ordering, and if S induces the simple path $\{p_1, p_2, \dots, p_k\}$, then every vertex $x <$ min (S) is dominated by $\{p_1, p_2\}$, and every vertex $y >$ max (S) is dominated by $\{p_{k-1}, p_k\}$.*

Proof. We prove the lemma for vertices $x < \min(S)$. The remainder follows by a similar argument. One of $\{p_1, p_2\}$ must be $\min(S)$. This follows from the previous lemma. Any edge connecting a vertex $x < \min(S)$ and a vertex $z \in S - \{p_1, p_2\}$ covers p_1, implying that $\{x, p_1\} \in E$. □

Now, for each edge of G, we can check whether the endpoints dominate all vertices to the left or all vertices to the right. If so, then that edge is a candidate for the first or last edge in the dominating path. Now we need only calculate the shortest paths between these pairs of edges. We describe how this is done in the following algorithm. When the algorithm stops, s is the minimum cardinality of a connected dominating set of G.

ALGORITHM 2.

```
$s := n$
**If** *any single vertex dominates* $V$ **then** $s := 1$; **stop**
**For** *each edge* $\{u, v\} \in E$
    **if** $\{u, v\}$ *dominates* $[1, \min\{u, v\}]$ $([\max\{u, v\}, n])$
```

```
        then the vertices $u$ and $v$, and the edge $\{u, v\}$, are labeled $L$ $(R)$
        if $\{u, v\}$ is labeled both L and R
        then $s := 2$; stop
        if either $u$ or $v$ is labeled both L and R
        then $s := 3$
    if $s = 3$ then stop
    $D :=$ the distance matrix for $G$
    For each vertex $u$ with label $L$:
        $d :=$ the minimum distance from $u$ to any vertex $v$ with label $R$
        $s := \min\{s, d + 3\}$.
```

THEOREM 4.4. CONNECTED DOMINATING SET *is solvable in time $O(n^3)$ for cocomparability graphs.*

Proof. Let s^* be the minimum cardinality of a connected dominating set for G. It is clear that, if $s^* = 1$ or $s^* = 2$, then s is correctly calculated. If $s^* = 3$, then it must be that two edges, which share a vertex, dominate all vertices of G. s is correctly calculated in this case, also. If $s^* \geqq 4$, then no vertex or edge will be labeled both L and R by the algorithm. Let $\mathbf{L} = \{u \in V: u \text{ is marked } L\}$ and $\mathbf{R} = \{u \in V: u \text{ is marked } R\}$.

Suppose that s is incorrectly calculated by the algorithm. Suppose that $4 \leqq s < s^*$. From the algorithm, we see there must exist a path from a vertex $u \in \mathbf{L}$ and a vertex $v \in \mathbf{R}$ of length $s - 3$, i.e., with $s - 2$ vertices. This path, together with any neighbour of u in $G_{\mathbf{L}}$ and any neighbour of v in $G_{\mathbf{R}}$ is a connected dominating set, by our earlier lemmas.

Suppose that $s > s^* \geqq 4$. From our earlier lemmas, we have seen that there must exist a minimum connected dominating set that induces a simple path beginning with an edge that dominates all to the left and ending with an edge that dominates all to the right. Such a path surely begins with a vertex of $\mathbf{L}$ and ends with a vertex of $\mathbf{R}$. The algorithm examines all such shortest paths and hence will certainly find s correctly.

The labeling of the vertices and edges can be done in $O(mn)$ time, and an all-pairs shortest path algorithm requires at most $O(n^3)$ time (see [26]). The remaining operations can be done in $O(n^2)$ time. This gives an overall time complexity of $O(n^3)$. □

We note that slight modifications to the dominating set algorithm of § 5 would yield an $O(n^6)$ dynamic programming algorithm for connected domination. The details are mentioned in § 6.

5. Domination. We show that our techniques can be extended to the well-known domination problem itself.

LEMMA 5.1. *Every cocomparability graph has a dominating set S such that each connected component of G_S is either an induced path on at least two vertices or an isolated vertex.*

Proof. This is a consequence of the key lemma in the previous section. If one of the connected components of G_S, say C, is not an induced path, then there exists an induced path C' such that $N[C'] \supseteq N[C]$. Hence $S' = (S - C) \cup C'$ is a dominating set, also. Eventually, S' may have a smaller number of components than S, since C' and a few components of $C - S$ may be connected in $G_{S'}$. If such a connected component is not an induced path, we apply the lemma again. We can obviously proceed with this procedure until we get a dominating set in which all connected components are induced paths of length at least two or isolated vertices in G_S. □

First, we note that different components of a dominating set cannot overlap in the labeling, since otherwise one would cover a vertex of the other and therefore we would have an edge between the components. This problem is different from independent domination in that we must consider a much greater number of cases and more of the vertices of a subsolution S corresponding to a state $Z \in \{0, 1, 2, \ldots, n\}^5$. The components indicate

z_1 number of vertices in current S;
z_2 third last vertex of S (in the order of the path or the components);
z_3 second last vertex of S;
z_4 last vertex of S;
z_5 last processed vertex.

Note that z_2, z_3, and z_4 are not always in this order in the labeling; exceptions occur in an induced path with backward steps. We first consider the cases in which we include a vertex i in S with $z_2 \neq 0$, i.e., $|S| \geqq 3$, for the state $Z = [z_1, z_2, z_3, z_4, i-1]$. For each configuration of z_2, z_3, and z_4, we will have three or four different possible ways of including i. We give all possible new states for INCLUDE (Z, i), defining the main rules for our algorithm. The word(s) connected and/or total and/or independent in parentheses indicate(s) that this inclusion applies to connected and/or total and/or independent domination, respectively, also. The diagrams in Fig. 4 correspond to similarly numbered cases.

1. $\{z_2, z_3\} \in E$, $\{z_3, z_4\} \in E$, $z_2 < z_3$. The last component of S is a path of at least three vertices with no backward step among them.
   (a) $\{z_2, i\} \notin E$, $\{z_3, i\} \notin E$, $\{z_4, i\} \in E$. We can include i as the next vertex of the path: INCLUDE $(Z, i) = [z_1 + 1, z_3, z_4, i, i]$ (total, connected);
   (b) $\{z_3, i\} \notin E$, $\{z_4, i\} \notin E$, ($\{z_2, i\} \in E$ would imply that $\{z_4, i\} \in E$). For every j with $z_3 < j < z_4$, $\{j, z_4\} \in E$, $\{j, i\} \in E$, $\{j, z_3\} \notin E$, $\{j, z_2\} \notin E$, we do a backward step $z_4 - j$ and a forward one $j - i$ adding two vertices to the path, namely, j and i. Thus we remain with the following last vertices of the path in the last component of S: $z_2 - z_3 - z_4 - j - i$: INCLUDE $(Z, i) \ni [z_1 + 2, z_4, j, i, i]$ for every j satisfying the conditions (total, connected);
   (c) $\{z_3, i\} \notin E$, $\{z_4, i\} \notin E$. We can always create a new component of S: INCLUDE $(Z, i) = [z_1 + 1, z_3, z_4, i, i]$ (total);
   (d) $\{z_3, i\} \notin E$, $\{z_4, i\} \notin E$. For every j with $z_3 < j < z_4$, $\{j, z_4\} \in E$, $\{j, i\} \notin E$, $\{j, z_3\} \notin E$, $\{j, z_2\} \notin E$, we do a backward step $z_4 - j$, adding one vertex to the path containing z_4. We then add vertex i, beginning a new component. INCLUDE $(Z, i) \ni [z_1 + 2, z_4, j, i, i]$ for every j satisfying the conditions (total).

In all cases, we consider configurations (a)–(c), below; in some cases, we must also consider configuration (d):

(a) include i in the component containing z_4, with a forward step;

(b) include a vertex j and the vertex i in the component containing z_4, with a backward step followed by a forward one;

(c) include i creating a new component, i.e., i has no neighbours in the last component;

(d) end the component containing z_4 with a backward step to a vertex j and then begin a new component with i.

We henceforth simply list the states in INCLUDE (Z, i) for each of these three or four possibilities.

2. $\{z_2, z_3\} \in E$, $\{z_3, z_4\} \in E$, $z_3 < z_2$.
   (a) INCLUDE $(Z, i) = [z_1 + 1, z_3, z_4, i, i]$ if $\{z_4, i\} \in E$, $\{z_3, i\} \notin E$, $\{z_2, i\} \notin E$ (total, connected);
   (b) INCLUDE $(Z, i) = \{[z_1 + 2, z_4, j, i, i]$: $z_2 < j < z_4$, $\{i, j\} \in E$, $\{j, z_4\} \in E$, $\{j, z_3\} \notin E$, $\{j, z_2\} \notin E\}$ if $\{z_4, i\} \notin E$, $\{z_3, i\} \notin E$ (total, connected);

(c) INCLUDE $(Z, i) = [z_1 + 1, z_3, z_4, i, i]$ if $\{z_3, i\} \notin E$, $\{z_4, i\} \notin E$ (total);

(d) INCLUDE $(Z, i) = \{[z_1 + 2, z_4, j, i, i]$: $z_2 < j < z_4$, $\{i, j\} \notin E$, $\{j, z_4\} \in E$, $\{j, z_3\} \notin E$, $\{j, z_2\} \notin E\}$ if $\{z_4, i\} \notin E$, $\{z_3, i\} \notin E$ (total).

3. $\{z_2, z_3\} \in E$, $\{z_3, z_4\} \notin E$.

(a) INCLUDE $(Z, i) = [z_1 + 1, z_3, z_4, i, i]$ if $\{z_4, i\} \in E$, $\{z_3, i\} \notin E$, $\{z_2, i\} \notin E$, and $\{z_2, z_3, z_4, i\}$ dominates $[\max \{z_2, z_3\}, z_4]$ (total);

(b) INCLUDE $(Z, i) = \{[z_1 + 2, z_4, j, i, i]$: $\max \{z_2, z_3\} < j < z_4$, $\{j, i\} \in E$, $\{j, z_4\} \in E$, $\{j, z_3\} \notin E$, $\{j, z_2\} \notin E$, $\{z_2, z_3, z_4, j\}$ dominates $[\max \{z_2, z_3\}, j]\}$ if $\{z_4, i\} \notin E$ (total);

(c) INCLUDE $(Z, i) = [z_1 + 1, z_3, z_4, i, i]$ if $\{z_4, i\} \notin E$ and $\{z_2, z_3, z_4\}$ dominates $[\max \{z_2, z_3\}, z_4]$;

(d) not applicable because the vertices of a P_2 component should appear in the same order in $\{z_2, z_3, z_4\}$ and in the cocomparability ordering.

4. $\{z_2, z_3\} \notin E$, $\{z_3, z_4\} \in E$.

(a) INCLUDE $(Z, i) = [z_1 + 1, z_3, z_4, i, i]$ if $\{z_4, i\} \in E$, $\{z_3, i\} \notin E$ (total);

(b) INCLUDE $(Z, i) = \{[z_1 + 2, z_4, j, i, i]$: $z_3 < j < z_4$, $\{j, i\} \in E$, $\{j, z_4\} \in E$, $\{j, z_3\} \notin E\}$ if $\{z_3, i\} \notin E$, $\{z_4, i\} \notin E$ (total);

(c) INCLUDE $(Z, i) = [z_1 + 1, z_3, z_4, i, i]$ if $\{z_4, i\} \notin E$, $\{z_3, i\} \notin E$ (total);

(d) INCLUDE $(Z, i) = \{[z_1 + 2, z_4, j, i, i]$: $z_3 < j < z_4$, $\{j, i\} \notin E$, $\{j, z_4\} \in E$, $\{j, z_3\} \notin E\}$ if $\{z_3, i\} \notin E$, $\{z_4, i\} \notin E$ (total).

5. $\{z_2, z_3\} \notin E$, $\{z_3, z_4\} \notin E$.

(a) INCLUDE $(Z, i) = [z_1 + 1, z_3, z_4, i, i]$ if $\{z_4, i\} \in E$, $\{z_3, i\} \notin E$ and $\{z_3, z_4, i\}$ dominates $[z_3, z_4]$;

(b) INCLUDE $(Z, i) = \{[z_1 + 2, z_4, j, i, i]$: $z_3 < j < z_4$, $\{j, i\} \in E$, $\{j, z_4\} \in E$, $\{j, z_3\} \notin E$, $\{z_3, z_4, j\}$ dominates $[z_3, j]\}$ if $\{z_4, i\} \notin E$;

(c) INCLUDE $(Z, i) = [z_1 + 1, z_3, z_4, i, i]$ if $\{z_4, i\} \notin E$, and $\{z_3, z_4\}$ dominates $[z_3, z_4]\}$ (independent);

(d) not applicable.

(Note that cases 5(c), 6, 7(c), and 9(c) are the only ones applying to independent domination, indicating that this dynamic programming approach would work for independent domination as well.)

Before considering the correctness, we give the corresponding rules for $|S| < 3$.

6. $z_4 = 0$. INCLUDE $(Z, i) = [1, 0, 0, i, i]$; any vertex may begin a dominating set (connected, total, independent).

7. $z_3 = 0$, $z_4 \neq 0$.

(a) INCLUDE $(Z, i) = [2, 0, z_4, i, i]$ if $\{z_4, i\}$ dominates $[1, z_4]$ and $\{z_4, i\} \in E$ (total, connected);

(b) INCLUDE $(Z, i) = \{[3, z_4, j, i, i]$: $j < z_4$, $\{j, z_4\} \in E$, $\{j, i\} \in E$, $\{z_4, j\}$ dominates $[1, j]\}$ if $\{z_4, i\} \notin E$ (total, connected);

(c) INCLUDE $(Z, i) = [2, 0, z_4, i, i]$ if $\{z_4, i\} \notin E$, and z_4 dominates $[1, z_4]$ (independent);

(d) not applicable.

8. $z_2 = 0$, $z_3 \neq 0$, $\{z_3, z_4\} \in E$.

(a) INCLUDE $(Z, i) = [z_1 + 1, z_3, z_4, i, i]$ if $\{i, z_4\} \in E$, $\{i, z_3\} \notin E$ (total, connected);

(b) INCLUDE $(Z, i) = \{[z_1 + 2, z_4, j, i, i]$: $z_3 < j < z_4$, $\{j, i\} \in E$, $\{j, z_4\} \in E$, $\{j, z_3\} \notin E\}$ if $\{z_4, i\} \notin E$, $\{z_3, i\} \notin E$ (total, connected);

(c) INCLUDE $(Z, i) = [z_1 + 1, z_3, z_4, i, i]$ if $\{z_4, i\} \notin E$, $\{z_3, i\} \notin E$ (total);

(d) INCLUDE $(Z, i) = \{[z_1 + 2, z_4, j, i, i]$: $z_3 < j < z_4$, $\{j, i\} \notin E$, $\{j, z_4\} \in E$, $\{j, z_3\} \notin E\}$ if $\{z_4, i\} \notin E$, $\{z_3, i\} \notin E$ (total).

9. $z_2 = 0$, $z_3 \neq 0$, $\{z_3, z_4\} \notin E$.
   (a) INCLUDE $(Z, i) = [z_1 + 1, z_3, z_4, i, i]$ if $\{z_4, i\} \in E$, $\{z_3, i\} \notin E$ and $\{z_3, z_4, i\}$ dominates $[z_3, z_4]$;
   (b) INCLUDE $(Z, i) = \{[z_1 + 2, z_4, j, i, i]: z_3 < j < z_4, \{j, i\} \in E, \{j, z_4\} \in E, \{j, z_3\} \notin E, \{z_3, z_4, j\}$ dominates $[z_3, j]\}$ if $\{z_4, i\} \notin E$;
   (c) INCLUDE$(Z, i) = [z_1 + 1, z_3, z_4, i, i]$ if $\{z_4, i\} \notin E$, and $\{z_3, z_4\}$ dominates $[z_3, z_4]$ (independent);
   (d) not applicable.

The exclusion of i is always possible and gives EXCLUDE $(Z, i) = [z_1, z_2, z_3, z_4, i]$.

It is not difficult to check that, for a new state $Z' = [z'_1, z'_2, z'_3, z'_4, z'_5]$, any corresponding set S' dominates $[1, z'_3]$; i.e., the vertices in the labeling are dominated up to the second last vertex of S'. The algorithm ensures that, for $\{z'_2, z'_3\} \notin E$, this interval is dominated by two, three, or four vertices of S'. The following lemma shows that we can indeed restrict our attention to at most four vertices.

LEMMA 5.2. *Let $S \subseteq V$ such that each connected component of G_S is either an induced path on at least two vertices or an isolated vertex. Let $s, t \in S$ be two consecutive vertices of S in the labeling contained in two different components of G_S. Then S dominates $[s, t]$ if and only if one of the following is true*:

(i) *$\{s, t\}$ dominates $[s, t]$ for s and t isolated vertices in G_S,*

(ii) *$\{s', s, t\}$ dominates $[s, t]$ for t isolated and s with neighbour s' in a path of G_S,*

(iii) *$\{s', s, t', t\}$ dominates $[s, t]$ for s with neighbour s' in a path of G_S and t with neighbour t' in a path of G_S.*

Proof. One direction is obvious. We prove that we need not check all vertices of S. The main fact is that any vertex $a \in [s, t]$ that is dominated by $v \in S$, $v \notin \{s', s, t', t\}$ is also dominated by $\{s, t\}$, since the edge $\{a, v\}$ must cover one of $\{s, t\}$, forcing $\{a, s\} \in E$ or $\{a, t\} \in E$ by the cocomparability ordering. □

In a similar way, also shown for independent and connected domination, we get the following lemma for the domination of vertices outside min (S) and max (S).

LEMMA 5.3. *Let S be a dominating set such that the components of G_S are induced paths on at least two vertices or isolated vertices. Let $s =$ min (S) and $t =$ max (S) and let s' and t' be adjacent to s and t, respectively, if such vertices exist. Then S dominates $[1, s]([t, n])$ if and only if one of the following holds*:

(i) *$\{s\}(\{t\})$ dominates $[1, s]([t, n])$ if $s(t)$ isolated,*

(ii) *$\{s, s'\}(\{t, t'\})$ dominates $[1, s]([t, n])$ if $s(t)$ belongs to an induced path in G_S.*

These lemmas verify the correctness of all the conditions made for defining INCLUDE (Z, i) in all the different cases. (Note that the termination is checked inside the algorithm.)

Now we are prepared to present the dynamic programming algorithm for the general case, noting that the general nonstarting computation applies if $z_2 \neq 0$. We emphasize that we do not repeat the conditions that must be satisfied such that INCLUDE (Z, i) is nonempty. Also, the organization of the arrays is similar to Algorithm 1, except that we now need arrays of type $[0 \cdots n, 0 \cdots n, 0 \cdots n]$, and the address for storing the component z_1 of Z, if z_1 is less than the previous stored value, is given by $[z_2, z_3, z_4]$.

```
ALGORITHM 3. Start with array A containing only [0, 0, 0, 0, 0], s = |V|.
    For i := 1 to n do
        For each state Z = [z_1, z_2, z_3, z_4, i − 1] in array A do
            Compute INCLUDE(Z, i).
        For each state $Z' = [z'_1, z'_2, z'_3, z'_4, i] \in$ INCLUDE$(Z, i)$ do
            If $z'_2$ and $z'_3$ form a backward step, i.e., $z'_3 < z'_2$,
            and $\{z'_2, z'_3\}$ dominates $[z'_2, n]$
            then $s := \min\{s, z'_1 - 1\}$
            else
                If either the new second last and last vertices, $z'_3$ and
                $z'_4 = i$, are last vertices of a path and dominate $[i, n]$;
                or the last vertex $z'_4 = i$ is isolated and dominates $[i, n]$
                and $\{z'_2, z'_3, z'_4 = i\}$ dominates $[\max\{z'_2, z'_3\}, i]$
                then $s := \min\{s, z'_1\}$
                else Store $Z'$ in the array $B$
        Store EXCLUDE$(Z, i)$ in the array $B$
    $A := B$.
```

THEOREM 5.4. DOMINATING SET *is solvable in time* $O(n^6)$ *for cocomparability graphs.*

Proof. The time bound is implied by the fact that, in each of the n iterations, at most $O(n^3)$ different states are stored in the array B. We can check each condition for INCLUDE (Z, i) in time $O(n)$ and we get at most n new states Z' from any Z in array A.

Let S be a minimum dominating set of the cocomparability graph G. By Lemma 5.1, we can assume that the connected components of S are induced paths on at least two vertices or isolated vertices in G_S. By Lemma 4.2, we know that any induced path cannot have two consecutive backward steps. Therefore INCLUDE (Z, i) checks all cases. If the vertices of S have the cocomparability ordering $s_1, s_2, \dots, s_r$, then, by Lemma 5.2, any interval $[s_j, s_{j+1}]$ not covered by any component of G_S is dominated by the vertices s_j, s_{j+1} and possibly the neighbours of s_j and s_{j+1} in G_S. Furthermore, by Lemma 5.3, $[1, s_1]$ and $[s_r, n]$ are dominated by s_1 or $\{s_1, s_2\} \in E$, and s_r or $\{s_{r-1}, s_r\} \in E$, respectively. Therefore the algorithm would produce a sequence of states corresponding to the inclusion of exactly the vertices in S. Finally, after inclusion of s_n, $z'_1 = |S|$ would be considered for $s := \min\{z'_1, s\}$, with the result that $s \leqq |S|$.

The algorithm does not delete any state that could lead to S. Let Z be the state corresponding to vertices of S in the interval $[1, i]$ and suppose that the algorithm deletes Z, replacing it with Z'. This implies that $z_1 > z'_1$. Then, however, we would get a dominating set S' with $|S'| < |S|$ if we proceed on Z' exactly as we would have done on Z to get S. This is a result of the fact that only the last three vertices of a set are important for the computation. □

6. Total domination. We show that this variant of the domination problem also becomes polynomial time solvable when restricted to cocomparability graphs. We use the results of the previous section. We also show how to use the algorithm for the connected case. The key Lemma 4.1 gives us the following lemma.

LEMMA 6.1. *For a cocomparability graph G with no isolated vertices, there exists a minimum total dominating set S such that each connected component of G_S is an induced path on at least two vertices.*

The proof is similar to the proof of Lemma 5.1 and is therefore omitted.

We would be in enormous trouble if we had to check all possible starting edges for the path for each new component of S, as in the connected domination algorithm. Therefore, we cannot use the shortest path approach of § 4. Fortunately, the dynamic programming approach of § 5 can be modified to work for total domination.

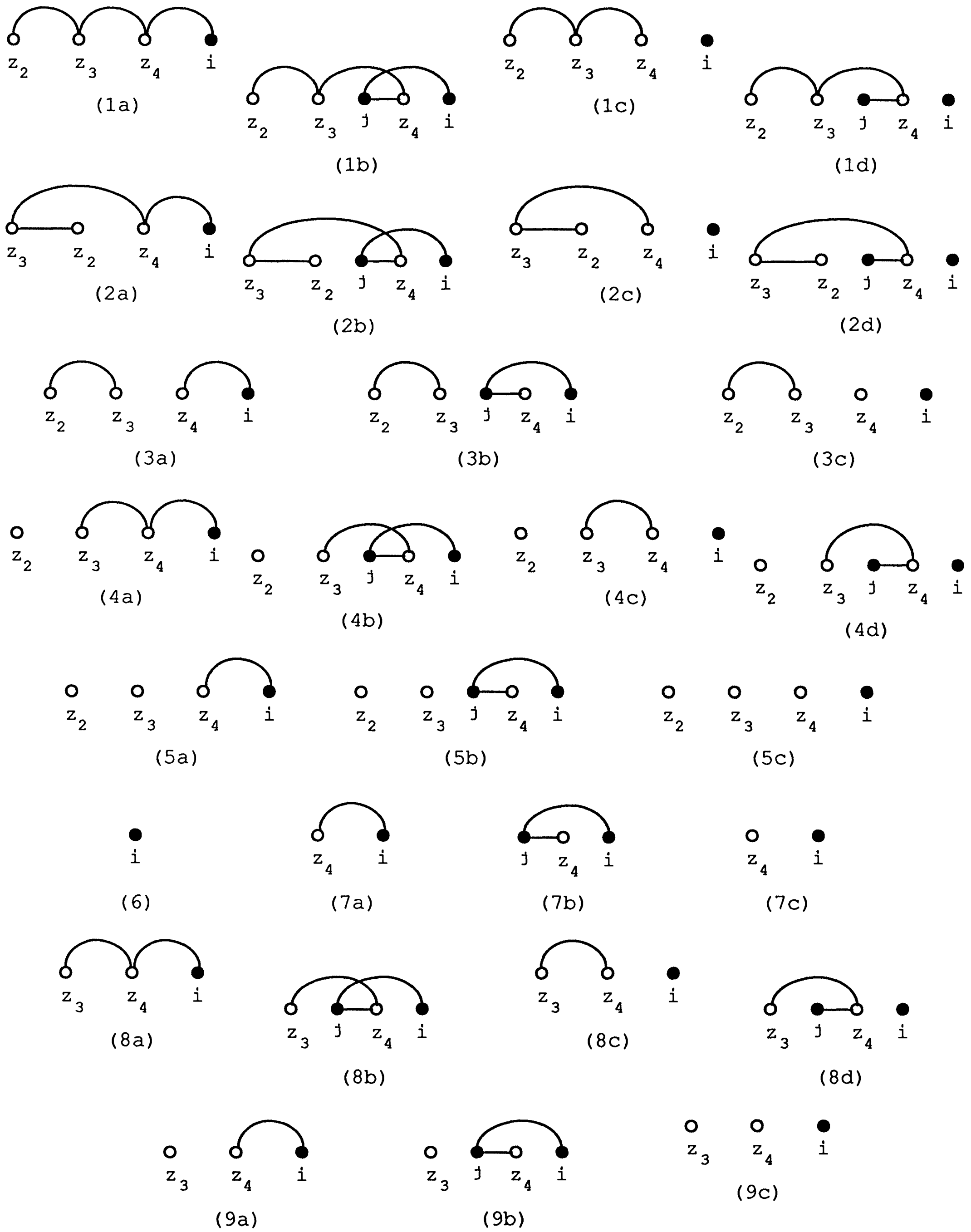


FIG. 4. *All cases for including vertex i.*

During the construction of the total dominating set S by a linear scan through the labeling and dynamic programming, whenever the considered state corresponds to a set S in which the last component has at least two vertices, we are allowed to start a new component in S if the current vertex i is not adjacent to the last and the second last vertex in the induced path of the previous component. This is the main difference from the domination problem, where we can always create a new component by including i. For the termination, we must make sure that we do not stop with a single vertex as last component of S.

These are the only differences from the domination problem itself; thus we get an algorithm for total domination working in exactly the same manner and with the same states as Algorithm 3. We have only to avoid a few INCLUDE (Z, i)'s, which would definitely produce an isolated vertex in S. To assist the reader, we enumerate the rules from the previous section that apply to total domination. We call the rules $\text{INCLUDE}_{\text{total}}$ to indicate the difference from INCLUDE:

$$\text{INCLUDE}_{\text{total}} = \{1(a)\text{–}1(d), 2(a)\text{–}2(d), 3(a), 3(b), 4(a)\text{–}4(d), 6, 7(a), 7(b), 8(a)\text{–}8(d)\}.$$

We obviously need not change EXCLUDE. This leads to the following algorithm for total domination.

```
ALGORITHM 4. Start with array A containing only $[0, 0, 0, 0, 0]$, $s = |V|$.
    For $i := 1$ to $n$ do
        For each state $Z = [z_1, z_2, z_3, z_4, i-1]$ in array A do
            Compute $\text{INCLUDE}_{\text{total}}(Z, i)$.
            For each state $Z' = [z'_1, z'_2, z'_3, z'_4, i] \in \text{INCLUDE}_{\text{total}}(Z, i)$ do
                If $z'_2$ and $z'_3$ form a backward step, i.e., $z'_3 < z'_2$,
                and $\{z'_2, z'_3\}$ dominates $[z'_2, n]$
                then $s := \min\{s, z'_1 - 1\}$
                else
                    If the new second last and last vertices $z'_3$ and $z'_4 = i$,
                    are last vertices of a path and dominate $[i, n]$
                    then $s := \min\{s, z'_1\}$
                    else Store $Z'$ in the array B
            Store EXCLUDE$(Z, i)$ in the array B
        $A := B$.
```

THEOREM 6.2. TOTAL DOMINATING SET *is solvable in time $O(n^6)$ for cocomparability graphs.*

The proof is similar to the proof of Theorem 5.4 and is therefore omitted.

Remark 6.3. Using a corresponding $\text{INCLUDE}_{\text{conn}} = \{1(a), 1(b), 2(a), 2(b), 6, 7(a), 7(b), 8(a), 8(b)\}$ instead of $\text{INCLUDE}_{\text{total}}$ in Algorithm 4 yields a similar $O(n^6)$ algorithm for CONNECTED DOMINATING SET on cocomparability graphs.

7. Clique domination.

In this section, we show that finding dominating cliques is NP-hard for cocomparability graphs, although the problem is trivial on interval and comparability graphs. (For both interval graphs and comparability graphs, if the graph has a dominating clique, then it has a dominating clique of size at most two.)

THEOREM 7.1. MINIMUM DOMINATING CLIQUE (*i.e., given graph G and integer k, does G have a dominating clique of size $\leqq k$?*) *is* NP-*complete on cocomparability graphs* (*in fact, on cobipartite graphs with one pendant vertex*).

Proof. We transform HITTING SET $= \{(M, \mathcal{M}, k)\colon \mathcal{M} \subseteq \wp(M)$, k integer, there is a hitting set $H \subseteq M$ with $|H| \leqq k$, such that for every $A \in \mathcal{M}$ holds $H \cap A \neq \emptyset\}$ ([SP 8] of [18]) to our problem.

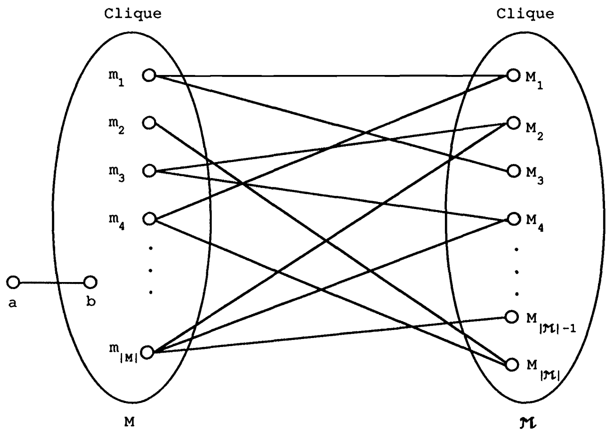


FIG. 5. *The constructed graph.*

For an instance $(M, \mathcal{M}, k)$ of HITTING SET, we construct the following graph $G = (V, E)$ (see Fig. 5):

$$V = M \cup \mathcal{M} \cup \{a, b\},$$

$$\begin{aligned} E = {} & \{\{m_i, M_j\} : m_i \in M, M_j \in \mathcal{M}, m_i \in M_j\} \\ & \cup \{\{m_i, m_j\} : m_i, m_j \in M, i \neq j\} \\ & \cup \{\{M_i, M_j\} : M_i, M_j \in \mathcal{M}, i \neq j\} \\ & \cup \{\{b, m_i\} : m_i \in M\} \cup \{\{a, b\}\}. \end{aligned}$$

We must show that $(M, \mathcal{M}, k) \in \mathrm{HS} \Leftrightarrow (G, k+1) \in \mathrm{MDC}$.

If $(M, \mathcal{M})$ has a hitting set H of size at most k, then $H \cup \{b\}$ is a clique and a dominating set, since it contains a vertex belonging to M_i for each $M_i \in \mathcal{M}$.

If G has a dominating clique C of size at most $k + 1$, it must contain b, and therefore the remaining vertices of C must belong to M, but none of them is in $\mathcal{M}$. Since C dominates $\mathcal{M}$, each $M_i \in \mathcal{M}$ has at least one neighbour in C, which is surely different from b. Hence $|C| \leqq k + 1$ gives $C - \{b\}$ as a hitting set of size at most k in $(M, \mathcal{M})$.

This completes the NP-completeness proof, but we still must show that G is a cocomparability graph. The corresponding labeling is given in Fig. 6. □

Even the existence problem DOMINATING CLIQUE (i.e., given graph G, does G have a dominating clique?) remains NP-complete. The proof is similar to the one for weakly triangulated graphs [4].

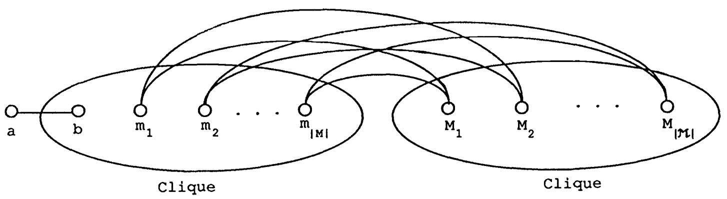


FIG. 6. *Labeling of the graph G for* MINIMUM DOMINATING CLIQUE.

THEOREM 7.2. DOMINATING CLIQUE *is* NP-*complete on cocomparability graphs*.

Proof. We transform MONOTONE 3-SAT (mentioned in the comments to the problem 3-SAT [LO 2] in [18]) to our problem.

Let W be an instance of MONOTONE 3-SAT, i.e., $W = W_1, W_2, \dots, W_r$, where

$$W_i = (x_{i_1}^{\alpha_{i_1}} \vee x_{i_2}^{\alpha_{i_2}} \vee x_{i_3}^{\alpha_{i_3}}) \text{ for } \alpha_{i_j} \in \{0, 1\}$$

and, for a fixed i, either all $\alpha_{i_j} = 0$ or all $\alpha_{i_j} = 1$ (each clause W_i contains either only negated or only nonnegated variables). H contains the variables $x_1, x_2, \dots, x_l$. As usual, x_i^0 means $\sim x_i$ and x_i^1 means x_i.

We construct a cocomparability graph $G = (V, E)$ from W, below:

$$V = \{W_i : i \in \{1, 2, \dots, m\}\} \cup \{x_j : j \in \{1, 2, \dots, l\}\}$$
$$\cup \{\sim x_j : j \in \{1, 2, \dots, l\}\}.$$

Then let $W_1, W_2, \dots, W_r$ be the clauses containing only nonnegated and $W_{r+1}, W_{r+2}, \dots, W_m$ the clauses containing only negated variables, without loss of generality $1 \leqq r < m$. Thus we can define

$$\begin{aligned} E = {} & \{\{W_i, W_j\} : 1 \leqq i, j \leqq r, i \neq j\} \\ & \cup \{\{W_i, W_j\} : r + 1 \leqq i, j \leqq m, i \neq j\} \\ & \cup \{\{x_j^{\alpha_j}, x_k^{\alpha_k}\} : j \in \{1, 2, \dots, l\}, k \in \{1, 2, \dots, l\}, j \neq k, \alpha_j, \alpha_k \in \{0, 1\}\} \\ & \cup \{\{x_j^{\alpha_j}, W_i\} : x_j^{\alpha_j} \text{ is a literal in } W_i\}. \end{aligned}$$

Note that $\{x_j, \sim x_j\} \notin E$ for all $j \in \{1, 2, \dots, l\}$.

Let S be a truth assignment satisfying W, i.e., $S(W_i) = 1$ for all $i \in \{1, 2, \dots, m\}$. If $S(x_j) = \alpha_j, j \in \{1, 2, \dots, l\}$, then $C = \{x_j^{\alpha_j} : j \in \{1, 2, \dots, l\}\}$ is obviously a clique, and it is also a dominating set, since each clause W_i contains a literal with value one under S and this is contained in C.

Now let C be a dominating clique of G. Assume that C contains a vertex W_i. Assume that W_i contains only negated variables, i.e., $r + 1 \leqq i \leqq m$. Therefore C cannot contain any vertex x_j for $j \in \{1, 2, \dots, l\}$ or a vertex W_k containing only nonnegated variables. Consequently, W_1 is not dominated by C. Similarly, the case in which W_i contains only nonnegated variables implies that W_m is not dominated by C.

Hence we have $C \subseteq \{x_j : j \in \{1, 2, \dots, l\}\} \cup \{\sim x_j : j \in \{1, 2, \dots, l\}\}$. Since C is a clique, it cannot contain x_j and $\sim x_j$ for any $j \in \{1, 2, \dots, l\}$. Thus, for each $x_j^{\alpha_j}$ in C, we set $S(x_j) = \alpha_j$. No matter how we eventually extend S to a truth assignment on all variables, we get $S(W) = 1$, since C dominates $\{W_1, W_2, \dots, W_m\}$.

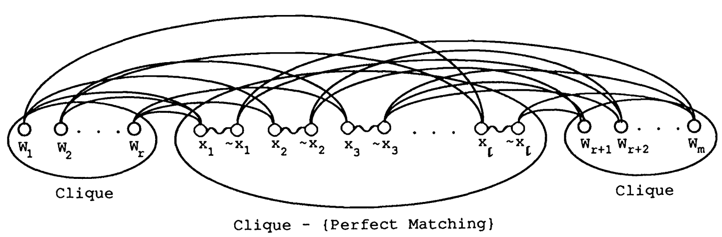


FIG. 7. *Labeling of the graph G for* DOMINATING CLIQUE.

Finally, we show that G is a cocomparability graph by giving the corresponding labeling (see Fig. 7).

The only fact worth mentioning is that there is no edge in G from a clause to a variable covering the nonedge $\{x_j, \sim x_j\} \notin E$, since clauses on the left of the variables contain only nonnegated variables and clauses on the right of the variables contain only negated variables. □

8. Conclusions. We have studied dominating set problems for cocomparability graphs, making use of a certain labeling for these graphs. We have extended polynomial time algorithms for domination, independent domination, connected domination, and total domination for interval graphs and permutation graphs to cocomparability graphs, and we have shown that the dominating clique problem is NP-complete for cocomparability graphs. We believe that forbidden ordered subgraph considerations may lead to extensions of other polynomial time algorithms to larger classes of graphs.

The clustering problem remains of unknown complexity for interval graphs and permutation graphs and also for cocomparability graphs [8].

The Hamiltonian path and Hamiltonian cycle problems have polynomial time algorithms for cocomparability graphs [11]–[13].

An important area for future research is the investigation of a general approach for dealing with families of graphs that have a forbidden ordered subgraph characterization. This is related to acyclic, antisymmetric orientations of the edges of a graph, as considered in [20].

It has been observed that the complexity of the domination problem on perfect graphs closely resembles that of the isomorphism problem on these families of graphs. Indeed, for various perfect graph families, both problems have polynomial time algorithms, whereas, for other families, isomorphism is isomorphism-complete, while domination is NP-complete. As pointed out by Derek Corneil, cocomparability graphs provide the first example for a class of perfect graphs for which the domination problem is polynomial, while the isomorphism problem remains isomorphism-complete.

Acknowledgments. We thank Derek Corneil and the Department of Computer Science at the University of Toronto for their hospitality.

Note added in proof. After this work was submitted for publication, the time bounds for some of the problems were improved. (See K. Arvind and C. Pandu Rangan, *Efficient algorithms for domination problems on cocomparability graphs*, Technical Report TR-TCS-90-18, Indian Institute of Technology, Department of Computer Science and Engineering, November 1990, submitted.)

REFERENCES

[1] C. Berge, *Graphs*, North-Holland, Amsterdam, 1985.

[2] A. A. Bertossi, *Total domination in interval graphs*, Inform. Process. Lett., 23 (1986), pp. 131–134.

[3] K. S. Booth and J. H. Johnson, *Dominating sets in chordal graphs*, SIAM J. Comput., 11 (1982), pp. 191–199.

[4] A. Brandstädt and D. Kratsch, *Domination problems on permutation and other graphs*, Theoret. Comput. Sci., 54 (1987), pp. 181–198.

[5] ———, *On the restriction of some* NP-*complete graph problems to permutation graphs*, in Proc. of FCT '85, Springer-Verlag, Berlin, New York, 1985, pp. 53–62.

[6] C. J. Colbourn and A. Lubiw, private communications, 1989.

[7] C. J. Colbourn and L. Stewart, *Permutation graphs: Connected domination and Steiner trees*, Discrete Math., 86 (1990), pp. 179–189.

[8] D. G. CORNEIL AND Y. PERL, *Clustering and domination in perfect graphs*, Discrete Appl. Math., 9 (1984), pp. 27–39.

[9] D. G. CORNEIL AND L. STEWART, *Dominating sets in perfect graphs*, Discrete Math., 86 (1990), pp. 145–164.

[10] P. DAMASCHKE, *Forbidden ordered subgraphs*, in Topics in Combinatorics and Graph Theory, R. Bodendieck and R. Henn, eds., Physica-Verlag, Heidelberg, 1990, pp. 219–229.

[11] P. DAMASCHKE, J. S. DEOGUN, D. KRATSCH, AND G. STEINER, *Finding Hamiltonian paths in cocomparability graphs using the bump number algorithm*, Order, 8 (1992), pp. 383–391.

[12] J. S. DEOGUN AND G. STEINER, *Hamiltonian cycle is polynomial on cocomparability graphs*, Discrete Appl. Math., 39 (1992), pp. 165–172.

[13] ———, *Polynomial algorithms for Hamiltonian cycle in cocomparability graphs*, SIAM J. Comput., to appear.

[14] A. K. DEWDNEY, *Fast turning reductions between problems in* NP, Report 71, University of Western Ontario, 1981.

[15] M. FARBER, *Independent domination in chordal graphs*, Oper. Res. Lett., 1 (1982), pp. 134–138.

[16] ———, *Domination, independent domination, and duality in strongly chordal graphs*, Discrete Appl. Math., 7 (1984), pp. 115–130.

[17] M. FARBER AND J. M. KEIL, *Domination in permutation graphs*, J. Algorithms, 6 (1985), pp. 309–321.

[18] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of* NP-*Completeness*, W. H. Freeman, New York, 1979.

[19] M. C. GOLUMBIC, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[20] C. T. HOANG AND B. REED, *P_4-comparability graphs*, Discrete Math., 74 (1989), pp. 173–200.

[21] D. S. JOHNSON, *The* NP-*completeness column: An ongoing guide*, J. Algorithms, 6 (1985), pp. 434–451.

[22] J. M. KEIL, *Total domination in interval graphs*, Inform. Process. Lett., 22 (1986), pp. 171–174.

[23] J. PFAFF, R. LASKAR, AND S. T. HEDETNIEMI, NP-*Completeness of Total and Connected Domination and Irredundance for Bipartite Graphs*, Tech. Report 428, Clemson University, 1983.

[24] G. RAMALINGAM AND C. P. RANGAN, *Total domination in interval graphs revisited*, Inform. Process. Lett., 27 (1988), pp. 17–21.

[25] J. SPINRAD, *On comparability and permutation graphs*, SIAM J. Comput., 14 (1985), pp. 658–670.

[26] R. E. TARJAN, *Data Structures and Network Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1983.

[27] K. WHITE, M. FARBER, AND W. R. PULLEYBLANK, *Steiner trees, connected domination, and strongly chordal graphs*, Networks, 15 (1985), pp. 109–124.

SIAM J. DISC. MATH.
Vol. 6, No. 3, pp. 418–427, August 1993

007

COMPLEXITY OF THE FORWARDING INDEX PROBLEM*

RACHID SAAD†

Abstract. The forwarding index problem is, given a connected graph G and an integer k, finding a way of connecting each ordered pair of vertices by a path so that every vertex is an internal point of at most k such paths. Such a problem arises in the design of communication networks and parallel architectures, a model of parallel computation being represented by a network of processors or machines processing and forwarding (synchronous) messages to each other and a physical constraint on the number of messages that can be processed by a single machine. In this paper, the author proves that the forwarding index problem is NP-complete even if the diameter of the graph is 2, thereby answering a question of F. Chung et al. [*IEEE Trans. Inform. Theory*, 33 (1987), pp. 224–232] concerning the complexity of the problem.

Key words. NP-completeness, network, routing, forwarding index

AMS subject classifications. 68C25, 68E10

1. Introduction. A routing R of a graph G of order n is a set of $n(n-1)$ elementary paths specified for all ordered pairs of vertices of G. The load (or charge) of a vertex v for a given routing R of G, denoted by $\xi(G, R, v)$ or $\xi(R, v)$ if the graph G is understood, is the number of paths p of R passing through v, such that v is not an end vertex of p. The forwarding index of a network (G, R) is defined to be $\xi(G, R) = \max_{v \in V(G)} \xi(G, R, v)$. These definitions provide an appropriate theoretical framework in which some "network problems" can be discussed. Indeed, each vertex of G an be viewed as an element treating and sending data or messages through paths of G to all other vertices, and the aim in the design of communication networks is then to minimize $\xi(G, R)$ so as to prevent the overload of a vertex. Another relevant field of application concerns parallel architectures. A model of parallel computation is given by a network of processors forwarding to each other synchronous messages (to be processed), and the goal here is to minimize the size of a maximum "queue" on a vertex (the messages received by a vertex form a queue and are processed one after another, which induces some delay to be minimized). This, stated more formally, yields the following problem.

The forwarding index problem.

Instance: G a graph; k an integer.

Question: Does there exist a routing R of G such that for all $x \in G$, $\xi(R, x) \leqq k$?

This problem will be denoted by FI.

In [3] and [4] some upper bounds on the forwarding index are given as functions of some parameters of the graph (mainly, its connectivity and minimum degree). However, no efficient algorithm was obtained for FI. In [1] the authors raised the question of the complexity of FI. The aim of this paper is to prove the following theorem.

THEOREM 1. FI *is* NP-*complete*.

Let us first recall some notions from graph theory. The girth of a graph $G = (V, E)$ is the minimum length of a cycle in G. An xy-path of G is a path of G connecting the vertex x to y. An xy-path P of G is said to be inclusion-minimal if the subgraph induced by P does not properly contain an xy-path, i.e., if every xy-path included in the subgraph induced by P uses all the vertices of P. Observe in this respect that all the paths involved in an optimal routing for an instance of FI can be supposed to be inclusion-minimal.

* Received by the editors March 25, 1991; accepted for publication (in revised form) August 11, 1992.

† Laboratoire de recherche en informatique, Bât 490, Université Paris-Sud, 91405 Orsay Cedex, France.

The degree of a vertex x of G is the number of its neighbours and is denoted by $\delta(x)$. For a pair x, y of vertices of G, the distance from x to y is the minimum length of an xy-path in G and is denoted by $d(x, y)$.

Let us now introduce some notation. Let R be a routing of G and let W be a subset of V. R_W will denote the routing R restricted to the pairs of vertices of W. R_{xy} will denote the route of R connecting x to y. Let $X_1, X_2, \ldots, X_k$ be subsets of vertices of G. We say that R_{xy} is in $xX_1X_2 \cdots X_ky$ and we note $R_{xy} \in xX_1X_2 \cdots X_ky$, if $R_{xy} = xx_1x_2 \cdots x_ky$ for some $x_1, x_2, \ldots, x_k$ in $X_1, X_2, \ldots, X_k$, respectively.

2. Preliminary results. To prove the theorem, we will reduce the NP-complete three-dimensional matching problem (3DM) to FI. The reduction will occur through a series reductions to intermediate problems as follows:

$$3\text{DM} \prec \text{PART1} \prec \text{PART2} \prec \text{FI}' \prec \text{FI}.$$

$\prec$ denotes the usual polynomial reduction, and PART1, PART2, and FI′ are defined below. The main intuition underlying the proof of our theorem is that "if we can route near pairs of vertices by short disjoint paths, then certainly we can obtain good routings of the graph." Hence, a reasonable candidate for a reduction to FI would be the problem of partitioning the vertices of a graph into paths of length 2. However, we need for our proof more precise information on the "short paths" involved. Thus, the following problems are considered.

Problem PART1.

Instance: A graph G of maximum degree $\Delta \leqq 3$ and girth $g \geqq 5$.

Question: Does there exist a partition of the vertices of G into paths of length 2?

Problem PART2.

Instance: A 3-regular graph G of girth $g \geqq 5$.

Question: Does there exist a partition of the vertices of G into paths of length 2?

Note that, for every instance of PART2, we know exactly the number of paths of length 2 in the instance as well as the number of such paths passing through any fixed vertex x of the graph. We consider now for the needs of the proof the following slight generalisation of FI.

The forwarding index problem 2.

Instance: $G = (V, E)$ a graph, k an integer and a subset of vertices $X \subset V$.

Question: Does there exist a routing R of G, such that

$$\forall x \in X, \ \xi(G, R, x) = 0,$$

$$\xi(G, R) \leqq k?$$

This latter problem will be denoted by FI′.

We recall the definition of the three-dimensional matching problem.

Problem 3DM.

Instance: $X \times Y \times Z$ a Cartesian product of three sets of equal cardinalities, and $U \subset X \times Y \times Z$.

Question: Does there exist a subset M of U, called a perfect three-dimensional matching, such that every element of $X \cup Y \cup Z$ belongs to exactly one element of M?

It is well known that 3DM is NP-complete.

In a first step, we establish the NP-completeness of PART2 (this is done in Lemmas 1 and 2 by a reduction from 3DM). In a third lemma, we prove that FI′ reduces polynomially to FI. Next, we prove that PART2 reduces polynomially to FI′, which will conclude our proof. Now we can start the proof of our theorem.

LEMMA 1. 3DM $\prec$ PART1.

Proof. Given an instance of 3DM $U \subset (X \times Y \times Z)$, we will construct a graph G of maximum degree $\Delta \leqq 3$ and girth $g \geqq 5$ such that a solution to PART1 on G implies a solution to 3DM on U and conversely.

For each $x \in X$, let us denote by n_x the number of occurrences of x in the triples of U; that is, $n_x = |\{u \in U | x \in u\}|$; n_y and n_z are defined likewise. We can reduce to the case where $n_x, n_y, n_z \geqq 2$ for all x, y, z.

Each element $x \in X$ will be represented by a graph H_x defined as follows (see Fig. 1): $H_x = (V_x, E_x)$, $V_x = S_x \cup S'_x \cup S''_x \cup C_x$.

S_x (respectively, S'_x, S''_x) induces an independent set of order n_x. C_x induces a chordless cycle of length $3n_x + 1$. The vertices of S_x (respectively, S'_x; respectively, S''_x) are labeled x_i (respectively, x'_i; respectively, x''_i) for $i = 1, 2, \ldots, n_x$. There is an edge between x_i and x''_j if and only if $i = j$. Similarly, there is an edge between x'_i and x''_j if and only if $i = j$. The vertices of C_x are labeled $x^0_1, x^0_2, \ldots, x^0_{3n_x+1}$. There is an edge between x_i and x^0_r if and only if $r = 3i - 2$. This concludes the description of H_x. Similarly, we construct a graph H_y for each $y \in Y$ and a graph H_z for each $z \in Z$.

The triples of U are labeled $u_1, u_2, \ldots, u_{|U|}$ and ordered accordingly. For each triple $u_i \in U$, we construct a cycle C_{u_i} of length 6 with vertices labeled cyclically by u^1_i, $u^2_i, \ldots, u^6_i$. See Fig. 2.

The graph G instance of PART1 is constructed as follows: Take the (vertex-)disjoint union of all the preceding components (namely, the H_x, H_y, H_z, C_{u_i} for all $x \in X$, $y \in Y$, $z \in Z$, $u_i \in U$). Now, for each $u_i = (x, y, z) \in U$, do the following: Consider the list $U(x)$ (respectively, $U(y)$, $U(z)$) of all the triples containing x (respectively, y, z) as ordered by the induced order of U on $U(x)$ (respectively, $U(y)$, $U(z)$).

If u_i is the i_1^{th} triple in $U(x)$, then add the edge $x_{i_1}u^1_i$.

If u_i is the i_2^{th} triple in $U(y)$, then add the edge $y_{i_2}u^3_i$.

If u_i is the i_3^{th} triple in $U(z)$, then add the edge $z_{i_3}u^5_i$ (see Fig. 3, below).

The so-obtained graph G is of maximum degree $\Delta = 3$ and of girth $\geqq 5$. Furthermore, G admits a partition into paths of length 2 if and only if $(X \times Y \times Z, U)$ admits a perfect three-dimensional matching.

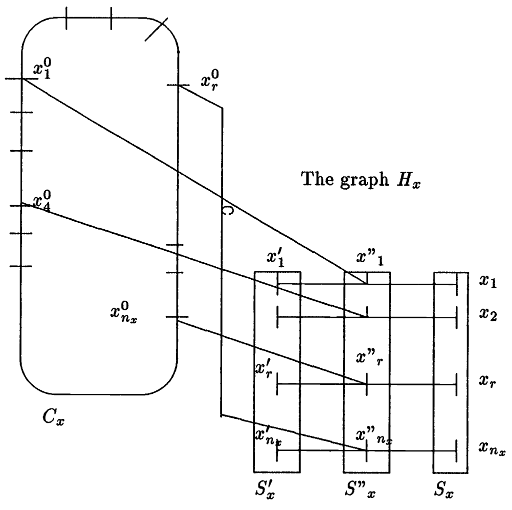


FIG. 1

The triple C_{u_i}.

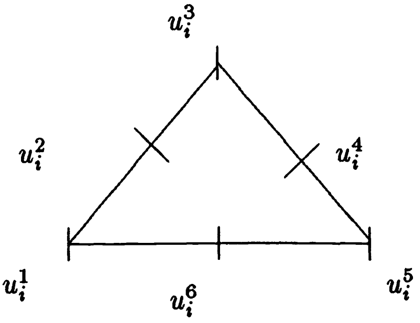


FIG. 2

Indeed, suppose that $(X \times Y \times Z, U)$ admits a perfect three-dimensional matching; let M be this matching and consider the following set P of disjoint paths of length 2: For each $u_i \in M$ (where $u_i = (x, y, z)$, u_i is the i_1^{th} triple containing x, the i_2^{th} triple containing y, and the i_3^{th} triple containing z), $x_{i_1} u_i^1 u_i^2 \in P$, $y_{i_2} u_i^3 u_i^4 \in P$, $z_{i_3} u_i^5 u_i^6 \in P$.

For each $j \neq i_1$, $x_j' x_j'' x_j \in P$, whereas $x_{i_1}' x_{i_1}'' x_{3i_1-2}^0 \in P$.

For each $j \neq i_2$, $y_j' y_j'' y_j \in P$ and $y_{i_2}' y_{i_2}'' y_{3i_2-2}^0 \in P$.

Similarly, for each $j \neq i_3$, $z_j' z_j'' z_j \in P$ and $z_{i_3}' z_{i_3}'' z_{3i_3-2}^0 \in P$. We add all the paths realizing the unique partition into paths of length 2 of

$$(C_x - \{x_{3i_1-2}^0\}) \cup (C_y - \{y_{3i_2-2}^0\}) \cup (C_z - \{z_{3i_3-2}^0\}).$$

For each $u_i \notin M$, $u_i^1 u_i^2 u_i^3 \in P$ and $u_i^4 u_i^5 u_i^6 \in P$. P forms a partition of G into paths of length 2. Conversely, suppose that G admits a partition into paths of length 2 and let P be such a partition. Then, for all $x \in X$ (respectively, $y \in Y$, $z \in Z$), P contains only one path of the form $x_i' x_i'' x_{3i-2}^0$ (respectively, $y_i' y_i'' y_{3i-2}^0$, $z_i' z_i'' z_{3i-2}^0$). Indeed, for all x (re-

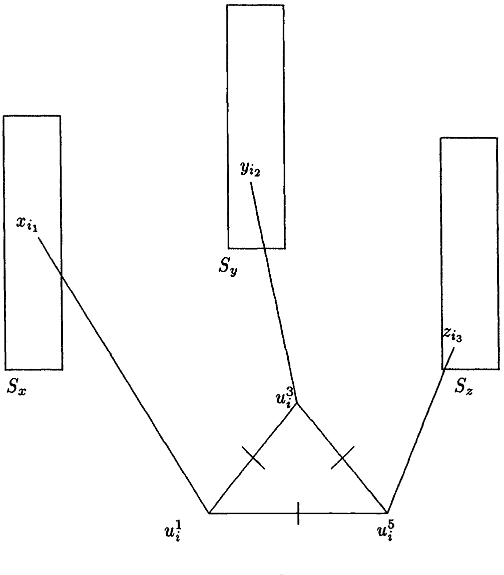


FIG. 3

spectively, y, z), P must contain at least one such path; otherwise, the cycle C_x (respectively, C_y, C_z) of length $3n_x + 1 = 1 \bmod 3$ cannot be covered by P. Moreover, P cannot contain more than two such paths; otherwise, a path of length $= 2 \bmod 3$ will be disconnected from C_x and G, which then cannot be covered by P. As a consequence, there exists for each x a *unique* j such that, for some $i \leqq |U|$, $x_j u_i^1 u_i^2 \in P$ or $x_j u_i^1 u_i^6 \in P$. We can suppose by relabeling the vertices of C_{u_i} that $x_j u_i^1 u_i^2 \in P$. As C_{u_i} is of length 6 with three internal points, we must have necessarily $y \in Y$, $z \in Z$ and j_2 and j_3 such that $y_{j_2} u_i^3 u_i^4 \in P$, $z_{j_3} u_i^5 u_i^6 \in P$ and $u = (x, y, z)$ by construction. Then let M be the following set of triples: $M = \{u_i \in U \mid \exists j,\ x_j u_i^1 u_i^2 \in P\}$. M is a perfect three-dimensional matching of $(X \times Y \times Z, U)$.

LEMMA 2. PART1 $\prec$ PART2.

Proof. Given an instance G of PART1, we construct an instance G'' of PART2, which has a solution if and only if G has a solution to PART1.

Consider first the two graphs H and H' (see Figs. 4 and 5). H is of girth 5, and, for each h in $V(H)$, $\delta(h) = 3$ if $h \neq h_1, h_2$ and $\delta(h) = 2$ for $h = h_1$ or $h = h_2$. Moreover, H can be partitioned into paths of length 2. Similarly, H' is of girth 5, and, for each of its vertices $h' \neq h_0$, $\delta(h') = 3$, whereas $\delta(h_0) = 1$.

Now G'' is constructed from G as follows: (i) Take two copies of G: G and G', (ii) For each vertex x (respectively, x') of degree 1 in G (respectively, G'), do the following: Take a copy of H and join x to the vertices h_1 and h_2 of this H.

For each vertex x of degree 2 in G, consider its counterpart x' in G', take a copy of H', and join x and x' to the vertex h_0 in this H'. The so-constructed graph is 3-regular of girth 5. Furthermore, G admits a partition into paths of length 2 if and only if G'' admits such a partition. Indeed, if G admits a partition into paths of length 2, then so do G', H, H', and G''.

Conversely, let P be a partition of G'' into paths of length 2 and x in $V(G)$ such that $\delta_G(x) = 1$. Let us denote by H_x the copy of H that has been attached to x in (i). Then P restricted to H_x forms a partition of H_x into paths of length 2 because if P contains a path of the form $c = xh_1z$ or $c = xh_2z$ or gxh_1 or gxh_2, with $z \in H_x$, $g \in V(G)$, then $(H_x - c)$ cannot be partitioned into paths of length 2. The same holds if $\delta_G(x) = 2$ and H'_x is the copy of H' attached to x in (ii). Hence P restricted to $G \cup G'$ is also a partition of $G \cup G'$. As a consequence, G admits a partition into paths of length 2.

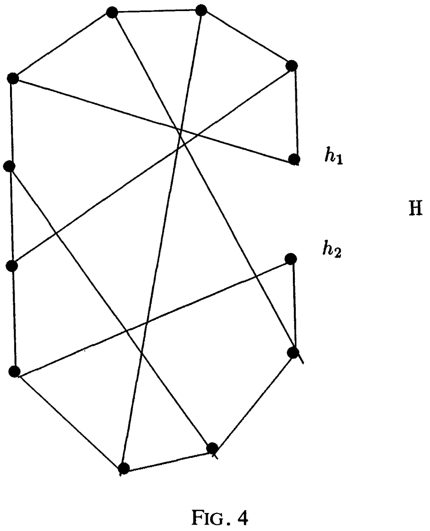


FIG. 4

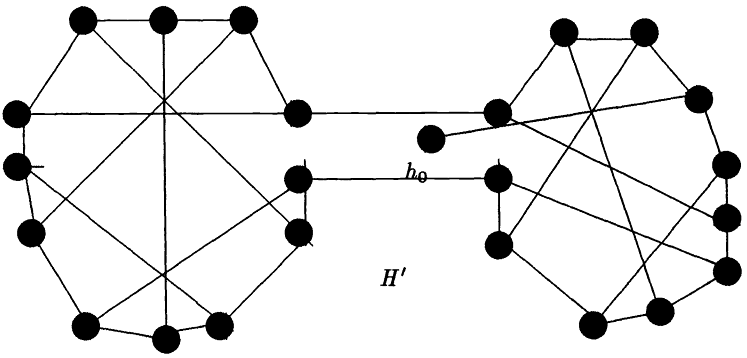


FIG. 5

LEMMA 3. $\mathrm{FI}' \prec \mathrm{FI}$.

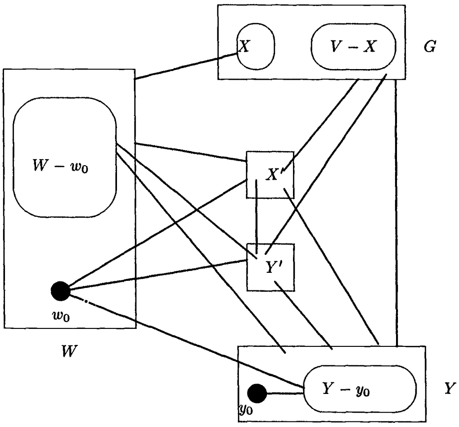
Proof. Let $(G = (V, E), X \subset V, k \in N)$ be an instance of FI′ and consider the following graph G':

$$G' = (V', E'),\ V' = V \cup X' \cup Y \cup Y' \cup W, \quad \text{with } |X'| = |X|,\ |W| = k|X|,$$
$$|Y| = |Y'| = (n - |X|)(|X|), \quad \text{where } n = |V|.$$

Description of E': $V \subset V'$ induces the graph G. W induces the union of a clique with an isolated point, noted w_0. $Y \cup Y' \cup X' \cup (W - \{w_0\})$ induces a clique. $Y \cup Y' \cup X' \cup \{w_0\}$ induces a clique minus one single edge $w_0 y_0$, $y_0 \in Y$. All the edges occur between X and W. All the edges occur between V and $X' \cup Y' \cup Y$. This completes the description of G' (see Fig. 6).

Then G admits a routing R satisfying (a) $\xi(G, R) \leqq k$, and (b) for all $x \in X$, $\xi(G, R, x) = 0$ if and only if G' admits a routing R' satisfying $\xi(G', R') \leqq k$.

Indeed, let R be a routing on G satisfying (a) and (b) and let R' be the following routing on G':



The graph G'.

FIG. 6. *Bold lines denote the occurrence of all edges between two sets of vertices.*

(i) $R'_V = R$,

(ii) For all $w \in (W \setminus \{w_0\})$, R'_{w_0w} (respectively, R'_{ww_0}) is of length 2 and passes through X (respectively, X'). As $|W| = k|X| = k|X'|$, all these charges can be uniformly distributed so that all the vertices of X, (respectively, X') will have charge k except one single vertex x_0 (respectively, x'_0) that will have charge $k - 1$,

(iii) For all w in W, v in $V \setminus X$, R'_{wv} passes through Y and is of length 2.

As here again all these charges can be uniformly distributed on Y and as $|W|\,|V \setminus X| = k|Y|$, all the vertices of Y will have charge k. Similarly, by letting all the paths R'_{vw} pass through Y' (with v in $V \setminus X$, w in W), all the vertices of Y' will have charge k.

(iv) $R'_{w_0y_0} = w_0x_0y_0$ and $R'_{y_0w_0} = y_0x'_0w_0$, x_0 and x'_0 being the only vertices of $X \cup X' \cup Y \cup Y'$ of charge less than k.

The routing R' satisfies $\xi(G', R', x) \leqq k$ for each x in V'.

Conversely, let R' be a routing on G' satisfying $\xi(G', R') \leqq k$. Then, as $X \cup X' \cup Y \cup Y'$ separates both W from $V - X$ and $\{w_0\}$ from $W - \{w_0\}$ and as any path from w_0 to y_0 must also pass through $X \cup X' \cup Y \cup Y'$, the total sum of the charges induced on $X \cup X' \cup Y \cup Y'$ by the routes of $R' - R'_V$ is greater than or equal to $2|W|(|V| - |X|) + 2(|W| - 1) + 2 = |X \cup X' \cup Y \cup Y'|k$. As a consequence, all the vertices of $X \cup X' \cup Y \cup Y'$ are "saturated" (that is, they have charge k) by these routes of R'. In particular, the vertices of X are saturated by these routes of R' that are not in R'_V. Therefore, R'_V satisfies (a) and (b).

3. Main result. We refer to Theorem 1 in § 1.

Proof. It suffices to prove that PART2 $\prec$ FI′. Let G be an instance of PART2. We can suppose that $n = |V(G)| = 0 \bmod 3$. We can also suppose that $n = 6 \bmod 8$; otherwise, we take eight disjoint copies of G and we add a 3-regular graph of girth 5 and order $= 6 \bmod 8$ that admits a partition into paths of length 2 (there exists such a graph of order 30). In the following, $n = 6 \bmod 8$ and $n = 0 \bmod 3$.

Let us then consider the following instance of FI′: $I = (G', X, k)$, where $G' = (V'E')$ with $V' = V_1 \cup V_2 \cup H_1 \cup H_2 \cup Y_1 \cup Y_2$; $|V_1| = |V_2| = |V(G)|$; $|H_2| = 2$, $|H_1| = 8$; $|Y_1| = 2 \bmod 4$, and

$$\frac{3n(n-10)}{8} + \frac{n}{2} + \frac{n|Y_1|}{4} = \frac{\left(6n - \frac{n}{3}\right)}{2} + (n+8)|Y_1| + s \quad \text{with } s \leqq 3n + 7.$$

This can also be written $|Y_1| = 4r + 2$ and

$$\frac{3n(n-10)}{8} + \frac{n}{2} - \frac{\left(6n - \frac{n}{3}\right)}{2} - 2(n+8) + \frac{n}{2} = (3n+8)r + s.$$

So $|Y_1|$ and s are polynomial and are obtained, respectively, as quotient and remainder of Euclidean division. Moreover, we can suppose that $|Y_1| \geqq 6$, for otherwise n will be bounded by a constant.

$|Y_2|$ satisfies

$$2n + 33 + 2|Y_2| = \frac{3n(n-10)}{8} + \frac{n}{2} + \frac{n|Y_1|}{4}$$

($|Y_2|$ is well defined because the right-hand term is odd). The subgraph of G' generated by V_1 is isomorphic to G and will therefore be called G. The subgraph of G' generated by V_2 is isomorphic to the graph U on $n = |V(G)|$ vertices obtained from G in the following way: Take a copy of G and add all the edges xy such that the distance

$d_G(x, y) \geqq 3$. H_1 and H_2 are isomorphic to K_8 and K_2, respectively. Y_1 and Y_2 are cliques. All the edges occur between V_2 and H_2, Y_1 and Y_2, Y_1 and H_2, Y_2 and H_2, H_1 and H_2, H_1 and V_1, V_1 and Y_2. Two vertices of V_1 and V_2 are linked by an edge if and only if they correspond to each other in the natural pairing between V_1 and V_2 (recall that G is a partial subgraph of U). For every vertex x of V_1, we denote by x' the counterpart of x in V_2. All the edges occur between U and Y_1 except s of them. This completes the description of G'. (See Fig. 7, below.)

Let us put $X = V_2 \cup Y_1 \cup Y_2$ and $k = 2n + 33 + 2|Y_2|$. Then G admits a partition into paths of length 2 if and only if (G', X, k) admits a routing R satisfying

(a) for all $x \in X$, $\xi(R, x) = 0$,

(b) for all $x \in V' - X$, $\xi(R, x) \leqq k$.

Indeed, suppose that G admits a partition P into paths of length 2 and consider the following routing R on G'.

(1) If $xyz \in P$, we take $R_{x'z'}$ in $x'H_2z'$ (that is, x' being the counterpart of x in U, we let $R_{x'z'}$ pass through H_2) and we take $R_{z'x'} = z'zyxx'$. (We let only one ordered pair of the pair $\{x', z'\}$ pass through H_2.)

(2) If $xyz \notin P$, where xyz is a path of length 2 in G, then $R_{x'z'} \in x'H_2z'$ and $R_{z'x'} \in z'H_2x'$. Moreover, we choose R so that all the charges (or loads) induced on H_2 by the routes of (1) and (2) be uniformly distributed on H_2 (this is possible since $|H_2| = 2$, the number of such routes is $6n - n/3 = 0 \bmod 2$), and all the edges occur between U and H_2. So we have routed all the ordered pairs of vertices of U. This partial routing is well defined because G contains no C_4: There exist no xy_1z and xy_2z in G.

(3) If $xy \in E(G)$, $R_{x'y} = x'xy$ and $R_{xy'} = xyy'$.

(4) If $\{x, z\}$ is a pair of vertices at distance 2 in G (that is, $d_G(xz) = 2$), then there exists only one y such that the path xyz is in G, and we take

$$xyz \in R, \quad zyx \in R, \quad x'xyz \in R, \quad zyxx' \in R, \quad z'zyx \in R, \quad xyzz' \in R.$$

In this way, as G is 3-regular of girth 5, every vertex y of G, considered as an internal point of three paths of length 2 in G, will be charged by $1 + 3 \times 6 = 19$ routes of (1) and (4). Moreover, as each time we route a pair (x, z') or (z', x) (with x in G and z' in U), we use the vertex x, the charge on every vertex of G is $2(n - 1) + 19$.

(5) Let (x, y) be an ordered pair of vertices of G such that $d_G(xy) \geqq 3$ (there are $n(n - 4) - 6n = n(n - 10)$ such ordered pairs). Then the ordered pairs (x, y), (y, x), (x', y), (y, x'), (x, y'), (y', x) are routed as follows:

$$R_{yx} \in yH_1x, \quad R_{xy} \in xH_1y, \quad R_{x'y} \in x'xH_1y,$$
$$R_{yx'} \in yH_1xx', \quad R_{xy'} \in xH_1yy', \quad R_{y'x} \in y'yH_1x.$$

Moreover, we choose R so that all the corresponding charges be uniformly distributed on H_1. This is possible since the number of the corresponding ordered pairs of vertices is $3n(n - 10) = 0 \bmod 8$, and all the edges occur between H_1 and G. We have then routed all the vertices of $G \cup U$.

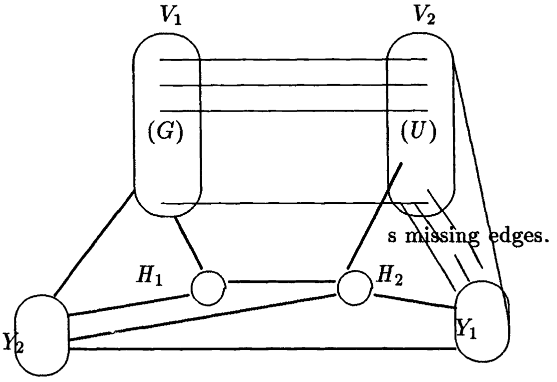
(6) If $u \in H_1 \in Y_2$, $x' \in U$, then $uxx' \in R$ and $x'xu \in R$.

(7) If $h_2 \in H_2$ and $x \in G$, then $R_{h_2x} \in h_2H_1x$ and $R_{xh_2} \in xH_1h_2$. Furthermore, R is chosen in such a way as to distribute all these changes uniformly on H_1, which is possible since $2|H_2|\,|V(G)|$ is a multiple of 8.

(8) Similarly, if $y_1 \in Y_1$ and $x \in G$, $R_{y_1x} \in y_1H_2H_1x$, $R_{xy_1} \in xH_1H_2y_1$, and R is chosen so as to distribute the charges of these routes uniformly on H_1 and H_2, which is possible because $|Y_1|$ and n are even.

(9) If $y_1 \in Y_1$, $x' \in U$ and $x'y_1 \notin E(G')$, then $R_{y_1x'} \in y_1H_2x'$ and $R_{x'y_1} \in x'H_2y_1$; again these charges are uniformly distributed on H_2.

This concludes the definition of the routing R.



The graph G'.

FIG. 7. *Bold lines denote the existence of all edges between two sets of vertices.*

Let us count the charges on each vertex. On each vertex x of G, R induces a charge equal to the sum of the charges induced on x by the routes (1)–(6), which yields

$$19 + 2(n-1) + 2|Y_2| + 2|H_1| = 2n + 33 + 2|Y_2| = k.$$

On each vertex h_1 of H_1, R induces a charge equal to the sum of the charges induced on h_1 by the routes (5), (7), and (8) as follows:

$$\frac{3n(n-10)}{8} + \frac{n}{2} + \frac{2n|Y_1|}{8} = k.$$

On each vertex of H_2, R induces the charge

$$\frac{6n - \frac{n}{3}}{2} + (n+8)|Y_1| + s = k.$$

R induces no charge on $X = V_2 \cup Y_1 \cup Y_2$. Thus (a) and (b) are satisfied by R.

Conversely, suppose that there exists a routing R solution of FI$'$ for the instance (G', X, k). We can suppose that the paths of R are inclusion-minimal. First, we have the following facts:

For each ordered pair of vertices $(y_1, g) \in Y_1 \times G$, R_{y_1g} and R_{gy_1} pass through H_2 and H_1.

For each ordered pair of vertices $(y_1, h_1) \in Y_1 \times H_1$, $R_{y_1h_1}$ and $R_{h_1y_1}$ pass through H_2.

For each ordered pair of nonadjacent vertices $(y_1, g') \in Y_1 \times U$, $R_{y_1g'}$ and $R_{g'y_1}$ pass through H_2. Let us put $W =$ the subgraph of G' induced by the set $V(G')\backslash(Y_1 \cup Y_2)$, and $u =$ the sum of the charges induced on H_1 by the routes of R connecting ordered pairs of vertices of W (that is, by R_W); similarly, u' is defined as the sum of the charges induced on H_2 by R_W. From the preceding facts, we deduce that

(*) $u \leqq 8k - 2n|Y_1| = 3n(n-10) + 4n$,

(**) $u' \leqq 2k - 2s - 2n|Y_1| - 8|Y_1| = 6n - n/3$.

From (**), we deduce that there are at least $n/3$ ordered pairs (g'_1, g'_2), g'_1 and g'_2 belonging to $V_2 = V(U)$, such that $R_{g'_1g'_2}$ passes through V_1. Let S be a set of $n/3$ such ordered pairs and S' its corresponding set of routes. All these routes have lengths greater than or equal to 4, so they use at least $3(n/3) = n$ charges on $V' - X$. Every other xy-path of R uses at least $d_{G'}(x, y) - 1$ charges on $V' - X$. Hence

$$\sum_{x \in V'} \xi(R, x) \geqq n + \sum_{(x,y) \notin S} (d_{G'}(x, y) - 1).$$

Now, computing the second-hand term of the inequality, we obtain precisely

$$n + \sum_{(x,y)\notin S} (d_{G'}(x, y) - 1) = k|V' - X|.$$

(This can be checked either by computing explicitly the number n_1 of nonadjacent ordered pairs of vertices in G' and the number n_2 of the ordered pairs of vertices at distance 3 in G', or by observing that, except for $n/3$ ordered pairs of vertices, the total contribution of which was n, all the routes of the routing constructed in the preceding page were shortest paths in G' and their total contribution, in terms of loads on $V' - X$, was precisely $k|V' - X|$.) Now, since (a) and (b) are satisfied by R, all the routes of S' are of length 4, and all the other ordered pairs of vertices of G' are routed by shorted paths in G'. This has the following consequences:

(i) H_2 is saturated by the $6n - n/3$ remaining ordered pairs of U together with the routes substracted from (**) whose shortest paths pass necessarily through H_2,

(ii) From (i) and (*), we deduce that H_1 is saturated by the $3n(n - 10) + 4n$ ordered pairs of vertices of the form (x, y), (x', y), (x, y') for (x, y) in V_1 with $d_G(x, y) \geqq 3$, or of the form (x, h_2) or (h_2, x) for x in V_1 and h_2 in H_2, whose shortest paths pass necessarily through H_1,

(iii) From (ii), we deduce that if $xyz \subset G$, then $R_{xz} = xyz$, $R_{zx} = zyx$, $R_{x'z} = x'xyz$, $R_{xz'} = xyzz'$, $R_{x'y} = x'xy$ and $R_{xy'} = xyy'$ (indeed, none of these ordered pairs can be routed by shortest paths through H_1 since H_1 is saturated by the paths of (ii)).

Moreover, from (i) and the fact that the routes of R are inclusion minimal, we deduce that if $y_2 \in Y_2$, $g' \in U$, then $R_{y_2g'} = y_2gg'$ and $R_{g'y_2} = g'gy_2$. Thus each vertex of G is charged by at least $2n + 32 + 2|Y_2|$ paths of R routing ordered pairs of vertices in $V' \times V' - S$. As a consequence, there remains on each vertex x of G at most one available charge to route the $n/3$ ordered pairs of S that together use n charges on G. The corresponding routes of S are therefore pairwise vertex-disjoint, and their restrictions to G induce a partition of G into paths of length 2. This ends the proof.

THEOREM 2. FI *remains* NP-*complete for graphs of diamater* 2.

Proof. The diameter of G' in the reduction of Theorem 1 is 2. Moreover, in the reduction of Lemma 3, the diameter of the graph corresponding to the instance of FI is 2. This proves the theorem.

Let MFI denote the version of FI corresponding to the case where the routes of R are shortest paths. Then, by contrast, we have the following result of [2].

THEOREM (see [2]). MFI *is polynomial for graphs of diameter* 2.

Now let SFI denote, as in [2], the version of FI corresponding to the case where the routes of R are symmetric; that is, $R_{xy} = R_{yx}$ for each x, y. Then, by modifying slightly the preceding proof, we obtain the NP-completeness of SFI.

Acknowledgments. I thank Prof. W. F. de la Vega and the unknown referee for helpful suggestions on writing.

REFERENCES

[1] F. R. K. CHUNG, E. COFFMAN, M. REIMAN, AND B. SIMON, *The forwarding index of communication networks*, IEEE Trans. Inform. Theory, 33 (1987), pp. 224–232.

[2] M. HEYDEMANN, J. OPATRNY, AND D. SOTTEAU, *Consistent routings and their complexity*, LRI Report No. 496, Universite Paris 11, Orsay, 1989.

[3] W. F. DE LA VEGA AND Y. MANOUSSAKIS, *The forwarding index for k-connected graphs*, Discrete Appl. Math., 1992, to appear.

[4] M. EL HADDAD, Y. MANOUSSAKIS, AND R. SAAD, *Antisymmetric routings of networks*, Networks, submitted.

SIAM J. DISC. MATH.
Vol. 6, No. 3, pp. 428–431, August 1993

008

PARITY SUBGRAPH, SHORTEST CYCLE COVER, AND POSTMAN TOUR*

CUN-QUAN ZHANG†

Abstract. Let $G = (V, E)$ be a simple graph such that the number of odd vertices of G is $|V_0|$ and the minimum odd degree is δ_0. This paper proves that the number of edges in a smallest parity subgraph of G is at most $|V| - \text{Min}\{\delta_0, |V| - |V_0|/2\}$. Consequently, some results about the shortest cycle cover problem due to Itai and Rodeh, Fan, Zhang, Raspaud, Zhao are generalized. If G is a 2-edge-connected simple graph such that either G admits a nowhere-zero 4-flow or G contains no subdivision of the Petersen graph, then the total length of a shortest cycle cover of G is at most $|E| + |V| - \text{Min}\{\delta_0, |V| - |V_0|/2\}$.

Key words. parity subgraph, cycle cover, postman tour

AMS subject classifications. 05C38, 05C70, 05C05

1. Introduction. We follow the terminology and notation of Bondy and Murty [BM]. All graphs we will consider in this paper are two-edge-connected and simple. Note that the *cycles* in this paper are closed simple paths.

The *shortest cycle cover* problem (SCC) is to find a family F of cycles covering every edge of G such that the total length of F is as small as possible. The SCC might be an NP-complete problem (it was conjectured by Itai, Lipto, Papadimitriou, and Rodeh [IL]). Upper bounds of solutions of the SCC for various graphs have been studied extensively ([AT], [BJJ], [F1], [F2], [FP], [IL], [IR], [Z1], etc.). In this paper the following results will be generalized. (Refer to Younger [Y] for the definition of nowhere-zero integer flows.)

THEOREM A ([IR]; see also [F1]). *Let $G = (V, E)$ be a simple graph admitting a nowhere-zero 4-flow. Then the total length of an SCC of G is at most $|E| + |V| - 1$.*

THEOREM B [F1]. *Let $G = (V, E)$ be a simple graph admitting a nowhere-zero 4-flow. Then the total length of an SCC of G is at most $|E| + |V| - 2$.*

THEOREM C [Z1]. *Let $G = (V, E)$ be a simple graph admitting a nowhere-zero 3-flow. Then the total length of an SCC of G is at most $|E| + |V| - 3$.*

(*Note*: Theorem C was initially conjectured by Fan [F1] and was recently generalized by Raspaud [R] and Zhao [ZC].)

THEOREM D [R]. *Let $G = (V, E)$ be a simple graph admitting a nowhere-zero 4-flow and $G \neq K_4$. Then the total length of an SCC of G is at most $|E| + |V| - 3$.*

THEOREM E [ZC]. *Let $G = (V, E)$ be a simple graph admitting a nowhere-zero 4-flow. Then, except for a family of counterexamples, the total length of an SCC of G is at most $|E| + |V| - 4$.*

Let $G = (V, E)$ be a simple graph. Denote

$$V_0 = \{v: v \in V(G) \text{ and } d(v) \text{ is odd}\}$$

and

$$\delta_0 = \text{Min}\{d(v): v \in V(G) \text{ and } d(v) \text{ is odd}\},$$

where δ_0 is the minimum odd degree of G.

* Received by the editors July 24, 1991; accepted for publication (in revised form) May 22, 1992. This research was partially supported by National Science Foundation grant DMS-8906973.

† Department of Mathematics, West Virginia University, Morgantown, West Virginia 26506.

A better upper bound for the problem of SCC is given in the following theorem.

THEOREM 1. *Let $G = (V, E)$ be a simple graph admitting a nowhere-zero 4-flow. Then the total length of an* SCC *of G is at most* $|E| + |V| - \text{Min}\{\delta_0, |V| - |V_0|/2\}$.

A graph admitting a nowhere-zero 4-flow is two-edge-connected. Therefore $\delta_0 \geqq 3$. It is obvious that $|V| - (|V_0|/2) \geqq |V|/2$ for any graph. Thus the upper bounds in Theorems A, B, C, and D are improved in Theorem 1, and Theorem E is also generalized when $\delta_0 \geqq 4$ and $|V| \geqq 8$.

Let H be a spanning subgraph of G. The degrees of a vertex v in H and G are denoted by $d_H(v)$ and $d_G(v)$, respectively. A spanning subgraph H of G is called a *parity subgraph* of G if $d_H(v) \equiv d_G(v) \pmod 2$ for every vertex v of G. A parity subgraph H is called *smallest* if the number of edges in H is the minimum. Theorem 1 is a corollary of the following main theorem of the paper.

THEOREM 2. *Let $G = (V, E)$ be a simple graph with the minimum odd degree δ_0. Then the number of edges in a smallest parity subgraph of G is at most* $|V| - \text{Min}\{\delta_0, |V| - (|V_0|/2)\}$.

In some sense, Theorem 2 is the best possible one. A complete graph K_{2n} has $|V_0| = 2n$ and the smallest parity subgraph of K_{2n} is a perfect matching consisting of n edges. A complete bipartite graph $K_{3,n}$ with $n \geqq 3$ is two-connected and $\delta_0 = 3$. The smallest parity subgraph of $K_{3,n}$ has three components and consists of n edges.

A closed trail covering all edges of G is called a *postman tour* of G. The *Chinese postman problem* (CPP) is to find the shortest postman tour of G (see [EJ] or [BM]). The follow proposition is obvious.

PROPOSITION. *The optimum solution of* CPP *of G is* $|E(G)| + |E(H)|$, *where H is a smallest parity subgraph of G.*

The CPP is solvable by a polynomial algorithm [EJ], whereas the SCC might be an NP-complete problem. The relation between the CPP and the SCC have been studied by many mathematicians ([BJJ], [GF], [IR], [AZ], [AGZ], [Z1], etc.). Obviously the length of an optimum solution of the CPP is not greater than the total length of a solution of the SCC. We say that the CPP is equivalent to the SCC (CPP = SCC) for a graph G if the length of an optimum solution of the CPP is equal to the total length of a solution of the SCC (see [Z1] or [Z2]). The Petersen graph is an example for which the CPP is not equivalent to the SCC (see [IR], [S], [GF], or [AZ]). The following theorems have been found.

THEOREM F [AGZ]. *If a two-edge-connected graph G contains no subdivision of the Petersen graph, then* SCC = CPP.

THEOREM G [Z1], [J]. *If a graph G admits a nowhere-zero 4-flow, then* SCC = CPP.

With Theorems F and G and the proposition, Theorem 1 and the following theorem become corollaries of Theorem 2.

THEOREM 3. *Let $G = (V, E)$ be a two-edge-connected simple graph such that G contains no subdivision of the Petersen graph and minimum odd degree is δ_0. Then the total length of an* SCC *of G is at most* $|E| + |V| - \text{Min}\{\delta_0, |V| - (|V_0|/2)\}$.

2. Proof of Theorem 2.

Because the symmetric difference of a parity subgraph and a cycle is still a parity subgraph, we have the following lemma.

LEMMA. *If H is a smallest parity subgraph of a graph G, then, for any cycle C of G, we must have that*

$$|E(H) \cap E(C)| \leqq |E(C) \setminus E(H)|.$$

Proof of Theorem 2. Let H be the smallest parity subgraph of G. Because H is smallest, by the lemma, H is acyclic. That is, each component of H is a tree. Because G

is simple, by the lemma again, each edge of $E(G)\backslash E(H)$ joins a pair of vertices of distinct components of H.

We only need to show that

$$c(H) \geqq \operatorname{Min}\left\{\delta_0,\ |V| - \frac{|V_0|}{2}\right\}.$$

Denote the set of endvertices (degree 1 vertices) of a component T of H by $L(T)$. Note that each endvertex of a component T of H is an odd vertex of G. If each component T of H is of order 1 or 2, then the number of components of order 1 is $|V| - |V_0|$ and the number of components of order 2 is $|V_0|/2$. Hence

$$c(H) = |V| - |V_0| + \frac{|V_0|}{2} = |V| - \frac{|V_0|}{2}.$$

So we assume that there is a component T of H such that $|V(T)| \geqq 3$.

Let T be a component of H with $|V(T)| \geqq 3$ and $x_1, x_2 \in L(T)$. Assume that

$$|N(x_1) \cap V(T_\mu)| + |N(x_2) \cap V(T_\mu)| \leqq 2$$

for each component T_μ of H. Then

$$d(x_1) + d(x_2) \leqq 2c(H).$$

Because $d(x_1)$ and $d(x_2) \geqq \delta_0$, we have that

$$c(H) \geqq \delta_0.$$

We now assume that if T is a component of H with $|V(T)| \geqq 3$ and $x_1, x_2 \in L(T)$, then there is a component T^* of H such that

$$|N(x_1) \cap V(T^*)| + |N(x_2)| \cap V(T^*)| \geqq 3.$$

We claim that either

$$N(x_1) \cap V(T^*) = \varnothing$$

or

$$N(x_2) \cap V(T^*) = \varnothing.$$

Suppose that

$$N(x_1) \cap V(T^*) \neq \varnothing \qquad \text{and} \qquad N(x_2) \cap V(T^*) \neq \varnothing.$$

Because

$$|N(x_1) \cap V(T^*)| + |N(x_2) \cap V(T^*)| \geqq 3,$$

let $y \in N(x_1) \cap V(T^*)$ and $y' \in N(x_2) \cap V(T^*)$ such that $y \neq y'$. Let Q' be the path in T^* joining y and y'. Let Q be the path in T joining x_1 and x_2. Because the distance between x_1 and x_2 is at least 2 in T, $|E(Q)| \geqq 2$. Then the cycle $x_1Qx_2y'Q'yx_1$ contains at least three edges of H and two edges of $G\backslash E(H)$, this contradicts the lemma and proves our claim.

Because $|N(x_1) \cap V(T^*)| + |N(x_2) \cap V(T^*)| \geqq 3$ and either $N(x_1) \cap V(T^*) = \varnothing$ or $N(x_2) \cap V(T^*) = \varnothing$, we must have that either

$$|N(x_1) \cap V(T^*)| \geqq 3$$

or

$$|N(x_2) \cap V(T^*)| \geqq 3.$$

In summary, for each component T of H with $|V(T)| \geqq 3$, there must be an endvertex x of T and another component T^* of H with $|V(T^*)| \geqq 3$ such that

$$|N(x) \cap V(T^*)| \geqq 3.$$

Construct a directed graph D such that $V(D)$ is the set of all components of H of the order at least 3 and a vertex T_i dominates another vertex T_j in D if $|N(x) \cap V(T_j)| \geqq 3$ for some $x \in L(T_i)$. Because the outdegree $d_D^+(T_\mu) \geqq 1$ for each $T_\mu \in V(D)$, D contains a cycle $C = T_0 \cdots T_{r-1}T_0$. For each T_i, let x^i be an endvertex of T_i with $|N(x^i) \cap V(T_{i+1})| \geqq 3$ (mod r). Let P_i be the longest path in T_{i+1} joining x^{i+1} and $N(x^i) \cap V(T_{i+1})$ (mod r). Because $|N(x^i) \cap V(T_{i+1})| \geqq 3$ for each i (mod r) and x^{i+1} is an endvertex of T_{i+1}, the length of each P_i is at least 2. Therefore the cycle

$$x^0 P_0 x^1 P_1 x^2 \cdots P_{r-1} x^0$$

contains at least $2r$ edges of H and r edges of $G \backslash E(H)$. This contradicts the lemma and completes the proof of the theorem. □

REFERENCES

[AGZ] B. Alspach, L. Goddyn, and C. Q. Zhang, *Graphs with the circuit cover property*, Trans. Amer. Math. Soc., to appear.

[AT] N. Alon and M. Tarsi, *Covering multigraphs by simple circuits*, SIAM J. Alg. Discrete Math., 6 (1985), pp. 345–350.

[AZ] B. Alspach and C. Q. Zhang, *Cycle coverings of cubic multigraphs*, Discrete Math., to appear.

[BJJ] C. Bermond, B. Jackson, and F. Jaeger, *Shortest covering of graphs with cycles*, J. Combin. Theory Ser. B, 35 (1983), pp. 297–308.

[BM] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, Macmillan, London/Elsevier, New York, 1976.

[EJ] J. Edmonds and J. Johnson, *Matching, Euler tours and the Chinese postman*, Math. Programming, 5 (1973), pp. 88–124.

[F1] G. Fan, *Integer flows and cycle covers*, J. Combin. Theory Ser. B, 54 (1992), pp. 113–122.

[F2] ———, *Tutte's 3-flow conjecture and short cycle covers*, J. Combin. Theory Ser. B, 57 (1993), pp. 36–43.

[FP] P. Fraisse, *Cycle covering in bridgeless graphs*, J. Combin. Theory Ser. B, 39 (1985), pp. 146–152.

[GF] M. Guan and H. Fleischner, *On the minimum weighted cycle covering problem for planar graphs*, Ars Combin., 20 (1985), pp. 61–68.

[IL] A. Itai, R. J. Lipto, C. H. Papadimitriou, and M. Rodeh, *Covering graphs with simple circuits*, SIAM J. Comput., 10 (1981), pp. 746–750.

[IR] A. Itai and M. Rodeh, *Covering a graph by circuits*, Automata, Languages and Programming, Lecture Notes in Comput. Sci., 62 (1978), pp. 289–299.

[J] B. Jackson, *Shortest circuit covers and postman tours in graphs with a nowhere zero 4-flow*, SIAM J. Comput., 19 (1990), pp. 659–665.

[R] A. Raspaud, *Cycle covers of graphs with a nowhere-zero 4-flow*, preprint.

[S] P. D. Seymour, *Sums of circuits*, in Graph Theory and Related Topics, J. A. Bondy and U. S. R. Murty, eds., Academic Press, New York, 1979, pp. 341–355.

[Y] D. H. Younger, *Integer flows*, J. Graph Theory, 7 (1983), pp. 349–357.

[Z1] C. Q. Zhang, *Minimum cycle coverings and integer flows*, J. Graph Theory, 14 (1990), pp. 537–546.

[Z2] ———, *Cycle covers and cycle decompositions of graphs*, Ann. Discrete Math., 55 (1993), pp. 183–190.

[ZC] C. Zhao, *Smallest (1, 2)-eulerian weight and shortest cycle covering*, J. Graph Theory, to appear.

SIAM J. DISC. MATH.
Vol. 6, No. 3, pp. 432–442, August 1993

COMPLEXITY RESULTS FOR POMSET LANGUAGES*

JOAN FEIGENBAUM[†], JEREMY A. KAHN[‡], AND CARSTEN LUND[§]

Abstract. Pratt [*Internat. J. Parallel Programming*, 15 (1986), pp. 33–71] introduced POMSETs (partially ordered multisets) to describe and analyze concurrent systems. A POMSET P gives a set of temporal constraints that any correct execution of a given concurrent system must satisfy. Let $L(P)$ (the *language of* P) denote the set of all system executions that satisfy the constraints given by P. This paper shows the following for finite POMSETs P, Q, and system execution x:

- The POMSET language membership problem (given x and P, is $x \in L(P)$?) is NP-complete.
- The POMSET language containment problem (given P and Q, is $L(P) \subseteq L(Q)$?) is Π_2^p-complete.
- The POMSET language equality problem (given P and Q, is $L(P) = L(Q)$?) is at least as hard as the graph-isomorphism problem.
- The POMSET language size problem (given P, how many x are in $L(P)$?) is span-P-complete.

Key words. computer-aided verification, partial orders, POMSETs

AMS(MOS) subject classifications. 68Q15, 68Q60

1. Introduction. Verification of concurrent systems has been studied as a formal language-containment problem for a number of years [1], [15], [6]. In this formulation, we are given a model M represented by a finite transition structure such as a finite state machine, automaton, or Petri net (sometimes termed an *implementation*), together with an abstraction A of the model, represented by an automaton or logic formula (sometimes termed a *specification*, defining a property to be proved about the model M). The verification problem consists of testing whether $L(M) \subseteq L(A)$, where $L(X)$ is the formal language associated with X. Typically, M is large and therefore defined implicitly in terms of components. An inherent difficulty in this approach is the computational complexity of the language containment test as a function of the size of the representation of M in terms of components. For example, if M is defined in terms of coordinating state machines, then the size of M grows geometrically with the number of components defining it, and the language containment problem is PSPACE-complete [7, AL6, p. 266]. This computational complexity issue has been addressed by a number of heuristics, notably homomorphic reduction [11], [10], inductive methods [3], [12], binary decision diagrams [5], [4], [17], and partial orders [8], [14].

In this paper, we consider the language containment problem for POMSETs (partially ordered multisets), which were introduced by Pratt [13]. Both the implementation and the specification of a system can be represented by POMSETs as follows. Let Σ denote a finite set of *actions* that the system can perform. So actions are things like "send 0 to processor p," "receive message m from processor q," and "wait." Each *vertex* v in the POMSET P corresponds to a distinct *event*. Intuitively, an event is a logical "step" taken by the system. The *label* $l(v)$ is an element of Σ, and distinct vertices may have the

[*] Received by the editors October 21, 1991; accepted for publication (in revised form) June 5, 1992. A preliminary version of this paper was presented at the 3rd Annual Workshop on Computer Aided Verification, Aalborg, Denmark, July 1991.

[†] AT&T Bell Laboratories, Room 2C473, 600 Mountain Avenue, P.O. Box 636, Murray Hill, New Jersey 07974-0636. jf@research.att.com

[‡] Mathematics Department, University of California, Berkeley, California 94720. kahn@math.berkeley edu. This work was done while the author was a student at Harvard College and a summer intern at AT&T Bell Laboratories.

[§] AT&T Bell Laboratories, Room 2C324, 600 Mountain Avenue, P.O. Box 636, Murray Hill, New Jersey 07974-0636. lund@research.att.com. This work was done while the author was a postdoctoral fellow at the DIMACS Center at Rutgers University, supported by National Science Foundation grant STC-88-09648.

same label; this corresponds to the fact that a given action (say "send 0 to processor p") may be performed several times by the system during any execution. Each *arc* (v, w) in P represents a *constraint* of the form "event v must occur before event w in any execution of the system." For example, if $l(v)$ is "receive message m from processor p," and $l(w)$ is "if the value of register r is equal to m then signal processor q_1, else signal processor q_2," then the arc (v, w) has the obvious interpretation. The *language* of P is simply the set of all correct executions of the system.

The following example motivates the use of POMSETs. The language $L = \{ab_{i_1}b_{i_2}\dots b_{i_n}a\}$, where $i_1i_2\dots i_n$ is a permutation of $12\dots n$ and all of the b_i's are distinct, arises often in the description of concurrent processes. Its meaning is "perform action a, then perform each of the actions b_1 through b_n in any order, then perform action a again." A nondeterministic finite automaton (NFA) that accepts L must have at least 2^n states. POMSETs, however, offer a much more compact representation: The $(n + 2)$-node POMSET of Fig. 1 represents L.

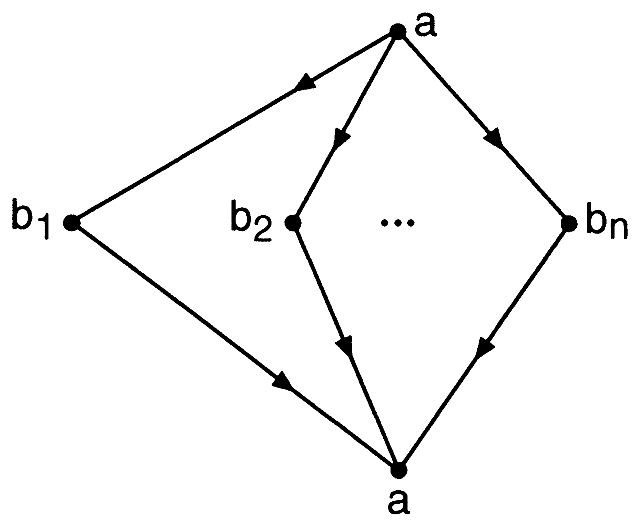


FIG. 1

Formally, the problem of interest is as follows.

POMSET language containment (PLC):
Input: Two POMSETs P and Q.
Question: Is the language of P a subset of the language of Q?

The POMSET P represents the implementation and Q the specification. We show that the PLC problem is Π_2^p-complete.

Note that P and Q are both finite POMSETs. Thus the languages in question are finite, and the strings in them are of finite length. If we were presenting an algorithm for PLC, this finiteness restriction would render the algorithm impractical because real concurrent systems produce infinite sets of infinite sequences. However, we are giving a lower bound on the complexity of PLC, and hence the finiteness restriction makes our result all the more meaningful: Even in this restricted case, the problem appears to be intractable.

We also give an NP-completeness result for the following simpler problem.

POMSET language membership (PLM):
Input: A POMSET P and a string x.
Question: Is x in the language of P?

Once again, the finiteness restriction only strengthens our result, because we are providing a lower bound rather than an algorithm.

The following problem formalizes the question of whether two specifications in fact specify the same system.

POMSET language equality (PLE):
Input: Two POMSETs P and Q.
Question: Is the language of P equal to the language of Q?

It is clear that PLE is in Π_2^p because it can be reduced to PLC. We show that PLE is at least as hard as graph isomorphism. Since graph isomorphism is in NP and is not even believed to be NP-complete, the exact complexity of PLE is still open.

Finally, we consider a problem that is interesting from a purely combinatorial and complexity-theoretic point of view.

POMSET language size (PLS):
Input: A POMSET P.
Question: What is the number of strings in the language of P?

We show that PLS is complete for the class span-P (cf. Köbler, Schöning, and Toran [9]).

2. Definitions and notation. Throughout this paper, P and Q denote (finite) POMSETs, and x denotes a (finite) string. We now fix these ideas precisely.

DEFINITION 2.1. A POMSET P is a triple (V, A, l). The *vertex set* $V(P)$ consists of a finite number n of distinct elements $\{v_1, \dots, v_n\}$ called the *events*. The *arc set* $A(P)$ consists of a set of ordered pairs (v, w), where v and w are distinct elements of V, called the *constraints*. The directed graph $(V(P), A(P))$ is acyclic. The mapping $l : V \rightarrow \Sigma$ assigns an *action* to each event in V, and $l(v)$ is called the *label* of vertex v.

Recall that a linear ordering of $V = \{v_1, \dots, v_n\}$ *extends* a partial ordering of V if, for all pairs v_i, v_j of distinct elements in V, $v_i < v_j$ in the partial ordering implies that $v_i < v_j$ in the linear ordering. Technically, a DAG (directed acyclic graph) may not be a partial ordering because it may not be transitively closed. When we say that a linear ordering of V extends the DAG (V, A), we mean that it extends the transitive closure of the DAG.

DEFINITION 2.2. The *language* $L(P)$ of a POMSET $P = (V, A, l)$ is a subset of Σ^n, where $n = |V(P)|$. The string $\sigma_1 \cdots \sigma_n$ is in $L(P)$ if there is a linear ordering $v_{i_1} \cdots v_{i_n}$ of the vertex set V that extends the DAG (V, A) and satisfies $l(v_{i_j}) = \sigma_j$, for $1 \leqq j \leqq n$.

For our result about the complexity of PLS, we will need the function classes #P (cf. Valiant [18]) and span-P (cf. Köbler, Schöning, and Toran [9]). A function f is in #P if there is a nondeterministic, polynomial-time machine, say M, such that $f(x)$ is equal to the number of accepting computations of M on input x. For example, the function that counts the number of satisfying assignments of a propositional formula is in #P, as is the function that counts the number of Hamiltonian cycles in a graph. A function g is in span-P if there is a nondeterministic, polynomial-time transducer, say N, that behaves as follows. On input x and rejecting computation path w, the output of N is simply "reject." On input x and accepting computation path w, the output is the value of a polynomial-time computable function, say ϕ, of x and w. Then $g(x)$ is the number of different values that ϕ can take on input x, i.e.,

$$g(x) = \#\{z : \exists \text{ an accepting path } w \text{ such that } \phi(x, w) = z\}.$$

3. PLC is Π_2^p-complete.

THEOREM 3.1. *The* PLC *problem is* Π_2^p*-complete.*

Proof. First, note that it is obvious that PLC is in Π_2^p. Suppose that we wish to know whether $L(P)$ is contained in $L(Q)$, where $V(P) = \{v_1, \ldots, v_n\}$ and $V(Q) = \{w_1, \ldots, w_n\}$. The following is a Π_2^p expression for $L(P) \subseteq L(Q)$: For all linear orderings $v_{i_1} \cdots v_{i_n}$, there exists a linear ordering $w_{j_1} \cdots w_{j_n}$ such that, if $v_{i_1} \cdots v_{i_n}$ extends $A(P)$, then $w_{j_1} \cdots w_{j_n}$ extends $A(Q)$ and $l(v_{i_k}) = l(w_{j_k})$ for $1 \leqq k \leqq n$. The hypothesis "if $v_{i_1} \cdots v_{i_n}$ extends $A(P)$" is equivalent to "if $l(v_{i_1}) \cdots l(v_{i_n}) \in L(P)$," and the conclusion "then $w_{j_1} \cdots w_{j_n}$ extends $A(Q)$ and $l(v_{i_k}) = l(w_{j_k})$ for $1 \leqq k \leqq n$" is equivalent to "$l(w_{i_1}) \cdots l(w_{i_n}) \in L(Q)$ and is equal to $l(v_{i_1}) \cdots l(v_{i_n})$."

It is also obvious that PLC is NP-hard, because PLM is the special case of PLC in which $L(P)$ contains just one string, and PLM is NP-complete (see §4 below).

We show completeness by reduction from the following Π_2^p-complete problem (cf. [7, p. 166]).

Normalized B_2^c:

Input: Two sets $\{w_1, \ldots, w_m\}$ and $\{y_1, \ldots, y_n\}$ of boolean variables and a set $\{c_1, \ldots, c_k\}$ of clauses. Each clause is of the form $a \Rightarrow b \vee c \vee d$, where a is either w_i or $\overline{w_i}$ for some i and each of b, c, and d is y_j or $\overline{y_j}$ for some j.

Question: Is it the case that, for every truth assignment to the w_i's, there exists some truth assignment to the y_j's such that every c_l is satisfied?

Given an instance $(W = \{w_1, \ldots, w_m\}, Y = \{y_1, \ldots, y_n\}, C = \{c_1, \ldots, c_k\})$ of normalized B_2^c, we construct an instance (P, Q) of PLC as follows.

In $V(P)$, there are three disjoint sets of vertices. The first group contains n vertices, labeled y_1 through y_n. The second group in $V(P)$ contains $2m + k$ vertices. For $1 \leqq i \leqq m$, there are two vertices in this group labeled w_i; we refer to them as "the positive w_i vertex" and "the negative w_i vertex." For $1 \leqq l \leqq k$, there is one vertex in the second group labeled c_l. The third group of vertices in $V(P)$ is of size $n + 3k$. There is one vertex in this group labeled y_j, for $1 \leqq j \leqq n$, and there are three vertices in the third group labeled c_l, for $1 \leqq l \leqq k$. For every clause c_l in which w_i appears on the left side of the implication, there is an arc in $A(P)$ from the positive w_i vertex to the second-group vertex labeled c_l; for every c_l in which $\overline{w_i}$ appears on the left side of the implication, there is an arc in $A(P)$ from the negative w_i vertex to the second-group vertex labeled c_l. Every w vertex in the second group is joined by an arc to every c vertex in the third group. The rest of the arcs that make up $A(P)$ can be seen in Fig. 2, where an example of this construction is given. The subscripts are omitted from the labels of some clause vertices to reduce clutter.

In $V(Q)$, there are two vertices labeled y_j, for $1 \leqq j \leqq n$, and two vertices labeled w_i, for $1 \leqq i \leqq m$. These are referred to as "the positive y_j (respectively, w_i) vertex" and "the negative y_j (respectively, w_i) vertex." $V(Q)$ also contains four vertices labeled c_l, for $1 \leqq l \leqq k$. One group of these c vertices is associated with the y vertices; each c vertex in this group has in-degree 1. For each clause c_l in which the literal y_j appears on the right side of the implication, there is an arc from the positive y_j vertex to a c_l vertex. Similarly, for each clause c_l in which the literal $\overline{y_j}$ appears on the right side of the implication, there is an arc from the negative y_j vertex to a c_l vertex. Note that each label c_l appears three times in this group, once for each literal in the clause. The second group of c vertices is associated with the w vertices; each c vertex in this group has in-degree 2. If w_i or $\overline{w_i}$ appears on the left side of the implication in clause c_l, then there are arcs from both the positive w_i vertex and the negative w_i vertex to the c_l vertex in

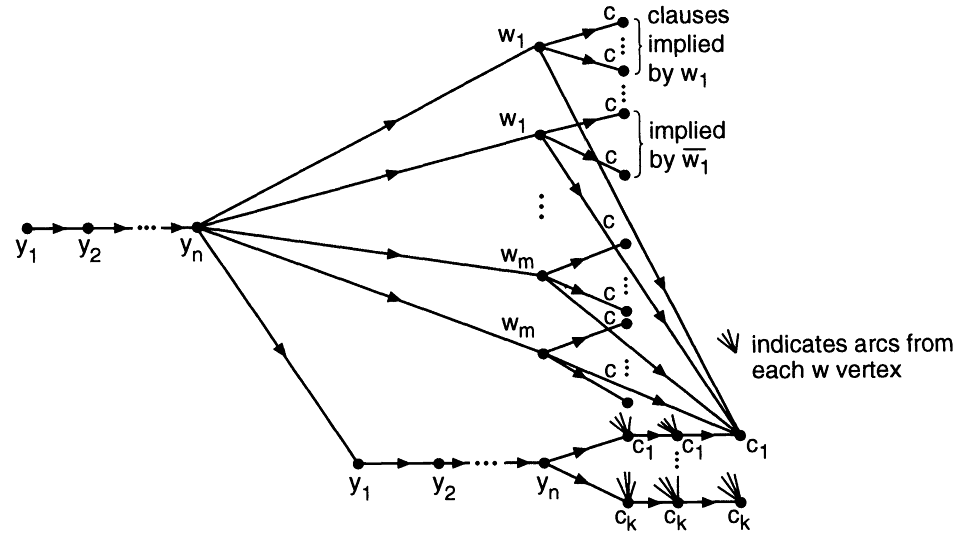


FIG. 2

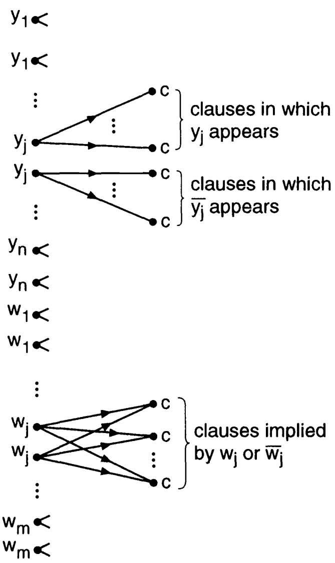

y1
y1
yj
yj
c
c
clauses in which yj appears
c
c
clauses in which ȳj appears
yn
yn
w1
w1
wj
wj
c
c
c
clauses implied by wj or w̄j
wm
wm

FIG. 3

the second group. See Fig. 3 for an example of this construction. Once again, subscripts are omitted from some clause vertices to reduce clutter.

Suppose that (P, Q) is a yes-instance of PLC; so $L(P)$ is contained in $L(Q)$. We must show that (W, Y, C) is a yes-instance of B_2^c. Choose an assignment of truth values to the variables in W. We will construct an assignment of truth values to the variables in Y that, together with the initial assignment to those in W, satisfies all the clauses in C.

Consider the string

$$x = y_1 \cdots y_n w_1 \cdots w_m c_{q_1} \cdots c_{q_t} y_1 \cdots y_n w_1 \cdots w_m c_{q_{t+1}} \cdots c_{4k}$$

in $L(P)$ that is formed as follows. The prefix $y_1 \cdots y_n$ comes from the first group of vertices in $V(P)$. In the first substring $w_1 \cdots w_m$, each w_i represents a choice between the positive w_i vertex and the negative w_i vertex within the second group in $V(P)$. The substring $c_{q_1} \cdots c_{q_t}$ corresponds exactly to the clauses that are nontrivial to satisfy: If a clause vertex v in the second group in $V(P)$ is adjacent to the positive w_i vertex and w_i is TRUE in the initial assignment, then $l(v)$ goes into the substring $c_{q_1} \cdots c_{q_t}$; similarly, if v is adjacent to the negative w_i vertex and w_i is FALSE in the initial assignment, then $l(v)$ goes into the substring $c_{q_1} \cdots c_{q_t}$. The rest of the string x is constructed in any way that is consistent with the constraints in $A(P)$, subject to y's, then w's, then c's.

Note that x is always in $L(P)$. Because (P, Q) is assumed to be a yes-instance of PLC, x is also in $L(Q)$. Consider the vertices $v(c_{q_1}), \ldots, v(c_{q_t})$ in $V(Q)$ that give rise to the substring $c_{q_1} \cdots c_{q_t}$ of x. These vertices must all be in the first group of c vertices in Q; that is, they must be in the group whose incoming arcs start with y's. This is because none of $c_{q_1}, \ldots, c_{q_t}$ is preceded in x by two occurrences of w_i, for any i. If $v(c_{q_l})$ is connected to the positive (respectively, negative) y_j vertex, then assign the variable y_j the value TRUE (respectively, FALSE). Assign arbitrary values to any remaining y variables. Note that no conflicts arise in making this assignment; that is, each y_j is assigned one value. This is because each y_j symbol appears once in the prefix of x, and hence only one of the two y_j vertices is used; if the y_j vertex that is used is adjacent to two vertices $v(c_{q_{l_1}})$ and $v(c_{q_{l_2}})$, then either y_j appears in both $c_{q_{l_1}}$ and $c_{q_{l_2}}$ or $\overline{y_j}$ appears in both $c_{q_{l_1}}$ and $c_{q_{l_2}}$. This assignment, together with the initial assignment to the w variables, satisfies all of the clauses in C. Because the initial assignment to the w variables was arbitrary, this shows that (W, Y, C) is a yes-instance.

Now suppose that (W, Y, C) is a yes-instance of normalized B_2^c. Let x be an arbitrary element of $L(P)$ in the corresponding instance of PLC. We must show that x is also in $L(Q)$.

We construct a truth assignment that corresponds to x as follows. Each symbol in x comes from a vertex in a linear ordering of $V(P)$ that extends $A(P)$. Take the first occurrence of w_i in x and see whether it corresponds to the positive w_i vertex or the negative w_i vertex. If positive, assign the variable w_i the value TRUE and, if negative, assign it FALSE. Because (W, Y, C) is a yes-instance, there must be an assignment of truth values to the y variables that, together with the assignment to the w's, satisfies every clause in C. This assignment to the y's corresponds to the prefix $y_1 \cdots y_n$ of x in a way that will become clear below. Denote by A the full assignment to y's and w's.

Call a y vertex or w vertex in $V(Q)$ "active" if it corresponds to the truth assignment A; e.g., the positive y_j vertex is active if and only if the variable y_j is TRUE in A. Now Q is the disjoint union of subPOMSETs Q_1 and Q_2, where Q_1 contains exactly the active y vertices and the c vertices that are connected by arcs from active y vertices, and Q_2 contains exactly the active w vertices and the c vertices that are connected by arcs from active w vertices.

The only nontrivial task involved in finding a linear ordering of $V(Q)$ that extends $A(Q)$ and gives rise to x is the following: Suppose that clause c_l contains the variable w_i and that the first occurrence of the symbol c_l in x falls between the two occurrences of the symbol w_i; what is the vertex in $V(Q)$ that gives rise to this first occurrence of c_l? By construction, this vertex can be found in $V(Q_1)$; that is, the active y vertices correspond to the prefix $y_1 \cdots y_n$ of x. Thus x is in the shuffle of $L(Q_1)$ and $L(Q_2)$, which is $L(Q)$. □

There are some special cases of PLC that are easily solved in polynomial time. For example, if each element of Σ occurs at most once as a label in each POMSET, then there is at most one bijection ϕ from $V(Q)$ to $V(P)$, given by the labels. If no such ϕ exists, $L(P) \not\subseteq L(Q)$. Otherwise, let $T(P)$ (respectively, $T(Q)$) be the transitive closure of $A(P)$ (respectively, $A(Q)$). It is easily seen that $L(P)$ is contained in $L(Q)$ if and only if, for every arc (v, w) in $T(Q)$, the arc $(\phi(v), \phi(w))$ is in $T(P)$. We call this the *unique-label case* of PLC.

Similarly, the *no-autoconcurrence case* of PLC is solvable in polynomial time. "No autoconcurrence" means that, if v and w are in $V(P)$ (respectively, $V(Q)$), and $l(v) = l(w)$, then either (v, w) or (w, v) is in $A(P)$ (respectively, $A(Q)$). The no-autoconcurrence case can be reduced to the unique-label case as follows: For each $a \in \Sigma$, let $v_1, \ldots, v_m$ be all of the vertices of POMSET P with label a. These vertices must be linearly ordered in $A(P)$, else there would be autoconcurrence. If the linear order is $v_{i1} < \cdots < v_{im}$, then relabel these vertices $l(v_{i1}) = a_{i1}, \ldots, l(v_{im}) = a_{im}$, where the a_{ij}'s are not in Σ. Do the same for all of the vertices with label a in Q, once again using the labels $a_{i1}, \ldots, a_{im}$.

4. PLM is NP-complete. The following theorem was obtained in collaboration with J. Kilian.

THEOREM 4.1. *The* PLM *problem is* NP-*complete.*

Proof. Once again, it is obvious that PLM is in NP. To verify that $x = \sigma_1 \cdots \sigma_n$ is in $P = (V, A)$, where $V = \{v_1, \ldots, v_n\}$, simply guess a linear ordering $v_{i_1} \ldots v_{i_n}$ of V and check that each arc in A joins a pair of vertices $v_{i_{j_1}}, v_{i_{j_2}}$ with $j_1 < j_2$ and that $l(v_i) = \sigma_i$ for each i.

We show completeness by reduction from the archetypal NP-complete problem 3SAT. Recall the statement of this problem.

Three satisfiability (3SAT):

Input: Clauses $c_1, \ldots, c_n$ on boolean variables $y_1, \ldots, y_m$. Each c_j is of the form $c_{j_1} \vee c_{j_2} \vee cj3$, where each c_{j_k} is either y_i or $\overline{y_i}$ for some i.

Question: Is there an assignment of truth values to the variables $y_1, \ldots, y_m$ that satisfies all of the clauses $c_1, \ldots, c_n$ simultaneously?

Given an instance $(C = \{c_1, \ldots, c_n\}, Y = \{y_1, \ldots, y_m\})$ of 3SAT, we construct an equivalent instance (x, P) of PLM as follows. The vertex set V of P contains two vertices, say v_{i_1} and v_{i_2}, for each variable y_i and three vertices, say w_{j_1}, w_{j_2}, and w_{j_3}, for each clause c_j. Vertices v_{i_1} and v_{i_2} have label y_i, and vertices w_{j_1}, w_{j_2}, and w_{j_3} all have label c_j. For each clause c_j, consider the variables (say y_r, y_s, and y_t) that occur in c_j. Put in exactly one of arcs (v_{r_1}, w_{j_1}) and (v_{r_2}, w_{j_1}) (respectively, $[(v_{s_1}, w_{j_2})$ and $(v_{s_2}, w_{j_2})]$ and $[(v_{t_1}, w_{j_3})$ and $(v_{t_2}, w_{j_3})]$), by choosing the first if y_r (respectively, y_s and y_t) occurs in c_j and the second if $\overline{y_r}$ (respectively, $\overline{y_s}$ and $\overline{y_t}$) occurs in c_j. The string in the PLM instance is

$$x = y_1 \cdots y_m c_1 \cdots c_n y_1 \cdots y_m c_1 c_1 c_2 c_2 \cdots c_n c_n.$$

See Fig. 4 for an example of this construction.

It is easily seen that (x, P) is a yes-instance of PLM if and only if (C, A) is a yes-instance of 3SAT. The key point is that the choice of vertices that map to the prefix $y_1 \cdots y_m$ of x corresponds exactly to the choice of truth values in the satisfying assignment and that this choice "covers" the first occurrence of each c_j symbol in x. □

Next, we show that a special case of PLM is solvable in polynomial time.

THEOREM 4.2. *There is a polynomial-time algorithm for the special case of* PLM *in which each label in* Σ *occurs at most twice in* x.

Proof. We exhibit a polynomial-time reduction from this case of PLM to 2SAT, which is solvable in polynomial time [7, §3.1.1]. The formulation of the 2SAT problem is iden-

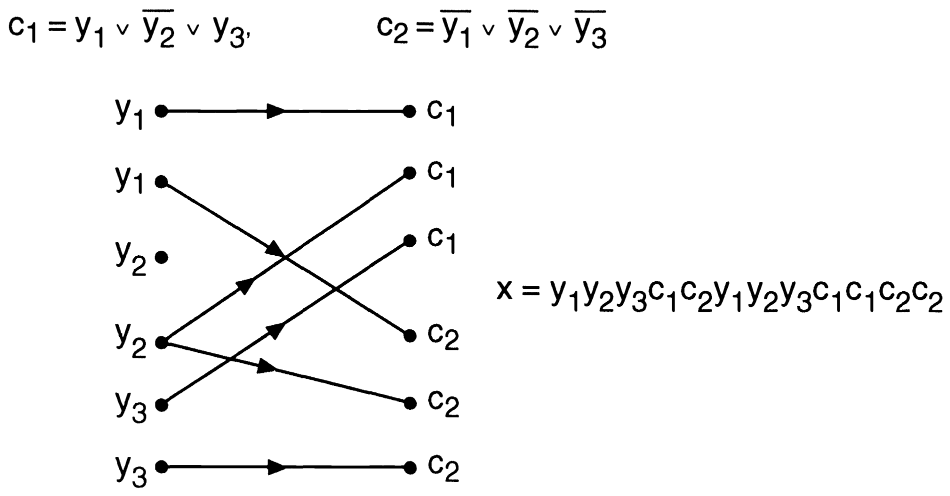

$c_1 = y_1 \vee \overline{y_2} \vee y_3$,
$c_2 = \overline{y_1} \vee \overline{y_2} \vee \overline{y_3}$
y_1
y_1
y_2
y_2
y_3
y_3
c_1
c_1
c_1
c_2
c_2
c_2
$x = y_1y_2y_3c_1c_2y_1y_2y_3c_1c_1c_2c_2$

FIG. 4

tical to that of the 3SAT problem (see above), except that each clause has two literals instead of three.

Let (x, P) be an instance of PLM in which each label occurs at most twice. We will construct a formula φ such that

$$x \in L(P) \quad \text{if and only if } \varphi \in \text{2SAT}.$$

Assume that $V(P) = \{1, 2, \ldots, n\}$. Each position i, $1 \leq i \leq |x|$ gives rise to a boolean variable z_i in φ. In what follows, we will use $<$ and $>$ to denote integer inequalities and $<_P$ and $>_P$ to denote inequalities in the partial order of the POMSET.

Let π be a linear ordering of $V(P)$ that may or may not be an extension of P. Assume that the word corresponding to π is x. Any such π corresponds to a truth assignment of the z_i's in the following way. If the label in the ith position of x only occurs once, then z_i is TRUE. If the label occurs twice, then observe that only two nodes, say n_1 and n_2 with $n_1 < n_2$, can correspond to position i. If π chooses n_1 to correspond to position i, then z_i is TRUE and otherwise it is FALSE.

The clauses of φ are of the following three types. The two first types ensure that any satisfying assignment of the variables corresponds to a linear ordering π of $V(P)$. The third type ensures that π extends P.

1. Suppose that label a occurs exactly once in x, in position i. We force z_i to be TRUE by adding the clause "z_i" to φ.
2. If label a occurs in positions i and j, there are two corresponding variables z_i and z_j. To ensure consistency, we add the clauses "$z_i \vee z_j$" and "$\overline{z_i} \vee \overline{z_j}$" to φ.
3. For each pair of positions $1 \leq i < j \leq n$ in the string x, we construct some clauses. The clauses will contain only the variables z_i and z_j. Each assignment b_1, b_2 of z_i and z_j determines two nodes n_i and n_j, by interpretation of the boolean variables. If $n_i >_P n_j$, then we add the clause "$(b_1 \oplus z_i) \vee (b_2 \oplus z_j)$," where $\oplus$ is exclusive or. These clauses ensure that all satisfying assignments correspond to linear orderings π in which the ith element follows the jth element in the $>_P$ ordering.

It is easy to see that φ is satisfiable if and only if (x, P) is a yes-instance of PLM and that the reduction can be done in polynomial time. □

Note that the PLM instances that are constructed in the proof of Theorem 4.1 have at most three occurrences of each label. Thus the (presumed) jump in complexity from polynomial-time solvability to NP-completeness occurs when the maximum number of occurrences of a label is increased from two to three.

5. PLE is as hard as graph isomorphism.

THEOREM 5.1. *The* PLE *problem is as hard as graph isomorphism.*

Proof. Let (G, H) be an instance of the graph isomorphism problem. We show how to construct, in polynomial time, a POMSET $P(G, H)$ and a string x such that

(A) For any permutation π of $V(G)$, $L(P(\pi(G), H)) = L(P(G, H))$, and

(B) The string x is in $L(P(G, H))$ if and only if $G \cong H$.

Statements (A) and (B) together imply that $L(P(H, H)) = L(P(G, H))$ if and only if $G \cong H$, and hence this construction is tantamount to a polynomial-time reduction from graph isomorphism to PLE.

Let $V(G) = \{v_1, \ldots, v_n\}$ and $V(H) = \{w_1, \ldots, w_n\}$. Then $V(P)$ consists of

1. $N(v_i, w_k)$, where $1 \leq i, k \leq n$. The label of $N(v_i, w_k)$ is w_k;
2. $N'(v_i, v_j, w_k, w_l)$, where $1 \leq i < j \leq n$ and $1 \leq k, l \leq n$. The label of $N'(v_i, v_j, w_k, w_l)$ is a;
3. $N''(v_i, v_j, w_k, w_l)$, where $\{v_i, v_j\}$ and $\{w_k, w_l\}$ are in $E(G)$ and $E(H)$, respectively. The label of $N''(v_i, v_j, w_k, w_l)$ is $b_{k,l}$.

The constraint set $A(P)$ consists of all possible arcs of the form $(N(v_i, w_k), N'(v_i, v_j, w_k, w_l))$ and $(N(v_j, w_l), N'(v_i, v_j, w_k, w_l))$ and all possible arcs of the form $(N(v_i, w_k), N''(v_i, v_j, w_k, w_l))$ and $(N(v_j, w_l), N''(v_i, v_j, w_k, w_l))$.

The string x is

$$w_1 \cdots w_n a^{\binom{n}{2}} b_{k_1,l_1} \cdots b_{k_m,l_m} w^* a^* b^*,$$

where $E(H) = \{\{w_{k_1}, w_{l_1}\}, \ldots, \{w_{k_m,l_m}\}\}$ and $w^* a^* b^*$ means simply "the right number of $w'_{k,l}$, a's, and $b_{k,l}$'s." Refer to Fig. 5 for an example of this construction.

It is easily seen that $P(G, H)$ satisfies property (A). For property (B), the intuition is as follows. Each symbol in the prefix $w_1 \cdots w_n$ of x corresponds to a vertex in H, which in turn corresponds to a Type-1 vertex in P. If $N(v_i, w_k)$ and $N(v_j, w_l)$ are two of these chosen vertices, then $w_k \neq w_l$. Furthermore, $v_i \neq v_j$, else we would not have satisfied enough constraints on Type-2 vertices to be able to put the substring $a^{\binom{n}{2}}$ into x. So the prefix $w_1 \cdots w_n$ actually gives us a bijection between $V(G)$ and $V(H)$. Finally, this bijection must be an isomorphism, else we would not have satisfied enough constraints on Type-3 vertices to be able to put the substring $b_{k_1,l_1} \cdots b_{k_m,l_m}$ into x. □

6. PLS is span-P-complete.

THEOREM 6.1. *The* PLS *problem is* span-P-*complete.*

Proof. Brightwell and Winkler [2] have shown that the following problem is #P-complete: Given a partial order, how many linear extensions does it have? Because PLS contains Brightwell and Winkler's problem as a special case (i.e., the case in which each label appears at most once), it is clear that PLS is #P-hard.

It is also clear that PLS is in span-P. The underlying nondeterministic, polynomial-time machine takes a POMSET P as input. The paths of the machine correspond to linear orderings of $V(P)$, and the accepting paths correspond to linear orderings

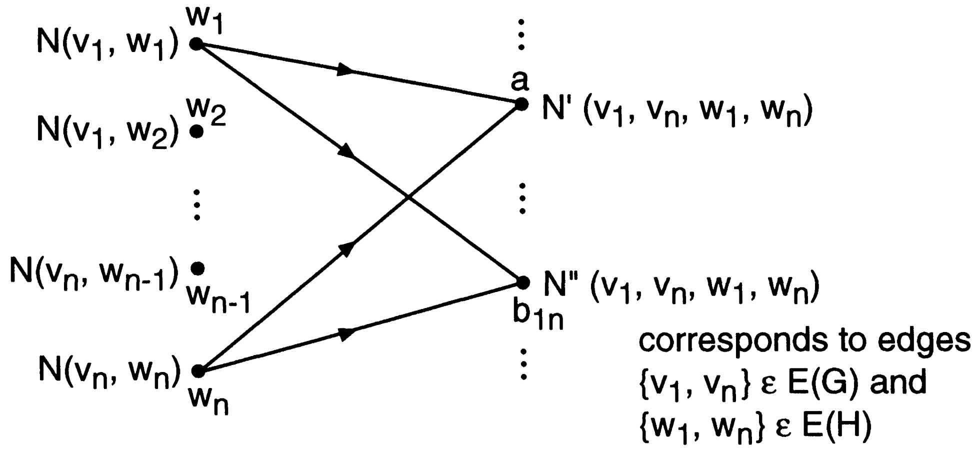


FIG. 5

that extend the DAG $(V(P), A(P))$. The output of the accepting path corresponding to $v_{i1} \cdots v_{in}$ is the string $l(v_{i1}) \cdots l(v_{in})$ in $L(P)$.

Thus, to prove Theorem 6.1, it suffices to show that span-P is contained in the function class $\mathrm{FP}^{\#P}$. (FP denotes the class of polynomial-time computable *functions*; this is a generalization of P, the class of polynomial-time decidable *sets*.) This is straightforward: The containment span-P $\subseteq \#\mathrm{P}^{NP}$ follows easily from the definitions of span-P and #P, and the containment $\#\mathrm{P}^{NP} \subseteq \mathrm{FP}^{\#P}$ is a special case of the main result of Toda and Watanabe [16]. □

Toda and Watanabe's theorem yields the stronger statement $\mathrm{FP}^{\#P} = \mathrm{FP}^{\text{span-P}}$, but we do not need this full generality to prove that PLS is span-P-complete.

7. Discussion. A natural next step to take is to identify interesting special cases of PLC and to develop algorithms for these cases. For these algorithms to be practical, they would have to test containment of infinite languages of infinite sequences. It is unclear how to represent such languages by POMSETs so as to facilitate language-containment testing. Some candidate representations are suggested in Pratt's original paper and in Probst and Li [14].

We propose the following notation. Each language is represented by a deterministic Büchi automaton A and a collection $P_1, \ldots, P_k$ of POMSETs. Assume that each P_i exhibits no autoconcurrence. Each transition of A is labeled by a POMSET P_i. The language given by $(A, P_1, \ldots, P_k)$ consists of all sequences $w_{i_1}\ w_{i_2} \cdots$, where $P_{i_1}\ P_{i_2} \cdots$ is in $L(A)$ and w_{i_j} is in $L(P_{i_j})$.

Suppose that $(A, P_1, \ldots, P_k)$ and $(B, Q_1, \ldots, Q_k)$ are two such representations. Note that an implicit one-to-one correspondence between the two collections of POMSETs is given by their subscripts. Form an automaton B' by starting with B and substituting for each transition label Q_i the corresponding label P_i. Then a sufficient, but not necessary, condition for the language given by $(A, P_1, \ldots, P_k)$ to be contained in the language given by $(B, Q_1, \ldots, Q_k)$ is: $L(A) \subseteq L(B')$ and, for each i, $L(P_i) \subseteq L(Q_i)$.

This test can be performed in polynomial time. We hope to investigate its applicability in future work.

Finally, there is a large gap between the known upper and lower bounds for PLE, and we would like to close it.

REFERENCES

[1] S. Aggarwal, R. P. Kurshan, and K. K. Sabnani, *A calculus for protocol specification and validation*, in Proc. 3rd Sympos. on Protocol Specification, Testing, and Verification, North-Holland, Amsterdam, 1983, pp. 19–34.

[2] G. Brightwell and P. Winkler, *Counting linear extensions*, Order, 8 (1991), pp. 225–242.

[3] M. C. Brown, E. M. Clarke, and O. Grumberg, *Reasoning about networks with many identical finite state processes*, Inform. Comput., 81 (1989), pp. 13–31.

[4] J. R. Burch, E. M. Clarke, K. L. McMillan, D. L. Dill, and J. Hwang, *Symbolic model checking*: 10^{20} *states and beyond*, Inform. Comput., 98 (1992), pp. 142–170.

[5] O. Coudert, C. Berthet, and J. C. Madre, *Verification of synchronous sequential machines based on symbolic execution*, in Automatic Verification of Finite State Systems, Lecture Notes in Computer Science, Vol. 407, Springer, Berlin, 1989, pp. 365–373.

[6] D. L. Dill, *Timing assumptions and verification of finite-state concurrent systems*, in Automatic Verification of Finite State Systems, Lecture Notes in Computer Science, Vol. 407, Springer, Berlin, 1989, pp. 197–212.

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

[8] P. Godefroid, *Using partial orders to improve automatic verification methods*, in Computer-Aided Verification 90, DIMACS Series on Discrete Mathematics and Theoretical Computer Science, Vol. 3, American Mathematical Society, Providence, RI, 1991, pp. 321–340.

[9] J. Köbler, U. Schöning, and J. Toran, *On counting and approximation*, Acta Inform., 26 (1989), pp. 363–379.

[10] R. P. Kurshan, *Analysis of discrete event coordination*, in Stepwise Refinement of Distributed Systems, Lecture Notes in Computer Science, Vol. 430, Springer, Berlin, 1990, pp. 414–453.

[11] ———, *Reducibility in Analysis of Coordination*, Lecture Notes in Control and Information Sciences, Vol. 103, Springer, Berlin, 1987, pp. 19–39.

[12] R. P. Kurshan and K. L. McMillan, *A structural induction theorem for processes*, in Proc. 8th Ann. Sympos. on Principles of Distributed Computing, ACM, New York, 1989, pp. 239–247.

[13] V. Pratt, *Modelling concurrency with partial orders*, Internat. J. Parallel Programming, 15 (1986), pp. 33–71.

[14] D. Probst and H. Li, *Using partial order semantics to avoid the state explosion problem in asynchronous systems*, in Computer-Aided Verification 90, DIMACS Series on Discrete Mathematics and Theoretical Computer Science, Vol. 3, American Mathematical Society, Providence, RI, 1991, pp. 15–24.

[15] A. P. Sistla, M. Y. Vardi, and P. Wolper, *The complementation problem for Buchi automata with applications to temporal logic*, Theoret. Comput. Sci., 49 (1987), pp. 217–237.

[16] S. Toda and O. Watanabe, *Polynomial time* 1*-turing reductions from* #PH *to* #P, Theoret. Comput. Sci., 100 (1992), pp. 205–221.

[17] H. J. Touati, R. K. Brayton, and R. P. Kurshan, *Testing language containment for* ω*-automata using* BDD's, in Formal Methods in VLSI Design, ACM, New York, to appear.

[18] L. Valiant, *The complexity of computing the permanent*, Theoret. Comput. Sci., 8 (1979), pp. 189–201.

SIAM J. DISC. MATH.
Vol. 6, No. 3, pp. 443–459, August 1993

AN EFFICIENT PARALLEL ALGORITHM THAT FINDS INDEPENDENT SETS OF GUARANTEED SIZE*

MARK GOLDBERG[†] AND THOMAS SPENCER[‡]

Abstract. Every graph with n vertices and m edges has an independent set containing at least $n^2/(2m + n)$ vertices. This paper presents a parallel algorithm that finds an independent set of this size and runs in $O(\log^3 n)$ time on a CRCW PRAM with $O((m+n)\alpha(m, n)/\log^2 n)$ processors, where $\alpha(n, m)$ is a functional inverse of Ackerman's function. The ideas used in the design of this algorithm are also used to design an algorithm that, with the same resources, finds a vertex coloring satisfying certain minimality conditions.

Key words. Turán's theorem, independent set, NC, graph, parallel computation, deterministic

AMS(MOS) subject classifications. 68Q22, 68R10, 68R05

1. Introduction. This paper presents a fast parallel algorithm that, given a graph G, finds an independent set of G whose size is bounded from below. The bound depends on the number n of vertices and number m of edges of G and cannot be improved in these terms.

Since constructing a *maximum* independent set is NP-hard, it cannot be done using a polynomial algorithm unless P=NP. Johnson [13] proved that if there is a polynomial-time algorithm that finds an independent set whose size is within a constant factor of optimal, then there is a polynomial approximation scheme for the maximum independent set problem, that is, an algorithm that finds an independent set whose size is within $(1 - \epsilon)$ of optimal and whose running time is polynomial for any fixed $\epsilon > 0$. Nobody has yet devised such a polynomial approximation scheme, and it is somewhat unlikely that it exists. In particular, it seems to be unlikely that this approximation problem is in NC. In contrast to this, if we require that the algorithm only find a *maximal* independent set (the MIS problem), then the problem becomes polynomial. In fact, a maximal independent set can be easily found sequentially in a linear time. Karp and Wigderson [14] showed that MIS is in NC. Starting with their work, a number of parallel algorithms have been proposed to solve this problem [2], [10], [11], [16], [17]. Currently, the most efficient algorithm is presented in [11]; it runs in $O(\log^3 n)$ time on $O((n+m)/\log n)$ processors.

A common drawback of all NC-algorithms for MIS mentioned above is that occasionally they can find *too small* a set. Any graph with a large independent set and a vertex adjacent to all other vertices is a potential example of such a situation. This motivates the approach we take here; we require that the algorithm find a *sufficiently large* independent set. Besides the apparent theoretical interest in the question of whether this task can be accomplished in NC, it is conceivable that such an algorithm can be a useful subroutine for solving other problems. For example, in our earlier work [11], the crucial part of the algorithm for MIS is a subroutine that efficiently finds in $O(\log n)$ time a matching of sufficiently large, though not necessarily maximum, size. Note that finding a maximum matching is not known to be in NC.

*Received by the editors October 19, 1990; accepted for publication (in revised form) August 11, 1992.

[†]Department of Computer Science, Rensselaer Polytechnic Institute, Troy, New York 12180. goldberg@cs.rpi.edu. This author's work was supported in part by National Science Foundation grant CDA-8805910 and by National Security Agency grant MDA904-88-H-2037.

[‡]Department of Mathematics and Computer Science, University of Nebraska at Omaha, Omaha, Nebraska 68182. spencer@unocss.unomaha.edu. This author's work was supported in part by National Science Foundation grants CCR-8810609 and CDA-8805910 and by the University Committee on Research, University of Nebraska at Omaha. Most of this work was done while the author was at the Department of Computer Science, Rensselaer Polytechnic Institute, Troy, New York 12180.

Our interpretation of a *sufficiently large* independent set is based on Turán's theorem [19]. It states that every graph with n vertices and m edges contains an independent set of size at least $\lceil n^2/(2m+n) \rceil$; this bound cannot be improved in terms of n and m.

The following sequential algorithm finds an independent set of this size [9]:

```
$I \leftarrow \emptyset$;
while $G$ is not empty begin
    $v \leftarrow$ a vertex of minimum degree in $G$;
    $I \leftarrow I \cup \{v\}$
    delete $v$ and its neighbors from $G$;
end;
```

In this paper, we present a parallel algorithm that finds an independent set of Turán's size in $O(\log^3 n)$ time on a CRCW PRAM with $O((n+m)\alpha(m,n)/\log^2 n)$ processors, where $\alpha(m,n)$ is the inverse of Ackerman's function. We assume that whenever several processors write to the same location at the same time, one of them succeeds. Note that it was not known whether the problem of finding an independent set of Turán's size is in NC or even in RNC. The first parallel algorithm that finds an independent set of guaranteed size is due to Goldberg [8]. It finds an independent set of size at least $n^2/32m$ in $O(\log^2 n)$ time on an EREW PRAM with $O(n+m)$ processors, provided that $m \geqq n/2$. An alternative approach is to delete all vertices of degree at least $4m/n$ and then to find a maximal independent set of the remaining graph. This independent set must contain at least $(n/2)/(4m/n+1) = n^2/(8m+2n)$ vertices.

Our algorithm uses a graph partitioning subroutine that is of independent interest. In the graph partitioning problem, we are asked to divide the vertices of a given graph into two sets, A and A', so that the number of edges joining a vertex in A to a vertex in A' is maximized; such edges are said to be cut by the partition (A, A'). Erdös [7] proved that every connected graph has a partition that cuts $\lceil m/2 + n/4 - 1/4 \rceil$ edges. His proof leads to a linear-time sequential algorithm that appears to be hard to parallelize. For our purpose, we need a slightly better partition. We prove that the only connected graphs for which Erdös's bound cannot be improved are Δ-graphs.

We define *Δ-graph* to be either an isolated vertex or a connected graph in which every block is an *odd clique*.[1] If a connected component C of a graph G is a Δ-graph, then C is a *Δ-component* of G. A graph that has no Δ-components is *Δ-free*. A connected graph G is called a *near Δ-graph* if exactly one block of G is an even clique and all other blocks are odd cliques.

We call a partition (A, A') that cuts at least $\lceil m/2 + n/4 \rceil$ edges a *dividing partition*, provided that each part (A and A') contains at least one-fifth of the total vertices. It is easy to see that Δ-graphs do not have dividing partitions; we prove that all other graphs do. For every Δ-graph, our partition subroutine finds an optimal partition, and, for every Δ-free graph, it finds a dividing partition; on a CRCW PRAM with $O((n+m)\alpha(m,n)/\log n)$ processors it runs in $O(\log^2 n)$ time.

An algorithm that finds a dividing partition can also be applied to the *weighted vertex coloring problem*. In the weighted vertex coloring problem, we assign a positive integer to each vertex of a given graph so that no adjacent vertices are assigned the same number and the sum of the numbers assigned is minimized. The sum is called the *weight* of the coloring. A coloring is called *minimum* if it is of the minimal possible weight. A coloring is *minimal* if, for each color k, every vertex of color k is adjacent to some vertex of each color less than k.

[1] The term *odd* (*even*) *clique* means a clique with an odd (even) number of vertices.

We can prove that any minimal coloring has weight at most $m + n$ and that every graph that is the union of disjoint cliques has a minimum weight coloring of weight $m+n$. We call a coloring *light* if its weight is at most $m + n$. A minimal coloring can be found sequentially in linear time. There are no previously known parallel algorithms that find light vertex colorings. We present a parallel algorithm that finds a light vertex coloring in $O(\log^3 n)$ time on a CRCW PRAM with $O((n + m)\alpha(m, n)/\log^2 n)$ processors. Any coloring of this weight uses at most $\lceil\sqrt{2m}\,\rceil$ colors. Another parallel algorithm that colors graphs using $\lceil\sqrt{2m}\,\rceil$ colors is given in [8]; it runs in $O(\log^3 n)$ time on an EREW PRAM and uses $O(m + n)$ processors.

The bottleneck in all three algorithms is finding the blocks of a graph. Tarjan and Vishkin proposed an algorithm that effectively reduces the block-finding problem to the connected-components-finding problem [18]. If the connected-components algorithm of Cole and Vishkin [5] is used, the block-finding algorithm runs in $O(\log n)$ time on a CRCW PRAM with $O((n + m)\alpha(m, n)/\log n)$ processors. We call the resulting algorithm the CTV-algorithm. Using a more efficient algorithm or an algorithm for a different model of parallel computation (for example, see [12]) will lead to other results. If $T(n, m)$ and $P(n, m)$ are, respectively, the time and the number of processors required by the biconnected components algorithm, then the partitioning algorithm requires $O(T(n, m)\log n)$ time and $P(n, m)$ processors, and the independent set algorithm and weighted vertex coloring algorithm take $O(T(n, m)\log^2 n)$ time and $P(n, m)/\log n$ processors.

Note that, for the partitioning, we do not have sufficient resources to sort. This complicates several low-level subroutines. One such case is Step 4 of the procedure DIVIDE. If we were to use $\log n$ times more processors, Step 4 would be less complicated.

We follow the usual graph-theoretic terminology [3]. Our graphs are without loops or parallel edges. The vertices of a graph on n vertices are represented by integers $0, \cdots, n - 1$; the edges are given by a list of pairs $\{(i, j)\}$, where $0 \leqq i < j \leqq n - 1$. The *decomposition tree* $T = T(G)$ of a graph G is defined to be the tree whose vertex set is comprised of the blocks and the cut vertices[2] of G; two vertices of T are adjacent if one is a block and the other is a cut vertex belonging to the block. A *star* is a tree with at least two vertices, one of which, called *the center*, is adjacent to all of the other vertices.

Section 2 contains a description of PARTITION, a parallel algorithm that finds a dividing partition of a graph with no Δ-component. Section 3 describes applications of PARTITION to the problems of finding large independent sets in parallel and of finding weighted vertex colorings of small weight. The list processing steps that are used by PARTITION are described in §4. Finally, suggestions are made for further work.

2. The partitioning algorithm. We use divide-and-conquer to design our partitioning algorithm PARTITION.

LEMMA 1. *Let (B_1, B_2) be a partition of G and let (A_1, A'_1) and (A_2, A'_2) be dividing partitions of the subgraphs $G[B_1]$ and $G[B_2]$ induced on B_1 and B_2, respectively. Then either $\rho = (A_1 \cup A_2, A'_1 \cup A'_2)$ or $\sigma = (A_1 \cup A'_2, A'_1 \cup A_2)$ is a dividing partition of G.*

Proof. Since (A_1, A'_1) and (A_2, A'_2) are dividing partitions, each of the sets $A_1 \cup A_2$, $A'_1 \cup A'_2$, $A_1 \cup A'_2$, and $A'_1 \cup A_2$ contains at least one-fifth of the vertices of G. Every edge that is cut by (B_1, B_2) is also cut by one of ρ or σ; every edge cut by (A_1, A'_1) or (A_2, A'_2) is cut by both ρ and σ. Thus, the sum Σ of the numbers of edges cut by ρ and σ is at least $w + m_1 + n_1/2 + m_2 + n_2/2$, where w is the number of edges cut by (B_1, B_2), and m_i

[2] Cut vertices are sometimes referred to as *articulation points* and blocks are sometimes referred to as *biconnected components*.

(respectively, n_i) is the number of edges (respectively, vertices) in $G[B_i]$ ($i = 1, 2$). If m and n are, respectively, the number of edges and the number of vertices of G, then $m = m_1 + m_2 + w$ and $n = n_1 + n_2$. Thus, $\Sigma \geqq m + n/2$, implying that at least one of ρ and σ is a dividing partition of G. □

We can prove[3] that every Δ-free graph that does not contain a star as a connected component can be partitioned into two Δ-free subgraphs. With the use of this fact, every Δ-free graph can be split into disjoint stars. Since every star has a trivial dividing partition, we can obtain a dividing partition of the graph by repeated application of Lemma 1. For this approach to be effective in parallel, the partition into two Δ-free subgraphs must be quickly computable and must consist of two graphs of approximately equal size. Unfortunately, it is not clear how to do this. It is, however, possible to partition a Δ-free graph either into two approximately equally sized Δ-free subgraphs or into two parts; one of which is a Δ-free graph, and the other consists of a "large" star and a number of Δ-components. Thus, instead of splitting Δ-free graphs into disjoint stars, we split them into disjoint subgraphs, each containing a star large enough to guarantee the existence of a dividing partition.

The following two lemmas take care of the base case, that is, graphs that consist of stars possibly with some Δ-graphs. The processor counts given in these lemmas depend on the assumption that Δ-graphs are represented by a list of the vertices in each block and a list of the blocks to which each cut vertex belongs.

LEMMA 2. *Every connected Δ-graph G with n vertices and m edges has a partition (A, A') that cuts $\geqq m/2 + n/4 - 1/4$ edges such that $|A| + 1 = |A'|$. Such a partition can be constructed in $O(\log n)$ time on an* EREW PRAM *with $O(n/\log n)$ processors, provided that the decomposition tree is given.*

Proof. If G is an odd clique, then choosing A to be any $\lfloor n/2 \rfloor$ of the vertices gives the desired partition. If G has two or more blocks, the situation is more complicated. First, we partition each block b independently so that one part contains $\lfloor |b|/2 \rfloor$ vertices and the other contains $\lceil |b|/2 \rceil$ vertices. Moreover, we ensure that the parent of b is in the larger part. Next, we use the Eulerian tour technique [20] to combine these partitions into a single partition of the required size. To do this, we give each edge in the decomposition tree between a block and its parent a weight of zero. Consider an edge e in the decomposition tree between a cut vertex v and its parent, a block b. Then b's parent will be a cut vertex u. If, in the partition of b, u and v are in the same part, then the edge e has weight zero; otherwise, it has weight one. (If the root of the decomposition tree is a block c, then one of the vertices in c is chosen arbitrarily to act as the parent of c in the determination of the weights of the edges between c and its children.) To determine in which part each cut vertex v is, we calculate, using the Eulerian tour technique, the weighted length of the path from the root of the decomposition tree to v. If this length is even, v is in A; otherwise, it is in A'. The noncut vertices are assigned to the parts so that the partition of each block is respected. It is easy to prove by induction on the structure of the decomposition tree that this is the desired partition. □

LEMMA 3. *Let G be formed by taking the union of a star S with s vertices and d Δ-components and possibly adding edges from vertices in the Δ-components to the center of S. Then, if $s - 2 \geqq d$, G has a dividing partition. Furthermore, this partition can be found in $O(\log n)$ time on an* EREW PRAM *with $O(n/\log n)$ processors, provided that the blocks of the Δ-graphs are given.*

Proof. To find a dividing partition of G, we partition each Δ-component, as above, and then rename the parts, if necessary, so that the larger part is A'. If G contains

[3] We omit this proof here.

d Δ-components with the total of m' edges and n' vertices, then the partition cuts $\geqq m'/2 + n'/4 - d/4$ edges belonging to the blocks. We then place the center of the star in A or A' so as to cut at least half of the edges between the center of the star and the Δ-components. We place $\lceil (d+3s-2)/4 \rceil$ vertices other than the center into the part that does not contain the center; the remaining vertices of the star are placed to minimize the disbalance of the partition. Obviously, we can produce such a partition only if $\lceil (d+3s-2)/4 \rceil \leqq s-1$; this inequality is equivalent to the condition $s-2 \geqq d$. □

From the proofs of the lemmas, we can easily extract a procedure for building a dividing partition of a graph described in Lemma 3. We call this procedure ⋆PARTITION. Now we are ready to see how DIVIDE works. Given a Δ-free graph G, DIVIDE either produces a Δ-free partition or a partition (A, A') such that A is Δ-free and A' satisfies the assumptions of Lemma 3. Note that DIVIDE can treat each connected component of G separately; thus we can assume, without loss of generality, that G is connected. The procedure DIVIDE comprises the following four steps.

Step 1. Either find an induced star S with at least $n/3$ vertices such that $G - S$ has at most one isolated vertex, or create a partition (A, A') such that A and A' each contain at least $n/3$ vertices, $G[A']$ is connected, and $G[A]$ contains no isolated vertices. If the star is found, do Step 2; otherwise do Steps 3 and 4.

Step 2. Put S and all of the Δ-components of $G - S$ into A'. Put the rest of the graph into A. Return (A, A').

Step 3. Move one vertex from each Δ-component of $G[A]$ to $G[A']$. Choose the vertices to be moved so that $G[A']$ stays connected.

Step 4. If $G[A']$ is a Δ-graph, move between one and three vertices from A' to A, so that the resulting partition is Δ-free. Return (A, A').

Step 3 is fairly straightforward, but the other steps need to be explained in more detail.

Explanation of Step 1. DIVIDE starts by finding a rooted spanning tree T of G and a vertex v with at least $2n/3$ descendants, none of which have $2n/3$ or more descendants. Next, it rearranges T so that v is the root. It does this by reversing the direction of all the parent-child links between v and the root. That is, it makes v's parent its child, its grandparent its grandchild, and so on. Let $D = V - \{v\}$ be the set of proper descendants of v. Then DIVIDE finds the connected components of $G[D]$ and counts the number of isolated vertices among them. If there are $n/3$ or more isolated vertices in $G[D]$, then v is the center of a big star S, and the isolated vertices of $G[D]$ are the other vertices of S. Alternatively, suppose that there are fewer than $n/3$ isolated vertices in $G[D]$. In this case, DIVIDE looks for a Δ-free partition (A, A') of G. There are several subcases, depending on the sizes of the connected components of $G[D]$.

If $G[D]$ has a connected component C with at least $n/3$ and at most $2n/3$ vertices, then DIVIDE returns $(C, G - C)$.

Suppose that every connected component of $G[D]$ has less than $n/3$ vertices. In this case, DIVIDE makes a list of connected components of $G[D]$ that contain two or more vertices. Then, it calculates the minimum k so that the total size of the first k connected components in this list is at least $n/3$. These components become the set A, and the rest of the graph becomes A'. DIVIDE returns the partition (A, A').

Finally, if $G[D]$ has a connected component C with more than $2n/3$ vertices, it chooses A to be a subset of C. To construct this subset, it finds all the children of v that are in C and puts them in a list so that the children that are leaves of the spanning tree T of G are at the end of the list. If one of these children has at least $n/3$

descendants, that child and its descendants become A, and the rest of the graph becomes A'. Otherwise, DIVIDE calculates the minimum k so that the subtrees rooted at the first k children of v together contain at least $n/3$ vertices. Let D' be the set of vertices of these subtrees. If $G[D']$ has no isolated vertices, DIVIDE returns $(D', G - D')$. If $G[D']$ contains isolated vertices, then the subtrees that were not included in D' all consist of a single vertex. Thus, DIVIDE can add a subtree adjacent (in $G[D]$) to all of the isolated vertices of $G[D']$, while ensuring that $|D'| \leqq 2n/3$.

Explanation of Step 2. The hardest part of Step 2 is finding the Δ-components of $G - S$. The connected components are found using the Cole–Vishkin algorithm. In §4 we will see how to determine if a connected graph H is a Δ-graph in $O(\log n)$ time on a CRCW PRAM with $O((n + m)\alpha(m, n)/\log n)$ processors.[4]

Explanation of Step 4. The input to this step is a partition (A, A') with $G[A]$ Δ-free, $G[A']$ connected, and each part containing at least $2n/9$ vertices. If $G[A']$ is a Δ-graph, Step 4 constructs the desired partition by transferring between one and three vertices from A' to A. (If $G[A']$ is not a Δ-graph, Step 4 is not executed.) Step 4 comprises twelve substeps. Below, we describe these substeps and then prove their correctness. In §4, we show that Step 4 can indeed be executed on $O((n+m)\alpha(m, n)/\log n)$ processors in $O(\log n)$ time.

Substep 4.1. Find the connected components of $G[A]$ and the blocks of $G[A]$ and $G[A']$.

Substep 4.2. If one of the connected components of $G[A]$ is not a near Δ-graph, then find a vertex $x \in A'$ adjacent to this connected component and transfer x from A' to A.

Substep 4.3. Else, if there is a vertex $x \in A'$ that is adjacent to a vertex $v \in A$ that is not in an even clique, then transfer x from A' to A.

Substep 4.4. (*If Step* 4 *gets to this point*, *each connected component of* $G[A]$ *must be a near* Δ*-graph. Moreover*, *all vertices in* A *that are adjacent to a vertex in* A' *are in even blocks of* $G[A]$.) Else, find a block B of the original graph G that is not an odd clique.

Substep 4.5. If B contains no vertices from A', find a vertex $x \in A'$ that is adjacent to the connected component of $G[A]$ that contains B and transfer it to A.

Substep 4.6. Else, if B contains exactly one vertex x from A', transfer x from A' to A.

Substep 4.7. Else, if B contains two vertices that are not in the same block of $G[A']$, then find such two vertices x and y such that no block of $G[A']$ contains both of them and there is a block of $G[A]$ containing vertices adjacent to x and y; transfer x and y from A' to A. (*At this point*, *every connected component of* $G[A]$ *containing vertices in* B *must be adjacent to two or more vertices in* A'; *so* x *and* y *exist*.)

Substep 4.8. (*If Step* 4 *gets to this point*, $C' = B \cap A'$ *is a single block of* $G[A']$.) Else, if $|B \cap A| = 1$, then find $v = B \cap A$ and $x \in B \cap A'$, such that (v, x) is an edge, and transfer x from A' to A.

Substep 4.9. Let $C' = A' \cap B$. If $B \cap A$ contains vertices only from a single block of $G[A]$ and $|B \cap A| > 1$, then find three vertices $x, y, z \in C'$ such that x and y are each adjacent to some vertex in $B \cap A$ and z is not adjacent to all vertices in $B \cap A$. Transfer the vertices x, y, and z from A' to A. (*At this point, the vertex* z *exists, since* B *contains an odd number of vertices*; *so it is not a clique.*)

[4] The obvious algorithm would be to find the blocks of H and then check if they are odd cliques by seeing whether the vertices have the "right" degrees. This algorithm does not work with the resources stated because there does not seem to be any way to produce, for each vertex, a list that contains exactly once each block to which this vertex belongs (see §4).

Substep 4.10. (*If Step* 4 *gets to this point*, B *contains vertices from multiple blocks of* A.) Else, if every vertex of $C' = B \cap A'$ is adjacent to every vertex of $B \cap A$, then take any three vertices in C' and move them in A.

Substep 4.11. Else, find a vertex $x \in C'$ and an even block C_0, ($C_0 \cap B \neq \emptyset$) such that x is not adjacent to all vertices in C_0. If $C_0 \cap B$ consists of one vertex u only, then let y and z be any two vertices in C' adjacent to u. Move x, y, and z from A' to A.

Substep 4.12. Let $y, z \in C'$ be vertices adjacent to some vertex in C_0. If y or z is x, replace x with some vertex (other than y or z) in C'. Move x, y, and z from A' to A.

The next three lemmas prove the correctness of Step 4; the first two are quite obvious, so we omit their proofs. For all the lemmas, partition (A, A') satisfies the conditions of the input to Step 4.

LEMMA 4. *Moving any odd number of vertices from some block of* A' *to* A *or moving any two nonadjacent vertices from* A' *to* A *makes* $G[A']$ *a* Δ*-free graph.*

LEMMA 5. *If, for some set* R, $G[R] \cup x$ *is a* Δ*-graph, then every component of* $G[R]$ *is a near* Δ*-graph.*

LEMMA 6. *Step* 4 *produces a new partition* (A, A') *for which both* $G[A]$ *and* $G[A']$ *are* Δ*-free.*

Proof. Since G is connected, every component C of $G[A]$ contains vertices adjacent to some vertices in A'; therefore Substep 4.2 can indeed find a required vertex $x \in A'$. By Lemma 4, moving x into A makes $G[A']$ Δ-free, and, by Lemma 5, it does not create Δ-components in $G[A]$, provided that the premise of Substep 4.2 or Substep 4.3 holds. Thus, if the algorithm does not halt after completion of Substep 4.3, then (a) every component of $G[A]$ is a near Δ-graph; and (b) for every edge (u, x) with $u \in A$ and $x \in A'$, u belongs to an even block.

The analysis of Substeps 4.5 and 4.6 is also simple. If $|B \cap A'| = 0$, then B is the even block of one of the components of $G[A]$, a vertex x can be found, and moving it to A makes $G[A']$ Δ-free. Furthermore, moving x to A does not create a Δ-component in $G[A]$, since x can only be adjacent to one vertex in B, say u, which implies that x and u make an even block in $G[A]$. Similarly, if $|B \cap A'| = 1$, then Step 4 halts after execution of Substep 4.6 (Lemma 4 and the fact that B is not an odd block). Let us now assume that the premise of Substep 4.7 holds. We first prove that B contains two vertices, x and y, that are not in the same block of $G[A']$, but are adjacent to vertices of the same block of $G[A]$. Indeed, otherwise for every even block B_e of $G[A]$, all vertices in A' adjacent to B_e belong to the same block of $G[A']$, implying that every cut vertex of $G[A']$ is also a cut vertex of G. Obviously, this implication is eliminated by the premise of Substep 4.7.

If Step 4 does not halt after Substep 4.7, then, in addition to (a) and (b), the partition satisfies two more conditions, as follows: (c) the intersection $B \cap A'$ is a single block C' of $G[A']$, and (d) for every even block C of $G[A]$, $|B \cap C| = 0$, or 1, or $|C|$.

Both properties follow from the following fact, which we already used: If a vertex is adjacent to two vertices in a block, then it is itself in the block.

If v is the only vertex of $B \cap A$ (Substep 4.8), then moving to A any $x \in B \cap A'$ adjacent to v completes Step 4. Indeed, since B is a block of G, v must be the only vertex in $A \cap B$ adjacent to x.

Let us now assume that Step 4 does not halt after Substep 4.8, and the premise of Substep 4.9 holds. Since B is not an odd clique, it is obvious that the required three vertices exist, and their transferring to A yields a Δ-free partition.

If Substep 4.10 is executed, $G[A']$ becomes Δ-free by Lemma 4, and $G[A]$ is Δ-free since the vertices from different even blocks are not adjacent to each other.

Finally, reasoning similar to the above proves that, if Step 4 attempts Substep 4.11, then the resulting $G[A]$ and $G[A']$ are Δ-free. □

Now we are ready to prove that the procedure DIVIDE produces the required partition efficiently.

THEOREM 1. *If G has n vertices and $n \geqq 27$, then* DIVIDE *finds a Δ-free partition of G such that each part has at least $2n/9$ vertices or* DIVIDE *finds a partition where one part satisfies the requirements of Lemma* 3 *and the other is Δ-free and has at most $2n/3$ vertices. Furthermore,* DIVIDE *finds this partition in $O(\log n)$ time on a* CRCW PRAM *with $O((n+m)\alpha(m,n)/\log n)$ processors.*

Proof. First, let us see that DIVIDE produces an acceptable partition. If Step 2 is executed, $G - S$ contains at most $\lfloor 2n/9 \rfloor + 1$ Δ-components, since $G - S$ contains at most $2n/3$ vertices and at most one of the Δ-components of $G - S$ has fewer than three vertices. Therefore, A' satisfies the requirements of Lemma 3.

If Steps 3 and 4 are executed, then after Step 3, A contains at least $2n/9$ vertices, since each Δ-component of $G[A]$ has at least three vertices, and $|A| \geqq n/3$. Furthermore, $G[A]$ is Δ-free, and $G[A']$ is connected. Thus, Step 4 produces a Δ-free partition. Therefore, DIVIDE produces an acceptable partition.

Steps 3 and 4 can be done in $O(\log n)$ time on a CRCW PRAM with $O((n+m)/\log n)$ processors. Therefore, DIVIDE finds an acceptable partition in $O(\log n)$ time using $O((n+m)\alpha(m,n)/\log n)$ processors in the CRCW model. □

Finally, we describe the algorithm PARTITION, which delivers a dividing partition for every Δ-free graph:

```
function PARTITION($G$);
begin
  DIVIDE($G, G_1, G_2$);
  /* $G_1$ and $G_2$ are the resulting parts and $G_1$ is a
     $\Delta$-free graph */
  $(A_1, A_1') \leftarrow$ PARTITION($G_1$);
  if $G_2$ is not $\Delta$-free, then
     $(A_2, A_2') \leftarrow \star$PARTITION ($G_2$)
  else $(A_2, A_2') \leftarrow$ PARTITION($G_2$);
  $\rho \leftarrow (A_1 \cup A_2, A_1' \cup A_2')$; $\sigma \leftarrow (A_1 \cup A_2', A_1' \cup A_2)$;
  return $\rho$ or $\sigma$, whichever is bigger;
end;
```

The correctness of PARTITION follows from Lemmas 1–6. Putting them together with Theorem 1, we obtain the following result.

THEOREM 2. *A dividing partition can be found in $O(\log^2 n)$ time on a* CRCW PRAM *with $O((m+n)\alpha(m,n)/\log n)$ processors.*

3. Applications. The algorithm that finds a dividing partition can be used to find an independent set of a graph G with n vertices and m edges that contains at least $n^2/(2m+n)$ vertices. It turns out that, if (G_1, G_2) is a dividing partition of G, then one of the parts has an independent set of the necessary size. If, however, G has a Δ-component, then G may not have a dividing partition. Thus, IND, an algorithm that finds an independent set of Turán's size, must treat Δ-components as a special case. The treatment is based on the following.

LEMMA 7. *Let G be a Δ-graph and let I contain one noncut vertex from each block that has one. Also, let G' be the result of deleting I and its neighborhood from G. Then n',* *the number of vertices of G', satisfies $n' \leqq n/3$. Furthermore, if I' is an independent set of G' such that*

$$|I'| \geqq \frac{n'^2}{2m' + n'},$$

where m' is the number of edges of G', then $I \cup I'$ is an independent set of G such that

$$|I \cup I'| \geqq \frac{n^2}{2m + n}.$$

Proof. If G is a single clique, the lemma is obvious. Alternatively, let us assume that G has at least two blocks.

To prove that $n' \leqq n/3$, we consider the decomposition tree of G. Note that any vertex in a block containing a noncut vertex will not be in G'. Let b be the number of blocks containing a noncut vertex. Since each block of G has at least three vertices, there are at least $2b + 1$ vertices of G that are not in G'. Allocate each cut vertex to one of its children in the decomposition tree. Then each block has at most one cut vertex allocated to it. Each block that contains only cut vertices has at least two children. Therefore, there are at most b vertices in blocks containing only cut vertices and at most b vertices in G'. Thus, G' has at most $n/3$ vertices.

Since I contains only noncut vertices and at most one vertex from each block, I is independent. Furthermore, since G' contains no neighbors of I, $I \cup I'$ is independent.

It remains to estimate the size of $I \cup I'$. Suppose that one vertex of I and its neighborhood is deleted, leaving G'' with n'' vertices and m'' edges. It is enough to show that

$$\frac{n^2}{2m + n} \leqq 1 + \frac{n''^2}{2m'' + n''}.$$

The following computation that appeared in [9] does this. Suppose that the vertex to be deleted has degree d. Then all of its neighbors have degrees of at least d, so $n'' = n - d - 1$ and $m'' \leqq m - d(d+1)/2$. Note that

$$\begin{aligned} 1 + \frac{n''^2}{2m'' + n''} &\geqq 1 + \frac{(n-d-1)^2}{2\left(m - \frac{d(d+1)}{2}\right) + n - d - 1} \\ &= 1 + \frac{(n-d-1)^2}{2m - (d+1)^2 + n} \\ &= \frac{2m + n + n^2 - 2n(d+1)}{2m + n - (d+1)^2}. \end{aligned}$$

Now let us consider

$$1 + \frac{n''^2}{2m'' + n''} - \frac{n^2}{2m + n}.$$

Simplifying, we see that

$$\begin{aligned}
1 + \frac{n''^2}{2m''+n''} - \frac{n^2}{2m+n} \\
&\geqq \frac{2m+n+n^2-2n(d+1)}{2m+n-(d+1)^2} - \frac{n^2}{2m+n} \\
&= \frac{(2m+n)^2 - 2n(d+1)(2m+n) + n^2(d+1)^2}{(2m+n)(2m+n-(d+1)^2)} \\
&= \frac{(2m+n-n(d+1))^2}{(2m+n)(2m+n-(d+1)^2)} \\
&\geqq 0.
\end{aligned}$$

This completes the proof. □

To find a large independent set of a Δ-graph, IND computes I and G', as suggested in Lemma 7. If G' is empty, I is the desired set. If G' is a Δ-graph, IND repeats the process. Finally, if G' is neither an empty nor a Δ-graph, IND calls itself recursively to find I'.

The algorithm IND, given below, implements this idea:

```
function $IND(G)$;
begin
  $J \leftarrow \emptyset$;
  for each connected component $D$ of $G$ do begin
    if $D$ is a $\Delta$-component do begin
      while $D$ is a nonempty $\Delta$-graph do begin
        $I \leftarrow$ one noncut vertex from each block of $D$ that has one;
        $J \leftarrow J \cup I$;
        Delete $J$ and the neighborhood of $J$ from $D$;
        end;
      $J \leftarrow J \cup$ IND$(D)$;
  /* $J$ is a big independent set of the $\Delta$-components */
    end;
    else /* $D$ is not a $\Delta$-component */
  Delete the $\Delta$-components of $G$;
  if $G$ is empty then $I' := \emptyset$
  else begin
    $(G_1, G_2) \leftarrow$ a dividing partition of $G$;
    $n_i \leftarrow$ the number of vertices of $G_i$ $(i = 1, 2)$;
    $m_i \leftarrow$ the number of edges of $G_i$ $(i = 1, 2)$;
    if $n_1^2/(2m_1 + n_1) \geqq n_2^2/(2m_2 + n_2)$
      then $I' :=$ IND$(G_1)$
    else $I' :=$ IND$(G_2)$; end
  return $J \cup I'$;
end;
```

THEOREM 3. *The procedure* IND *finds an independent set of size* $\geqq n^2/(2m+n)$ *of a graph G with n vertices and m edges in* $O(\log^3 n)$ *time on a* CRCW PRAM *with* $O((n+m)\alpha(m,n)/\log^2 n$ *processors.*

Proof. The first thing to note is that IND can afford to treat the Δ-components separately, since, if $n_1+n_2=n$ and $m_1+m_2=m$, then

$$\lceil n_1^2/(2m_1+n_1)\rceil + \lceil n_2^2/(2m_2+n_2)\rceil \geqq \lceil n^2/(2m+n)\rceil .$$

It remains to be seen that IND finds an independent set of Turán's size on the Δ-free part of the graph. Suppose that this part has n vertices and m edges and that G_1 (G_2) has n_1 (n_2) vertices and m_1 (m_2) edges. Since (G_1, G_2) is a dividing partition, $n_1+n_2=n$ and $m_1+m_2 \leqq m-m/2-n/4$. By induction, IND returns a set of size at least $\max(n_1^2/(2m_1+n_1), n_2^2/(2m_2+n_2))$. Thus, if IND does not return a big enough set, then

$$\frac{n_i^2}{2m_i+n_i} < \frac{n^2}{2m+n}, \qquad (i=1,2),$$

so

$$(2m+n)n_i^2 < (2m_i+n_i)n^2.$$

Adding the previous inequalities for $i=1$ and $i=2$ together, we obtain

$$(2m+n)(n_1^2+n_2^2) < (2m_1+n_1+2m_2+n_2)n^2.$$

Since the partition is dividing, we have

$$\begin{aligned}(2m+n)(n_1^2+n_2^2) &< (2(m_1+m_2)+n)n^2\\ &\leqq \left(2\left(m-\frac{m}{2}-\frac{n}{4}\right)+n\right)n^2\\ &= \left(m+\frac{n}{2}\right)n^2.\end{aligned}$$

This is equivalent to

$$2(n_1^2+n_2^2) < (n_1+n_2)^2, \quad \text{so} \quad (n_1-n_2)^2 < 0,$$

which is impossible. Thus, IND returns a big enough independent set.

The most time-consuming part of IND is finding a dividing partition of G. Recall that PARTITION runs in $O(\log^2 n)$ time, so IND runs in $O(\log^3 n)$ time, since each recursive call is made on a graph with at most 4/5 as many vertices and at most half as many edges as the original graph. Furthermore, since there is only one recursive call, the technique of Brent [4] can be used to reduce the processor count by a factor of $\log n$, to $O((n+m)\alpha(m,n)/\log^2 n)$. □

The other application of PARTITION is finding a light coloring. The algorithm follows the same outline as IND, except that both parts of the partition need to be considered. To combine the colorings, we require that each part use different colors. It turns out that this gives a light coloring.

```
procedure COLOR($G$);
begin
   for each connected component $C$ of $G$ begin
      if $C$ is an isolated vertex then color it 1
      else begin
         if $C$ is a $\Delta$-graph then $\star$PARTITION($G, G_1, G_2$)
            else PARTITION($G, G_1, G_2$)
         /*($G_1, G_2$) $\leftarrow$ the resulting partition of $C$;*/
         COLOR($G_1$); COLOR($G_2$);
         $c \leftarrow$ the number of colors used for $G_1$;
         increase by $c$ every color used for $G_2$;
         compute the sizes of the color classes of the
            resulting coloring of $C$ and sort them in
            nonincreasing order;
         renumber the color classes according to the new
            order;
         end;
      end;
end;
```

Note that the partitions of the connected components C into G_1 and G_2 eliminate at least half of the edges. Furthermore, since isolated vertices are part of the base case, we can assume that $n \leqq 2m$. Thus, at each level of the recursion, the total problem size $(n + m)$ decreases by a constant fraction, so the same analysis as for IND shows that COLOR runs in $O(\log^3 n)$ time on a CRCW PRAM with $O((n + m)\alpha(m, n)/\log^2 n)$ processors. It is less obvious that COLOR finds a light coloring.

THEOREM 4. *The procedure* COLOR *finds a light coloring.*

Proof. The proof is by induction on the progress of the algorithm. If G is an isolated vertex, COLOR produces a light coloring. Note that, if two graphs have light colorings, these light colorings combine in the obvious way to form a light coloring of their union. Therefore, we can assume, without loss of generality, that G has a single connected component.

Let n, n_1, and n_2 be the number of vertices of G, G_1, and G_2, respectively, and let m, m_1, and m_2 be the number of edges of G, G_1, and G_2, respectively. If (G_1, G_2) is a dividing partition, $n_1 + n_2 = n$ and $m - m/2 - n/4 \geqq m_1 + m_2$, so $m/2 \geqq m_1 + m_2 + n/2$. Alternatively, if C is a Δ-graph, then $m - m/2 - n/4 + 1/4 = m_1 + m_2$, so $m/2 = m_1 + m_2 + n/2 - 1/2$. Note that Δ-graphs have an odd number of vertices, so, in either event, $m \geqq m_1 + m_2 + \lfloor n/2 \rfloor$.

By induction, COLOR produces light colorings for G_1 and G_2. Let W be the sum of the weights of these colorings. Then it is enough to prove that $\mathcal{C}$, the coloring that COLOR produces, has weight at most $2W - \lceil n/2 \rceil$. To prove this, we give each vertex a sequence number. The sequence numbers are chosen so that the sequence numbers assigned to each color class C of $\mathcal{C}$ are $1, 2, \ldots, p_C$. Let A_k be the set of vertices with sequence number k and let l be the number of such vertices. Then, the total weight assigned to A_k is $l(l-1)/2$. Suppose that the l_1 of these vertices came from G_1, so $l - l_1$ came from G_2. Then the total weight assigned to A_k by the colorings of G_1 and G_2 is

$W_k = l_1(l_1-1)/2+(l-l_1)(l-l_1-1)/2$. If we think of W_k as a function of l_1, its minimum value occurs when $l_1 = l/2$. Thus, $W_k \geqq 2(l/2)(l/2-1)/2 = l(l-2)/4 = l(l-1)/4-l/4$. Therefore, the total weight assigned to A_k by C is at most $2W_k - l/2$. Summing over the sequence numbers, we see that the weight of C is at most $2W - n/2$. Since the weight of C is an integer, it is also at most $2W - \lceil n/2 \rceil$. □

4. Implementation details. A graph is represented by an array of vertices and an array of edges. The *number* of a vertex is its position in the vertex array. Each vertex v has a pointer into the array of edges showing where the list of edges incident to v starts.

The most expensive part of the whole computation is finding spanning trees, the connected components, and blocks of graphs. These steps are done in $O(\log n)$ time on a CRCW PRAM with $O((m+n)\alpha(m,n)/\log n)$ processors by using the CTV-algorithm appropriately modified for our purposes. It is not hard to see how to do the other steps in $O(\log n)$ time on a PRAM with $O(n+m)$ processors, by the liberal use of a sorting routine. However, it is not obvious how the processor count can be reduced to $O((m+n)/\log n)$. The necessary techniques based on the parallel prefix computation are described here.

Often, IND and COLOR need to compute $G[A]$ for various $A \subset V$. For example, after DIVIDE finds a Δ-free partition (A, A'), it calculates $G[A]$ and $G[A']$. To do this, it is enough to renumber the vertices of the original graph G so that the vertices in A and the vertices in A' have consecutive numbers. With a parallel prefix computation, each vertex in A can compute the number of vertices in A with a lower original number; this will be the vertex's new number. Similarly, each vertex in A' can compute its new number. Since an n-element parallel prefix computation can be done in $O(\log n)$ time on an EREW PRAM with $O(n/\log n)$ processors [15], the computation of $G[A]$ and $G[A']$ is not an important contribution to the overall running time of DIVIDE.

Note that the technique described above works so that the algorithm runs in $O(\log n)$ time and uses $O(n/\log n)$ processors only when it divides the graph into a bounded number of pieces. Thus, when a routine divides a graph into its connected components, it needs to use a different technique.

LEMMA 8. *Given a graph G and rooted spanning trees of each connected component of G, it is possible to renumber the vertices of G so that each connected component consists of consecutively numbered vertices, in $O(\log n)$ time on an* EREW PRAM *with $O((n+m)/\log n)$ processors.*

Proof. Consider the routine RENUMBER that does the renumbering described above. Using the Eulerian tour technique [18], RENUMBER computes the number of vertices in each connected component. Then, it assigns to the root of each spanning tree the number of vertices in its connected component and to each other vertex the value zero. A parallel prefix computation determines the range of new numbers for each vertex. Actually, assigning the new vertex numbers and rearranging the edges lists can be done with an application of the Eulerian tour technique and some parallel prefix computations. The parallel prefix and Eulerian tour computations can be done in $O(\log n)$ time on an EREW PRAM with $O(n/\log n)$ processors [18], [15], [6]. □

Some procedures, including DIVIDE, need to be able to identify Δ- and near Δ-graphs. It is done by finding the blocks of the graph in question. Unfortunately, the straightforward use of the CTV-algorithm is not always helpful, since its output must be in a different form. Effectively, the CTV-algorithm returns a list of the edges in each block, while DIVIDE and the other procedures need a data structure based on the decomposition tree; we call it the *decomposition tree data structure*. In addition to the decomposition tree, this data structure includes the information showing, for every noncut vertex, the block it belongs to and pointers in both directions between every cut vertex and corresponding node[5] in the decomposition tree. Furthermore, every block node contains a list of the noncut vertices that belong to that block.[6]

LEMMA 9. *Given a connected graph G, a spanning tree T of G, and the name of the block that each edge belongs to, it is possible to build the decomposition tree data structure in $O(\log n)$ time on an* EREW PRAM *with $O((n+m)/\log n)$ processors.*

Proof. Consider the procedure BUILD_TREE that creates the decomposition tree. First, BUILD_TREE consults the adjacency list of each vertex v to see if all the edges incident to v are in the same block. If they are, v is not a cut vertex; otherwise, v is a cut vertex. The parent of the node corresponding to a cut vertex v is the block containing the edge from v to its parent in T. If v is the root of T, then the node corresponding to v is the root of the decomposition tree. To find the parents of the block nodes, BUILD_TREE forms a list of the tree edges in each block. For each block B, it then finds the endpoint v of these edges that is highest (closest to the root) in the spanning tree T. The cut node corresponding to v is the parent of B, unless v is not a cut vertex and is the root of T. In this case, B is the root of the decomposition tree. Recall that every edge of T has a parent endpoint and a child endpoint. To find the children of a block node B, BUILD_TREE makes a list of the child endpoints of the tree edges in B. The child endpoints that are cut vertices correspond to the children of B in the decomposition tree. The child endpoints that are not cut vertices are the noncut vertices in B. All of these computations can be done in $O(\log n)$ time on an EREW PRAM with $O((n+m)/\log n)$ processors. □

Now we show how to determine if a graph is a Δ-graph or a near Δ-graph.

LEMMA 10. *Given the decomposition tree data structure, an* EREW PRAM *with $O((m+n)/\log n)$ processors can determine if a connected graph G is a Δ-graph, a near Δ-graph, or neither in $O(\log n)$ time.*

Proof. We describe the procedure IS_DELTA that determines if a connected graph is a Δ-graph.[7] First, IS_DELTA determines the number of vertices in each block B by computing the number of noncut vertices in B and the number of cut nodes adjacent to B in the decomposition tree. If one of the blocks has an even number of vertices, then G is not a Δ-graph; otherwise, it might be one.

At this point, IS_DELTA needs to determine if every block is a clique. Let $|B|$ be the number of vertices in a block B. To determine if a block B is a clique, IS_DELTA first computes the degree of each noncut vertex in B. If one or more of these vertices has a degree that is not $|B|-1$, then B is not a clique, and IS_DELTA returns "no." Otherwise, IS_DELTA considers the edges incident to the cut vertices corresponding to the children of B in the decomposition tree and activates some of them. An edge (u,v) incident to a cut vertex u is activated if v is a noncut vertex in B or if v is a cut vertex and the father of the corresponding node in the decomposition tree is B. The edges of the form (u,w), where u corresponds to a child of B and where w corresponds to the father of B, are also activated. (If B is the root of the decomposition tree, there are no edges of this form.) If, for all the vertices u corresponding to children of B, the number of activated edges is $|B|-1$, then B is a clique; otherwise, it is not. All of these computations can be done with the resources allowed. □

[5] We will use the term *node* to refer a vertex in the decomposition tree; the term *vertex* refers to a vertex in G or a subgraph of G.

[6] Obviously, the list of children of a block node can contain duplicates. This is a misfeature of the data structure and complicates its use, but there does not seem to be any way to eliminate the duplicates without using too many resources.

[7] The procedure for determining if a graph is a near Δ-graph is similar and is omitted.

The techniques described above are enough to show that Step 1 can be implemented efficiently.

LEMMA 11. *Step* 1 *can be done in* $O(\log n)$ *time on a* CRCW PRAM *with* $O((n+m)\alpha(m,n)/\log n)$ *processors.*

Proof. The running time and processor count of the first step, and of the whole procedure, is dominated by the resources required to find the spanning tree T of G. This can be done in $O(\log n)$ time on a CRCW PRAM with $O((m+n)\alpha(m,n)/\log n)$ processors [5]. The number of descendants of each vertex in the spanning tree T can be found in $O(\log n)$ time on an EREW PRAM with $O(n/\log n)$ processors [18]. Given this information, v (which must be unique) can be found easily. Finding the connected components of $G[D]$ requires another application of the spanning tree algorithm. If the largest connected component of $G[D]$ has fewer than $n/3$ vertices, then the connected components that DIVIDE assigns to A can be identified by a parallel prefix computation. This computation can be done in $O(\log n)$ time on an EREW PRAM with $O(n/\log n)$ processors [18], [15]. If the largest connected component of $G[D]$ has between $n/3$ and $2n/3$ vertices, the partition can be found in $O(1)$ steps. Finally, if there is a connected component of $G[D]$ with more than $2n/3$ vertices, DIVIDE finds the portion of this component that it puts in A with a parallel prefix computation and a connected components computation. This is done in $O(\log n)$ time on a CRCW PRAM with $O((m+n)\alpha(m,n)/\log n)$ processors. □

The last difficult step is Step 4.

LEMMA 12. *Step* 4 *can be done in* $O(\log n)$ *time on a* CRCW PRAM *with* $O((n+m)\alpha(m,n)/\log n)$ *processors.*

Proof. Substep 4.1 can be done with the resources stated by the CTV-algorithm. The only other difficult substeps are Substeps 4.7, 4.9, and 4.11.

To do Substep 4.7, DIVIDE first calculates the blocks of $G[A \cap B]$. Then, for each block B_i, it finds L_i, the list of vertices in A' adjacent to B_i. Note that a vertex $v \in A'$ can appear in several of the L_i and even several times in the same L_i. However, the total length of the L_i is at most m.

The procedure processes each L_i in parallel; it looks for vertices x and y in L_i that are not part of the same block. First, it determines if any of the vertices in the list are not cut vertices of $G[A']$. If there is a vertex $x \in A' \cap B$ that is not a cut vertex, DIVIDE finds the block C'' of $G[A']$ containing x. Next, it looks for a vertex $y \in B \cap A'$ that is not in C''. If such a vertex exists, DIVIDE returns x and y; otherwise, the desired vertices do not exist.

Alternatively, all vertices in $B \cap A'$ are cut vertices. Given a vertex $v \in B \cap A'$, let $P(v)$ be the parent of the node in the decomposition tree corresponding to v; if v is the root of the decomposition tree, let $P(v)$ be null. DIVIDE calculates $P(v)$ for each vertex in $B \cap A'$. If all these parent blocks are the same, the desired vertices do not exist. Otherwise, DIVIDE finds two vertices x and y with different parent blocks. If the node corresponding to y is an ancestor of the node corresponding to x, DIVIDE switches x and y. If the node corresponding to x is not the parent of $P(y)$, the parent block of y, then x and y are the desired vertices. Otherwise, DIVIDE looks for a vertex z that is not in $P(y)$. (If z does not exist, the desired vertices do not exist.) If the parent node of $P(z)$ corresponds to y, then x and z are the desired vertices; otherwise, y and z are.

To do Substep 4.9, DIVIDE creates a list of the vertices in C' that are adjacent to some vertex in $B \cap A$. If this list is not all of C', DIVIDE chooses x and y to be the first two vertices on this list, and it chooses z to be some vertex not on the list. Alternatively, if all the vertices in C' are adjacent to some vertex in $B \cap A$, DIVIDE chooses z to be some vertex in C' not adjacent to all the vertices in $B \cap A$, and x and y to be two other vertices in C'.

Finally, we come to Substep 4.11. To do this, DIVIDE first identifies x by computing the number of vertices in $A \cap B$ adjacent to each vertex in $A' \cap B$ and choosing x to be one of the vertices adjacent to fewer than all of them. Once DIVIDE has chosen x, it can choose C_0 to be a block of $G[A \cap B]$ containing a vertex not adjacent to x. If there is a vertex in C_0 that is adjacent to two or more vertices in C', DIVIDE chooses that vertex to be u, it chooses v to be some other vertex in C_0 adjacent to some vertex z in C', and it chooses $y \neq z$ to be a vertex in C' adjacent to u. Alternatively, if every vertex in C_0 is adjacent to at most one vertex in C', DIVIDE chooses u and v so that the vertices that they are adjacent to in C' are different.

Therefore, DIVIDE executes Step 4 in $O(\log n)$ time on a CRCW PRAM with only $O((n+m)\alpha(n,m)/\log n)$ processors. □

Putting this all together, we obtain the following theorem.

THEOREM 5. *The procedure* DIVIDE *requires* $O(\log n)$ *time on a* CRCW PRAM *with* $O((n+m)\alpha(m,n)/\log n)$ *processors.*

5. Conclusions. One way to cope with hard problems is to find a solution that is not necessarily optimal but has a guaranteed quality. We present an efficient parallel algorithm that finds an independent set of a guaranteed size. It uses, as a subroutine, a procedure that finds a partition that cuts a guaranteed number of edges. The partitioning procedure can also be used to obtain an efficient parallel algorithm to find a light coloring. The problem of finding a fast parallel algorithm that finds a minimal coloring remains open, however.

Approaching other combinatorial problems similarly should lead to interesting results. In parallel, the matching problem may be hard; no deterministic NC-algorithm is known. Thus, we are led to the question of how big a matching we can be sure of finding deterministically in NC. If the edges have weights, we can ask a similar question about the weight of the matching. This question is particularly interesting, since matching is the bottleneck in Anderson and Aggarwal's algorithm, which finds a depth-first search tree of an undirected graph [1]. Therefore, studying algorithms that find big matchings may lead to a deterministic NC-depth-first search algorithm.

Acknowledgments. The authors thank H. Kierstead, for finding an error in an earlier version of DIVIDE, and the reviewers for several useful comments.

REFERENCES

[1] A. AGGARWAL AND R. ANDERSON, *A random* NC*-algorithm for depth first search*, in Proc. 19th Annual ACM Symposium on Theory of Computing, 1987, New York, pp. 325–334.

[2] 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.

[3] J. A. BONDY AND U. S. R. MURTY, *Graph Theory with Applications*, North–Holland, Amsterdam, 1979.

[4] R. P. BRENT, *The parallel evaluation of general arithmetic expressions*, J. Assoc. Comput. Mach., 2 (1974), pp. 201–208.

[5] R. COLE AND U. VISHKIN, *Approximate and exact parallel scheduling with applications to list, tree, and graph problems*, in Proc. 27th Annual Symposium on Foundations of Computer Science, 1986, Toronto, Canada, pp. 478–491.

[6] ———, *Approximate parallel scheduling.* I. *The basic technique with applications to optimal parallel list ranking in logarithmic time*, SIAM J. Comput., 17 (1988), pp. 128–142.

[7] P. ERDÖS, *On even subgraphs of graphs*, Mat. Lapok., 18 (1964), pp. 283–288. (In Hungarian.)

[8] M. GOLDBERG, *Parallel algorithms for three graph problems*, Congr. Numer., 54 (1986), pp. 111–121.

[9] M. Goldberg, S. Lath, and J. Roberts, *Heuristics for the graph bisection problem*, Tech. Report, 86-8, Rensselaer Polytechnic Institute, Troy, NY, 1986.

[10] M. Goldberg and T. Spencer, *A new parallel algorithm for the maximal independent set problem*, SIAM J. Comput., 18 (1989), pp. 419–427.

[11] ———, *Constructing a maximal independent set in parallel,* SIAM J. Discrete Math., 2 (1989), pp. 322–328.

[12] Y. Han and R. A. Wagner, *An efficient and fast parallel connected component algorithm*, J. Assoc. Comput. Mach., 37 (1990), pp. 626–642.

[13] D. Johnson, *Approximate algorithms for combinatorial problems*, J. Comput. System Sci., 9 (1974), pp. 256–278.

[14] R. M. Karp and A. Wigderson, *A fast parallel algorithm for the maximal independent set problem*, in Proc. 16th ACM Symposium on Theory of Computing, 1984, Washington, DC, pp. 266–272.

[15] R. E. Ladner and M. J. Fischer, *Parallel prefix computation*, J. Assoc. Comput. Mach., 27 (1980), pp. 831–838.

[16] M. Luby, *A simple parallel algorithm for the maximal independent set problem*, SIAM J. Comput., 15 (1986), pp. 1036–1053.

[17] ———, *Removing randomness in parallel computation without a processor penalty*, in Proc. 29th Annual Symposium on Foundation of Computer Science, 1988, White Plains, NY, pp. 162–173.

[18] R. Tarjan and U. Vishkin, *An efficient parallel biconnectivity algorithm*, SIAM J. Comput., 14, (1985), pp. 862–874.

[19] P. Turán, *On the theory of graphs*, Colloq. Math., 3 (1954), pp. 19–30.

SIAM J. DISC. MATH.
Vol. 6, No. 3, pp. 460–476, August 1993

EFFICIENT DETECTION AND PROTECTION OF INFORMATION IN CROSS TABULATED TABLES I: LINEAR INVARIANT TEST*

MING-YANG KAO† AND DAN GUSFIELD‡

Abstract. To protect sensitive information in a cross tabulated table, it is a common practice to suppress some of the cells in the table. A linear combination of suppressed cells is called a *linear invariant* if the combination has a unique feasible value. Intuitively, the information contained in an invariant is not protected even though the values of the suppressed cells are not disclosed. This paper gives a surprisingly efficient algorithm for testing whether a linear combination of suppressed cells is an invariant. In sequential computation, the algorithm runs in optimal linear time. In parallel computation, the algorithm runs in polylogarithmic time using a polynomial number of processors on a parallel random access machine. The algorithm exploits a linear algebraic structure of directed and undirected cycles in a mixed graph induced by a given table. This new structure also plays a crucial role in subsequent papers on other aspects of detecting and protecting sensitive information in a cross tabulated table.

Key words. statistical tables, linear algebra, graph theory, mixed graphs, cycle spaces, strong connectivity, parallel computation

AMS(MOS) subject classifications. 68Q22, 62A99, 05C99

1. Introduction.

Cross tabulated tables are extremely useful tools for organizing and exhibiting information. In particular, they are routinely used to report statistical data. To protect sensitive information in statistical reports, it is a common practice to suppress the values of certain sensitive cells in a table. There are two fundamental issues concerning the effectiveness of this practice. The *detection* issue is to decide whether an adversary can deduce significant information about the suppressed cells from the published data of a table. The *protection* issue is to study how a table maker can suppress a small number of cells in addition to the sensitive ones such that the resulting table does not leak significant information.

These two issues are of utmost concern to statistical agencies. As is briefly discussed in Denning [19], the research into these two issues was in fact started by statisticians. They applied linear programming techniques to various problems and obtained many helpful computational heuristics [36], [7], [17], [18], [16], [35], [37], [15], [34], [14].

The first algorithmic advance came when Gusfield introduced a graph theoretic approach and developed several algorithms that are vastly more efficient than linear programming techniques [24]–[26]. Chief among those results is an optimal linear-time sequential algorithm for finding all suppressed cells that have unique feasible values. Intuitively, these cells are in effect unprotected because their values can be precisely deduced from the published data.

Kao conducted the first systematic study of the area by introducing a linear algebraic approach to complement the graph theoretic one [28]. This paper reports two fundamental theorems, the *Strongly Connected Cycle Space Theorem* and the *Strongly Connected Table Basis Theorem*, developed for the combined approach of Kao [28]. The

*Received by the editors May 6, 1992; accepted for publication July 21, 1992. Part of this work was done while the authors were at the Department of Computer Science, Yale University, New Haven, Connecticut 06520.

†Department of Computer Science, Duke University, Durham, North Carolina 27706. This author's work was supported in part by National Science Foundation grants MCS-8116678 and DCR-8405478.

‡Department of Electrical and Computer Engineering, University of California, Davis, California 95616. This author's work was supported in part by National Science Foundation grant MCS-81-05894 and U.S. Census Bureau grant JSA 86-9.

cycle space theorem characterizes the relationship between the classic Z_2 vector spaces generated by directed and undirected cycles in a strongly connected mixed graph. The table basis theorem is a variant of the cycle space theorem for Euclidean spaces induced by feasible assignments to the suppressed cells of a table. These two fundamental theorems lay the foundation for the systematic study of Kao on detecting and protecting sensitive information in a cross tabulated table [28].

This paper demonstrates the usefulness of the above two theorems by using them to design a surprisingly efficient algorithm for the problem of testing whether a linear combination of suppressed cells has a unique feasible value. A linear combination of the suppressed cells is called a *linear invariant* if it has a unique feasible value. Intuitively, the information contained in a linear invariant is not protected even though the values of the suppressed cells are not disclosed. In sequential computation, the linear invariant test algorithm runs in optimal linear time. In parallel computation [21], [29], the algorithm runs in polylogarithmic time using a polynomial number of processors on a parallel random access machine.

To elaborate on the significance of the linear invariant test algorithm, a few definitions are in order. This paper studies two-dimensional tables that publish the following three types of data:

- the precise values of all cells except a set of sensitive ones, which are *suppressed*;
- an upper bound and a lower bound for each suppressed cell; and
- all row sums and column sums of the complete set of cells.

The suppressed cells may have real or integer values. The suppressed cells may have different bounds, and the bounds may be finite or infinite. The upper bound of a suppressed cell should be strictly greater than its lower bound; otherwise, the precise value of that cell is immediately known.

An *unbounded feasible assignment* to a table is an assignment of values to the suppressed cells such that each row or column adds up to its published sum. A *bounded feasible assignment* is an unbounded feasible assignment that also satisfies the bounds of the suppressed cells.

Formally, a *linear invariant* is a linear combination of suppressed cells that has the same value at all *bounded* feasible assignments. Similarly, an *invariant cell* is a suppressed cell that has the same value at all bounded feasible assignments. An invariant cell is in fact a special case of a linear invariant where the cell in question has coefficient 1 and all other suppressed cells have coefficients 0 in the linear combination.

Figure 1 provides an example of a complete table. Figure 2 gives a published version of that complete table. Let $E_{p,q}$ denote the cell at row p and column q. In the published table, $E_{6,i}$ is an invariant because it is the only suppressed cell in row 6. $E_{2,c}$ and $E_{3,c}$ are invariants for the following reasons. The sum of $E_{2,c}$ and $E_{3,c}$ is 19, and their values are between 0 and 9.5. Thus, both cells are forced to have the unique value 9.5.

Let R_p denote the sum of the suppressed cells in row p. Let C_q denote the sum of the suppressed cells in column q. Then R_p and C_q are clearly linear invariants.

The following linear combination is also an invariant: $2.5E_{1,a} + 1.5E_{1,b} + 3.5E_{2,a} + 2.5E_{2,b} + E_{2,d} + 1.5E_{2,e} + 3E_{2,f} + 3E_{2,g} + 4E_{2,h} + 2E_{2,i} + 2E_{3,d} + 2.5E_{3,e} + 2.5E_{4,f} + 2.5E_{4,g} + 2.5E_{5,f} + 2.5E_{5,g} + 3.5E_{5,h} + 1.5E_{5,i}$. This linear combination is an invariant because it can be expressed as a linear combination of the above-proven invariants: $3C_a + 2C_b + 0.5C_d + C_e + 2.5C_f + 2.5C_g + 3.5C_h + 1.5C_i - 0.5R_1 + 0.5R_2 + 1.5R_3 - 0.5E_{2,c} - 1.5E_{3,c} - 1.5E_{6,i}$.

Formally, the *linear invariant test problem* is that of testing whether an input linear combination of suppressed cells is a linear invariant. The problem can be solved by linear

row column index	a	b	c	d	e	f	g	h	i	row sum
1	9.5	4.5	1.5	7	1.5	1.5	5.5	2	3	36.0
2	4.5	9.5	9.5	4.5	4.5	9.5	9.5	9.5	4.5	65.5
3	6	1.5	9.5	0	9.5	6	5.5	2	5.5	45.5
4	2	1.5	4	7	1.5	4.5	9.5	5.5	2	37.5
5	1.5	5.5	4	6	5.5	0	0	4.5	9.5	36.5
6	2	3	3	4	6	5.5	2	2	9.5	37.0
column sum	25.5	25.5	31.5	28.5	28.5	27.0	32.0	25.5	34.0	

FIG. 1. *A complete table.*

row column index	a	b	c	d	e	f	g	h	i	row sum
1			1.5	7	1.5	1.5	5.5	2	3	36.0
2										65.5
3	6	1.5				6	5.5	2	5.5	45.5
4	2	1.5	4	7	1.5			5.5	2	37.5
5	1.5	5.5	4	6	5.5					36.5
6	2	3	3	4	6	5.5	2	2		37.0
column sum	25.5	25.5	31.5	28.5	28.5	27.0	32.0	25.5	34.0	

Note: Let $E_{p,q}$ denote the cell at row p and column q. The lower and upper bounds for all suppressed cells, except $E_{2,c}$ and $E_{3,c}$, are $-\infty$ and $+\infty$. The lower and upper bounds for $E_{2,c}$ and $E_{3,c}$ are 0 and 9.5.

FIG. 2. *A published table.*

programming. The given table is translated into a set of linear constraints such that

- each suppressed cell is a variable;
- each row or column sum induces an equation; and
- the upper and lower bounds of each suppressed cell yield a pair of inequalities.

To decide whether the given linear combination is a linear invariant, it suffices to treat the linear combination as an objective function and compute its maximum and minimum subject to the above constraints. Then the given linear combination is an invariant if and only if its maximum and minimum are equal. Let n denote the number of rows and columns. Let m denote the number of suppressed cells. Then there are m variables, n equations, and $2m$ inequalities.

In sequential computation, using the best-known algorithm for linear programming [40], this approach is not even strongly polynomial-time. It actually runs in $O((m+n)^{1.5}\cdot m\cdot L)$ time, where L is the maximum number of bits needed to represent the row and column sums and the bounds of the suppressed cells. In sharp contrast, given the suppressed cells, their bounds, and a bounded feasible assignment, the linear invariant test algorithm of this paper runs in optimal $O(m+n)$ time.

In parallel computation [21], [29], linear programming is log-space complete for P [23], [20] and is unlikely to have efficient parallel algorithms [11]. Again in sharp contrast, given the suppressed cells, their bounds, and a bounded feasible assignment, the linear invariant test algorithm of this paper runs in $O(\log^2 n)$ time using $M(n)$ processors on an exclusive-read exclusive-write parallel random access machine, where $M(n)$ is the number of arithmetic operations used to multiply two $n\times n$ matrices. Currently, the best known value for $M(n)$ is $O(n^{2.376})$ [12].

This paper is organized as follows. Section 2 proceeds to discuss the Strongly Connected Cycle Space Theorem. Section 3 describes the linear algebraic and graph theoretic approaches. Section 4 states the Strongly Connected Table Basis Theorem and uses the theorem to solve the linear invariant test problem. Section 5 uses the cycle space theorem to prove the table basis theorem. Section 6 concludes the paper with a brief discussion.

2. The Strongly Connected Cycle Space Theorem. The cycle space theorem characterizes the relationship between the classic Z_2 vector spaces generated by directed and undirected cycles in a strongly connected mixed graph. Section 2.1 reviews some basic facts about mixed graphs. Section 2.2 states the cycle space theorem and discusses its implications.

The proof of the cycle space theorem uses induction based on depth-first search. Section 2.3 reviews depth-first search in mixed graphs and proves some technical lemmas that are needed to prove the theorem. Section 2.4 gives the proof of the theorem.

2.1. Basics of mixed graphs. A *mixed* graph is one that may contain both directed and undirected edges. A *traversable* cycle (or path) of a mixed graph refers to one that can be traversed along the directions of its edges. A *direction-blind* cycle (path, tree, or forest) refers to one that disregards the directions of its edges; the word direction-blind is often omitted for brevity. A mixed graph is called *strongly connected* if for each pair of vertices x and y, there exists a traversable cycle containing both x and y.

An *edge-simple* cycle (or path) is one where no edge appears more than once. Let $\mathcal{H}$ be a mixed graph with m edges. An edge-simple cycle C of $\mathcal{H}$ is sometimes regarded as a vector α in the vector space Z_2^m, the m-fold Cartesian product of Z_2. Each edge $e\in\mathcal{H}$ corresponds to a dimension in Z_2^m, and the component of α at the dimension of e is 1 if and only if e is in C.

Based on the above convention, the *mod* 2 *sum* of two edge-simple cycles is the set of edges that appear in exactly one of the two given cycles [6]. Note that the mod 2 sum of two edge-simple direction-blind cycles always results in an edge-disjoint set of edge-simple direction-blind cycles. However, the mod 2 sum of two edge-simple traversable cycles may or may not be an edge-disjoint set of traversable cycles.

Let T be a direction-blind spanning forest of $\mathcal{H}$ with a spanning tree in each connected component. For each nontree edge $e \in \mathcal{H}$, let $B(e)$ be the cycle formed by e and the direction-blind tree path of T between the two endpoints of e. The cycles $B(e)$ are called the *fundamental cycles* of $\mathcal{H}$ with respect to T [6].

The *direction-blind cycle space* of $\mathcal{H}$, denoted by $CS(\mathcal{H})$, is the Z_2 vector space that consists of the mod 2 sums of edge-simple *direction-blind* cycles in $\mathcal{H}$. The *traversable cycle space* of $\mathcal{H}$, denoted by $TCS(\mathcal{H})$, is the Z_2 vector space that consists of the mod 2 sums of edge-simple *traversable* cycles in $\mathcal{H}$. Note that $TCS(\mathcal{H})$ is a subspace of $CS(\mathcal{H})$.

The following classic fact is extremely useful.

FACT 2.1 (Folklore [6]). *Let Ω be the set of all $B(e)$. Then, Ω is a basis of $CS(\mathcal{H})$.*

Remark. Let $\mathcal{H}'$ be a subgraph of $\mathcal{H}$. Let C be a vector in $CS(\mathcal{H}')$. Note that C can be regarded as a vector in $CS(\mathcal{H})$ by assigning 0 to the edges that are in $\mathcal{H}$ but not in $\mathcal{H}'$. This embedding also applies to the traversable cycle spaces of $\mathcal{H}'$ and $\mathcal{H}$.

2.2. ***CS* versus *TCS*.** This section states the cycle space theorem and discusses its implications. The proof of the theorem is given in §§2.3 and 2.4.

THEOREM 2.2 (The Strongly Connected Cycle Space Theorem). *Let $\mathcal{H}$ be a mixed graph. If $\mathcal{H}$ is strongly connected, then $TCS(\mathcal{H}) = CS(\mathcal{H})$.*

The above theorem immediately implies several interesting nontrivial facts. Two instances are mentioned here.

COROLLARY 2.3. *If $\mathcal{H}$ is strongly connected, then $CS(\mathcal{H})$ has a basis consisting of traversable cycles.*

COROLLARY 2.4. *If $\mathcal{H}$ is strongly connected, then a basis of $CS(\mathcal{H})$ is also a basis of $TCS(\mathcal{H})$, even if that basis contains nontraversable cycles.*

The above theorem has two useful equivalent forms that are stated in the following two theorems.

THEOREM 2.5. *Let $\mathcal{H}$ be a mixed graph. If every connected component of $\mathcal{H}$ is strongly connected, then $TCS(\mathcal{H}) = CS(\mathcal{H})$.*

Proof. Let $\mathcal{H}_1, \cdots, \mathcal{H}_k$ be the subgraphs of $\mathcal{H}$ induced by its connected components. By definition, each vector of $CS(\mathcal{H})$ is the sum of some edge-simple direction-blind cycles in $\mathcal{H}$. Because a cycle can appear in only one connected component, each vector of $CS(\mathcal{H})$ is the sum of some vectors from $CS(\mathcal{H}_1), \cdots, CS(\mathcal{H}_k)$. Therefore, $CS(\mathcal{H}) \subseteq \sum_{i=1}^{k} CS(\mathcal{H}_i)$. On the other hand, because each $\mathcal{H}_i$ is a subgraph of $\mathcal{H}$, clearly $CS(\mathcal{H}) \supseteq \sum_{i=1}^{k} CS(\mathcal{H}_i)$. Thus, $CS(\mathcal{H}) = \sum_{i=1}^{k} CS(\mathcal{H}_i)$. Similarly, $TCS(\mathcal{H}) = \sum_{i=1}^{k} TCS(\mathcal{H}_i)$. Next, by Theorem 2.2, for all $\mathcal{H}_i$, $TCS(\mathcal{H}_i) = CS(\mathcal{H}_i)$ because $\mathcal{H}_i$ is strongly connected. Consequently, $TCS(\mathcal{H}) = \sum_{i=1}^{k} TCS(\mathcal{H}_i) = \sum_{i=1}^{k} CS(\mathcal{H}_i) = CS(\mathcal{H})$. □

THEOREM 2.6. *Let $\mathcal{H}$ be a mixed graph. Let X be the set of edges in $\mathcal{H}$ that are not in any strongly connected components of $\mathcal{H}$. Then, $TCS(\mathcal{H}) = CS(\mathcal{H} - X)$.*

Proof. First, $TCS(\mathcal{H}) = TCS(\mathcal{H} - X)$ because a traversable cycle of $\mathcal{H}$ cannot contain any edges from X and because $TCS(\mathcal{H})$ is generated by traversable cycles. Next, from Theorem 2.5, $TCS(\mathcal{H} - X) = CS(\mathcal{H} - X)$ because every connected component of $\mathcal{H} - X$ is strongly connected. Thus, $TCS(\mathcal{H}) = CS(\mathcal{H} - X)$. □

2.3. Depth-first search and technical lemmas. The proof of the cycle space theorem uses induction on orders derived from the postorder numbering of depth-first search trees. Both depth-first search and the postorder numbering of ordered trees are extremely useful in computational graph theory. Their definitions and a detailed discussion can be found in standard textbooks on algorithms [3], [27], [4], [13], [38]. This section reviews key facts about these concepts and proves two technical lemmas needed for the proof of the cycle space theorem.

Let $\mathcal{H}$ be a strongly connected mixed graph.

FACT 2.7. *Conducting depth-first search in $\mathcal{H}$ produces a spanning tree where all the vertices are reachable from the root via traversable paths.*

Let T be a spanning tree of $\mathcal{H}$ induced by depth-first search. T is considered an ordered tree: for each vertex, the tree edges connecting x and its children in T are arranged in the order of visit by depth-first search.

The induction proof of the cycle space theorem uses two orders based on the postorder numbering of the ordered tree T. Both are denoted by $\prec$ and are defined as follows:

- The first order is a total one defined on the vertices: For all vertices x in $\mathcal{H}$, let $\#(x)$ denote the postorder number of x in T. For all vertices x and y, let $x \prec y$ if $\#(x) < \#(y)$.
- The second order is a partial one defined on the directed edges not in T: For all directed nontree edges $e = x \rightarrow y$ in $\mathcal{H}$, let $\#(e) = \#(x)$. For all directed nontree edges d and e, let $d \prec e$ if $\#(d) < \#(e)$.

An *undirected back edge* is an undirected edge not in T that connects between a vertex and an ancestor in T. An *directed back edge* is a directed nontree edge that points from a vertex to an ancestor. A *directed forward edge* is a directed nontree edge that points from a vertex to a descendant. A *directed cross edge* is a directed nontree edge between two vertices without an ancestor-descendant relationship.

FACT 2.8. *There are four types of nontree edges in $\mathcal{H}$: undirected back edges, directed back edges, directed forward edges, and directed cross edges.*

FACT 2.9. *If $e = x \rightarrow y$ is a directed cross edge, then $y \prec x$.*

For visual intuition, assume that $\mathcal{H}$ is drawn on an Euclidean plane: for every vertex x of $\mathcal{H}$, x is assigned the coordinate $(\#(x), h(x))$, where $h(x)$ denotes the height of x in T. For two distinct vertices x and y, x is said to be *strictly* to the right of y if x is to the right of y in the above drawing and x is not a ancestor of y in T.

The following facts provide useful intuition.

FACT 2.10. *For all vertices x and y, if $x \prec y$, then either y is an ancestor of x in T or y is strictly to the right of x.*

FACT 2.11. *All directed cross edges point strictly from right to left.*

Proof. The proof follows by Facts 2.9 and 2.10. □

The next two technical lemmas are used to prove the cycle space theorem.

LEMMA 2.12. *For each directed nontree edge $e = x \rightarrow y$, there is an edge-simple traversable path P in $\mathcal{H}$ from y to some vertex z such that*

1. *either $z = x$ or z is an ancestor of x in T;*
2. *z appears only once in P and is the only vertex in P with the above property; and*
3. *$d \prec e$ for all directed nontree edges $d \in P$.*

Proof. P and z are constructed as follows. Because $\mathcal{H}$ is strongly connected, there is an edge-simple traversable path $Q = w_1, \cdots, w_k$ from y to x with $w_1 = y$ and $w_k = x$. Let s be the smallest index in Q such that either $x \prec w_s$ or $x = w_s$. Note that s exists because $w_k = x$. Now let $P = w_1, \cdots, w_s$ and let $z = w_s$. The three properties of P and z are verified as follows.

Property 1. It suffices to show that z is an ancestor of x in T assuming $z \neq x$. Then, $x \prec z$ by the choice of s. By Fact 2.10, either z is an ancestor of x in T or z is strictly to the right of x. To prove the property by contradiction, assume that z is strictly to the right of x. There are two cases: (1) $s = 1$ or (2) $s > 1$. In case (1), $z = y$. Therefore, the nontree edge e is a directed cross edge pointing strictly from left to right, contradicting Fact 2.11. In case (2), $w_{s-1} \prec x$ by the minimality of s. By Fact 2.10, either w_{s-1} is an descendant of x in T or w_{s-1} is strictly to the left of x. Thus, the edge in P between w_{s-1} and w_s is either an undirected cross edge or a directed cross edge pointing strictly from left to right, contradicting Fact 2.8 or Fact 2.11. This finishes the proof of the first property.

Property 2. This property follows from the minimality of the index s.

Property 3. Because P is traversable, every directed nontree edge $d \in P$ points from some vertex w_i with $i < s$. By the minimality of s, $w_i \prec x$. Hence, $d \prec e$. This finishes the proof of the third property. □

LEMMA 2.13. *For every directed nontree edge $e \in \mathcal{H}$, there is an edge-simple traversable cycle C in $\mathcal{H}$ containing e such that $d \prec e$ for all directed nontree edges $d \in C$ with $d \neq e$.*

Proof. Let $e = x \rightarrow y$. Let P be an edge-simple traversable path in $\mathcal{H}$ from y to some vertex z such that Lemma 2.12 is satisfied. Let R be the tree path in T from z to x. By the first property in Lemma 2.12, R is traversable. Thus, the cycle C formed by e, P, and R is traversable. By the second property in Lemma 2.12, C is edge-simple. Next, because R is a tree path, all nontree edges $d \in C$ with $d \neq e$ are also in P. Consequently, by the third property in Lemma 2.12, $d \prec e$ for all nontree edges $d \in C$ with $d \neq e$. □

2.4. Proving the cycle space theorem. This section completes the proof of the cycle space theorem.

A basis Ω of $CS(\mathcal{H})$ is constructed as follows. For each nontree edge e, let $B(e)$ be the edge-simple direction-blind cycle formed by e and the tree path in T between the two endpoints of e. Let Ω be the set of all $B(e)$. By Fact 2.1, Ω is a basis of $CS(\mathcal{H})$.

A set Π of traversable cycles in $\mathcal{H}$ is constructed as follows. For each nontree edge e, let $C(e)$ be an edge-simple traversable cycle. If e is undirected, let $C(e) = B(e)$, which is traversable because every undirected nontree edge in $\mathcal{H}$ is a back edge. If e is directed, let $C(e)$ be a traversable cycle that satisfies Lemma 2.13. Now let Π be the set of all $C(e)$.

The following two decomposition lemmas are useful for induction.

LEMMA 2.14. *Let e be a nontree edge in $\mathcal{H}$. Let $u_1, \cdots, u_s$ be the undirected nontree edges in $C(e)$ other than e itself. Let $d_1, \cdots, d_t$ be the directed nontree edges in $C(e)$ other than e itself. Then, $B(e) = C(e) + \sum_{i=1}^{s} B(u_i) + \sum_{j=1}^{t} B(d_j)$.*

Proof. It suffices to show that $0 = C(e) + B(e) + \sum_{i=1}^{s} B(u_i) + \sum_{j=1}^{t} B(d_j)$. In the right-hand side of this equality, each nontree edge of $C(e)$ is canceled for appearing exactly once in $C(e)$ and exactly once in a fundamental cycle $B(e)$, $B(u_i)$, or $B(d_j)$. Consequently, the resulting vector of the right-hand side of this equality contains only tree edges. Because this vector is the sum of vectors in $CS(\mathcal{H})$, it is in $CS(\mathcal{H})$. The lemma then follows from the fact that the only vector in $CS(\mathcal{H})$ not containing any nontree edge is the zero vector. □

LEMMA 2.15. *Every $B(e) \in \Omega$ is a sum of cycles in Π.*

Proof. There are two cases based on whether e is directed or not. If e is undirected, then $B(e) = C(e)$ is already a member of Π. If e is directed, then the proof is by induction on $\prec$ for the directed nontree edges as follows.

Induction Hypothesis. For each directed nontree edge g with $g \prec e$, $B(g)$ is a sum of cycles in Π.

Induction Step. Refer to the equality in Lemma 2.14 as the *traversable decomposition* of $B(e)$. In the right-hand side of $B(e)$'s traversable decomposition, $B(u_i) = C(u_i)$ because u_i is undirected. Next, by the induction hypothesis, $B(d_j)$ is a sum of cycles in Π because $d_j \prec e$ by the choice of $C(e)$ based on Lemma 2.13. Therefore, $B(e)$ is a sum of cycles in Π. □

THEOREM 2.2 (The Strongly Connected Cycle Space Theorem). *Let $\mathcal{H}$ be a mixed graph. If $\mathcal{H}$ is strongly connected, then $TCS(\mathcal{H}) = CS(\mathcal{H})$.*

Proof. Clearly $TCS(\mathcal{H})$ is a subspace of $CS(\mathcal{H})$. Conversely, by Lemma 2.15, $CS(\mathcal{H})$ is a subspace of $TCS(\mathcal{H})$. These two facts prove the theorem. □

3. The graph theoretic and linear algebraic approaches to invariant testing. The linear algebraic approach to invariant testing is outlined in §3.1. The graph theoretic approach is described in §3.2. The two approaches are combined in §3.3. The discussion of the approaches are introductory in this section. They are further developed in §5.

3.1. The linear algebraic approach. Let $\mathcal{T}$ be a two-dimensional table as described in §1. The *bounded kernel of* $\mathcal{T}$, denoted by $BK(\mathcal{T})$, is the real vector space consisting of all linear combinations of $\alpha - \beta$, where α and β are two arbitrary *bounded* feasible assignments of $\mathcal{T}$. Similarly, the *unbounded kernel* of $\mathcal{T}$, denoted by $UK(\mathcal{T})$, is the real vector space consisting of all linear combinations of $\alpha - \beta$, where α and β are two arbitrary *unbounded* feasible assignments of $\mathcal{T}$.

Remark. Let $\mathcal{T}'$ be a table obtained from $\mathcal{T}$ by publishing the precise values of some of the suppressed cells in $\mathcal{T}$. Then, a vector in $UK(\mathcal{T}')$ can be regarded as a vector in $UK(\mathcal{T})$ by assigning 0 to the suppressed cells that are in $\mathcal{T}$ but not in $\mathcal{T}'$. This embedding also applies to the vectors in the bounded kernels of $\mathcal{T}'$ and $\mathcal{T}$.

Let F be a linear combination of the suppressed cells in $\mathcal{T}$. The linear invariant test problem for F can be recast into a linear algebraic problem based on the following lemma.

LEMMA 3.1. *Let $\mathcal{T}$ be a table. Let Ω be a basis of $BK(\mathcal{T})$. Let F be a linear combination of the suppressed cells in $\mathcal{T}$. Then F is a linear invariant if and only if $F(\gamma) = 0$ for all $\gamma \in \Omega$.*

Proof. The two directions of the lemma are proved as follows.

($\Rightarrow$) Assume that F is a linear invariant. Let γ be a vector in Ω. By definition, $\gamma = \sum_{i=1}^{h} a_i(\alpha_i - \beta_i)$ for some h, a_i, α_i, and β_i. Then, $F(\gamma) = \sum_{i=1}^{h} a_i F(\alpha_i - \beta_i) = \sum_{i=1}^{h} a_i(F(\alpha_i) - F(\beta_i))$. Note that for all i, $F(\alpha_i) - F(\beta_i) = 0$ because F is a linear invariant of $\mathcal{T}$ and because α_i and β_i are bounded feasible assignments of $\mathcal{T}$. Thus, $F(\gamma) = 0$ for all $\gamma \in \Omega$.

($\Leftarrow$) Assume that $F(\gamma) = 0$ for all $\gamma \in \Omega$. Let $\gamma_1, \cdots, \gamma_k$ denote the vectors of Ω. Let α and β be bounded feasible assignments of $\mathcal{T}$. By definition, $\alpha - \beta \in BK(\mathcal{T})$. Because Ω is a basis of $BK(\mathcal{T})$, $\alpha - \beta = \sum_{i=1}^{k} b_i\gamma_i$ for some b_i. So $F(\alpha - \beta) = \sum_{i=1}^{h} b_i F(\gamma_i)$. Notice that for all i, $F(\gamma_i) = 0$. Therefore, $F(\alpha - \beta) = 0$. Consequently, $F(\alpha) = F(\beta)$ for all $\alpha, \beta \in BK(\mathcal{T})$, and F is a linear invariant of $\mathcal{T}$. □

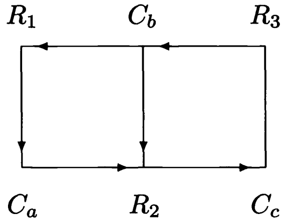
The above lemma can be used to test whether F is a linear invariant by evaluating F only at a basis of $BK(\mathcal{T})$. The computational complexity of this approach heavily depends upon how efficiently a basis of $BK(\mathcal{T})$ can be found. Computing a basis of $BK(\mathcal{T})$ seems very difficult. In contrast, computing a basis of $UK(\mathcal{T})$ appears less difficult because there are no bounds on cell values for $UK(\mathcal{T})$. In light of this difference, the key idea used in the linear invariant test algorithm of this paper is to relate the bases of $BK(\mathcal{T})$ and $UK(\mathcal{T})$ via a bipartite mixed graph $\mathcal{H}$ constructed from the input table $\mathcal{T}$. This mixed graph is defined in the next section.

3.2. The graph theoretic approach. The *suppressed graph* $\mathcal{H}$ of $\mathcal{T}$ is a bipartite mixed graph constructed below [25]. For each row or column of $\mathcal{T}$, there is a unique vertex in $\mathcal{H}$. Let $E_{i,j}$ denote the cell at row i and column j. For each suppressed cell $E_{i,j}$ of $\mathcal{T}$, there is a unique edge e in $\mathcal{H}$ between the vertex of row p and the vertex of column q. Recall that each suppressed cell in a table is accompanied by an upper and a lower bound on its possible value. The direction of the edge e is determined by the relationship between of the value and the bounds of the cell $E_{i,j}$ as follows:

- if the value is strictly between the bounds, then e is undirected;
- if the value is equal to the lower bound, then e points from the row vertex to the column vertex; and
- if the value is equal to the upper bound, then e points from the column vertex to the row vertex.

Figure 3 illustrates a table and its suppressed graph. The next theorem demonstrates part of the relationship between a table and its suppressed graph.

row column index	a	b	c	row sum
1	$\boxed{0}$	$\boxed{9}$	1	10
2	$\boxed{9}$	$\boxed{9}$	$\boxed{0}$	18
3	6	$\boxed{0}$	$\boxed{5}$	11
column sum	15	18	6	



In the above 3 × 3 table, the number in each cell is the value of that cell. A cell with a box is a suppressed cell. The lower and upper bounds of the suppressed cells are 0 and 9. The graph below the table is the suppressed graph of the table. Vertex R_p corresponds to row p, and vertex C_q to column q.

FIG. 3. *A table and its suppressed graph.*

THEOREM 3.2 (Gusfield [25]). *Let $\mathcal{T}$ be a table. Let $\mathcal{H}$ be the suppressed graph of $\mathcal{T}$. Then, a suppressed cell of $\mathcal{T}$ is not an invariant cell if and only if its corresponding edge in $\mathcal{H}$ is contained in a traversable cycle of $\mathcal{H}$.*

Proof. The proof is based on the fact that along a traversable cycle in $\mathcal{H}$, the values of the corresponding suppressed cells of $\mathcal{T}$ can be slightly adjusted to obtain another bounded feasible assignment. □

3.3. Combining the two approaches. The cycles of $\mathcal{H}$ can be related to the feasible assignments of $\mathcal{T}$ via two edge-labeling processes, *direction-blind labeling* and *traversable labeling*, described as follows.

3.3.1. Direction-blind labeling. This labeling process applies to the direction-blind cycles of $\mathcal{H}$. Because $\mathcal{H}$ is bipartite, every edge-simple direction-blind cycle of $\mathcal{H}$ is of even length. Consequently, the edges of an edge-simple direction-blind cycle of $\mathcal{H}$ can be alternately labeled with $+1$ and -1. Such a labeling is called a *direction-blind labeling*. Observe that every vector of $CS(\mathcal{H})$ can be decomposed into an edge-disjoint set of edge-simple direction-blind cycles of $\mathcal{H}$. Therefore, this labeling process can be extended to every vector of $CS(\mathcal{H})$ by direction-blindly labeling each cycle in a decomposition of that vector.

A direction-blindly labeled vector of $CS(\mathcal{H})$ can be regarded as an assignment to the suppressed cells of $\mathcal{T}$: if the corresponding edge of a suppressed cell is in the given labeled vector of $CS(\mathcal{H})$, then the value assigned to that cell is the label of the corresponding edge; otherwise, the value is 0. The following lemma describes a relationship between the direction-blindly labeled vectors of $CS(\mathcal{H})$ and the vectors of $UK(\mathcal{T})$.

LEMMA 3.3. *Every direction-blindly labeled vector of $CS(\mathcal{H})$ is also a vector of $UK(\mathcal{T})$.*

Proof. Let α be the original assignment to the suppressed cells in $\mathcal{T}$. Let β be a direction-blindly labeled vector of $CS(\mathcal{H})$. To show $\beta \in UK(\mathcal{T})$, it suffices to prove that $\alpha + \beta$ and α have the same row and column sums. Equivalently, it suffices to show as follows that the sum of β over the suppressed cells in each column or row L of $\mathcal{T}$ is 0. Observe that L is actually a vertex in $\mathcal{H}$. Also, the suppressed cells of L in $\mathcal{T}$ are the edges incident to L in $\mathcal{H}$. Therefore, the sum of β over the suppressed cells of L in $\mathcal{T}$ equals the sum of β over the edges incident to L in $\mathcal{H}$. Consequently, the alternate labeling rule of the direction-blind labeling process guarantees that both sums are 0. □

3.3.2. Traversable labeling. This labeling process applies to the traversable cycles of $\mathcal{H}$. Because $\mathcal{H}$ is a bipartite graph, the edges of an edge-simple traversable cycle of $\mathcal{H}$ can be alternately labeled $+1$ and -1 such that

- the undirected edges may be labeled $+1$ or -1;
- all directed edges from column vertices to row vertices are labeled -1; and
- all directed edges from row vertices to column vertices are labeled $+1$.

Such a labeling is called a *traversable labeling*. This labeling process can be extended to every vector of $TCS(\mathcal{H})$ that can be decomposed into an edge-disjoint set of edge-simple traversable cycles of $\mathcal{H}$. Note that not all vectors in $TCS(\mathcal{H})$ have such decompositions.

Because a traversable labeling is a special case of a direction-blind labeling, a traversably labeled vector of $TCS(\mathcal{H})$ can also be regarded as an assignment to the suppressed cells of $\mathcal{T}$. The following lemma describes a relationship between the traversably labeled vectors of $TCS(\mathcal{H})$ and the vectors of $BK(\mathcal{T})$.

LEMMA 3.4. *Every traversably labeled vector of $TCS(\mathcal{H})$ is also a vector in $BK(\mathcal{T})$.*

Proof. Let α be the original assignment to the suppressed cells in $\mathcal{T}$. Let β be a traversably labeled vector of $TCS(\mathcal{H})$. To show that $\beta \in BK(\mathcal{T})$, it suffices to find a *nonzero* number c such that $\alpha + c{\cdot}\beta$ is a bounded assignment of $\mathcal{T}$.

To choose c, an explanation of the traversable labeling process is helpful. In a traversable labeling, an edge e is labeled $+1$ only if either e is an undirected edge or e is a directed edge from a row vertex to a column vertex. In either case, the value of the suppressed cell corresponding to e is strictly less than the upper bound of that cell. Therefore, a new bounded feasible assignment can be obtained by increasing the value of the cell corresponding to e. Similarly, if e is labeled -1, a new bounded feasible assignment can be obtained by decreasing the value of the cell corresponding to e.

Now c is chosen as follows. Let $E_{i,j}$ denote the cell at row i and column j. Let $V_{i,j}$, $U_{i,j}$, and $L_{i,j}$ denote, respectively, the value, the upper bound, and the lower bound of a suppressed cell $E_{i,j}$. Let c be the minimum of $\{U_{i,j} - V_{i,j} \mid$ the edge corresponding to cell $E_{i,j}$ is labeled $+1$ in $\beta\} \cup \{V_{i,j} - L_{i,j} \mid$ the edge corresponding to cell $E_{i,j}$ is labeled -1 in $\beta\}$. If the minimum is $+\infty$, let c be an arbitrary positive number.

By the minimality of c, the three rules of the traversable labeling process concerning the edge directions imply that $\alpha + c{\cdot}\beta$ satisfies the bounds of all suppressed cells. The alternate labeling rule of the traversable labeling process guarantees that both $\alpha + c{\cdot}\beta$ and α yield the same row and column sums. □

4. A linear invariant test algorithm. Let $\mathcal{T}$ be a table and let $\mathcal{H}$ be its suppressed graph. Let F be a given linear combination of the suppressed cells in $\mathcal{T}$. As discussed in §3.1, Lemma 3.1 says that the linear invariant testing for F can be restricted to a basis of $BK(\mathcal{T})$ but does not suggest how such a basis can be efficiently found. The Strongly Connected Table Basis Theorem can provide such a simply found basis by exploiting the relationship between $CS(\mathcal{H})$ and $BK(\mathcal{T})$.

This section states the table basis theorem and applies it to the linear invariant test problem; the proof of the theorem is given in §5.

In the table basis theorem, a *direction-blindly labeled basis of* $CS(\mathcal{H})$ is a basis of $CS(\mathcal{H})$ together with a direction-blind labeling for each basis vector. A *traversably labeled basis of* $TCS(\mathcal{H})$ is a basis of $TCS(\mathcal{H})$ together with a traversable labeling for each basis vector.

THEOREM 4.1 (The Strongly Connected Table Basis Theorem). *Let $\mathcal{T}$ be a table. Let $\mathcal{H}$ be the suppressed graph of $\mathcal{T}$. Let $\mathcal{H}_1, \cdots, \mathcal{H}_k$ be the strongly connected components of $\mathcal{H}$. Let Ω_i be a direction-blindly labeled basis of $CS(\mathcal{H}_i)$. Then $\cup_{i=1}^{k} \Omega_i$ is a basis of $BK(\mathcal{T})$.*

Note that a basis for $CS(\mathcal{H}_i)$ can be easily found via Fact 2.1 by first computing a direction-blind spanning tree of $\mathcal{H}_i$ and then computing the fundamental cycles of that tree. In addition, these fundamental cycles are so well structured that it is very easy to direction-blindly label these cycles and evaluate F at the resulting vectors of $BK(\mathcal{H})$. Therefore, the table basis theorem provides an efficient way to compute a useful basis for $BK(\mathcal{T})$, leading to the following theorem.

THEOREM 4.2 (Linear Invariant Test). *Let $\mathcal{T}$ be a table. Let $\mathcal{H}$ be the suppressed graph of $\mathcal{T}$. Let n denote the number of rows and columns in $\mathcal{T}$. Let m denote the number of the suppressed cells in $\mathcal{T}$.*

1. *In sequential computation, given $\mathcal{H}$, the linear invariant test problem for $\mathcal{T}$ can be solved in $O(m + n)$ time.*

2. *In parallel computation, given $\mathcal{H}$, the linear invariant test problem for $\mathcal{T}$ can be solved in $O(\log^2 n)$ time using $M(n)$ processors on an exclusive-read exclusive-write parallel random access machine.*

To prove Theorem 4.2, an algorithm for the linear invariant test problem is presented in §4.1. The sequential and parallel implementations of this algorithm and their complexities are discussed in §4.2.

4.1. Describing the linear invariant test algorithm. Let $\mathcal{T}$ be a table. Let $\mathcal{H}$ be the suppressed graph of $\mathcal{T}$. Let F be a given linear combination of the suppressed cells in $\mathcal{T}$. Using Lemma 3.1, Theorem 4.1, and Fact 2.1, the linear invariant testing for F can be carried out as follows:

1. Compute the strongly connected components $\mathcal{H}_1, \cdots, \mathcal{H}_k$ of $\mathcal{H}$.

2. For each $\mathcal{H}_i$, first compute a direction-blind spanning tree T_i of $\mathcal{H}_i$. Next, let Ω_i be the set of all fundamental cycles induced by T_i. Then, direction-blindly label each cycle in Ω_i.

3. Evaluate F at each vector in the labeled Ω_i for all Ω_i. Then, F is a linear invariant of $\mathcal{T}$ if and only if $F(\gamma) = 0$ for all $\gamma \in \Omega_i$ and for all Ω_i.

Remark. In this algorithm, T_i need not be a structured tree such as a depth-first search or a breadth-first search tree. This structural freedom is essential for obtaining an efficient parallel implementation because structured trees often have high parallel complexity [33], [1], [2], [22].

Note that the total size of the bases Ω_i may far exceed $m + n$. Therefore, to achieve a sequential complexity of $O(m + n)$ time, the labeled cycles in the sets Ω_i must not be explicitly enumerated. It is shown below that the choice of fundamental cycles for the sets Ω_i allows efficient evaluation of F without explicitly enumerating the labelings of the sets Ω_i.

For each edge t in T_i, let $D(t)$ denote the depth of t in T_i, i.e., the number of edges in the tree path from the root to t. Similarly, for each vertex v in T_i, let $D(v)$ denote the depth of v in T_i, i.e., the number of edges in the tree path from the root to v.

For each nontree edge d in $\mathcal{H}_i$, let $B(d)$ be the fundamental cycle formed by d and the tree path of T_i between the endpoints of d. The *canonical labeling* of $B(d)$ is the unique direction-blind labeling of $B(d)$ where d is labeled $+1$. The canonical labelings of all $B(d)$ can be efficiently computed by labeling the edges of $\mathcal{H}_i$ as follows:

- Each nontree edge d of $\mathcal{H}_i$ is labeled by $+1$.
- Each tree edge t of T_i is labeled by $(-1)^{1+D(t)}$.

With these labelings, every nontree edge d in $\mathcal{H}_i$ is considered a real vector in the same way as a direction-blindly labeled cycle is in §3.3. Similarly, for a vertex u and a descendant v in T_i, the labeled tree path, denoted by $P(u, v)$, from u to v in T_i is considered a real vector. Based on these conventions, the next lemma shows how to use the above labeling scheme to compute the canonical labelings of all $B(d)$.

LEMMA 4.3. *Let r_i be the root of T_i. Let d be a nontree tree edge in $\mathcal{H}_i$ with endpoints x and y. Then, $B(d) = d + P(r_i, x)\cdot(-1)^{D(x)} + P(r_i, y)\cdot(-1)^{D(y)}$.*

Proof. Let z be the least common ancestor of x and y in T_i. Note that $D(x) = -D(y)$ because $\mathcal{H}_i$ is bipartite. Thus, by definition, $B(d) = d + P(z, x)\cdot(-1)^{D(x)} + P(z, y)\cdot(-1)^{D(y)}$. The lemma then follows from the fact that $(-1)^{D(x)} + (-1)^{D(y)} = 0$, $P(r_i, x) = P(r_i, z) + P(z, x)$, and $P(r_i, y) = P(r_i, z) + P(z, y)$. □

With Lemma 4.3, the values of F at each Ω_i can be computed by executing the following steps for $\mathcal{H}_i$:

1. For each edge e, let $W(e)$ be the coefficient of the suppressed cell e in F.
2. For each tree edge t, compute $W(t)\cdot(-1)^{1+D(t)}$, the *single-term value* of t.
3. For each vertex v, let $S(v)$ be the sum of the single-term values over the tree path from the root to v.
4. For each nontree edge d, compute $L(d) = S(x)\cdot(-1)^{D(x)} + S(y)\cdot(-1)^{D(y)} + W(d)$, where x and y are the endpoints of d.

LEMMA 4.4. *Let d be a nontree tree edge in $\mathcal{H}_i$. Then, $F(B(d)) = L(d)$.*

Proof. Let r_i be the root of T_i. Let x and y be the endpoints of d. The proof follows from Lemma 4.3 and the equalities $F(d) = W(d)$, $F(P(r_i, x)) = S(x)$, and $F(P(r_i, y)) = S(y)$. □

Summarizing the above discussion, Fig. 4 presents an algorithm for the linear invariant test problem. The correctness of the algorithm is stated in the next lemma.

LEMMA 4.5. *F is a linear invariant of $\mathcal{T}$ if and only if $L(d) = 0$ for all $\mathcal{H}_i$ and for all nontree edges d in $\mathcal{H}_i$.*

Proof. The proof follows from Lemma 3.1, Theorem 4.1, Fact 2.1, and Lemma 4.4. □

4.2. Implementations and complexities. This section discusses the sequential and parallel complexities of the linear invariant test algorithm in Fig. 4.

The following lemma analyzes the sequential complexity of the algorithm.

LEMMA 4.6. *The linear invariant test algorithm in Fig. 4 runs in $O(m + n)$ sequential time.*

Proof. The lemma follows directly from the following facts:

1. The strongly connected components of $\mathcal{H}$ can be found in $O(m + n)$ time.
2. A direction-blind spanning tree for each $\mathcal{H}_i$ can be computed in $O(m + n)$ total time.

```
**Procedure** Linear Invariant Test
**Input**: the suppressed graph $\mathcal{H}$ of a table $\mathcal{T}$ and a linear combination $F$ of
the suppressed cells in $\mathcal{T}$. Remark: $\mathcal{T}$ is not part of the input.
**Output**: a yes–no answer to whether $F$ is a linear invariant of $\mathcal{T}$.
**begin**
    1. Compute the strongly connected components $\mathcal{H}_1, \cdots, \mathcal{H}_k$ of $\mathcal{H}$.
    2. **for** each $\mathcal{H}_i$ **do**
        **begin**
            2-1. Compute a direction-blind spanning tree $T_i$ of $\mathcal{H}_i$.
            2-2. For each edge $e$, let $W(e)$ be the coefficient of the suppressed cell $e$ in $F$.
            2-3. For each tree edge $t$, let $D(t)$ be the depth of $t$ in $T_i$,
                 where the edges incident to the root are of depth 1.
            2-4. For each vertex $v$, let $D(v)$ be the depth of $v$ in $T_i$, where
                 the root is of depth 0.
            2-5. For each tree edge $t$, compute $W(t)\cdot(-1)^{1+D(t)}$, the
                 *single-term value* of $t$.
            2-6. For each vertex $v$, let $S(v)$ be the sum of the single-term
                 values over the tree path from the root to $v$.
            2-7. For each nontree edge $d$, compute $L(d)=S(x)\cdot(-1)^{D(x)}$
                 $+S(y)\cdot(-1)^{D(y)} + W(d)$, where $x$ and $y$ are the endpoints of $d$.
                 (Remark: $D(x) = -D(y)$ because $\mathcal{H}$ is bipartite.)
        **end**.
    3. **return** "$F$ is a linear invariant" if and only if $L(d) = 0$ for all $\mathcal{H}_i$ and
       for all nontree edges $d$ in $\mathcal{H}_i$.
**end**.
```

FIG. 4. *Linear invariant testing.*

3. The coefficients $W(e)$ for the edges in all $\mathcal{H}_i$ can be obtained in $O(m + n)$ total time.

4. The depths $D(t)$ of the tree edges, the depths $D(v)$ of the vertices, the single-term values $W(t)\cdot(-1)^{1+D(t)}$ of the tree edges, and the sums $S(v)$ for the vertices can be computed in $O(n)$ total time all by a top-down traversal in each T_i.

5. With the above computation done, the values $L(d)$ for the nontree edges can be found in $O(m)$ total time by processing each nontree edge in constant time.

6. The final output can be found by examining whether the values $L(d)$ are all zero in $O(m)$ total time. □

The following lemma analyzes the parallel complexity of the linear invariant test algorithm, and thus completes the proof of Theorem 4.2.

LEMMA 4.7. *The linear invariant test algorithm in Fig.* 4 *runs in* $O(\log^2 n)$ *parallel time using* $M(n)$ *processors on an* EREW PRAM.

Proof. The lemma follows directly from the following facts:

1. The strongly connected components of $\mathcal{H}$ can be found in $O(\log^2 n)$ time using $M(n)$ processors [29].

2. A direction-blind spanning tree for each strongly connected component can be computed in $O(\log^2 n)$ time using $O(m + n)$ or fewer processors by undirected connectivity algorithms [10], [39].

3. The coefficients $W(e)$ for the edges in all $\mathcal{H}_i$ can be obtained in constant time using $O(m)$ processors.

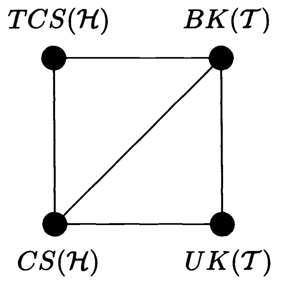


FIG. 5. *The proof scheme of the Strongly Connected Table Basis Theorem.*

4. The depths $D(t)$ of the tree edges and the depths $D(v)$ of the vertices can be computed in $O(\log n)$ time using $O(n/\log n)$ processors by tree contraction techniques [31], [5], [8], [9], [30].

5. With the depths obtained, the single-term values of the tree edges and the sums $S(v)$ for the vertices can be computed in $O(\log n)$ time using $O(n/\log n)$ processors also by tree contraction techniques.

6. With the above computation done, the values $L(d)$ for the nontree edges can be found in constant time using $O(m)$ processors.

7. The final output can be then found by examining whether the values $L(d)$ are all zero in $O(\log n)$ time using $O(m)$ processors. □

5. The Strongly Connected Table Basis Theorem. Let $\mathcal{T}$ be a table. Let $\mathcal{H}$ be the suppressed graph of $\mathcal{T}$. The table basis theorem characterizes the relationship between $CS(\mathcal{H})$ and $BK(\mathcal{T})$. Its proof exploits the five pairwise relationships among $CS(\mathcal{H})$, $TCS(\mathcal{H})$, $UK(\mathcal{T})$, and $BK(\mathcal{T})$ indicated in Fig. 5.

The CS–TCS relationship has been characterized by the cycle space theorem in §2.2. The CS–UK, UK–BK, and TCS–BK relationships are discussed in §§5.1–5.3, respectively. The proof of the table basis theorem is derived from the above four relationships in §5.4.

5.1. ***CS* versus *UK*.**

LEMMA 5.1. *Every direction-blindly labeled basis Ω of $CS(\mathcal{H})$ is also linearly independent in $UK(\mathcal{T})$.*

Proof. Let Ω' be the unlabeled version of Ω. An integer matrix for Ω and a corresponding Z_2 matrix for Ω' are constructed as follows. The vectors of Ω are regarded as integer vectors whose components are $+1$, 0, or -1. Let N be the integer matrix consisting of the vectors of Ω as rows. Similarly, the vectors of Ω' are regarded as Z_2 vectors. Let N' be the Z_2 matrix consisting of the vectors of Ω' as rows. Note that $N = N' \pmod 2$.

Let k be the number of vectors in Ω'. Because Ω' is a basis of $CS(\mathcal{H})$, N' contains a $k \times k$ submatrix J' such that $\det J' \neq 0 \pmod 2$. Let J be the submatrix of N corresponding to J'. Then $J = J' \pmod 2$ and $\det J = \det J' \neq 0 \pmod 2$. Therefore, $\det J$ is a nonzero integer and Ω is linearly independent in $UK(\mathcal{T})$. □

LEMMA 5.2. *$CS(\mathcal{H})$ has some direction-blindly labeled basis Ω such that every vector of $UK(\mathcal{T})$ is a linear combination of vectors in Ω.*

Proof. Ω is constructed as follows. Arbitrarily choose a spanning tree in each connected component of $\mathcal{H}$. Let $e_1, \cdots, e_k$ be the nontree edges of $\mathcal{H}$. For each e_i, let $B(e_i)$ be the cycle formed by e_i and the tree path between the endpoints of e_i. Let Ω be the set of all $B(e_i)$. By Fact 2.1, Ω is a basis of $CS(\mathcal{H})$. Now direction-blindly label Ω with each e_i labeled $+1$ in $B(e_i)$.

To finish the proof, it suffices to verify that every vector $\alpha \in UK(\mathcal{T})$ is a linear combination of vectors in Ω. Let c_i be the component of α at e_i. Let $\beta = \alpha - \sum_{i=1}^{k} c_i{\cdot}B(e_i)$. To prove that α is a linear combination of vectors in Ω, it suffices to show $\beta = 0$ as follows. Observe that e_i appears in $B(e_i)$ but not in the other fundamental cycles. Furthermore, in the right-hand side of the above equality, the components of α and $c_i{\cdot}B(e_i)$ at e_i cancel each other. Thus, the component of β at each e_i is 0. Next, by Lemma 3.3, the cycles $B(e_i)$ are all in $UK(\mathcal{T})$. Hence, β is also in $UK(\mathcal{T})$. Therefore, β has the following two properties: (1) the components of β at the nontree edges are all 0, and (2) for each vertex $x \in \mathcal{H}$, the sum of the components of β at the edges incident to x is also 0. These two properties together imply that the components of β are all 0. Consequently, $\beta = 0$ and $\alpha = \sum_{i=1}^{k} c_i{\cdot}B(e_i)$. □

THEOREM 5.3. *$CS(\mathcal{H})$ and $UK(\mathcal{T})$ have the same dimension. Furthermore, every direction-blindly labeled basis of $CS(\mathcal{H})$ is a basis of $UK(\mathcal{T})$.*

Proof. By Lemma 3.3, a direction-blindly labeled basis of $CS(\mathcal{H})$ is a subset of $UK(\mathcal{T})$. Therefore, Lemmas 5.1 and 5.2 together imply the dimension equality. The dimension equality and Lemma 5.1, in turn, imply the basis embedding. This finishes the proof. □

5.2. *UK* versus *BK*.

LEMMA 5.4. *Every traversably labeled basis of $TCS(\mathcal{H})$ is linearly independent in $BK(\mathcal{T})$. Consequently, $\dim TCS(\mathcal{H}) \leq \dim BK(\mathcal{T})$.*

Proof. The same as the proof of Lemma 5.1. □

Note that the proof of Lemma 5.2 does not seem to extend to $TCS(\mathcal{H})$ and $BK(\mathcal{T})$ because it is difficult to find a basis for $TCS(\mathcal{H})$ consisting of traversable cycles as well-structured as fundamental cycles.

THEOREM 5.5. *If every connected component of $\mathcal{H}$ is strongly connected, then $BK(\mathcal{T}) = UK(\mathcal{T})$.*

Proof. Because $BK(\mathcal{T})$ is a subspace of $UK(\mathcal{T})$, it suffices to show $\dim UK(\mathcal{T}) \leq \dim BK(\mathcal{T})$ as follows. First, $\dim UK(\mathcal{T}) = \dim CS(\mathcal{H})$ by Theorem 5.3. Second, $\dim CS(\mathcal{H}) = \dim TCS(\mathcal{H})$ by Theorem 2.5. Third, $\dim TCS(\mathcal{H}) \leq \dim BK(\mathcal{T})$ by Lemma 5.4. Thus, $\dim UK(\mathcal{T}) \leq \dim BK(\mathcal{T})$. □

THEOREM 5.6. *Let X be the set of edges in $\mathcal{H}$ that are not in its strongly connected components. Let $\mathcal{T}'$ be the same table as $\mathcal{T}$ except that the cells corresponding to the edges in X are also published. Then $BK(\mathcal{T}) = UK(\mathcal{T}')$.*

Proof. First, from Theorem 3.2, $BK(\mathcal{T}) = BK(\mathcal{T}')$. Next, because the suppressed graph of W is $\mathcal{H} - X$, by Theorem 5.5, $BK(\mathcal{T}') = UK(\mathcal{T}')$. Therefore, $BK(\mathcal{T}) = UK(\mathcal{T}')$. □

5.3. *TCS* versus *BK*.

THEOREM 5.7. *Every direction-blindly labeled basis of $TCS(\mathcal{H})$ is also a basis of $BK(\mathcal{T})$.*

Proof. Let X be the set of edges in $\mathcal{H}$ that are not in its strongly connected components. Let $\mathcal{T}'$ be the same table as $\mathcal{T}$, except that the cells corresponding to the edges in X are also published. First, from Theorem 2.6, every direction-blindly labeled basis of $TCS(\mathcal{H})$ is also a direction-blindly labeled basis of $CS(\mathcal{H} - X)$. Second, because the suppressed graph of W is $\mathcal{H} - X$, by Theorem 5.3, every direction-blindly labeled basis of $CS(\mathcal{H} - X)$ is a basis of $UK(\mathcal{T}')$. Third, from Theorem 5.6, every basis of $UK(\mathcal{T}')$ is also a basis of $BK(\mathcal{T})$. Thus, the theorem is true. □

5.4. *CS* versus *BK*.

THEOREM 4.1 (The Strongly Connected Table Basis Theorem). *Let $\mathcal{T}$ be a table. Let $\mathcal{H}$ be the suppressed graph of $\mathcal{T}$. Let $\mathcal{H}_1, \cdots, \mathcal{H}_k$ be the strongly connected components of $\mathcal{H}$. Let Ω_i be a direction-blindly labeled basis of $CS(\mathcal{H}_i)$. Then $\cup_{i=1}^{k}\Omega_i$ is a basis of $BK(\mathcal{T})$.*

Proof. Let X be the set of edges in $\mathcal{H}$ that are not in its strongly connected components. Because the graphs $\mathcal{H}_i$ are the connected components of $\mathcal{H} - X$, the set $\cup_{i=1}^{k}\Omega_i$ is a direction-blindly labeled basis of $CS(\mathcal{H} - X)$. Then, by Theorem 2.6, $\cup_{i=1}^{k}\Omega_i$ is a direction-blindly labeled basis of $TCS(\mathcal{H})$. Finally, from Theorem 5.7, $\cup_{i=1}^{k}\Omega_i$ is a basis of $BK(\mathcal{T})$. This finishes the proof of the theorem. □

6. Discussion. Sometimes it is very useful to consider the notion that a linear combination of suppressed cells is called an invariant if and only if it has the same value at all bounded feasible assignments whose cell values are all *integers*. The integrality constraint in this notion does not change the nature of the linear invariant test problem. The set of linear constraints induced by a table is *totally unimodular* [32]. Therefore, given a table and a linear combination of suppressed cells, if the original assignment to the suppressed cells consists only of integers, then the maximum and minimum values of the given combination can be achieved at bounded feasible assignments with integer cell values. Consequently, if the original assignment to the suppressed cells consists only of integers, then the two notions of a linear invariant are equivalent.

Acknowledgments. The first author is grateful to Mike Fischer for providing financial support and other help at Yale University. The authors both wish to thank Larry Cox and Gordon Sande for helpful discussions and pointers. The authors also wish to thank the two anonymous referees for their extremely prompt, thorough, and helpful reports.

REFERENCES

[1] A. AGGARWAL AND R. J. ANDERSON, *A random* NC *algorithm for depth first search*, Combinatorica, 8 (1988), pp. 1–12.

[2] A. AGGARWAL, R. J. ANDERSON, AND M. Y. KAO, *Parallel depth-first search in general directed graphs*, SIAM J. Comput., 19 (1990), pp. 397–409.

[3] A. AHO, J. HOPCROPFT AND J. ULLMAN, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

[4] ———, *Data Structures and Algorithms*, Addison-Wesley, Reading, MA, 1983.

[5] R. J. ANDERSON AND G. L. MILLER, *Deterministic parallel list ranking*, Algorithmica, 6 (1991), pp. 859–868.

[6] C. BERGE, *Graphs*, 2nd revised ed., North-Holland, New York, 1985.

[7] G. J. BRACKSTONE, L. CHAPMAN, AND G. SANDE, *Protecting the confidentiality of individual statistical records in Canada*, in Proc. Conf. of the European Statisticians 31st Plenary Session, Geneva, 1983.

[8] R. COLE AND U. VIDHKIN, *The accelerated centroid decomposition technique for optimal tree evaluation in logarithmic time*, Algorithmica, 3 (1988), pp. 329–346.

[9] ———, *Faster optimal prefix sums and list ranking*, Inform. and Comput., 81 (1989), pp. 334–352.

[10] ———, *Approximate parallel scheduling. Part* II: *Applications to optimal parallel graph algorithms in logarithmic time*, Inform. and Comput., 91 (1991), pp. 1–47.

[11] S. A. COOK, *A taxonomy of problems with fast parallel algorithms*, Inform. and Control, 64 (1985), pp. 2–22.

[12] D. COPPERSMITH AND S. WINOGRAD, *Matrix multiplication via arithmetic progressions*, J. Symbolic Comput., 9 (1990), pp. 251–280.

[13] T. H. CORMEN, C. L. LEISERSON, AND R. L. RIVEST, *Introduction to Algorithms*, The MIT Press, Boston, 1991.

[14] L. COX, *Disclosure analysis and cell suppression*, in the Proc. Amer. Statist. Assoc., Social Statistics Section, 1975, pp. 380–382.

[15] ———, *Suppression methodology in statistics disclosure*, in the Proc. Amer. Statist. Assoc., Social Statistics Section, 1977, pp. 750–755.

[16] L. COX, *Automated statistical disclosure control*, in the Proc. Amer. Statist. Assoc., Survey Research Method Section, 1978, pp. 177–182.

[17] ———, *Suppression methodology and statistical disclosure control*, J. Amer. Statist. Assoc., Theory and Method Section, 75 (1980), pp. 377–385.

[18] L. H. COX AND G. SANDE, *Techniques for preserving statistical confidentiality*, in Proc. of the 42nd Session of the International Statistical Institute, the International Association of Survey Statisticians, 1979.

[19] D. DENNING, *Cryptography and Data Security*, Addison-Wesley, Reading, MA, 1982.

[20] D. DOBIKIN, R. J. LIPTON, AND S. REISS, *Linear programming is log-space hard for P*, Inform. Process. Lett., 8 (1979), pp. 96–97.

[21] S. FORTUNE AND J. WYLLIE, *Parallelism in random access machines*, in Proc. 10th Annual ACM Sympos. on the Theory of Computing, 1978, pp. 114–118.

[22] H. GAZIT AND G. L. MILLER,*An improved parallel algorithm that computes the* BFS *numbering of a directed graph*, Inform. Process. Lett., 28 (1988), pp. 61–65.

[23] L. M. GOLDSCHLAGER, R. A. SHAW, AND J. STAPLES, *The maximum flow problem is log space complete for P*, Theoret. Comput. Sci., 21 (1982), pp. 105–111.

[24] D. GUSFIELD, *Optimal mixed graph augmentation*, SIAM J. Comput., 16 (1987), pp. 599–612.

[25] ———, *A graph theoretic approach to statistical data security*, SIAM J. Comput., 17 (1988), pp. 552–571.

[26] ———, *A faster algorithm for finding compromised data in* 2-d *tables*, in Proc. IEEE Sympos. on Research in Security and Privacy, 1990, pp. 86–94.

[27] E. HOROWITZ AND S. SAHNI, *Fundamentals of Data Structures*, Computer Science Press, Rockville, MD, 1976.

[28] M. Y. KAO, *Systematic protection of precise information on two dimensional cross tabulated tables*, Ph.D. thesis, Yale University, New Haven, CT, 1986.

[29] R. KARP AND V. RAMACHANDRAN, *A survey of parallel algorithms for shared-memory machines*, in Handbook Theoret. Comput. Sci., J. van Leeuwen, ed., Elsevier Science Publishers B. V., New York, 1990, pp. 869–941.

[30] S. R. KOSARAJU AND A. L. DELCHER, *Optimal parallel evaluation of tree-structured computations by raking*, in Proc. 3rd Aegean Workshop on Computing: VLSI Algorithms and Architectures, J. H. Reif, ed., Lecture Notes in Computer Science 319, Springer-Verlag, Berlin, New York, 1988, pp. 101–110.

[31] G. L. MILLER AND J. H. REIF, *Parallel tree contractions and its applications*, in Proc. 26th Annual IEEE Sympos. on Foundations of Computer Science, 1985, pp. 478–489.

[32] K. MURTY, *Linear and Combinatorial Programming*, John Wiley, New York 1976.

[33] J. H. REIF, *Depth-first search is inherently sequential*, Inform. Process. Lett., 20 (1985), pp. 229–234.

[34] G. SANDE, *Towards automated disclosure analysis for establishment based statistics*, Tech. Rep., Statistics Canada, 1977.

[35] ———, *A theorem concerning elementary aggregations in simple tables*, Tech. Rep., Statistics Canada, 1978.

[36] ———, *Automated cell suppression to preserve confidentiality of business statistics*, Statist. J. United Nations, 2 (1984), pp. 33–41.

[37] ———, *Confidentiality and polyhedra, an analysis of suppressed entries on cross tabulations*, Tech. Rep., Statistics Canada, unknown date.

[38] R. SEDGEWICK, *Algorithms*, Addison-Wesley, Reading, MA, 1988.

[39] Y. SHILOACH AND U. VISHKIN, *An $O(\log n)$ parallel connectivity algorithm*, J. Algorithms, 3 (1982), pp. 57–67.

[40] P. M. VAIDYA, *Speeding-up linear programming using fast matrix multiplication*, in Proc. 30th Annual IEEE Sympos. on Foundations of Computer Science, IEEE Computer Society, Washington, DC, 1989, pp. 332–337.

SIAM J. DISC. MATH.
Vol. 6, No. 3, pp. 477–490, August 1993

THE LATTICE STRUCTURE OF FLOW IN PLANAR GRAPHS*

SAMIR KHULLER†, JOSEPH (SEFFI) NAOR‡, AND PHILIP KLEIN§

Abstract. Flow in planar graphs has been extensively studied, and very efficient algorithms have been developed to compute max-flows, min-cuts, and circulations. Intimate connections between solutions to the planar circulation problem and with "consistent" potential functions in the dual graph are shown. It is also shown that the set of integral circulations in a planar graph very naturally forms a distributive lattice whose maximum corresponds to the shortest path tree in the dual graph. Further characterized is the lattice in terms of unidirectional cycles with respect to a particular face called the root face. It is shown how to compactly encode the entire lattice and it is also shown that the set of solutions to the min-cost flow problem forms a sublattice in the presented lattice.

Key words. flows, planar graphs, lattice structure, solution encoding

AMS(MOS) subject classifications. 68R10, 05C38, 90B10, 90C35, 90C27

1. Introduction.

Maximum flow has been one of the most well-studied problems in the area of algorithms (both in the fields of computer science and operations research) over the last 40 years. It has applications in solving efficiently a large set of problems, e.g., many VLSI problems, transportation problems, and communication networks.

Flow in planar graphs has been extensively studied, and very efficient algorithms have been developed to compute max-flows, min-cuts, and circulations. There is a wealth of ideas in solving these problems efficiently for this class of graphs [FF], [H], [HJ], [IS], [J], [JV], [MN], [R], [KN]. (The algorithms vary for different versions of the same basic flow problem.) Very efficient parallel and sequential algorithms can also be developed by exploiting the planar structure of the graph.

Recently, Miller and Naor [MN] have noted that the most general formulation of the maximum flow problem in planar graphs must allow for the existence of many sources and sinks. Unlike the case of arbitrary graphs, where sources and sinks can be merged, for planar networks there is no obvious reduction of the multiple source/sink problem to a single source/sink problem. Miller and Naor further showed that the case where the demands of the sources and sinks are fixed is equivalent to a *circulation problem* (with lower bounds on edge capacities). They also gave an efficient algorithm for computing a circulation.

Our objective is to study the *structure* of the set of *integral solutions* to the circulation problem for planar graphs. We review the relation between solutions to the planar circulation problem and consistent potential functions in the dual graph and show that there is a one-to-one correspondence between them. We then show that the set of circulations in a planar graph very naturally forms a distributive lattice (under an appropriate definition of meet and join operations for the lattice). We further characterize the lattice in terms of unidirectional cycles with respect to a particular face called the root face. We show that the top (bottom) element of the lattice is the circulation in which there are no

*Received by the editors March 25, 1991; accepted for publication (in revised form) June 25, 1992.

†Department of Computer Science, University of Maryland, College Park, Maryland 20742. This author was partially supported by National Science Foundation grants CCR-8906949, CCR-9103135, and CCR-9111348. Part of this research was done while this author was supported by an IBM Graduate Fellowship at Cornell University, Ithaca, New York.

‡Computer Science Department, Technion, Haifa 32000, Israel. Part of this work was done while this author was at Stanford University, Stanford, California and was supported by Office of Naval Research contract N00014-88-K-0166.

§Department of Computer Science, Brown University, Providence, Rhode Island 02912. This author's research was supported by National Science Foundation grant CCR-9012357.

clockwise (counterclockwise) residual cycles around the root face. It turns out that the flow functions computed by [H], [HJ], [J], and [MN] correspond to the top element of the circulation lattice. (This is essentially a matter of notation; if we reverse the direction of the dual edges, their algorithms will be computing the bottom element in the lattice.) It is interesting to note that, if a planar graph also contains vertex capacities, i.e., there is a limit on the amount of flow that is allowed to go through a vertex, then the set of feasible circulations does *not* form a lattice (see §3.4).

The lattice representing all feasible circulations is clearly of *exponential* size, since there are exponentially many solutions to the circulation problem. We provide a compact encoding of the entire lattice by providing a directed acyclic graph (dag) such that the predecessor-closed subsets of this partial order correspond to elements in the lattice. Although this dag may be large, its size depends on the maximum edge-capacity—we can represent it succinctly, in polynomial size. This compact encoding of the partial order provides, in turn, a compact encoding of the lattice elements.

The *minimum cost circulation* problem is that of obtaining a circulation of minimum cost in a network whose edges have both capacities and costs per unit of flow. The problem is equivalent to the transshipment problem and has wide applicability to a variety of optimization problems [AMO]. One of the motivations for the research reported here is to investigate new approaches to solving the minimum cost circulation problem in planar networks more efficiently than in arbitrary graphs. For planar networks, we interpret the cost function as a function on the lattice, and we show that it is a modular function. It follows that the set of solutions to the minimum cost circulation problem forms a sublattice. We show that this sublattice consists of the feasible circulations for a network derived from the original network. This allows us to provide a succinct representation for the set of minimum cost circulations as well. Minimizing a modular function over a lattice is a well-known problem in operations research and can be solved by computing the minimum cut (max-flow) in a network obtained from the dag corresponding to the circulation lattice. However, this approach does not directly yield a new polynomial-time min-cost circulation algorithm, since the dag is so large. At the end of this paper, we briefly discuss two possible approaches to obtaining such an algorithm.

Other examples of problems whose solution set has similar structure are the stable marriage problem and the minimum cut problem. Picard and Queyranne [PQ] have shown that the set of all minimum cuts forms a distributive lattice where the join and meet operations are defined as intersection and union, respectively. The structure of the solution set of the stable marriage problem has been extensively investigated in the book by Gusfield and Irving [GI]. They show that the set of all stable marriages also forms a distributive lattice and show that a compact encoding of the entire solution set is possible. They provide many applications of the lattice structure, e.g., computing an egalitarian stable marriage solution.

2. Preliminaries and terminology. We begin by defining the *circulation* problem. Consider a directed graph G, with each edge e having an integral *lower* and *upper* bound on its capacity, denoted respectively by $\ell(e)$ and $u(e)$ ($[\ell, u]$). When we speak of the capacity of an edge without specifying whether it is a lower or upper capacity, we mean its upper capacity.

We are required to find a flow function $f : E \to Z$ that is feasible in that the following two conditions are satisfied:

Capacity constraints. For all $e \in E : \ell(e) \leq f(e) \leq u(e)$ (the flow on each edge is between the lower and upper bounds on its capacity);

Conservation constraints. For all $v \in V : \sum_{e \in \mathrm{in}(v)} f(e) = \sum_{e \in \mathrm{out}(v)} f(e)$. (The flow into each node equals the flow out of the node.)

The circulation problem is that of finding a feasible flow function (such a flow function may not even exist). In the *maximum flow* problem, two distinguished vertices are added to the graph, a source and a sink, and the aim is to maximize the amount of flow entering the sink. Note that a flow problem in which the flow value is prespecified can be reduced to a circulation problem.

We will henceforth be restricting our attention to planar graphs. Let $G = (V, E)$ be a directed embedded planar graph. The graph partitions the plane into connected regions called *faces*. For each edge $e \in E$, let $D(e)$ be the corresponding *dual edge* connecting the two faces bordering e. Let $D(G) = (F, D(E))$ be the *dual graph* of G, where F is the set of faces of G and $D(E) = \{D(e) | e \in E\}$. The dual graph is planar, too, but may contain self loops and multiple edges. We refer to graph G as the *primal graph*.

There is a one-to-one correspondence between primal and dual edges; the direction of a primal edge e induces a direction on $D(e)$. We use a right-hand rule: If the right-hand's thumb points in the direction of e, then the index finger points in the direction of $D(e)$ (with the palm facing downward). We refer to dual edges as capacitated as well, where the capacity of edge $D(e)$ is equal to that of edge e (see Fig. 1).

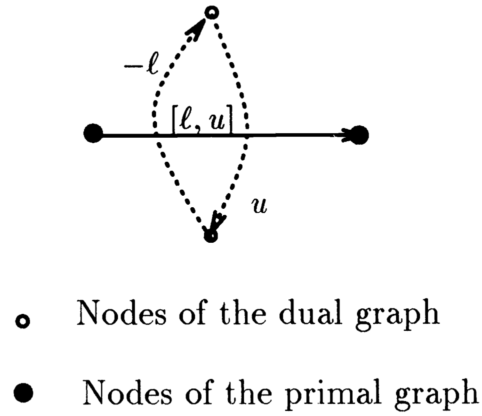


FIG. 1. *Construction of directed dual graph.*

We have the following equivalence rules that relate the orientation of an edge $e = (v \to w)$, the sign of its flow $f(e)$, and its lower and upper capacity bounds:

1. The edge $v \to w$ with flow $f(e)$ is equivalent to the edge $w \to v$ with flow $-f(e)$;
2. The edge $v \to w$ with capacities $[\ell, u]$ is equivalent to the edge $w \to v$ with capacities $[-u, -\ell]$;
3. The edge $v \to w$ with capacities $[\ell, u]$ is equivalent to two antiparallel edges: $v \to w$ of capacities $[0, u]$ and $w \to v$ with capacities $[0, -\ell]$;
4. Let e_1 and e_2 be two parallel edges that are oriented in the same direction with capacities $[\ell_1, u_1]$ and $[\ell_2, u_2]$, respectively. The two edges can be replaced by one edge with capacity $[\ell_1 + \ell_2, u_1 + u_2]$ and flow $f(e_1) + f(e_2)$.

The *residual graph* is defined with respect to a given circulation. Let $e = (v \to w)$ be an edge with capacities $[\ell, u]$ and flow f. In the residual graph, e is replaced by two darts, $v \to w$ with capacities $[0, u - f]$ and $w \to v$ with capacities $[0, f - \ell]$. A directed cycle is said to be *residual* with respect to the given circulation if every edge in the cycle has positive upper capacity in the residual graph.

Miller and Naor [MN] have shown that, for planar graphs, the maximum flow problem should be formulated with respect to many sources and sinks. They show how to reduce this problem to a circulation problem with lower bounds on the edges if the demands of the sources and sinks are given. This is done by returning the flow back from the sinks to the sources via a spanning tree.

An important tool for computing flow functions in planar graphs is the notion of a potential function $p : F \rightarrow Z$ in the dual graph. This function was first introduced by Hassin [H], and its usage was later elaborated by [HJ], [J], and [MN]. Let e be an edge in the graph G and let $D(e) = (g, h)$ be its corresponding edge in the dual graph such that $D(e)$ is directed from g to h. The potential difference over e is defined to be $p(h) - p(g)$. The following proposition, proved in [H] and [J], can be easily verified.

PROPOSITION 2.1. *Let $C = c_1, \dots, c_k$ be a cycle in the dual graph and let $f_1, \dots, f_k$ be the potential differences over the cycle edges. Then $\sum_{i=1}^{k} f_i = 0$.*

It follows from the proposition that the sum of the potential differences over all the edges adjacent to a primal vertex is zero.

A potential function is defined to be *consistent* if the potential difference over each edge is between the upper and lower bounds on the capacity (i.e., $\ell \leq p(h) - p(g) \leq u$). Such a potential function induces a circulation in the graph by defining the flow on an edge as the potential difference over it. Clearly, the flow on the edge satisfies the capacity constraints; by using the previous proposition, it is easy to see that the flow conservation constraints are also satisfied. Once we fix the potential of some particular face as zero, all the other potentials can be normalized with respect to this face. We will assume that all consistent potential functions are normalized, and we call the face whose potential is set to zero the *root face*. We will assume that the planar embedding is such that the root face is the infinite face.

How is a consistent potential function computed? Let us consider the dual graph where each edge $D(e)$ with capacity $[\ell, u]$ is split by rule 3 above, into two antiparallel edges, where one edge has capacity u, and the other capacity $-\ell$. Miller and Naor [MN] show that, if a solution to a circulation problem exists, then there cannot be negative cycles in the dual graph. Hence, a natural way for computing a consistent potential function would be the following: Choose an arbitrary face as the root face; the potential of face h is defined to be the length of the shortest path in the dual graph from the root face to h. It follows from properties of shortest paths that the resulting potential function is indeed consistent. We will refer to this potential function as the *shortest path* potential function.

It is easy to see that there is a one-to-one correspondence between consistent (and normalized) potential functions and circulations. Given a legal circulation C, a corresponding potential function can be constructed. To do so, the capacity of every edge is replaced by its actual flow in C. That is, if the flow on edge e is of value $f(e)$, then edge $D(e)$ is replaced by two parallel edges in opposite directions, where one edge has capacity $f(e)$, and the other has capacity zero. The potential of face h is the length of the shortest path in the dual graph from the root face to h. It is easy to see that this potential function induces circulation C.

We henceforth view a potential function as a vector where the entries correspond to the potentials of the faces, and the potential of the *root face* (an arbitrary but fixed face) is always equal to zero. Since there is a one-to-one correspondence between circulations and potential vectors, we will use both terms to refer to a circulation. We also assume that all potential values are integral, as we are interested only in integral solutions to the circulation problem.

3. The lattice structure. Our aim in this section is to investigate the structure of the set of legal circulations in G. We will show that this set forms a distributive lattice and also explore its structure. Given two consistent vectors P_1 and P_2, we say that $P_1 \geq P_2$ if, for all components i, $P_1(i) \geq P_2(i)$. We say that circulation C_1 *dominates* C_2, if, for their corresponding potential vectors P_1 and P_2, $P_1 \geq P_2$. We use the term $\mathcal{P}$ to refer to the set of all consistent potential vectors. It is easy to see that $\mathcal{P}$ is a partial order under the dominance relation (also written as $(\mathcal{P}, \preceq)$).

We now show that the partial order $(\mathcal{P}, \preceq)$ is, in fact, a *distributive lattice*. A distributive lattice is a partial order in which the following hold:

1. Each pair of elements has a greatest lower bound (g.l.b.), or *meet*, denoted by $a \wedge b$, so that $a \wedge b \preceq a$, $a \wedge b \preceq b$, and there is no element c such that $c \preceq a$, $c \preceq b$ and $a \wedge b \prec c$;
2. Each pair of elements has a least upper bound (l.u.b.), or *join*, denoted by $a \vee b$, so that $a \preceq a \vee b, b \preceq a \vee b$, and there is no element c such that $a \preceq c, b \preceq c$ and $c \prec a \vee b$;
3. The *distributive* laws hold, namely, $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$ and $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$.

We show that $(\mathcal{P}, \preceq)$ is a distributive lattice by presenting appropriate definitions for meet and join.

Given two circulations C_1 and C_2 (represented as P_1 and P_2), we define the meet as the circulation induced by the potential vector $P_m = \min(P_1, P_2)$. Clearly, the face at zero potential in both circulations stays at zero potential. Every face g is assigned a potential equal to $\min(P_1(g), P_2(g))$, where $P_i(g)$ is the potential of g in C_i. Similarly, the join is defined as $P_j = \max(P_1, P_2)$.

The following theorem shows that P_m and P_j are consistent potential vectors, assuming that P_1 and P_2 are consistent.

THEOREM 3.1. *The partial order $(\mathcal{P}, \preceq)$ is a distributive lattice, with the meet and join defined appropriately.*

Proof. We first show that the meet and join are consistent potential assignments. Let g and h be faces in the dual graph bordering primal edge e. The potential across e is $p_1(h) - p_1(g)$ and $p_2(h) - p_2(g)$, respectively, in each circulation. If $p_1(h) \leq p_2(h)$ and $p_1(g) \leq p_2(g)$, then the meet is clearly consistent. (Similarly, if p_2 is the smaller potential for both g and h.) If $p_1(h) \leq p_2(h)$ and $p_2(g) \leq p_1(g)$, then it follows that $\ell \leq p_1(h) - p_2(g) \leq u$ (since $p_1(h) - p_1(g) \geq \ell$, and $p_2(h) - p_2(g) \leq u$). The last case is when $p_2(h) \leq p_1(h)$ and $p_1(g) \leq p_2(g)$, and it follows that $\ell \leq p_2(h) - p_1(g) \leq u$ (since $p_2(h) - p_2(g) \geq \ell$ and $p_1(h) - p_1(g) \leq u$). Hence P_m is a consistent potential assignment.

The proof that P_j is a consistent potential assignment is almost identical. It is also easy to see that they are the g.l.b. and l.u.b., respectively. This establishes that $(\mathcal{P}, \preceq)$ is a lattice.

Let a, b, and c be any integers. Then $\min(a, \max(b, c)) = \max(\min(a, b), \min(a, c))$ and $\max(a, \min(b, c)) = \min(\max(a, b), \max(a, c))$. Hence, the distributive laws hold for the lattice $\mathcal{P}$. □

It is easy to see that a lattice has a unique minimum and maximum, P_b and P_t, referred to as *bottom* and *top*, respectively. We now provide a simple characterization for them. Let us denote by P the shortest path potential vector, in which the potential of a face is exactly its distance from the root face in the dual graph. The following lemma shows that P corresponds precisely to the top of the lattice.

LEMMA 3.2. *The potential vector P_t is equal to P.*

Proof. The shortest path problem can be cast as a linear program, with a variable x_h for each face h, in which the objective is to maximize $\sum_h x_h$, subject to $x_r = 0$, where r is the root face and subject to inequalities $x_g \leq x_h + u(e)$ for each edge $e = h \to g$ of the dual graph. Any vector x satisfying these constraints is a circulation. The top element of the lattice clearly maximizes the objective function over all lattice elements. □

The following lemma, which characterizes the bottom of the lattice, follows by symmetry.

LEMMA 3.3. *Let a potential vector P be computed as follows: The potential of face h in P is the length of the shortest path from h to r, the root face, multiplied by -1. Then the vector P is equal to P_b.*

3.1. Eliminating lower bounds. The existence of the lattice provides us with a simple way of getting rid of lower bounds on edges. To do so, we define a new lattice $\mathcal{P}'$ by normalizing the vectors in $\mathcal{P}$ with respect to P_b, the bottom element of the lattice. Each vector $P \in \mathcal{P}$ is replaced by a new vector P', where $P' = P - P_b$, and subtraction is performed componentwise. This is the same as computing the residual graph with respect to P_b.

LEMMA 3.4. *Let G' be the residual graph of G with respect to the circulation P_b. Then* (1) *the lower bounds on the capacity of the edges in G' are zero, and* (2) *the lattice of feasible circulations in G' is isomorphic to that in G.*

Proof. By the additivity property of flow, each circulation $P \in \mathcal{P}$ can be written as the sum of two circulations, P_b and some other circulation Q. Hence, the lemma follows. □

3.2. Unidirectional cycles and the lattice. In this section, we establish a connection between the lattice and unidirectional cycles. Recall that we assumed the planar embedding was such that the infinite face is the root face. Each simple cycle divides the sphere into two nonempty disjoint sets of faces, called regions. The region containing the root face is designated the *exterior* region; the other region is *interior*. In a traversal of a directed cycle, all faces that border the cycle on its right are in the same region, the cycles *right-hand region*.

DEFINITION 3.5. *A directed cycle is* clockwise *if the cycles right-hand region is interior. Otherwise, the cycle is* counterclockwise.

Let us adopt the following convention that follows from the right-hand rule defined in §2. Pushing positive flow through a directed cycle C is equivalent to increasing the potentials of the faces in C's right-hand region.

A circulation is said to be *maximal* in the clockwise direction ("clockwise maximal," for short) if there are no clockwise residual cycles with respect to the circulation. "Maximal in the counterclockwise direction" is defined similarly.

We begin by characterizing the top and bottom of the lattice.

THEOREM 3.6. *A circulation is clockwise maximal if and only if it corresponds to P_t. A circulation is counterclockwise maximal if and only if it corresponds to P_b.*

Proof. We consider the first statement; the second follows by symmetry. First, we show that P_t is clockwise maximal. Let Γ be any clockwise cycle; we show that Γ is not residual with respect to P_t because some edge of Γ has zero residual capacity.

By Lemma 3.2, P_t is the shortest path vector. Let T be the shortest path tree in the dual graph, rooted at the root face. Since T spans all faces, there must be some face h

in the interior of Γ whose parent g in T is in the exterior of Γ. Then $P(h) = P(g) + b$, where b is the capacity of the edge $D(e) = g \to h$ in the dual graph. Thus b is also the capacity of the edge $e \in \Gamma$ in the primal graph. However, the flow $f(e)$ defined by P is $P(h) - P(g) = b$, so the residual capacity is zero.

Conversely, suppose that P is a circulation with respect to which there exists a clockwise residual cycle Γ. Since Γ is clockwise, the interior does not contain the root face. Since Γ is residual, we can therefore increase the potentials of all faces in the interior by some positive amount without violating the constraints. Hence P is not the top element of the lattice. □

It is tempting to believe that the dominance relation in the lattice can be stated in terms of saturating clockwise cycles. That is, if $P_1 < P_2$, then circulation P_2 can be obtained from P_1 by saturating clockwise cycles. Unfortunately, the following counterexample shows that this is not true. Let c_1 and c_2 be clockwise cycles such that c_1 is contained in the interior of c_2. We construct two circulations P_1 and P_2 such that $P_1 < P_2$. To construct P_1, take P_b and push one unit of flow in the cycle c_1 in the clockwise direction. To construct P_2, take P_b and push one unit of flow in the cycle c_2 in the clockwise direction. Obviously, $P_1 < P_2$, but the only way to obtain circulation P_2 from P_1 is to push a unit of flow in the cycle c_2 in the clockwise direction and, in the cycle c_1, in the counterclockwise direction.

However, in the next section, we will show that every circulation can be obtained from P_b by pushing flow through a set of clockwise cycles.

3.3. The Region-Growing Algorithm. In this section, we give a generic algorithm for obtaining any circulation P, from P_b the bottom circulation of the lattice, by saturating only clockwise cycles. We assume that circulation P is given as input to the algorithm. This algorithm will be used to prove that the difference between P_b (or P_t) and any other circulation is a unidirectional set of cycles. The algorithm can also be used to obtain the top or bottom elements of the lattice from any given circulation. In §5.2 we briefly discuss a possible approach to computing a minimum cost circulation based on the idea of the Region-Growing Algorithm.

The Region-Growing Algorithm.

```
1. Let $Q \leftarrow P_b$
2. Let $R$ be the set of faces on which $Q$ and $P$ agree; $\Delta \leftarrow \min_i\{P(i) - Q(i) | i \in F - R\}$
3. For all faces $f \in F - R : Q(f) \leftarrow Q(f) + \Delta$
4. If $Q \neq P$, then Goto Step 2.
```

The correctness of the algorithm is trivial, and clearly the algorithm can be modified to use the top of the lattice as the initial value of Q. We now prove that the algorithm has an interesting property.

LEMMA 3.7. *The region R remains connected during all stages of the algorithm.*

Proof. By the construction of Lemma 3.4, we can assume without loss of generality that all capacity lower bounds in G are zero and that P_b is the all-zeros circulation. Let f be the assignment of flows to edges defined by the circulation P. Let G' be the graph obtained from G by setting upper bounds as follows:

$$u'(e) = \begin{cases} f(e) & \text{if } f(e) > 0, \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to check that P is the shortest path potential vector for G'. Thus the potential P assigns to a face f is the distance of f from the root face in a graph with nonnegative edge-lengths. Now consider the Region-Growing Algorithm. Because we started with the zero circulation, at any point t in the algorithm, R consists of all faces whose potentials in P are no more than some value, say v_t. That is, R consists of faces whose distance from the root is no more than v_t. Clearly, R is therefore connected. □

The last lemma provides us with the following view of the Region-Growing Algorithm. Initially, R only contains the root face. At each step: Push a certain amount of flow (Δ) on the boundary of R in the clockwise direction; annex to R the faces bordering it whose potential has reached the desired value.

For example, to obtain P_t from any circulation P, we will run the algorithm "backward." Initially, R only contains the root face. At each step, the boundary of R is saturated, and the faces bordering saturated edges are annexed to R. (A saturated edge e is an edge, whose flow has either reached its upper bound or its lower bound, and no more flow can be added to it in the clockwise direction.)

It is easy to see that for an efficient implementation of this algorithm, all we need is a shortest-path tree in the residual dual graph. This tree can be computed in $O(n\sqrt{\log n})$ by Frederickson's algorithm [F], where n is the number of faces in the graph.

3.4. Vertex capacities. An interesting version of planar flow is the case where vertices as well as edges have capacity constraints [KN]. Vertex capacities may arise in various contexts such as computing vertex disjoint paths in graphs [KS] and in various network situations when the vertices denote switches and have an upper bound on their capacities. For the case of general graphs, this problem can be reduced to the version with only edges having capacity constraints by a simple idea of "splitting" vertices into two and forcing all the flow to pass through a "bottleneck" edge in-between. In planar graphs, this reduction may *destroy* the planarity of the graph and thus cannot be used.

The following example shows that the set of feasible circulations with vertex capacity constraints does not form a lattice when circulations are represented by potential vectors. This may explain in part why it is harder to design efficient algorithms for this case.

Let $G = (V, E)$ be a planar graph with five vertices where v_1, v_2, v_3, v_4 form a directed anticlockwise cycle and v_5 is connected to all other vertices as follows: The edges from v_1 and v_3 are directed toward v_5, and the edges from v_5 to v_2 and v_4 are directed away from v_5. The capacity of vertex v_5 is c, and the capacity of each edge is $4c$.

We choose the following two feasible circulations. In circulation C_1, the flow on edges v_1, v_5 and v_5, v_2 is c, and the flow on edges v_3, v_5 and v_5, v_4 is zero. In circulation C_2, the flow on edges v_1, v_5 and v_5, v_2 is zero and the flow on edges v_3, v_5 and v_5, v_4 is c. (The flow on the edges on the cycle v_1, v_2, v_3, v_4 is not important in both C_1 and C_2.)

It is easy to see that either the meet or the join of C_1 and C_2 will generate the circulation in which the flow on edges v_1, v_5 and v_5, v_2 is c and the flow on edges v_3, v_5 and v_5, v_4 is also c. Clearly, this circulation is infeasible (due to the capacity at v_5 being violated).

4. The partial order. There is a partial order associated with every distributive lattice. We will investigate the partial order that is associated with the lattice $\mathcal{P}$. Our exposition will follow Gusfield and Irving [GI, Chap. 2] and Grätzer [G, Chap. 2].

Let $\mathcal{P}[f = i]$ denote the set of all circulations such that the potential of face f is equal to i. Obviously, $\mathcal{P}[f = i]$ induces a sublattice of $\mathcal{P}$. We call a lattice element *irreducible* if, for some face f and potential value i, it is the bottom element of $\mathcal{P}[f = i]$. Let $I(\mathcal{P})$ denote the set of all irreducible elements of the lattice. We define the partial order $(I(\mathcal{P}), \preceq)$ as the partial order on $I(\mathcal{P})$ where the dominance relation is inherited from $\mathcal{P}$. Clearly, $I(\mathcal{P})$ has a unique minimum and maximum since it contains both P_b and P_t.

For a partial order R, a subset S is said to be *closed* in R if, for every $s \in S$, the predecessors of s are also in S. The following theorem is proved in [G, Thm. 9, p. 72] and [GI, Thm. 2.2.1].

THEOREM 4.1. *Define a mapping from the closed subsets of $I(\mathcal{P})$ into $\mathcal{P}$ by $S \mapsto \vee S$. Then this mapping is one-to-one and onto. Moreover, if closed subsets S and S' of $I(\mathcal{P})$ correspond to circulations P and P', respectively, then P dominates P' if and only if $S \subseteq S'$.*

It is clear that $I(\mathcal{P})$ may have exponential size if the capacities are exponential. However, we will see that a different partial order can be constructed such that Theorem 4.2 still holds, yet this partial order has a more regular structure, which enables us to represent it *succinctly*.

The elements P_1 and P_2 are called *consecutive* elements in the lattice $\mathcal{P}$ if P_2 covers P_1; i.e., there is no element Q such that $P_1 < Q < P_2$. Suppose that elements P_1 and P_2 are consecutive and that $P_1 < P_2$. The *minimal difference* between P_1 and P_2 is defined to be the pair (f, i), where f is the face on which P_1 and P_2 differ and i is the potential of f in P_1. (Obviously, the potential of face f in P_2 is $i + 1$.) We denote by $\mathcal{D}$ the set of all minimal differences in $\mathcal{P}$.

Note that we can assume without loss of generality that consecutive elements differ in only one face, since we can assume that there are no edges in the graph whose lower bound on the capacity is equal to the upper bound. One way of seeing this follows from §3.1, where lower bounds are eliminated, and then such edges have zero capacity and can be removed from the graph.

LEMMA 4.2. *Suppose that the potential of face f in P_b and P_t is p and q, respectively. Then, for all i, $p \leq i \leq q$, there exist consistent potential vectors in which face f has potential i.*

Proof. The proof follows from the Region-Growing Algorithm. Run the algorithm so as to obtain the top element of the lattice. For some k, at the end of step $k - 1$, $Q(f) \leq i$, yet at the end of step k, $Q(f) \geq i$. By decreasing Δ appropriately at step k, $Q(f) = i$, and the potential vector obtained is consistent. □

The last lemma implies that, for consecutive elements P_1 and P_2 where $P_1 < P_2$, the potential of face f in P_2 is bigger than its potential in P_1 by precisely one unit. Hence, we can denote a minimal difference by (f, i); i.e., the potential of face f is increased from i to $i + 1$.

A *maximal chain* in a lattice is a chain of consecutive elements that starts at P_b and ends at P_t. An interesting property of distributive lattices is that each maximal chain contains *all* the minimal differences. The minimal differences appear on each maximal chain in some order, and each minimal difference appears exactly once.

We can now define the partial order $(K(\mathcal{P}), \preceq)$. Let $D_1, D_2 \in \mathcal{D}$; then $D_1 < D_2$ if and only if D_1 precedes D_2 on every maximal chain in $\mathcal{P}$. The motivation for defining this partial order follows from Theorem 19 of [G, p. 75], which states that every distributive lattice is isomorphic to a ring of sets. A ring of sets is a distributive lattice where the elements are subsets defined over a base set and the join and meet operations are respectively defined as intersection and union. Let $\mathcal{R}$ denote the ring of sets isomorphic to $\mathcal{P}$. Then, the closed subsets of the partial order $(K(\mathcal{P}))$ are, in fact, in one-to-one correspondence with the elements of $\mathcal{R}$. This leads us to the next theorem, whose proof follows from [GI, Thm. 2.4.4] and which relates the partial orders $I(\mathcal{P})$ and $K(\mathcal{P})$.

THEOREM 4.3. *There is a one-to-one correspondence between the closed subsets of $I(\mathcal{P})$ and $K(\mathcal{P})$.*

We are now ready to simplify the partial order $K(\mathcal{P})$ and define a dag $T(\mathcal{P}) = (\mathcal{D}, E)$, which has a succinct description. The vertex set of $T(\mathcal{P})$ is again $\mathcal{D}$, the set of minimal differences. The edge set of $T(\mathcal{P})$ is defined as follows:

- For face f that takes potential values between p and q, there is a directed chain $(f,p) \to (f,p+1), \to \cdots \to (f,q-1)$ (such a chain is called an f-chain);
- For adjacent faces f and g that take potential values between p_1 and q_1, and p_2 and q_2, respectively, and the edge from f to g has capacity b. There is a "ladder" between the f-chain and the g-chain: $(f,p_1) \to (g,p_1+b), (f,p_1+1) \to (g,p_1+b+1), \dots, (f,x) \to (g,x+b)$, where $x = \min\{q_1 - 1, q_2 - b - 1\}$.

The next theorem relates the closed subsets of the dag $T(\mathcal{P})$ and the elements of $\mathcal{P}$. The intuitive reason for its correctness follows from the fact that the shortest path information can be completely recovered from the constraints on adjacent faces.

THEOREM 4.4. *There is a one-to-one correspondence between the closed subsets of $T(\mathcal{P})$ and the elements of $\mathcal{P}$.*

Proof. Let S be any closed subset of $T(\mathcal{P})$. Let f be any face and denote the lower and upper bounds on its potential values by p and q. Since S is a closed subset, the intersection between the f-chain and S is a subchain starting at (f,p) and ending at (f,x_f), where $x_f \le q-1$. The potential vector corresponding to S is defined as follows: for each face f, assign its potential to be x_f. To see that this is a consistent potential vector, let f and g be any two adjacent faces where the capacity of the edge from f to g is b. If $x_g - x_f > b$, then S cannot be a closed subset, since $(g,x_g) \in S$ where as $(f,x_g - b) \notin S$.

The correspondence in the other direction is proved very similarly. Given a consistent potential vector where face f has potential x_f, the closed subset S is constructed as follows: For each f-chain, the subchain from (f,p) to $(f,x_f - 1)$ belongs to S. Again, for any adjacent faces f and g, since $x_g - x_f \le b$, S is a closed subset. □

We now consider the simple example in Fig. 2(a). The face f_1 is chosen to be the root face. The bottom element of the lattice corresponds to the smallest possible potential vector, which is $(0,1,-4)$ (these are the potentials of the faces f_1, f_2, f_3, respectively. We can now modify the graph by eliminating lower bounds on the edge capacities (by constructing the residual graph with respect to P_b). We now get the graph in Fig. 2(b). We construct its dual graph in Fig. 2(c). This is the graph for which we would like to encode all feasible potential vectors. The range of potentials for both f_2 and f_3 can easily be seen to be $0 \dots 2$. We thus construct the two chains $(f_i,0), (f_i,1)$ $(i = 2,3)$ (see Fig. 3). The ladder edges are added from (f_3,j) to (f_2,j) $(j = 0,1)$. This gives us $T(\mathcal{P})$, whose closed subsets encode all the feasible solutions. Clearly, there are six closed subsets of this dag, i.e., $\Phi, \{A\}, \{A,B\}, \{A,C\}, \{A,B,C\}, \{A,B,C,D\}$. Each closed subset corresponds to a set of minimal differences, which we can add to P_b to generate an integer circulation. These closed subsets are in one-to-one correspondence with the set of circulations of the graph, namely, $(0,1,-4), (0,1,-3), (0,2,-3), (0,1,-2), (0,2,-2), (0,3,-2)$. These circulations are obtained by adding the minimal differences to the potential vector P_b (bottom of the lattice). It is easy to see that the shortest path potential vector corresponds to the top of the lattice.

In the stable marriage problem, it was shown that every partial order can be associated with some instance of the problem [GI]. An interesting question is whether there exists a subset of the set of dags that has some "nice" characterization such that there exists a planar circulation instance that can be associated with each dag in the subset. Unfortunately, it seems that dags corresponding to planar circulation instances have very specialized structure: (i) There is a one-to-one correspondence between faces in

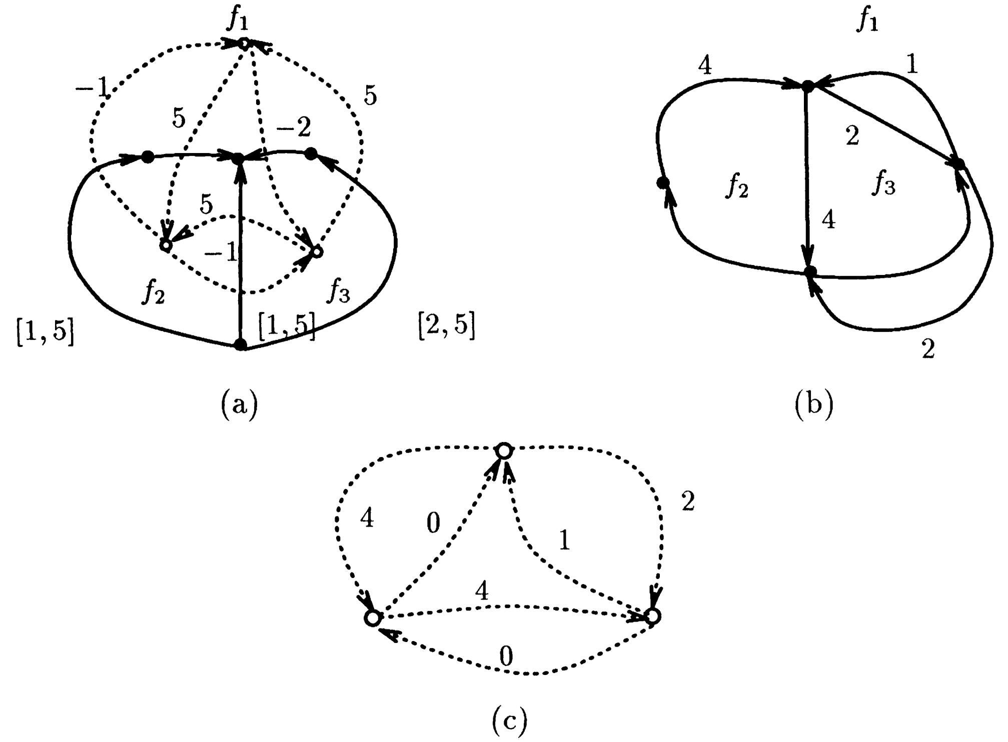


FIG. 2. *Figure to illustrate example*: (a) *graph and its dual graph*; (b) *residual graph with respect to* P_b; (c) *dual graph from residual graph.*

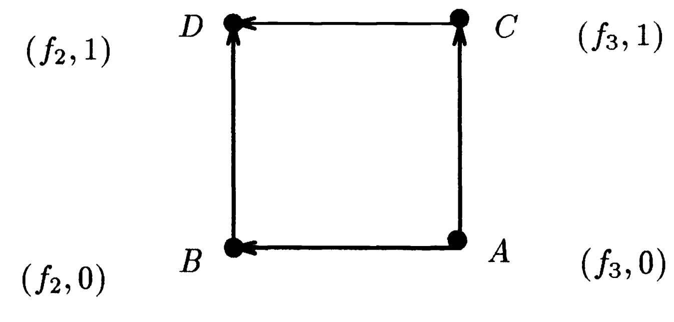


FIG. 3. *Figure to illustrate* dag $T(\mathcal{P})$.

the planar graph and subsets of vertices in the dag that induce an acyclic tournament, either directly or by implication (the f-chains); (ii) There is a special structure connecting these subsets (the "ladders") that must correspond to capacities in the circulation instance.

5. Minimum cost flow. In the minimum cost circulation problem, each edge has, in addition to its capacity, an associated cost $c(e)$ (sometimes written as c when the edge is clear from the context). The costs on the edges are assumed to be antisymmetric and may be positive as well as negative. The aim is to compute a feasible circulation such that the *cost* is minimized, where the cost is defined as

$$\sum_{e \in E} f(e)c(e).$$

In a planar graph, the cost of a circulation can be expressed as a function of the potentials of the faces and *face costs*. The cost of a face g is defined as follows. Traverse the boundary of the face clockwise, since the graph is directed some edges are traversed in the forward direction and some in the reverse direction; see below:

$$c(g) = \sum_{e \in \mathrm{forward}(g)} c(e) - \sum_{e \in \mathrm{reverse}(g)} c(e).$$

The cost of the circulation is $\sum_{e \in E} f(e)c(e)$ and is the same as $\sum_{g \in F} p(g)c(g)$.

Let f be a function defined on a lattice $\mathcal{L}$ and let $a, b \in \mathcal{L}$. The function f is called *modular* if

$$f(a) + f(b) = f(a \vee b) + f(a \wedge b).$$

The next proposition is immediate.

PROPOSITION 5.1. *The cost function of a planar circulation is modular.*

We now show that the solutions to the minimum cost circulation problem form a sublattice. Let f denote the cost function in a circulation and let $P_1, P_2 \in \mathcal{P}$ be any two minimum cost circulations. Since $f(P_1) + f(P_2) = f(P_1 \vee P_2) + f(P_1 \wedge P_2)$, the cost of $f(P_1 \vee P_2)$ and $f(P_1 \wedge P_2)$ must be minimum as well.

5.1. Representing the minimum cost solutions. Having shown that the minimum cost circulations of G form a sublattice, we now describe how to construct a network G' whose feasible circulations are exactly the minimum cost circulations of G. Hence, it follows that the machinery discussed in §4 for representing the set of feasible circulations can also be applied to represent the set of minimum cost circulations; i.e., a partial order whose closed subsets correspond to the minimum cost circulations can be constructed.

For any network G (not just for a planar network), once we have a single minimum cost circulation C, we can represent all minimum cost circulations as $C + \{C' : C' \text{ is a circulation in } G'\}$, where G' is a network derived from G and C. This representation is analogous to the representation of all solutions to a linear system or differential equation as a single solution, plus the set of solutions to the homogeneous equations.

To compute G', we first derive reduced edge-costs $\hat{c}$ from the original costs c in the residual graph of G with respect to C. Reduced edge-costs are all nonnegative and have the property that the cost of any cycle in G_C is the same whether we use original costs or reduced costs. In particular, any cycle in G_C that has zero-cost with respect to the original costs also has zero-cost with respect to the reduced costs and hence contains only edges that have zero reduced cost. Let G' be the subgraph of G_C consisting of edges with zero reduced cost.

LEMMA 5.2. *The set of min-cost circulations in G is $\{C + C' : C'$ a circulation in $G'\}$.*

Proof. Let C' be any circulation in G'. Since G' is a subgraph of G_C, C' consists of a collection of cycles of flow in G_C such that $C + C'$ is a circulation in G. Since G' contains only edges with zero-reduced cost, every cycle in C' has zero cost. Hence the cost of $C + C'$ is the same as that of C and is hence minimum.

Conversely, let C_1 be any min-cost circulation in G. Then $C_1 - C$, being the difference between two circulations, is itself a circulation in G_C and is hence composed of cycles of flow. If any such cycle of flow had negative cost, it could be added to C to reduce C's cost, so there are no negative cycles. Similarly, there are no positive-cost cycles, else they could be subtracted from C_1 to reduce its cost. Thus the difference $C_1 - C$ consists of a collection of zero-cost cycles of flow. By the remarks above, each such cycle consists of edges with zero reduced cost, so $C_1 - C$ is a circulation in G'. □

For completeness, we describe one standard construction for computing reduced costs. Since C is min-cost, every cycle in G_C has nonnegative cost [FF]. Obtain an auxiliary graph from G_C by adding a node s and zero-cost edges from s to every original node. Next, in the auxiliary graph compute shortest path distances $d(v)$ of each node v from the added node s, using an algorithm, e.g., Floyd–Warshall, that depends only on the nonexistence of negative cycles.

Now, for each edge $e = uv$ in G_C, we define the *reduced cost* $\hat{c}(e) = c(e) + d(u) - d(v)$. By Bellman's equations, $d(v) \leq c(e) + d(u)$, so each reduced edge-cost is nonnegative. Furthermore, it is easy to check that the cost of any cycle in G_C is the same whether we use original edge-costs or reduced edge-costs, since the $d(v)$'s cancel out as we traverse the cycle.

5.2. Directions for future research. An outstanding open question is whether a better algorithm for computing minimum cost circulations in planar graphs can be found. In what follows, we will outline two possible approaches for this problem.

Minimizing a modular function defined on a lattice is a well-known problem in operations research. We briefly review its solution. (See, e.g., [GI, pp. 130–133], [Ir], [To] for more details and proofs.) Let the cost of every vertex in the dag $T(\mathcal{P})$ be equal to the cost of the corresponding face in the original graph. It is easy to see that the minimum cost circulation problem can be restated as the problem of finding the predecessor-closed set of minimum cost in $T(\mathcal{P})$, where the cost of a closed set is defined to be the sum of the costs of its members. The problem of computing the minimum cost closed set can be reduced to computing the minimum cut in the following graph, denoted by T:

- Connect all positive cost vertices to a source and all negative cost vertices to a sink;
- The capacity assigned to edges adjacent to the source or sink is equal to the absolute value of the cost of the vertices to which they are adjacent;
- All other edges have infinite capacity.

Solving this problem directly, by computing a maximum flow, would take too long, because the graph T is too big. However, some algorithm based on maximum flow would be interesting for the following reason. Most algorithms for computing the minimum cost circulation have the following form: They start from an initial circulation and generate circulations of smaller cost until a minimum cost circulation is obtained. In the dag $T(\mathcal{P})$, a one-to-one correspondence can be established between its closed sets (or the circulations in $\mathcal{P}$) and the cuts separating the source from the sink in the graph T. Thus, all these algorithms can be viewed as algorithms that implicitly compute the minimum cut in T. On the contrary, an algorithm that computes the minimum cut in T via a maximum flow can be considered a dual algorithm to all other minimum cost circulation algorithms. The question therefore arises: Can the special structure of T be exploited to find the maximum flow much more quickly, say by considering only intermediate solutions (feasible flows) of a certain form ?

A different approach to computing a minimum cost circulation follows from the Region-Growing Algorithm. What happens when this algorithm is applied to the minimum cost circulation problem? At each step of the algorithm, we must first decide whether to push some amount of flow in the clockwise direction and then decide which faces to annex to R. These decisions will depend on the cost of the boundary of R in the clockwise direction. The easy case is when the cost is negative: Then the boundary is saturated, and the faces that border on saturated edges are annexed to R. The difficulty arises when the boundary has positive cost. Then there is no gain in pushing more flow on the boundary in the clockwise direction. From the existence of the Region-Growing Algorithm, we know that there is at least *one* face f that borders R and can be annexed to it; i.e., the potential of f has reached its value in some optimal solution. Does there exist a simple criterion for determining which face that is?

Another intriguing research question concerns dynamic computation of feasible or minimum cost circulations in planar networks. Suppose that we are given a circulation and assume that the capacities are changed on edges of a single face. It follows from the correspondence between circulations and shortest paths that it is easy to derive a new circulation from an old one in $O(n\sqrt{\log n})$ time [F] by using the notion of reduced cost. What can be said about the analogous problem for minimum cost flow?

Acknowledgments. We would like to thank Tomas Feder and Dan Gusfield for helpful discussions.

REFERENCES

[AMO] R. K. AHUJA, T. L. MAGNANTI, AND J. B. ORLIN, *Network Flow: Theory, Algorithms and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[F] G. N. FREDERICKSON, *Fast algorithms for shortest paths in planar graphs, with applications*, SIAM J. Comput., 16 (1987), pp. 1004–1022.

[FF] L. R. FORD AND D. R. FULKERSON, *Maximal flow through a network*, Canad. J. Math., 8 (1956), pp. 399–404.

[G] G. GRÄTZER, *Lattice Theory: First Concepts and Distributive Lattices*, W. H. Freeman, San Francisco, CA, 1971.

[GI] D. GUSFIELD AND R. W. IRVING, *The Stable Marriage Problem*, MIT Press, Cambridge, MA, 1990.

[H] R. HASSIN, *Maximum flows in (s, t) planar networks*, Inform. Process. Lett., 13 (1981), p. 107.

[HJ] R. HASSIN AND D. B. JOHNSON, *An $O(n \log^2 n)$ algorithm for maximum flow in undirected planar networks*, SIAM J. Comput., 14 (1985), pp. 612–624.

[I] M. IRI, *Structural theory for the combinatorial systems characterized by submodular functions*, in Progress in Combinatorial Optimization, Academic Press, New York, 1984, pp. 197–219.

[II] H. IMAI AND K. IWANO, *Efficient sequential and parallel algorithms for planar minimum cost flow*, SIGAL, Internat. Sympos. on Algorithms, Tokyo, 1990.

[IS] A. ITAI AND Y. SHILOACH, *Maximum flow in planar networks*, SIAM J. Comput., 8 (1979), pp. 135–150.

[J] D. B. JOHNSON, *Parallel algorithms for minimum cuts and maximum flows in planar networks*, J. Assoc. Comput. Mach., 34 (4) (1987), pp. 950–967.

[JV] D. B. JOHNSON AND S. VENKATESAN, *Using divide and conquer to find flows in directed planar networks in $O(n^{1.5} \log n)$ time*, in Proc. of the 20th Ann. Allerton Conf. on Communication, Control and Computing, University of Illinois, Urbana-Champaign, IL, 1982, pp. 898–905.

[KN] S. KHULLER AND J. NAOR, *Flow in planar graphs with vertex capacities*, in Proc. of Integer Programming and Combinatorial Optimization Conf., Waterloo, Ontario, Canada, 1990, pp. 367–383; Algorithmica, Special Issue on Network Flow, to appear.

[KS] S. KHULLER AND B. SCHIEBER, *Efficient parallel algorithms for testing k-connectivity and finding disjoint s-t paths in graphs*, SIAM J. Comput., 20 (1991), pp. 352–375.

[MN] G. L. MILLER AND J. NAOR, *Flow in planar graphs with multiple sources and sinks*, in Proc. of the 30th Ann. Sympos. on Foundations of Computer Science, Raleigh, NC, 1989, pp. 112–117.

[PQ] J. PICARD AND M. QUEYRANNE, *On the structure of all minimum cuts in a network and its applications*, Math. Programming Study, 13 (1980), pp. 8–16.

[R] J. H. REIF, *Minimum s-t cut of a planar undirected network in $O(n \log^2 n)$ time*, SIAM J. Comput., 12 (1983), pp. 71–81.

[T] D. TOPKIS, *Minimizing a submodular function on a lattice*, Oper. Res., 26 (1978), pp. 305–321.

SIAM J. DISC. MATH.
Vol. 6, No. 3, pp. 491–500, August 1993

$n+1$ SEGMENTS BEAT n*

JENŐ TÖRŐCSIK†

Abstract. Motivated by the elementary fact that the sum of the diagonals of a quadrilateral is less than its perimeter the following question was raised: Do the three largest sides exceed the diagonals? More generally, given n arbitrary segments in the plane, can one select $n + 1$ other segments whose endpoints are among the endpoints of the given segments and whose total length is at least as large as the total length of the given segments? Not only will the existence of these segments be shown, but also a fast $O(n \log n)$ algorithm to select them will be given. For quadrilaterals, two stronger inequalities will also be proved.

Key words. inequalities, computational geometry

AMS(MOS) subject classifications. 51M16, 51M04, 51M25, 68U05, 26D99

1. Introduction.

1.1. A new inequality for quadrilaterals. It is well known that the sum of the diagonals of a quadrilateral is not greater than its perimeter. This consequence of the triangle inequality was first proved thousands of years ago, and it also can be found in the works of many well-known scholars, among them Hadamard [H], Borel [B], and Grévy [G]. It might come as a surprise that a much stronger statement can be made.

THEOREM 1.1. *The sum of the diagonals of a quadrilateral is not greater than the sum of its three largest sides.*

With more effort, we will prove the following two stronger results.

THEOREM 1.2. *The sum of the diagonals of a quadrilateral is not greater than the sum of its two largest sides and its smallest side.*

THEOREM 1.3. *The sum of the diagonals of a quadrilateral is not greater than the sum of its largest side, the side opposite to it, and* $2\sqrt{2} - 2 \approx .83$ *times the average of the other two sides.*

If we only assume the triangle inequality, then these theorems do not necessarily hold; we must assume the Euclidean metric (see Remark 3.2).

1.2. The general result. Theorem 1.1 can be reformulated in the following way. Given two segments AC, BD in the plane, we can select three different segments from the four others determined by the points A, B, C, D with the total length at least $|AC| + |BD|$. This leads to another more general question. Given n arbitrary segments in the plane, can we select $n + 1$ other segments, whose endpoints are among the n segments' endpoints and whose total length is at least as large as the original n segments'? We can prove the following theorem.

THEOREM 1.4. *Given* $n \geq 2$ *arbitrary segments in the plane, with not necessarily distinct labeled endpoints, these labeled endpoints define* $\binom{n}{2} - n$ *new segments. From these new segments, we can select* $n + 1$ *segments distinct from each other and whose total length is at least as large as the total length of the given* n *segments. Here two segments are said to be distinct if the labels of their endpoints are not the same. Equality holds if and only if* $n = 2$ *and the two segments coincide.*

The main idea of the proof of Theorem 1.4 is the following selection algorithm. Let us mention that this algorithm takes time $O(n \log n)$, although the number of possible segments is $\binom{n}{2} - n$.

*Received by the editors January 23, 1990; accepted for publication (in revised form) May 6, 1992.

†Department of Computer Science, Eötvös University, Múzeum krt. 6-8, Budapest, H-1088, Hungary. ecom001@hueco.uni-wien.ac.at

SELECTION ALGORITHM. *Suppose that $n \geq 3$. Let A_1B_1 be the longest of the n segments. Label the others in order of their angles with A_1B_1.*[1] *Let $k \geq 2$ be the smallest k such that $A_1B_1 < \sqrt{3}A_kB_k$. Select the longest side of $A_1A_iB_1B_i$ for $i = 2, 3, \dots, k-1$, the two longest sides of $A_1A_kB_1B_k$, and the longest side of $A_iA_{i+1}B_iB_{i+1}$ for $i = k, k+1, \dots, n$ (index arithmetic* mod n). *If there is no such k, then it is enough to select n segments. (By a side we mean a segment different from A_iB_i!)*

2. Proofs.

2.1. Proof of Theorem 1.1. Label the sides a, b, c, d in order around the quadrilateral, where a is the largest, and the diagonals are e and f (Fig. 1).

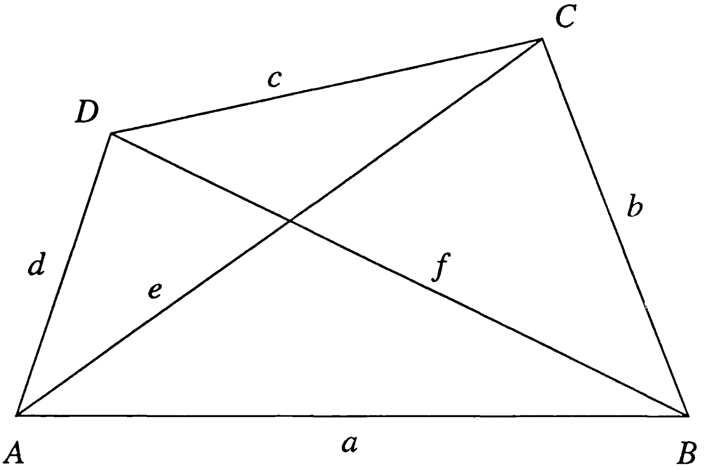


FIG. 1. *Notation. The largest side is a.*

If $a \leq e$ or $a \leq f$, then $b + c \leq f$, and $c + d \leq f$ proves the theorem. (Here we might have selected the "larger of the remaining sides.")

If $a < e$ and $a < f$, then the quadrilateral is convex, and

$$0 < (e - a)(f - a). \tag{1}$$

Ptolemy's theorem [BDJMV, Thm. 15.4] states that, for any quadrilateral, $ef \leq ac + bd$. From $b \leq a$,

$$ef \leq ac + ad. \tag{2}$$

Combining (1) and (2), $a(e + f) < a(a + c + d)$. So

$$e + f < a + c + d. \tag{3}$$

□

2.2. Proof of Theorem 1.2. We use the notation of the previous proof; a denotes the largest side again. By symmetry, we can suppose that $d \leq b$.

When inequality (1) holds, then the previous proof proves Theorem 1.2.

Therefore, by symmetry, we must only prove Theorem 1.2 *when $f \geq a \geq e$, and a is the largest side.* We can also suppose that *d is the smallest side*, otherwise $b + c \geq f$, $a \geq e$ proves Theorem 1.2.

[1] Translate the segments so that the midpoints will be the same as the midpoint of A_1B_1. Go around and label the endpoints by A_1, A_2, $A_3, \dots A_n, B_1B_2, \dots, B_n$ in order.

We have $e > b$, otherwise $b \geq e$, $a + d > f$ proves Theorem 1.2.

Denote by l the line perpendicular from C to DB. Let C' be the intersection of line l with the 60°-angle arc around A of radius AB with endpoint B. Since $e > b$, this intersection exists (see Fig. 2). Denote by b', c', and e' the lengths of $BC', C'D$, and AC', respectively.

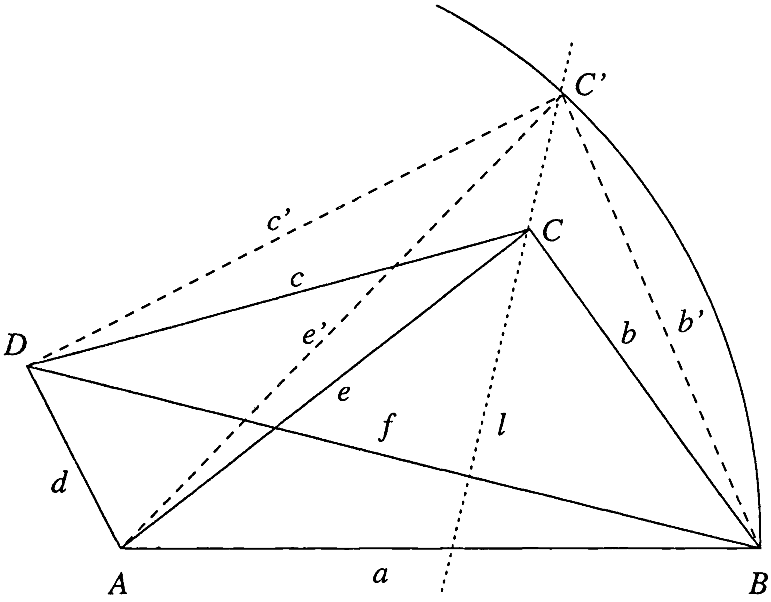


FIG. 2. *Line l has to intersect the arc when $e > b$.*

So $e' = a$. Therefore $0 = (e' - a)(f - a)$. Since $a > b'$, we can prove similarly to the proof of Theorem 1.2 that

$$e' + f < a + c' + d. \tag{4}$$

Consider the two hyperbolas, both with foci A and D, passing through C and C', respectively. We see that

$$c' - e' < c - e. \tag{5}$$

Combining (4) and (5), $e + f < a + c + d$. □

2.3. Proof of Theorem 1.3.

LEMMA 2.1. *If the angle and the lengths of the diagonals are given, then the sum of the largest side, the side opposite to it, and $2\sqrt{2} - 2 \approx .83$ times the average of the other two sides is minimal when the quadrilateral is a parallelogram.*

Proof. Label the vectors of the sides of the quadrilateral by $\mathbf{a} = \overrightarrow{AB}$, $\mathbf{b} = \overrightarrow{BC}$, $\mathbf{c} = \overrightarrow{CD}$, and $\mathbf{d} = \overrightarrow{DE}$ ($\mathbf{a}$ is not necessarily the longest side). Translate $\mathbf{e} = \overrightarrow{AC}$ and $\mathbf{f} = \overrightarrow{BD}$, the diagonals of the quadrilateral, to the diagonals of a parallelogram. Denote by $\mathbf{a}'$, $\mathbf{b}'$, $\mathbf{c}'$, and $\mathbf{d}'$ the sides of the parallelogram, i.e., the vectors of the images of $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, and $\mathbf{d}$, respectively. Therefore

$$|\mathbf{a}'| + |\mathbf{c}'| = |\mathbf{a}' - \mathbf{c}'| = |\mathbf{e} - \mathbf{f}| = |\mathbf{a} - \mathbf{c}| \leq |\mathbf{a}| + |\mathbf{c}|.$$

Similarly, $|\mathbf{b}'| + |\mathbf{d}'| \leq |\mathbf{b}| + |\mathbf{d}|$. We can suppose that we labeled the sides of the quadrilateral in such a way that $\mathbf{a}'$ is the longest side of the parallelogram. So

$$|\mathbf{a}'| + |\mathbf{c}'| + (2\sqrt{2} - 2)\frac{|\mathbf{b}'| + |\mathbf{d}'|}{2} \leq |\mathbf{a}| + |\mathbf{c}| + (2\sqrt{2} - 2)\frac{|\mathbf{b}| + |\mathbf{d}|}{2},$$

which is not more than the sum of the largest side, the side opposite to it, and $2\sqrt{2} - 2$ times the average of the other two sides of the quadrilateral. Lemma 2.1 is proved. □

So it is enough to prove Theorem 1.3 for a parallelogram. Label its vertices by A, B, C, and D, the midpoints of AC and AB by O and F, respectively.

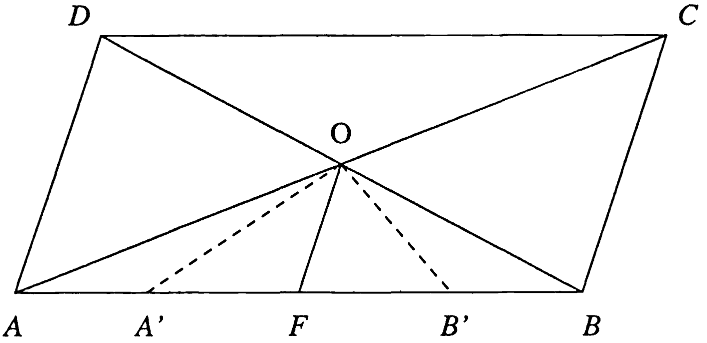


FIG. 3

Let $AB \geq BC$ (see Fig. 3). It would suffice to show that

$$(2\sqrt{2} - 2)OF + AB > AO + OB. \tag{6}$$

We will minimize the difference of the two sides in this inequality by moving A and B closer to F. If A' and B' lie on the segments AF and FB, respectively, then

$$\begin{aligned}(2\sqrt{2} - 2)OF + AB - AO - OB &= (2\sqrt{2} - 2)OF + A'B' + (AA' - AO) + (BB' - OB)\\ &\geq (2\sqrt{2} - 2)OF + A'B' - A'O - OB'.\end{aligned}$$

Since the angle $\angle AOB$ is not acute, it is enough to prove inequality (6) when $\angle AOB$ is a right angle. In this case, the length of AB is two times the length of OF. If we fix the length of AB (and thus the length of OF), then $AO + OB$ is maximal when AB is perpendicular to OF. In this case, $(2\sqrt{2} - 2)OF + AB = AO + OB$. So inequality (6) holds. □

2.4. Proof of Theorem 1.4. If $n = 2$, we have the two diagonals of a quadrilateral. So assume that $n \geq 3$.

We will prove that the total length of the segments selected by the Selection Algorithm is larger than the total length of the original n segments. Let us use the notation of the Selection Algorithm.

For $2 \leq i < k$, A_iB_i is not longer than the longest side of $A_1A_iB_1B_i$, where by a side we mean a segment different from A_jB_j. This follows from $A_1B_1 \geq \sqrt{3}A_iB_i$, since, in this case, A_i and B_i cannot both be in the interior of the intersection of the circles around A_1 and B_1 with radii A_iB_i. *So we can suppose that $k = 2$, i.e., that $A_1B_1 < \sqrt{3}A_2B_2$.*

We can prove similarly to the proof of Lemma 2.1 that, if the angle and the lengths of A_iB_i and A_jB_j are given, then the largest side of $A_iA_jB_iB_j$ and also the sum of its two largest sides is minimal if $A_iA_jB_iB_j$ is a parallelogram. *So we can suppose that the segments A_iB_i have the same midpoint O.*

If we dilated $A_{i+1}B_{i+1}$ to $A'_{i+1}B'_{i+1}$ from O, where

$$OA_{i+1} < OA'_{i+1} = OB'_{i+1} = OA_1,$$

then $A_{i+1}B_{i+1}$ would have increased at least as much as the increase of the longest side of $A_iB_iA_{i+1}B_{i+1}$ plus the increase of the longest side of $A_{i+1}B_{i+1}A_{i+2}B_{i+2}$. This simply follows from the triangle inequality (see Fig. 4).

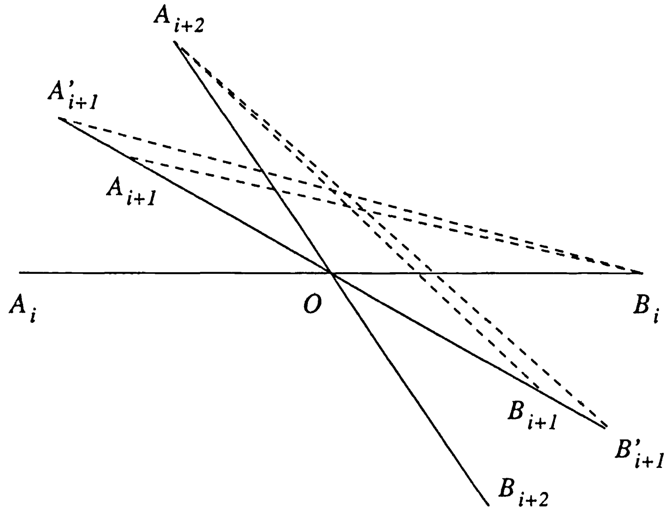


FIG. 4. *Dilate $A_{i+1}B_{i+1}$ to the length of A_1B_1.*

Since we select the longest side of $A_iA_{i+1}B_iB_{i+1}$ for $i = 2, 3, \dots, n$, *we can suppose that each segment except A_2B_2 is dilated to the length ofA_1B_1.*

The following lemma will be extremely useful in the rest of the proof.

LEMMA 2.2. *If A_iB_i and $A_{i+2}B_{i+2}$ with the same length are fixed and $A_{i+1}B_{i+1}$ is rotated between them around O and is not longer than the previous two, then the sum of the largest sides of $A_iA_{i+1}B_iB_{i+1}$ and $A_{i+1}A_{i+2}B_{i+1}B_{i+2}$ is minimal at the following two positions of $A_{i+1}B_{i+1}$:*

(i) *$A_{i+1}B_{i+1}$ is on the line A_iB_i or on $A_{i+2}B_{i+2}$ in the case of $\angle A_iOA_{i+2} < 90°$,*

(ii) *$A_{i+1}B_{i+1}$ is perpendicular to A_iB_i or $A_{i+2}B_{i+2}$ in the case of $\angle A_iOA_{i+2} \geq 90°$.*

Proof. First, suppose that $\angle A_iOA_{i+1} \leq 90°$ and $\angle A_iOA_{i+2} \leq 90°$. By symmetry, we can suppose $\angle A_iOA_{i+1} \geq \angle A_iOA_{i+2}$. If we rotate $A_{i+1}B_{i+1}$ around O to $A'_{i+1}B'_{i+1}$ such that $\angle A_iOA_{i+1} \leq \angle A_iOA'_{i+1} \leq 90°$, then the sum of the largest sides of $A_iB_iA_{i+1}B_{i+1}$ and $A_{i+1}B_{i+1}A_{i+2}B_{i+2}$ becomes smaller (see Fig. 5). To prove this, label by A''_{i+1} the intersection of the ray OA'_{i+1} and the circle around $A_{i+1}B_iB_{i+2}$, by B'_i and B''_i the intersections of $B_{i+2}A_{i+1}$ and $B_{i+2}A''_{i+1}$ with the circle arc from B_{i+2} to B_i such that

$$\angle B_{i+2}B'_iB_i = \angle B_{i+2}B''_iB_i = \tfrac{1}{2}\angle B_{i+2}A_{i+1}B_i.$$

So

$$B_{i+2}A_{i+1}+A_{i+1}B_i = B_{i+2}B'_i > B_{i+2}B''_i = B_{i+2}A''_{i+1}+A''_{i+1}B_i > B_{i+2}A'_{i+1}+A'_{i+1}B_i.$$

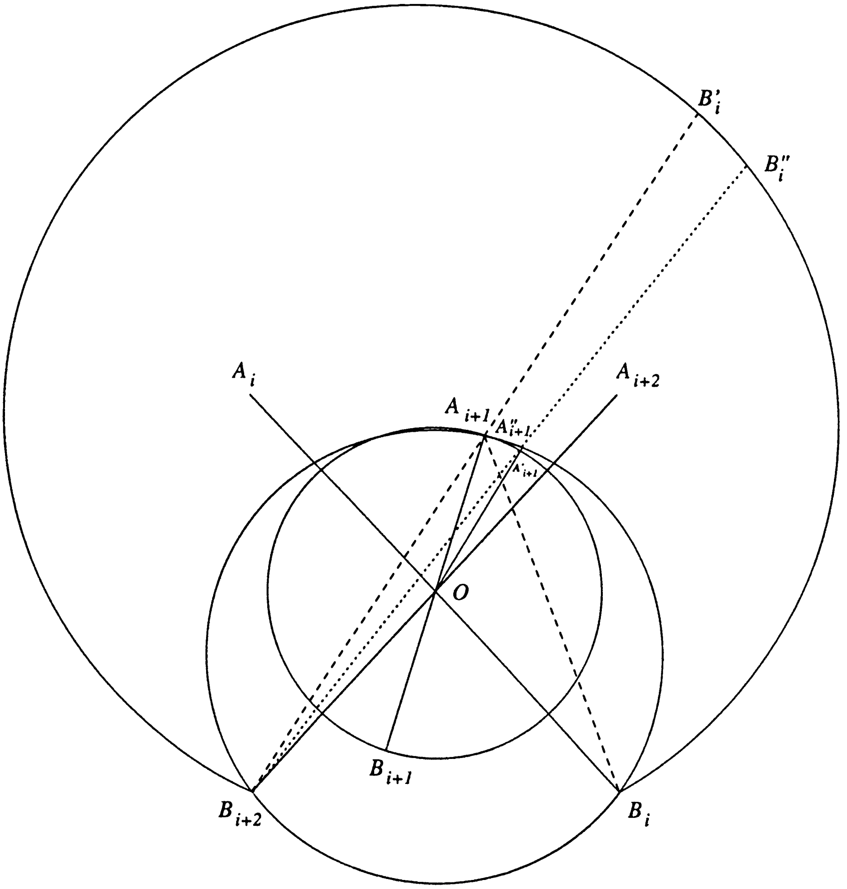


FIG. 5. *Rotate $A_{i+1}B_{i+1}$ to decrease the sum.*

Therefore, when $\angle A_iOA_{i+2} < 90°$, then (i) holds, and, when $\angle A_iOA_{i+2} \geq 90°$, then we can suppose that $\angle A_iOA_{i+1} \geq 90°$ or $\angle A_{i+1}OA_{i+2} \geq 90°$. By symmetry, we can suppose the latter case (see Fig. 6). If we rotate $A_{i+1}B_{i+1}$ around O, then $A_{i+1}B_i$ and $A_{i+1}A_{i+2}$ become smaller as A_{i+1} gets closer to A_{i+2}.

So the sum of the longest sides is minimal when $A_{i+1}B_{i+1}$ is perpendicular to A_{i+2} B_{i+2}; i.e., (ii) holds. □

Let us continue to determine the conditions under which the sum of the selected $n+1$ segments is minimal.

Case 1. A_3 is not nearer to A_1 than to B_1.

Label by e and f the perpendiculars to A_1B_1 and A_3B_3 at O, respectively.

If A_2B_2 is in the nonobtuse angle of A_1B_1 and f (see Fig. 7(a)), (or of A_3B_3 and e (see Fig. 7(b))), then the two longest sides of $A_1A_2B_1B_2$ and the longest side of A_2A_3 B_2B_3 are minimal if A_2B_2 is coincident with f (with e). *So we can suppose that A_2B_2 lies on either e or f or between them.*

Hence, the sum of the two longest sides of $A_1A_2B_1B_2$ and the longest side of A_2A_3 B_2B_3 is equal to $A_1B_2 + B_1A_2 + A_2B_3$ (see Fig. 8(a)). A_1B_2 is minimal if A_2B_2 is on e, and, by (ii) in Lemma 2.2, $B_1A_2 + A_2B_3$ is minimal if A_2B_2 is on e or on f. *So we can suppose that A_2B_2 is on e.*

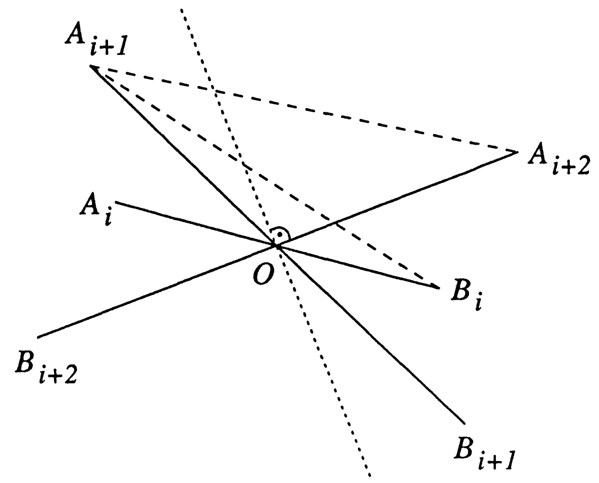


FIG. 6. *When the angle* $A_{i+1}OA_{i+2} \geq 90°$.

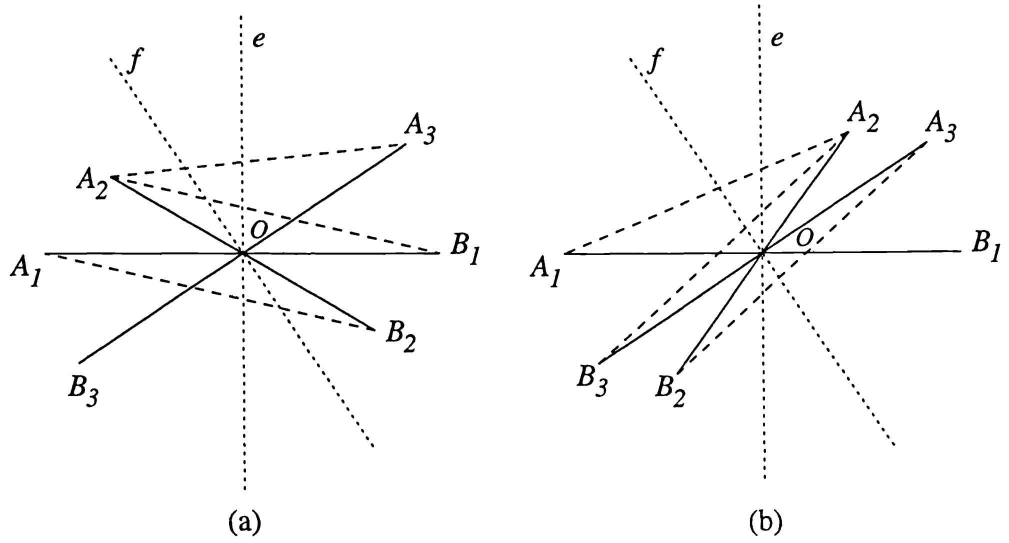


FIG. 7. *The sum is minimal when* (a) A_2B_2 *is on* f, *and* (b) A_2B_2 *is on* e.

The sum of the longest sides of $A_2A_3B_2B_3$ and $A_3A_4B_3B_4$ is equal to $B_4A_3 + A_3B_2 = (B_4A_3 + A_3B_2') + (A_3B_2 - A_3B_2')$, where B_2' is on the ray OB_2 and $OB_2' = OA_3$ (see Fig. 8(b)). Both part of this sum are minimal if A_3 is on the line A_2B_2 (the first part from (i) in the Lemma 2.2, the second part from $A_3B_2 - A_3B_2' \geq -B_2B_2' = A_3'B_2 - A_3'B_2'$, where A_3 is on the line A_2B_2 and $OA_3' = OA_3$). *So we can suppose that* A_3B_3 *is collinear with* A_2B_2 *and therefore with* e.

Thus Case 1 falls under Case 2.

Case 2. A_3 is not nearer to B_1 than to A_1 (see Fig. 9(a)).

Hence, the sum of the two longest sides of $A_1A_2B_1B_2$ and the longest side of $A_2A_3B_2B_3$ is equal to $A_1B_2 + B_1A_2 + A_2B_3$. A_1B_2 is minimal if A_2B_2 is collinear with A_3B_3, and, by (i) in Lemma 2.2, $B_1A_2 + A_2B_3$ is minimal if A_2B_2 is collinear with A_3B_3 or with A_1B_1. *So we can suppose that* A_2B_2 *is collinear with* A_3B_3 (see Fig. 9(b)).

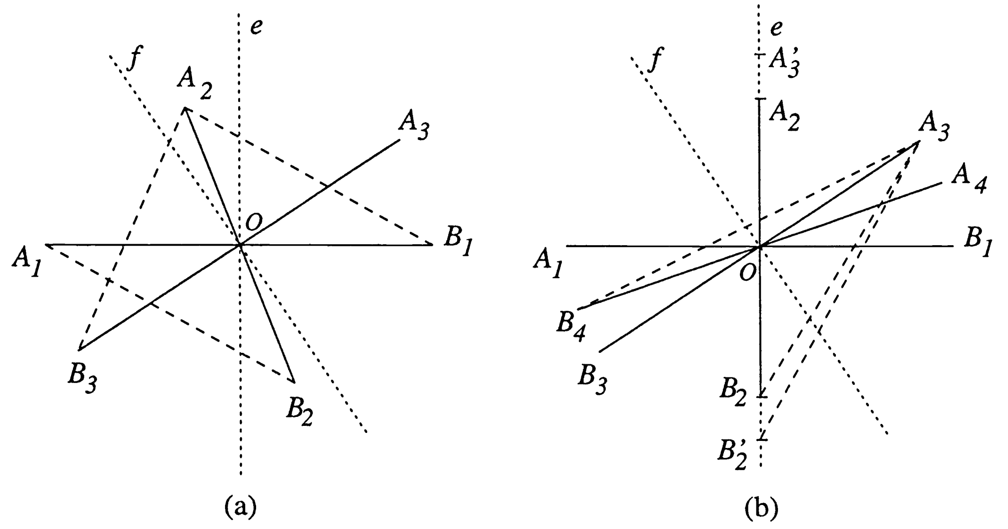


FIG. 8. *The sum is minimal when* (a) A_2B_2 *is on* e, *and* (b) A_3B_3 *is on* e.

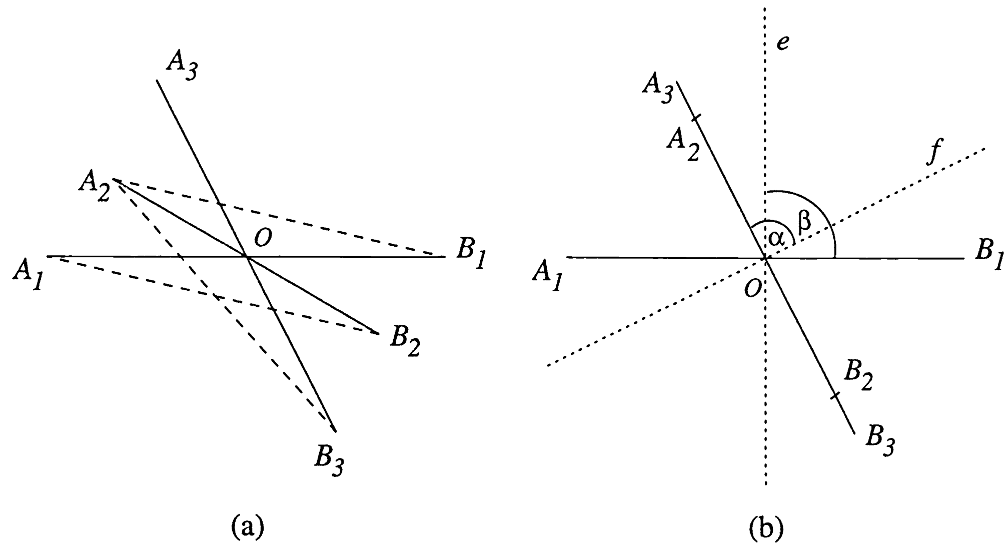


FIG. 9. (a) *We can rotate* A_2B_2 *onto line* A_3B_3. (b) *When* $n \geq 6$, *we can use induction on* n.

If $n = 3$, then the sum of the two longest sides of $A_1A_2B_1B_2$ and the longest side of $A_3A_1B_3B_1$ is minimal when A_1B_1 is perpendicular to the straight line $A_3A_2B_2B_3$. So we get Fig. 11(b).

If $n \geq 4$, label the sectors α and β shown in Fig. 9(b). *We can suppose that there is at most one segment in sector* α; otherwise, by (i) in Lemma 2.2, we could suppose that A_4B_4

is coincident with A_3B_3, so we could use induction on n. Similarly, *we can suppose that there is at most one segment is in sector β. So it is enough to prove the result for* $4 \leq n \leq 5$.

If $n = 5$, then A_4B_4 is in α, A_5B_5 is β, and none of them are $\alpha \cap \beta$. By (ii) in Lemma 2.2, we can suppose that A_4B_4 is on e, and, after this, by (i) in Lemma 2.2, we can suppose that A_5B_5 is coincident with A_4B_4. *So it is enough to prove the result for* $n = 4$.

If $n = 4$, then, by (ii) in Lemma 2.2, we can suppose that A_4B_4 is perpendicular to A_1B_1 (see Fig. 10).

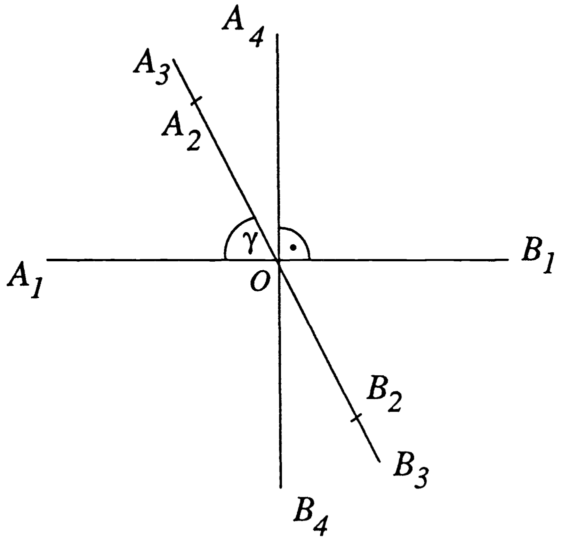


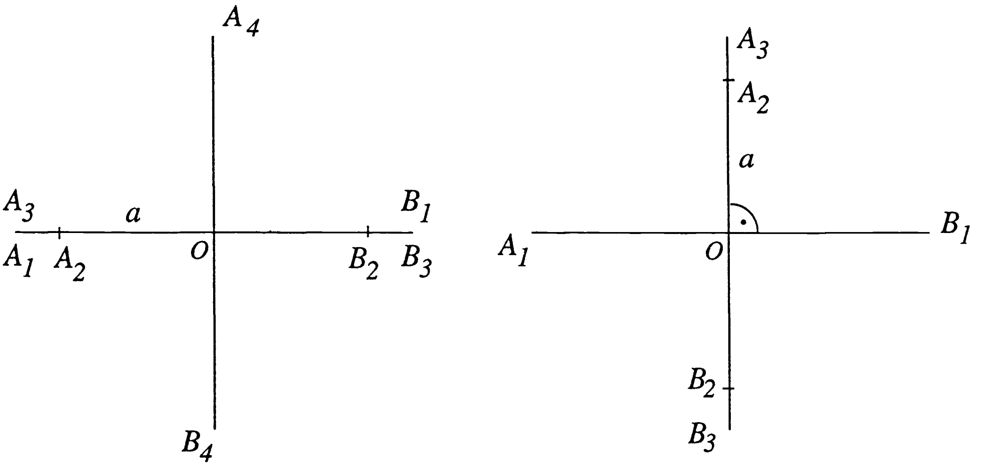
FIG. 10. *The sum is a concave function of* γ.

Let us examine the sum of the five chosen segments when the straight line $A_3A_2B_2B_3$ rotates around O such that the angle of it and A_1B_1 increases from 0° to 90°. Denote this angle by γ. Among the lengths of the five chosen segments, only the lengths of A_1B_2, A_2B_1, and A_4B_3 are changing. The length of these segments is a concave function of γ. The sum of concave functions is a concave function. So the sum of the segments is a concave function of γ. *So the minimum can be attained only when* $\gamma = 0°$ *or* $\gamma = 90°$. If $\gamma = 0°$, we get Fig. 11(a). If $\gamma = 90°$, then A_3B_3 is coincident with A_4B_4, so, by deleting one of them, we get the case where $n = 3$.

Simple calculation will complete the proof. What remained to be examined are the cases of Figs. 11(a) and 11(b). Fix the length of OA_1 to be 1. Denote the length of OA_2 by a. We have $a > 1/\sqrt{3}$, since $A_1B_1 < \sqrt{3}A_2B_2$. In the case of Fig. 11(a), the sum of the chosen five segments is $3a + 3 + 2\sqrt{2}$, which is larger than $2a + 6$, which is the sum of the original four segments. The minimum of their difference is $1/\sqrt{3} - 3 + 2\sqrt{2} > 0.4$. In the case of Fig. 11(b), the sum of the chosen four segments is $2\sqrt{a^2+1} + 1 + a + \sqrt{2}$, which is larger than $2a + 4$, which is the sum of the original three segments. By differentiating the difference of these sums, we get that the minimum of this difference is at $a = 1/\sqrt{3}$, when it is larger than 0.14. □

3. Remarks.

Remark 3.1. Theorems 1.2 and 1.3 are strict in the sense that there is a quadrilateral where the sum of the diagonals is larger than the sum of its largest side, the side opposite to it, and the smaller of the two remaining sides.



(a)

(b)

FIG. 11

For example, take a quadrilateral inscribed to a circle whose a and b sizes have the same length and the angle between them is less than 60°. By Ptolemy's theorem, $ef = ac+bd = a(c+d)$. Combining this with $O > (e-a)(f-a)$, we get $a(e+f) > a(a+c+d)$, so $e+f > a+c+d$.

Remark 3.2. If we only assume the triangle inequality, then Theorem 1.1 does not hold.

For example, take the geometry on the surface of a sphere. Let us choose A, B, C, and D to be distinct points on a great circle such that both of the straight lines AC and BD pass through the center of the sphere. So the sum of the diagonals of the spherical quadrilateral $ABCD$ is equal to its perimeter; so it is more than the sum of its three largest sides.

REFERENCES

[H] J. HADAMARD, *Lecons de Géométrie élémentaire,* Librairie Armand Colin, Paris, 1898, p. 23.

[B] É. BOREL, *Géométrie,* Librairie Armand, Paris, 1921, p. 128.

[G] A. GRÉVY, *Géométrie plane,* Librairie Vuibert, Paris, 1925, p. 35.

[BDJMV] O. BOTTEMA, R. Z. DJORDJEVIC, R. R. JANIC, D. S. MITRINOVIC, AND P. M. VISAC, *Geometric Inequalities,* Wolters-Nordhoff, Groningen, the Netherlands, 1969, p. 128.

SIAM J. DISC. MATH.
Vol. 6, No. 3, pp. 501–521, August 1993

REPLICATING TESSELLATIONS*

ANDREW VINCE†

Abstract. A theory of replicating tessellation of $\mathbb{R}^n$ is developed that simultaneously generalizes radix representation of integers and hexagonal addressing in computer science. The tiling aggregates tesselate Euclidean space so that the $(m+1)$st aggregate is, in turn, tiled by translates of the mth aggregate, for each m in exactly the same way. This induces a discrete hierarchical addressing systsem on $\mathbb{R}^n$. Necessary and sufficient conditions for the existence of replicating tessellations are given, and an efficient algorithm is provided to determine whether or not a replicating tessellation is induced. It is shown that the generalized balanced ternary is replicating in all dimensions. Each replicating tessellation yields an associated self-replicating tiling with the following properties: (1) a single tile T tesselates $\mathbb{R}^n$ periodically and (2) there is a linear map A, such that $A(T)$ is tiled by translates of T. The boundary of T is often a fractal curve.

Key words. tiling, self-replicating, radix representation

AMS(MOS) subject classifications. 52C22, 52C07, 05B45, 11A63

1. Introduction. The standard set notation $X + Y = \{x + y : x \in X, y \in Y\}$ will be used. For a set $T \subset \mathbb{R}^n$ denote by $T_x = x + T$ the translate of T to point x. Throughout this paper, Λ denotes an n-dimensional lattice in $\mathbb{R}^n$. A set T *tiles* a set R *by translation by* lattice Λ if $R = \bigcup_{x \in \Lambda} T_x$ and the intersection of the interiors of distinct tiles T_x and T_y is empty. Such a tiling is called *periodic*.

In this paper, $A : \Lambda \to \Lambda$ will be an endomorphism of Λ, often given by a nonsingular $n \times n$ matrix. Given A, a finite subset $S \subset \Lambda$, containing 0, is said to *induce a replicating tessellation* or simply a *rep-tiling* of Λ if (1)

$$S_m = \sum_{i=0}^{m} A^i(S)$$

tiles Λ by translation by the sublattice $A^{m+1}(\Lambda)$ for each $m \geq 0$, and (2) every point of Λ is contained in S_m for some m. The pair (A, S) will be called a *replicating tiling pair* or simply *rep-tiling pair*. This definition of rep-tiling is related to the rep-k tiles of Golomb [11], Dekking [5], Bandt [1], and others as described later in this introduction.

The definition of rep-tiling can be restated in terms of the Voronoi cells of the lattice. Recall that a lattice Λ determines a tessellation by polytopal Voronoi cells where the *Voronoi cell* of the lattice point x is defined by $\{y \in \mathbb{R}^n : |y - x| \leq |y - z| \text{ for all } z \in \Lambda\}$. Let V_m denote the union of the Voronoi cells corresponding to the lattice points of S_m. The definition of rep-tiling is equivalent to (1) V_m tiles $\mathbb{R}^n$ by translation by the sublattice $A^{m+1}(\Lambda)$, for each $m \geq 0$, and (2) every point of $\mathbb{R}^n$ lies in V_m for some m. The set S_m (or the corresponding V_m) is called the *m-aggregate* of the pair (A, S). If S induces a replicating tessellation, then the $(m+1)$-aggregate is tiled by $|S|$ copies of the m-aggregate for each $m \geq 0$. More precisely, $S_0 = S$ and S_{m+1} is the disjoint union

$$S_{m+1} = \bigcup_{x \in A^{m+1}(S)} x + S_m$$

for all $m \geq 0$. Hence, we have the term "replicating."

* Received by the editors May 6, 1992; accepted for publication (in revised form) July 21, 1992.

† Department of Mathematics, University of Florida, Gainesville, Florida 32611. e-mail: vince@math.ufl.edu

Given $A : \Lambda \to \Lambda$ and S, a *finite address* of a lattice point $x \in \Lambda$ is a finite sequence $s_0\, s_1 \ldots s_m$ such that $x = \sum_{i=0}^{m} A^i s_i$ where $s_i \in S$. The m-aggregate is then the set of lattice points whose address has at most $(m+1)$ digits.

PROPOSITION 1. *Given endomorphism $A : \Lambda \to \Lambda$, the set $S \subset \Lambda$ induces a rep-tiling of Λ if and only if every lattice point in Λ has a unique finite address.*

Proof. Condition (2) in the definition of a rep-tiling is equivalent to every lattice point having a finite address. Given condition (2), condition (1) in the definition is equivalent to the finite address being unique. This is proved as part of Proposition 2 in §2. □

Before proceeding with the theory, consider the following three examples. The first has applications to computer arithmetic and the representation of numbers by symbol strings [20], [21]. The third has applications to data addressing in computer vision and remote sensing [6], [18], [25].

Example 1 (Radix representation in $\mathbb{Z}$). The lattice Λ is the one-dimensional integer lattice $\mathbb{Z}$, and A is multiplication by an integer b. By Proposition 1, a finite subset S of $\mathbb{Z}$ induces a rep-tiling of $\mathbb{Z}$ if every integer x has a unique base b *radix* representation $x = \sum_{i=0}^{m} s_i b^i$, where $s_i \in S$. With $S = \{0, 1, \ldots, b-1\}$ and $b \geq 2$, the Fundamental Theorem of Arithmetic states that every nonnegative integer (but no negative integer) has such a unique radix representation. The m-aggregate, in this case, is the set of integers $\{0, 1, \ldots, b^m - 1\}$, and, clearly, each aggregate is tiled by b copies of the previous aggregate. With $b \leq -2$, *every* integer has a unique radix representation. With $b = 3$ and $S = \{-1, 0, 1\}$, the radix representation is called *balanced ternary*. Every integer has a unique representation in the balanced ternary system. Although $S = \{-1, 0, 4\}$ is also a complete set of residues modulo 3, the number -2 has no base 3 radix representation with coefficients in the set $S = \{-1, 0, 4\}$. Unique representation, in a more general setting, is a main topic of this paper.

Knuth [20] gives numerous reference to alternative positional number systems dating back to Cauchy, who noted that negative digits make it unnecessary for a person to memorize the multiplication table past 5 × 5. For a given positive integer base b, Odlyzko [22] gives necessary and sufficient conditions for a set S of positive real numbers to have the property that every real number can be represented in the form $\pm \sum_{i=-N}^{\infty} s_i b^{-i}$, $s_i \in S$. The unique representation of integers is investigated by Matula [21].

Example 2 (Radix representation in $\mathbb{Z}[i]$). Gilbert [7]–[9] extends radix representation to algebraic numbers. For example in the Gaussian integers $\mathbb{Z}[i] = \{a + bi : a, b \in \mathbb{Z}\}$, let $\beta = -1 + i$. Every Gaussian integer has a unique radix representation of the form $\sum_{i=0}^{m} s_i \beta^i$, where $s_i \in S = \{0, 1\}$. (This will be proved in §7.) In the terminology of this paper, if A is complex multiplication by β, then $S = \{0, 1\}$ induces a rep-tiling of the square lattice in the plane.[1] The first aggregate is the union of two translates of the zero aggregate; the second aggregate is the union of two translates of the first aggregate; in general the $(i+1)$st aggregate is the union of two translates of the ith aggregate. Using Voronoi cells to represent the lattice points, Fig. 1 illustrates how the aggregates fit together like jigsaw pieces. By contrast, with the value $\beta = 1 + i$ replacing $-1 + i$, a rep-tiling is not induced because the Gaussian integer i has no radix representation with coefficients in S.

The base β arithmetic in the Gaussian integers resembles usual arithmetic except in the carry digits. For example, $1 + 1 = 0011$ because $\beta = -1 + i$ satisfies the polynomial $x^3 + x^2 - 2$, i.e., $2 = \beta^2 + \beta^3$. So $1 + 1$ results in carrying 011 to the next three places to

[1] Katai and Szabo [16] show that for base $\beta = -k + i$, where k is a positive integer, every Gaussian integer has a unique radix representation with coefficients in $S = \{0, 1, \ldots, k^2\}$.

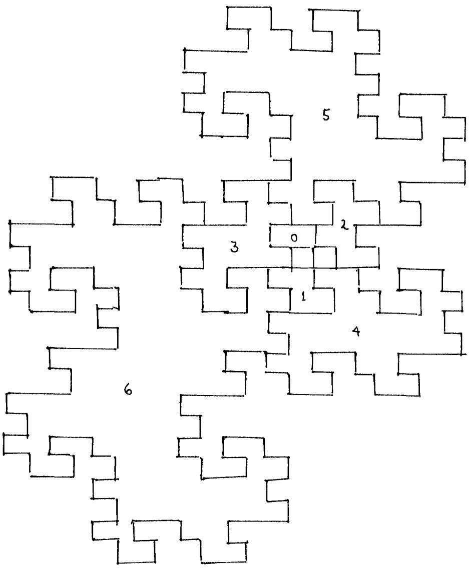


FIG. 1. *Radix representaion base* $-1+i$. *Copies of the first seven aggregates are indicated.*

the right. The ring structure of radix representation in algebraic number fields is further discussed in §7.

Example 3 (Hexagonal tiling). This example is a two-dimensional analogue of the balanced ternary of the first example. The lattice Λ is the hexagonal lattice in the plane shown in Fig. 2, and the endomorphism A is given by the matrix

$$A=\begin{pmatrix}\frac{5}{2} & -\frac{\sqrt{3}}{2}\\ \frac{\sqrt{3}}{2} & \frac{5}{2}\end{pmatrix},$$

which is a composition of an expasnsion by a factor of $\sqrt{7}$ and an $\arctan(\sqrt{3}/2)$ rotation. The set S, consisting of the origin and the six points located at the sixth roots of unity, induces a replicating tessellation. The 0-aggregate consists of the seven cells in Fig. 3(a). The set of cells in Fig. 3(b) is a first aggregate and is the union of seven translates of the zero aggregate. The second aggregate in Fig. 3(c) is, in turn, the union of seven translates of the first aggregate. In general, the $(i+1)$st aggregate is the union of seven translates of the ith aggregate. The entire plane can be tessellated by translated copies of the ith aggregate for any i in such a way that aggregates in the tessellation are nested in the manner described above. Moreover, every hexagon lies in some aggregate. In the unique finite address $s_0 s_1 \ldots s_m$ of a cell, the digit s_i indicates the relative position of that particular cell in the ith aggregate level. Replicating hexagonal tiling is generalized to higher dimensions in §7. From a computer science point of view, hexagonal addressing is an efficient addressing system that allows for addition and multiplication of addresses based on simple sum, product, and carry tables [6], [18]. In fact, one firm has developed a planar database management system based on hexagons (Gibson and Lucas [6]).

We consider two main questions.

Question 1. Given $A : \Lambda \to \Lambda$ and a finite set of lattice points S, does S induce a rep-tiling of Λ?

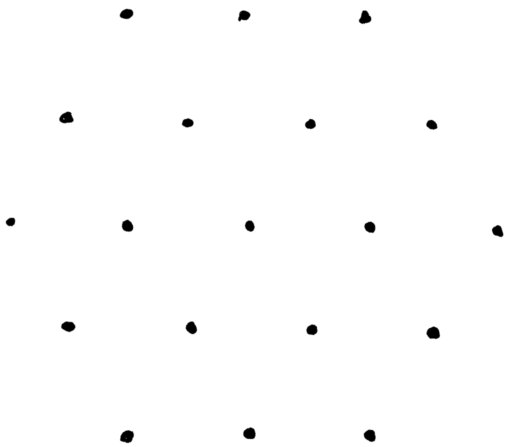


FIG. 2. *Hexagonal lattice.*

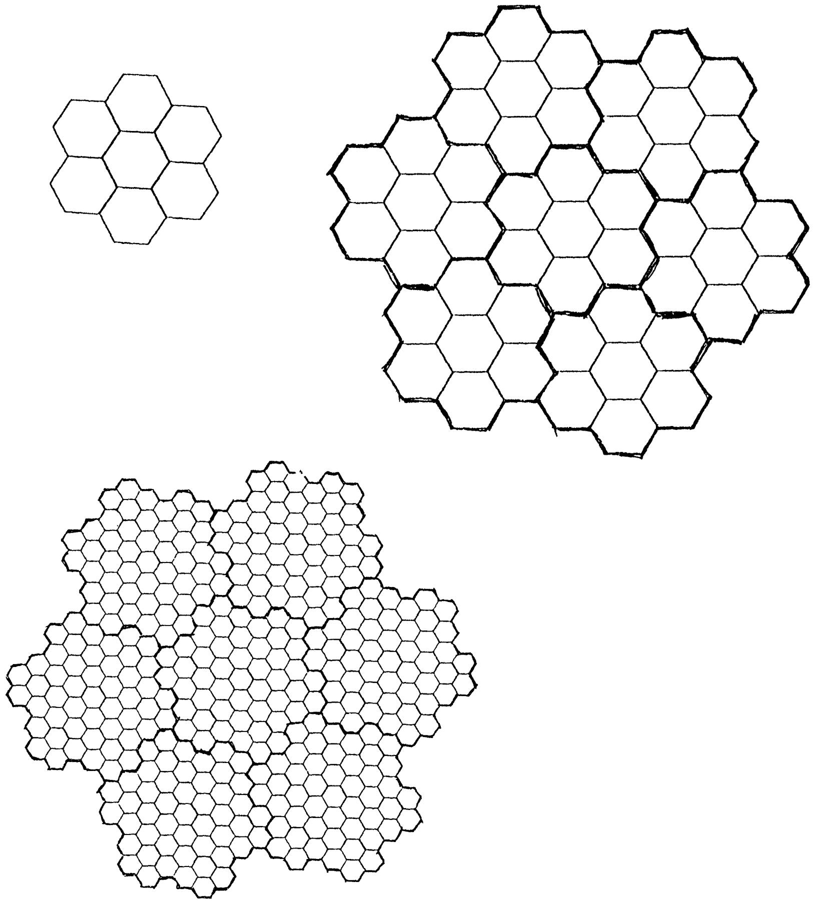


FIG. 3. *Zero, first, and second aggregates.*

Question 2. Given $A : \Lambda \to \Lambda$, does there exist some finite subset S of Λ, such that S induces a rep-tiling of Λ?

Necessary conditions for (A, S) to be a rep-tiling pair are that S be a fundamental domain for A (Proposition 2); in particular $|S| = |\det A|$. Also necessary is that A be a linear expansive map (Proposition 4), which means that the modulus of each eigenvalue of A is greater than 1. If A is linear expansive but (A, S) is not a rep-tiling pair, then, in general, not every lattice point has a finite address. However, every lattice point x does have an infinite repeating address that converges, in a certain sense, to x. This is proved in §5, where the A-adic integers are defined in analogy to the classical number theoretic p-adic integers; the A-adics are applied in §6. Section 6 contains three theorems giving various necessary and sufficient conditions for S to induce a rep-tiling, thus providing answers to Question 1. An efficient algorthm to determine whether (A, S) is a rep-tiling pair is based on one of these theorems. A fourth theorem in §6 states that, for a large class of matrices A, those with sufficiently large singular values (at least two in dimension 2), the set S of lattice points in the Voronoi region of a certain sublattice of Λ serves as a fundamental domain such that S induces a rep-tiling. This provides an answer to Question 2. The existence of an efficient algorithm, given $A : \Lambda \to \Lambda$, to decide whether or not there exists a finite set S that induces a replicating tessellation, is open.

A periodic tiling of $\mathbb{R}^n$ by translation of a single tile T by the lattice Λ is called *self-replicating* if there exists a linear expansive map $A : \Lambda \to \Lambda$, such that for each $x \in \Lambda$,

$$A(T_x) = \bigcup_{w \in S(x)} T_w$$

for some set $S(x) \subset \Lambda$. This self-replicating property originated with Golomb [11] who defined a figure to be *rep-k* if k congruent figures tile a similar figure. For example, a triangle is rep-k for k a perfect square. In this paper, tiling is restricted to lattice tiling, but similarity is generalized to allow any linear expansive map A. Giles [10] discusses the construction of rep-k figures whose boundary has Hausdorff dimension between 1 and 2, including the rep-7 Gosper "flowsnake" and the rep-16 Mandelbrot "square snowflake." The work of Dekking [4], [5], Bandt [1], Kenyon [17], and Gröchenig and Madyeh [12] all deal with the self-replicating property and use a construction similar in principle to Theorem 1. The notion of a self-replicating tiling of $\mathbb{R}^n$ is due to Kenyon [17], although the definition in [17] does not require that the tiling by translations of T be periodic, i.e., a lattice tiling. Kenyon shows that, on the line, the tiling is forced to be periodic, but not necessarily periodic in dimensions greater than one.

The main point here is that each rep-tiling pair (A, S) induces a self-replicating periodic tiling. The construction is as follows. Let

$$E_m = \sum_{i=1}^{m} A^{-i}(S).$$

Note that the E_m are nested and let

$$E = \bigcup_{m=1}^{\infty} E_m \quad \text{and} \quad T := T(A, S) = \overline{E},$$

where $\overline{E}$ denotes the closure of E.

THEOREM 1. *If (A, S) is a rep-tiling pair for Λ, then*

(1) $T = T(A, S)$ is compact and is the closure of its interior.

(2) *T tiles $\mathbb{R}^n$ periodically by translation by the lattice Λ.*

(3) *The tiling is self-replicating.*

The above construction of self-replicating tessellations is applied to the second and third examples in Figs. 4 and 5. Note that the tile in Fig. 4 is rep-2; the union of the two tiles (viewed at an angle $\pi/4$) is similar to the original tile. Also, the tile in Fig. 5 is rep-7; the union of the seven tiles is similar to the original tile. The proof of Theorem 1, given in §4 of this paper, is shorter and simpler than the proof of a similar theorem by Kenyon [17, Thm. 11], but the hypotheses in [17] are slightly less restrictive. Nevertheless, essentially very periodic self-replicating tiling can be obtained by the construction above (Theorem 2).

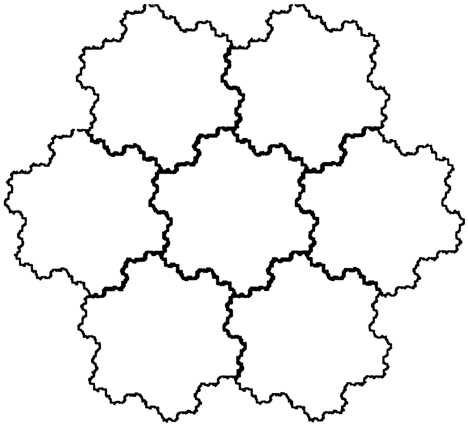


FIG. 4. *Self-replicating tessellation by tile $T(A, S)$, where $A = \begin{pmatrix} \frac{5}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{5}{2} \end{pmatrix}$ acts on the hexagonal lattice and S consists of the origin and the sixth roots of unity.*

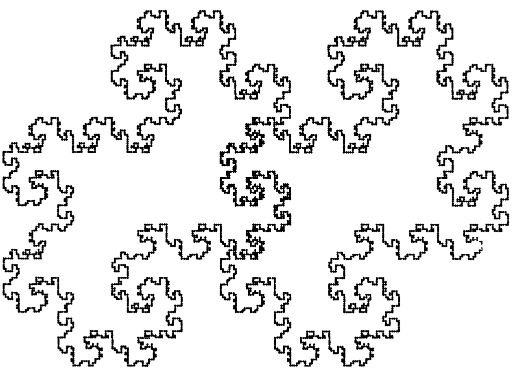


FIG. 5. *Self-replicating tessellation by tile $T(A, S)$ where A is multiplication by $-1 + i$ acting on the square lattice and S consists of the origin and the point* $(1, 0)$.

Section 7 of this paper examines the algebra, as well as the geometry, of replicating tessellations. A construction is given in which the lattice has a ring structure that allows for both addition and multiplication of finite addresses. This construction generalizes radix representation where the base is an algebraic integer and the hexagonal tessellation used in image processing. It is shown that one important example, the generalized balanced ternary, provides replicating tessellations in all dimensions. Subjects not treated in this paper, but of related interest, include L codes and ambiguity [3], [14], [23].

2. Fundamental domain. The notation $AX = A(X)$ will be used hereafter. For a lattice Λ, both Λ and $A\Lambda$ are abelian groups under addition. Define a *fundamental domain* S to be a set of coset representatives of the quotient $\Lambda/A\Lambda$. Indeed, if V is the union of the Voronoi cells corresponding to the points of such a set S, then V is a fundamental domain (Dirichlet domain) for the group of isometries of $\mathbb{R}^n$ that are translations by vectors in $A(\Lambda)$. If S is a fundamental domain, then [15]

$$|S| = |\det A|.$$

In Example 1 of the Introduction, $A = (b)$ and $\Lambda/A\Lambda = \mathbb{Z}/b\mathbb{Z}$. So a fundamental domain S, in this case, is a complete set of residues modulo $|b|$ and $|S| = |b|$. In Example 3 of the Introduction, $|S| = det(A) = 7$, corresponding to the seven lattice points in the 0-aggregate.

PROPOSITION 2. *Let $A : \Lambda \to \Lambda$ be an endomorphism.*

(1) *If S induces a rep-tiling of Λ, then S must be a fundamental domain.*

(2) *If S is a fundamental domain, then* (i) $S_m = \sum_{i=0}^{m} A^i(S)$ *tiles Λ by translation by the sublattice $A^{m+1}(\Lambda)$ for all $m \geq 0$, and* (ii) *the finite address of a lattice point, if it exists, is unique.*

Proof. Condition (1) in the definition of rep-tiling, with $m = 0$, is equivalent to S being a fundamental domain. To show (2), assume S is a fundamental domain and, by way of contradiction, assume the existence of lattice point with two distinct finite addresses. Thus $\sum_{i=0}^{m} A^i s_i = \sum_{i=0}^{m} A^i t_i$ for some $s_i, t_i \in S$ and, without loss of generality, $s_0 \neq t_0$. But this implies that $s_0 \equiv t_0 (\mathrm{mod} A\Lambda)$, a contradiction. To show that $S_m = \sum_{i=0}^{m} A^i(S)$ tiles Λ by translation, note that $\{S_x \; : \; x \in A\Lambda\}$ tiles Λ by translates of S. Iterate to obtain successive tilings

$$\begin{aligned} \Lambda &= S + A\Lambda \\ &= S + A(S + A\Lambda) = S + AS + A^2\Lambda = S_2 + A^2\Lambda \\ &\dots \\ &= (S + AS + \cdots + A^m S) + A^{m+1}\Lambda = S_m + A^{m+1}\Lambda. \end{aligned}$$ □

According to Proposition 2, if S is a fundamental domain then the finite address of a lattice point, if it exists, is unique. It is a consequence of topics in §5 that every lattice point has a unique infinite address, which coincides with the finite address in the case that all digits after a certain position are zero.

3. Equivalent tessellations. Matrix transformations $A : \Lambda \to \Lambda$ and $B : \Gamma \to \Gamma$ of lattices Λ and Γ, respectively, are said to be equivalent if there exists an invertible matrix Q, such that $B = QAQ^{-1}$ and $\Gamma = Q\Lambda$. Proposition 3 essentially states that questions about replicating tessellations are invariant under equivalence.

PROPOSITION 3. *Assume that $A : \Lambda \to \Lambda$ and $B : \Gamma \to \Gamma$ are equivalent via matrix Q.*

(1) *S is a fundamental domain for A if and only if QS is a fundamental domain for B.*

(2) $s_0\, s_1 \ldots s_m$ *is the finite address of* $x \in \Lambda$ *if and only if* $Qs_0\, Qs_1 \ldots Qs_m$ *is the finite address of* $Qx \in \Gamma$.

(3) S *induces a rep-tiling of* Λ *if and only if* QS *induces a rep-tiling of* Γ.

Proof. Concerning (1), there is a partition $\Lambda = S + A\Lambda$ if and only if there is a partition $\Gamma = Q\Lambda = QS + QA\Lambda = QS + B\Gamma$. Concerning (2), $x = \sum_{i=0}^{m} A^i s_i$ if and only if $Qx = \sum_{i=0}^{m} QA^i s_i = \sum_{i=0}^{m} B^i(Qs_i)$. Statement (3) follows from statement (2). □

Remark. Since equivalence is essentially a change of basis for the matrix A, there exist equivalent matrices in several canonical forms. By changing to a basis of the lattice Λ itself, an equivalent integer matrix $B : \mathbb{Z}^n \to \mathbb{Z}^n$ is obtained. Hence, from the point of view of replicating tessellations, there is no loss of generality in assuming that A is an integral matrix acting on the cubic lattice $\mathbb{Z}^n$. In particular, the characteristic and minimal polynomials for A have integral coefficients. Similarly, we can obtain equivalent matrices in Jordan canonical form or rational canonical form. As an example, consider the matrix A associated with the hexagonal tiling in the Introduction. Then

$$B = \begin{pmatrix} 2 & 1 \\ -1 & 3 \end{pmatrix}, \quad J = \begin{pmatrix} 2+\omega_1 & 0 \\ 0 & 2+\omega_2 \end{pmatrix}, \quad R = \begin{pmatrix} 0 & -7 \\ 1 & 5 \end{pmatrix}$$

are equivalent integral, Jordan, and rational forms, where ω_1 and ω_2 are the complex third roots of unity and the lattice for R is $\mathbb{Z}^2$. The lattice for the Jordan canonical form is actually a two-dimensional real lattice in $\mathbb{C}^2$.

Recall that a linear expansive map A is one for which each eigenvalue is greater than 1.

PROPOSITION 4. *If* (A, S) *is a rep-tiling pair, then* A *must be a linear expansive map.*

Proof. Assume that A has an eigenvalue of modulus $\epsilon < 1$. By the remark above, the $n \times n$ matrix A may be assumed in Jordan canonical form. Assume J is an $m \times m$ Jordan block of A of the form $\epsilon I + N$ corresponding to eigenvalue ϵ, where N is the nilpotent matrix consisting of all 0's except 1's just below the diagonal. (Without loss of generality, assume that J is the topmost block of A.) Let T be the projection of the fundamental domain S on the first m coordinates. For $t \in T$, an easy calculation shows that each entry in the matrix J^k is $O(k^m \epsilon^k)$. Since $|t| < C$ for all $t \in T$ and some bound C, then also $|J^k t| = O(k^m \epsilon^k)$ and $|\sum_{i=0}^{\infty} J^i t_i| < \sum_{i=0}^{\infty} O(i^m \epsilon^i) < c$ for some constant c. Hence, the component consisting of the first m coordinates of any finite address is bounded. However, there exists lattice points where this component is arbitarily large.

Next, assume that A has an eigenvalue λ_0 of modulus 1. By the remark above, A may be assumed to be an integer matrix. Let $p(x) \in \mathbb{Z}[x]$ be a factor of the characteristic polynomial, irreducible over $\mathbb{Z}$, with root λ_0. Assume that $p(x)$ has a root λ with modulus not equal to 1. If $|\lambda| < 1$, then we are done by the paragraph above, so assume $|\lambda| > 1$. There exists a Galois automorphism ϕ of $\mathbb{Q}[x]$ fixing $\mathbb{Q}$ elementwise, such that $\phi(\lambda_0) = \lambda$. Now $\phi(\lambda_0)\phi(\overline{\lambda_0}) = 1$ implies $|\phi(\overline{\lambda_0})| < 1$. Again, an eigenvalue has modulus less than 1, a contradiction. Therefore, all roots of $p(x)$ have modulus 1. However, Polya and Szegö [24, p. 145] prove the following result due to Kronecker [19]: If $p(x)$ is an irreducible monic polynomial with integer coefficients such that all roots lie on the unit circle, then the roots of $p(x)$ are roots of unity. This implies that A^j has eigenvalue 1 for some positive integer j. Because A is an integer matrix, the corresponding eigenvector x can be taken to have integer coordinates. If $s_0 s_1 \ldots s_r$ is the finite address of x, then the finite address of $x = A^j x$ is $00\ldots 0 s_1 s_2 \ldots s_r$, where the initial segment has j 0's. Now x has two finite addresses, which contradicts the assumption that the finite address is unique. □

The condition that A be expansive is necessary for (A, S) to be a rep-tiling pair, but it is not sufficient. There are matrices A satisfying the eigenvalue condition that admit *no* fundamental domain S for which S induces a rep-tiling. In dimension 1, for example, $A = (2)$ is such a matrix. (This is the only example in dimension 1.) Examples in all dimensions are given in the comments after Corollary 1 of Theorem 3 in §6. Nevertheless, Proposition 4 is sharp in the sense that, for any $\epsilon > 0$, there exists a matrix A_ϵ and a fundamental domain S_ϵ, such that A_ϵ has an eigenvalue of modulus a where $|a - 1| < \epsilon$, and such that (A_ϵ, S) is a rep-tiling pair. The generalized balanced ternary is proved, in the remark at the end of §7, to be such an example.

4. Self-replicating tilings. The proof of Theorem 1 appears in this section, as well as the proof of the converse, Theorem 2.

THEOREM 1. *If (A, S) is a rep-tiling pair for Λ and $T = T(A, S)$, then*

(1) *T is compact and is the closure of its interior;*

(2) *T tiles $\mathbb{R}^n$ periodically by translation by the lattice Λ; and*

(3) *the tiling is self-replicating.*

Proof. Since, by Proposition 4, all the eigenvalues of A^{-1} are less than 1, the set E_m is bounded, the bound independent of m; therefore, T is compact. Consider statement (2) of Theorem 1. Condition (1) in the definition of a rep-tiling pair (A, S) guarantees that E_m tiles $A^{-m}(\Lambda)$ by translation by Λ. Therefore, E tiles $\bigcup_{m=1}^{\infty} A^{-m}(\Lambda)$ by translation by Λ. The facts that $\bigcup_{m=1}^{\infty} A^{-m}(\Lambda)$ is dense in $\mathbb{R}^n$ and E is bounded imply $\mathbb{R}^n = \bigcup_{x \in \Lambda} T_x$. To show that the intersection of the interiors of distinct tiles is empty, it suffices to prove that $\mu(T_x \cap T_y) = 0$, where μ is Lebesgue measure. By condition (2) in the definition of a rep-tiling pair, there is an integer m such that $x, y \in S_m$. From the definition of E it follows that $A^{m+1}(E) = \bigcup_{w \in S_m} E_w$, which implies that $A^{m+1}(T) = \bigcup_{w \in S_m} T_w$. Now $(\det A)^{m+1} \mu(T) = \mu(A^{m+1}(T)) = \mu(\bigcup_{w \in S_m} T_w) \leq \sum_{w \in S_m} \mu(T_w) = (detA)^{m+1} \mu(T)$ implies that $\mu(\bigcup_{w \in S_m} T_w) = \sum_{w \in S_m} \mu(T_w)$. This, in turn, implies that $\mu(T_x \cap T_y) = 0$. Concerning statement (3) in the theorem it follows as above, with $m = 1$, that $A(T) = \bigcup_{w \in S} T_w$. Then $A(T_x) = \bigcup_{w \in S_{Ax}} T_w$.

Consider statement (1) in the theorem. To prove that T is the closure of its interior, it suffices to show that each point $x \in E$ in an interior point of T. Let F^0 denote the interior of a point set F. Assume that $0 \in T^0$. Since there is a nonnegative integer m such that $x \in E_m$, we have $z := A^m x \in A^m(E_m) = S_{m-1} \subset \Lambda$. Therefore, $x \in T^0$ if and only if $z \in (A^m T)^0$ if and only if $0 \in (-z + A^m T)^0$. Since $z + E \subseteq A^m E$, we have $T \subseteq -z + A^m T$. Therefore, if $0 \in T^0$, then $x \in T^0$. Recall that T is compact and that $\mathbb{R}^n = \bigcup_{x \in \Lambda} T_x$. Hence, to show $0 \in T^0$, it suffices to prove that $0 \notin T_x$ for all $x \in \Lambda - \{0\}$. Assume, by way of contradiction, that $0 \in T_x$ for some $x \in S_k - S_{k-1}$ for some fixed $k \geq 1$. Let $L_m = \{A^m S_m + A^{m-1} s_{m-1} + \cdots + s_0 \; : \; s_i \in S, \, s_m \neq 0\}$. We claim that $\lim_{m \to \infty} \min_{y \in L_m} |y| = \infty$. To see this, note that for any $R > 0$ there exists an integer m, such that $\bigcup_{i=0}^{m} L_i$ includes all points of Λ within a sphere of radius R. Since finite addresses are unique, the points of L_{m+1} lie outside the sphere. Next, choose B such that $|y| < B$ for all $y \in T$ and choose m_0 such that $|y| > 2B$ for all $y \in L_m$, $m \geq m_0$. Let $\alpha = \sup_{x \in \mathbb{R}^n} |Ax|/|x|$ and $\epsilon = B/\alpha^{m_0 - k}$. From the choice of x, there is a $y \in E$, such that $|z| < \epsilon$, where $z = x + y$. Now $A^{m_0 - k} z \in L_{m_0} + E$ implies that $|A^{m_0 - k} z| > 2B - B = B$, which in turn implies that $|z| > B/\alpha^{m_0 - k} = \epsilon$, a contradiction. □

The periodic, self-replicating tiling by a single tile $T(A, S)$ given by Theorem 1 is said to be *induced* by the rep-tile pair (A, S). The next result states that essentially every periodic self-replicating tiling is induced by a rep-tiling pair.

THEOREM 2. *Consider a periodic, self-replicating tiling by a single tile T. If* (1) *T is compact and is the closure of its interior, and* (2) *the origin is contained in the interior of T, then the tiling is induced by a rep-tiling pair.*

Proof. Let A and Λ be the linear expansive map and the lattice, respectively, associated with the periodic, self-replicating tiling. Let S be the finite subset of Λ, such that $A(T) = \bigcup_{x\in S} T_x$. We claim $A(\Lambda) \subseteq \Lambda$. To see this, let x be any point of Λ and note that $\bigcup_{w\in S(x)} T_w = A(T_x) = Ax + A(T) = Ax + \bigcup_{w\in S} T_w = \bigcup_{w\in Ax+S} T_w$. Because this equality involves only finitely many compact tiles, $Ax + S = S(x)$. In particular, $Ax \in S(x) \subset \Lambda$.

We next prove that (A, S) is a rep-tiling pair. To show that S is a fundamental domain for A, consider the Lebesgue measure on both sides of the equation $A(T) = \bigcup_{x\in S} T_x$. This gives $|S| = \det(A)$. Since S has the correct number of elements, it suffices to show that no two elements of S are congruent $\mathrm{mod}(A\Lambda)$. Assume, by way of contradiction, that $s = s' + Ax$ for some $x \in \Lambda - \{0\}$. Then s lies in the interior of T_s and hence in the interior of $A(T)$. Also, s' lies in the interior of $T_{s'}$, and hence s lies in the interior of $Ax+A(T) = A(T_x)$. However, the intersection of the interiors of T and T_x is empty, and hence, the same is true for $A(T)$ and $A(T_x)$, a contradiction. By Proposition 2, condition (1) in the definition of a rep-tiling pair is satisfied.

Iterating $A(T) = \bigcup_{x\in S} T_x$, we obtain $A^m(T) = \bigcup_{x\in S_{m-1}} T_x$. Because 0 lies on the interior of T, for any lattice point x there is an integer m such that $T_x \subset A^m(T) = \bigcup_{w\in S_{m-1}} T_w$. This implies that $x \in S_{m-1}$, proving condition (2) in the definition of a rep-tiling pair.

It remains to prove that $T = T(A, S)$. By the formula in the paragraph above, $S_{m-1} \subset A^m(T)$, which implies $E_m = A^{-m}(S_{m-1}) \subset T$ for all $m \geq 0$. This, in turn, implies $T(A, S) \subseteq T$. Since $T(A, S)$ tiles $\mathbb{R}^n$ by translation by Λ, the interior points of T satisfy $T^0 \subseteq T(A, S)$. Therefore $T = \overline{T^0} \subseteq T(A, S)$. □

Theorem 2 is false without the assumption that 0 is contained in the interior of some tile T. For example, the tiling of $\mathbb{R}$ by translates of the unit interval $T = [0, 1]$ is not induced by a rep-tile pair. This tiling is indeed induced by the pair (A, S) where $A = (3)$ and $S = \{0, 1, 2\}$, but (A, S) is not a rep-tiling pair because the negative integers have no base 3 radix representation with digit set S, i.e., -1 belongs to no aggregate. However, the tiling of $\mathbb{R}$ by translates of $T = [-\frac{1}{2}.\frac{1}{2}]$ *is* induced by the rep-tile pair (A, S) where $A = (3)$ and $S = \{-1, 0, 1\}$. It is an open question whether every periodic, self-replicating tiling is induced, up to a translation, by a rep-tiling pair.

5. A-adic integers.

It is assumed here that S is a fundamental domain for the matrix $A : \Lambda \to \Lambda$ and that A is expansive. This implies, in particular, that $|\det A|$ is an integer greater than or equal to 2.

LEMMA 1. *If A is a linear expansive map, then*

$$\bigcap_{i=0}^{\infty} A^i\Lambda = \{0\}.$$

Proof. If all eigenvalues have modulus greater than 1, then examination of the Jordan canonical form shows that $A^m x \to \infty$ for all nonzero x. □

For $x \in \Lambda$, let $\nu = \nu(x)$ denote the greatest integer ν, such that $x \in A^\nu\Lambda$. By Lemma 1, ν is finite except when $x = 0$, in which case we set $\nu(0) = \infty$. Then

$$|x| = \frac{1}{|\det A|^{\nu(x)}}$$

has the property that $|x| = 0$ if and only if $x = 0$ and thus defines a norm on Λ, and $d(x, y) = |x - y|$ defines a metric we call the *A-adic metric*. Two lattice points are close in the corresponding topology if their difference lies in $A^m\Lambda$ for large m. If $A = (p)$ is a one-dimensional matrix, then this reduces to the classical p-adic metric where two integers are close if their difference is divisible by a large power of p. The completion of Λ with respect to the A-adic metric will be called the *A-adic integers* and denoted $\overline{\Lambda}$. (Alternatively, the A-adic integers can be defined as an inverse limit of the system $(\{\Lambda/A^k\Lambda\}, \{f_{jk}\})$, where $f_{jk} : \Lambda/A^k\Lambda \to \Lambda/A^j\Lambda$, $j < k$ is defined by $f_{jk}\bar{x}_k = \bar{x}_j$, where $x_j \equiv x_k \text{mod} A^j\Lambda$.) Note that $\Lambda \subseteq \overline{\Lambda}$. If S is any set of coset representatives for $\Lambda/A\Lambda$, then, just as for the case of the ordinary p-adic integers, there is a unique canonical representation of each A-adic integer in the form $\sum_{i=0}^{\infty} A^i s_i$, where $s_i \in S$, which will be abbreviated $s_0\, s_1\, s_2 \ldots$ and called the *A-adic address*. The partial sums in this canonical form converge to the A-adic integer in the A-adic metric.

A simple recursive algorithm to determine the A-adic address $s_0 s_1 s_2 \ldots$ of a lattice point is obtained by iteration using the partition $\Lambda = \{S + x : x \in A\Lambda\}$, from the assumption that S is a fundamental domain. This process is analogous to finding the base b digits in the radix representation of a given integer.

ALGORITHM A. The ith entry s_i, $i = 0, 1, \ldots$, in the A-adic address of a lattice point $x_0 \in \Lambda$ is the unique element of S, such that

$$s_i \equiv x_i \pmod{A\Lambda},$$

where

$$x_{i+1} = A^{-1}(x_i - s_i).$$

The A-adic address $s_0 s_1 \ldots$ of a lattice point is called *finite* if $s_i = 0$ for all i sufficiently large. The A-adic address of every lattice point is finite if and only if (A, S) is a rep-tiling pair. For $A = (3)$ and $S = \{-1, 0, 4\}$, which is in the first example of the Introduction, Algorithm A yields A-adic addresses:

$$1 = (4)(-1) = 4 + (-1)3,$$
$$-2 = 444\ldots.$$

Since -2 has no finite address, S does not induce a replicating tessellation on $\mathbb{Z}$. For the matrix

$$A = \begin{pmatrix} 2 & 1 \\ -1 & 3 \end{pmatrix} : \mathbb{Z}^2 \to \mathbb{Z}^2$$

with

$$S = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \ldots \begin{pmatrix} 6 \\ 0 \end{pmatrix} \right\}$$

the A-adic address of $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ is

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}\begin{pmatrix} 6 \\ 0 \end{pmatrix}\begin{pmatrix} 2 \\ 0 \end{pmatrix}\begin{pmatrix} 3 \\ 0 \end{pmatrix}\begin{pmatrix} 3 \\ 0 \end{pmatrix}\cdots.$$

Again, this shows that S does not induce a replicating tessellation of $\mathbb{Z}^2$.

If a lattice point x has an A-adic address with repeating string $s_{i+1} \dots s_{i+q}$, we say that x has a *repeating address*. Although a lattice point may not have a finite A-adic address, the next result shows that every lattice point has a repeating A-adic address.

LEMMA 2. *The* A*-adic address of any point in* Λ *is repeating.*

Proof. According to Algorithm A, when x_i takes on a value a second time the address repeats. Hence it is sufficient to show that the sequence $\{x_i\}$ is bounded. Iterating the formula in the algorithm gives $x_m = A^{-m}x - \sum_{i=0}^{m-1} A^{-m+i}s_i$. If $1/\alpha$ is the eigenvalue of A with the least modulus, then α is the eigenvalue of A^{-1} with the greatest modulus. Choose a real number a such that $1 > a > |\alpha|$. A calculation using the Jordan canonical form suffices to show that all entries of the matrix $(1/aA^{-1})^m$ tend to 0 as $m \to \infty$. This implies that $|A^{-m}x| < a^m|x|$ for m sufficiently large. Hence, there is a constant c such that $|A^{-m}x| < ca^m|x|$, where c is independent of m. This implies, for any m, a bound $|x_m| < c|x| + \frac{tc}{1-a}$, where $t = \max\{|s| \, : \, s \in S\}$. □

Note that the proof above gives an upper bound on the number of iterations in Algorithm A necessary to determine whether or not a given lattice point x has a finite address. For example, in dimension 1 with $A = (b)$ the bound is $|x| + \max\{|s| \, : \, s \in S\}/(|b| - 1)$.

6. Necessary and sufficient conditions for replicating tessellations. This section contains several necessary and sufficient conditions for the existence of replicating tessellations, thus providing some answers to the two main questions posed in the Introduction. Again it is assumed throughout that S is a fundamental domain for A and that A is expansive. Note that the matrix $(I - A^m)$ is nonsingular for any positive integer m. Otherwise, 1 would be an eigenvalue of A^m, and hence A would have an eigenvalue of modulus 1.

THEOREM 3. *Given* $A : \Lambda \to \Lambda$, *the following statements are equivalent*:

(1) S *induces a rep-tiling of* Λ.

(2) $(I - A^{m+1})^{-1}S_m$ *contains no nonzero lattice point for* $m = 0, 1, \dots$.

(3) $(I - A^{m+1})^{-1}S_m$ *contains only lattice points with finite address for* $m = 0, 1, \dots$.

Proof. (3) ⇒ (1) Assume S does not induce a replicating tessellation on Λ. According to Lemma 2 some lattice point y has a repeating address where the repetition is not zeros. If there is an initial segment y_0 of length q before the address begins repreating, then $y - y_0 \in A^q\Lambda$ and $x = A^{-q}(y - y_0) \in \Lambda$ consists of that portion of y that repeats from the beginning. Let $s_0, s_1, \dots, s_m$ be the repeating digits in the address of x. Then $(I - A^{m+1})x = \sum_{i=0}^{m} A^i s_i$, and therefore $x \in (I - A^{m+1})^{-1}S_m$, where x does not have finite address.

(2) ⇒ (3) Clearly, if $(I - A^{m+1})^{-1}S_m$ contains a lattice point without finite address, then it contains a nonzero lattice point.

(2) ⇒ (1) Finally, assume that $(I - A^{m+1})^{-1}S_m$ contains a nonzero lattice point x. Then $(I - A^{m+1})x = \sum_{i=0}^{m} A^i s_i$ with $s_i \in S$. The lattice point y whose infinite address consists of the digits $s_0, s_1, \dots, s_m$ repeated satisfies the same equation $(I - A^{m+1})y = \sum_{i=0}^{m} A^i s_i$. Since $I - A^{m+1}$ is nonsingular, $x = y$ has a repeating (not finite) address, and therefore, S does not induce a rep-tiling of Λ. □

Theorem 3 implies, in particular, that if (A, S) is a rep-tiling pair, then S cannot contain any nonzero element of $(I - A)\Lambda$. In dimension 1, if $A = (b)$, then S can contain no integer divisible by $b - 1$, a result given in [21]. For example, with $S = \{-2, 0, 2\}$ there exist integers with no finite base 3 radix representation. Moreover, we have the following result.

COROLLARY 1. *Given* $A : \Lambda \to \Lambda$, *if* $\det(I - A) = \pm 1$, *then* A *admits no fundamental domain* S *such that* S *induces a rep-tiling of* Λ.

Proof. If $\det(I-A) = \pm 1$, then $(I-A)\Lambda = \Lambda$. Therefore, S must contain a nonzero element of $(I-A)\Lambda$. □

Examples of such matrices acting on the cubic lattice that admit no fundamental domain include all matrices of the form

$$\begin{pmatrix} 0 & -m \\ 1 & m \end{pmatrix} \qquad \text{and} \qquad 2I + H,$$

where m is an integer and H is strictly upper trianglular.

Theorem 4, based on Theorem 3, states that only lattice points within a bounded region need be tested for the existence of a finite address. This leads to an efficient algorithm to determine, given $A : \Lambda \to \Lambda$ and a fundamental domain S, whether or not S induces a replicating tessellation of Λ.

LEMMA 3. *The sets* $(I - A^{m+1})^{-1}S_m$, $m = 0, 1, \ldots$ *are contained in some ball centered at the origin whose radius is independent of m.*

Proof. Let $H = A^{-1}$ and let $H = Q^{-1}\overline{H}Q$, where $\overline{H}$ is the Jordan canonical form of H and Q is an appropriate nonsingular matrix. Suppose that c is a constant, such that $\overline{H}QS$ is contained in a "box" $B(c) = \{(y_1, y_2, \ldots, y_n) : |y_i| \leq c, i = 1, 2, \ldots, n\}$. Furthermore, let $C = \frac{c}{a}(1/(1-a))^n$, where a is the modulus of the largest eigenvalue of H. Note that $a < 1$, since it is assumed that the moduli of all eigenvalues of A are greater than one. Each entry in $\overline{H}^m$ approaches 0 as $m \to \infty$. Hence, for m large enough, say $m > m_0$, we have $B(C) \subseteq (\overline{H}^{m+1} - I)B(2C)$. For $m \leq m_0$, there is a constant K, such that $(H^{m+1} - I)^{-1}(B(c) + \overline{H}B(c) + \cdots + \overline{H}^m B(c)) \subseteq B(K)$. Let B' be the larger of the two boxes $B(2C)$ and $B(K)$ and let $B = Q^{-1}B'$. In the statement of the lemma, take any ball containing B.

Now $(I - A^{m+1})^{-1}S_m \subseteq B$ if and only if $S_m \subseteq (I - A^{m+1})B$. Multiplying by H^m gives the sufficient condition $B(c) + \overline{H}B(c) + \cdots + \overline{H}^m B(c) \subseteq (\overline{H}^{m+1} - I)B'$. This is true by definition for $m \leq m_0$. For $m > m_0$, it suffices to examine the situation on each Jordan block of $\overline{H}$ of the form $J = \alpha I + N$, where α is an eigenvalue of H and N is the nilpotent component of the Jordan block. An upper bound on the modulus of any coordinate of $B(c) + JB(c) + \cdots + J^m B(c)$ is

$$c\sum_{k=0}^{m}\sum_{j=0}^{n}\binom{k}{j}|\alpha|^{k-j} \leq c\sum_{j=0}^{n}\sum_{k=0}^{\infty}\binom{j+k}{j}a^k = c\sum_{j=0}^{n}\left(\frac{1}{1-a}\right)^{j+1} \leq \frac{c}{a}\left(\frac{1}{1-a}\right)^n = C.$$

Therefore, we have $B(c) + \overline{H}B(c) + \cdots + \overline{H}^m B(c) \subset B(C) \subset (\overline{H}^{m+1} - I)B(2C) \subset (\overline{H}^{m+1} - I)B$. □

THEOREM 4. *There exists a ball B centered at the origin, with radius depending only on A and S, such that* (A, S) *is a rep-tiling pair if and only if each lattice point in B has a finite address.*

Proof. If (A, S) is a rep-tiling pair, then every lattice point in B has a finite address because every lattice point does. The converse follows from Theorem 3 and Lemma 3. □

For particular cases, it is possible to give an explicit value for the radius of the ball B. An efficient algorithm to determining whether or not (A, S) is a rep-tiling pair is obtained by applying Algorithm A to each of the finite number of lattice points in B. Then (A, S) is a rep-tiling pair if and only if each of these A-adic addresses is finite. Two examples are considered.

Similarities. Consider the case where $A = bU$, where U is an isometry and the real number b is greater than 1 in absolute value. Call such a matrix a *similarity*. The techniques of Theorem 4 yield the following version.

COROLLARY 2. *Given a similarity $A : \Lambda \to \Lambda$, a set S induces a rep-tiling of Λ if and only if every lattice point in the ball of radius* $\max\{|s| \, : \, s \in S\}/(|b| - 1)$ *centered at the origin has a finite address.*

Applying this result to the one-dimensional case gives the following corollary, which is proved by other means in [21].

COROLLARY 3. *Every integer has a unique base b radix representation with digit set S if and only if every integer in the interval*

$$\left[-\frac{\max\{|s| \, : \, s \in S\}}{|b| - 1}, \frac{\max\{|s| \, : \, s \in S\}}{|b| - 1}\right]$$

has such a representation.

Diagonalizable matrices. If matrix A is diagonalizable, for example, if the minimal polynomial of A is irreducible over the integers, then the following result is obtained by the methods of Theorem 4. Let Q be a nonsingular matrix and D a diagonal matrix such that $D = QAQ^{-1}$. By a *box* is meant a set of the form $\{(z_1, z_2, \dots, z_n) \, : \, |z_i| \le c_i, \, i = 1, 2, \dots, n\}$ for some constants c_i. Let B_{QS} be the smallest box containing QS.

COROLLARY 4. *With notation as above, for the diagonalizable matrix A, the set S induces a rep-tiling of Λ if and only if each lattice point in $Q^{-1}(B_{QS})$ has a finite address.*

Example. Consider the matrix

$$A = \begin{pmatrix} 4 & -1 \\ -1 & 6 \end{pmatrix}$$

acting on the square lattice $\mathbb{Z}^2$. A fundamental domain S has cardinality $\det(A) = 23$; let S be the set of 23 circled latticed points in Fig. 6(a). Calculation shows that $Q^{-1}(B_{QS})$ is the rectangle indicated in the figure. It is routine to check that all lattice points within this rectangle are in the first aggregate (having, in fact, finite addresses of length at most two). By Corollary 4, this constitutes a proof that S induces an aggregate tessellation of $\mathbb{Z}^2$.

Theorem 5 essentially states that if, in adding two elements of S there is only one carry digit in the finite address of the sum, then S induces a replicating tessellation. More specifically, the sum and difference of any two vectors in the fundamental domain lie in the first aggregate.

THEOREM 5. *Given $A : \Lambda \to \Lambda$ and a fundamental domain S, if*

(1) *some aggregate contains a basis for the lattice Λ and*

(2) $S \pm S \subseteq S + A(S)$,

then S induces a rep-tiling of Λ.

Proof. By Proposition 2 it is sufficient to show that every lattice point has a finite address. Since some aggregate S_m contains a basis, every element of Λ is the sum of a finite number of elements with finite address of the form $\pm s_0 s_1 \dots s_m$, possibly with summands repeated many times. The proof is by induction on the number k of summands. It is clearly true for $k = 0$. Assume that every lattice point that is the sum of $k - 1$ terms has a finite address and let x be the sum of k elements of S. By induction, the sum of the first $k - 1$ of these k terms has the form $x = s_0 s_1 \dots s_q$, where $s_i \in S$. Let $y = t_0 t_1 \dots t_m$ be the kth term, where $t_i \in \pm S$. Since $S \pm S \subseteq S + AS$, addition of x and y is performed on the respective addresses from the left, where the number of carries to

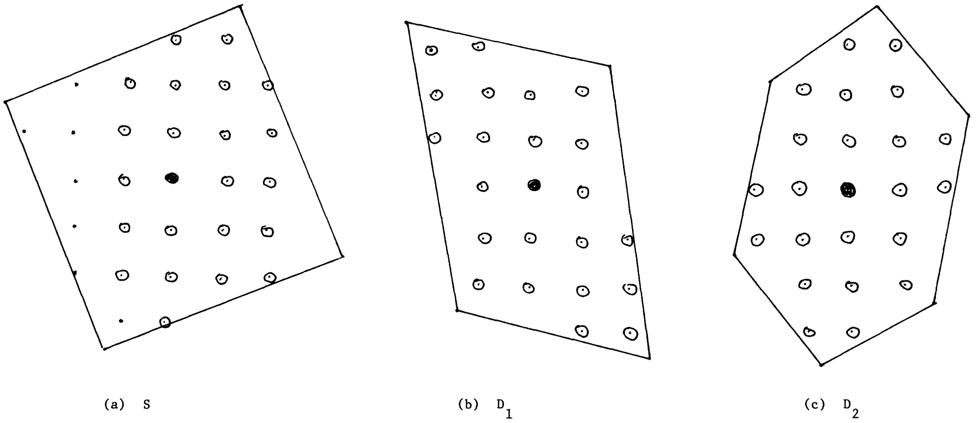


FIG. 6. *Fundamental domains for matrix A in the lattice* $\mathbb{Z}^2$.

the next digit to the right (e.g., place $i+1$) is one less than the number of summands at place i. It is not hard to deduce that the number of carries never exceeds $m+1$, and at place $\max(m,q)+i$ the number of carries does not exceed $m-i+1$. Hence, all digits after place $\max(m,q)+m+1$ are zero. □

For an n-dimensional lattice Λ, let G be the group of isometries of $\mathbb{R}^n$ generated by the n translations, taking the origin to each of n basis vectors of Λ. A *Dirichlet domain* is a subset $F \subset \mathbb{R}^n$, such that $\mathbb{R}^n$ is the disjoint union of the images of F under G. It is well known that there is a Dirichlet domain $\mathbf{V}_\Lambda$ whose closure is the Voronoi region of the lattice Λ. Call this Dirichlet domain $\mathbf{V}_\Lambda$ the *Voronoi domain* of Λ. The radius of the largest ball centered at the origin and contained in the Voronoi region is called the *packing radius* of Λ, and the radius of the smallest ball centered at the origin containing the Voronoi region is called the *covering radius* of Λ. Note that the packing radius is half the length of a minimum norm vector in Λ.

Theorem 6 states that, for a large class of matrices $A : \Lambda \to \Lambda$, there exists some fundamental domain S such that S induces a rep-tiling of Λ. In fact, a number of distinct viable fundamental domains can be obtained as the sets of lattice points contained in Voronoi domains of certain sublattices of Λ.

LEMMA 4. $\Lambda \cap \mathbf{V}_{A\Lambda}$ *is a fundamental domain for* $A : \Lambda \to \Lambda$.

Proof. Let $D = \Lambda \cap \mathbf{V}_{A\Lambda}$. By definition, $\{\mathbf{V}_{A\Lambda} + Ax : x \in \Lambda\}$ is a partition of $\mathbb{R}^n$. Hence $\{D_x : x \in A\Lambda\}$ is a partition of Λ. This is equivalent to saying that D is a fundamental domain for $A : \Lambda \to \Lambda$. □

LEMMA 5. *Given* $A : \Lambda \to \Lambda$, *let* $D = \Lambda \cap \mathbf{V}_{A\Lambda}$. *Assume that*

(1) *the set of minimum norm vectors in* $A\Lambda$ *contains a basis for* $A\Lambda$, *and*

(2) *all singular values of A are greater than* $3R/r$, *where* r *is the packing radius and* R *is the covering radius of* $A\Lambda$.

Then D induces a rep-tiling of Λ. *In the one- and two-dimensional cases, the bound* $3R/r$ *can be improved to* 2.

Proof. The proof uses Theorem 5 by showing that (1) D contains a basis for Λ, and (2) $D \pm D \subseteq D + AD$. To prove (1) we show that if $v_1, \dots, v_n$ constitute a basis of minimum norm vectors of $A\Lambda$, then $A^{-1}v_1, \dots, A^{-1}v_n$ is a basis for Λ contained in D. The condition on the singular values of A implies that $|A^{-1}x| < r/3R|x| < \frac{1}{2}|x|$ for all $x \in \mathbb{R}^n$. Therefore, $|A^{-1}v_i| < \frac{1}{2}|v_i| \leq r$, which implies that $A^{-1}v_i \in D$ for all i.

Concerning the second condition, if $x \in D \pm D$, then $|x| \leq 2R$. By Lemma 4, we know that D is a fundamental domain for A, and hence, $x = s + Ay$, where $s \in D$ and $y \in \Lambda$. It now suffices to show that $y \in D$. But $y = A^{-1}(x - s)$ implies $|y| < r/3R(|x|+|s|) \leq r/3R(2R+R) = r$. Therefore, $y \in D$. The improvement in dimensions 1 and 2 is obtained by showing directly that $|Ay| \leq 2r$, and hence $|y| < r$ if all singular values of A are greater than 2. □

A similarity of the form bU, where U is an isometry and $b > 3\sqrt{n}$ ($b > 2$ in dimensions 1 and 2), satisfies the hypotheses of Lemma 5 if the lattice Λ itself has a basis of minimum norm vectors. Applying Lemma 5 in dimension 1 gives: if $|b| > 2$, then every integer has a unique base b radix representation with digits in $D = \{-\lfloor |b| - 1/2 \rfloor, \dots, \lfloor |b|/2 \rfloor\}$. This is also proved in [21]. As another example, let D consist of the 9 lattice points with coordinates 0 or ±1. Applying Lemma 5 to

$$A = \begin{pmatrix} 0 & -3 \\ 3 & 0 \end{pmatrix} : \mathbb{Z}^2 \to \mathbb{Z}^2$$

implies that D induces a rep-tiling of $\mathbb{Z}^2$.

The first assumption in Lemma 5, concerning the minimum norm vectors, may very well fail in general. To remedy this situation, merely transform $A\Lambda$ to a lattice Λ_0 known to be generated by the minimum norm vectors.

THEOREM 6. *Given $A : \Lambda \to \Lambda$, let Q be any nonsingular matrix such that lattice $\Lambda_0 = Q(A\Lambda)$ is generated by its minimum norm vectors. If all singular values of QAQ^{-1} are greater than $3R/r$, where r is the packing radius and R is the covering radius of Λ_0, then $D = \Lambda \cap Q^{-1}\mathbf{V}_{\Lambda_0}$ induces a rep-tiling of Λ. In the one- and two-dimensional cases, the bound $3R/r$ can be improved to 2.*

Proof. Let $A_0 = QAQ^{-1}$. By definition, $A : \Lambda \to \Lambda$ and $A_0 : Q\Lambda \to Q\Lambda$ are equivalent. By Proposition 3 of §3, (A, D) is a rep-tiling pair for Λ if and only if (A_0, QD) is a rep-tiling pair for $Q\Lambda$. But $QD = Q\Lambda \cap \mathbf{V}_{\Lambda_0} = Q\Lambda \cap \mathbf{V}_{QA\Lambda} = Q\Lambda \cap \mathbf{V}_{A_0 Q\Lambda}$. The theorem now follows directly from Lemma 5 applied to A_0. □

Note that, in Theorem 6, if B is the matrix whose columns are a basis for Λ_0, then we can take

$$Q = BA^{-1},$$
$$D = \Lambda \cap AB^{-1}\mathbf{V}_{\Lambda_0},$$

in which case $QAQ^{-1} = BAB^{-1}$.

COROLLARY 5. *Given $n \times n$ matrix $A : \Lambda \to \Lambda$, let C be the Voronoi domain of the cubic lattice (the closure of C is a unit cube centered at the origin) and let $D = \Lambda \cap AC$. If all singular values of A are greater than $3\sqrt{n}$, then D induces a replicating tessellation of Λ. In the one- and two-dimensional cases, the bound $3\sqrt{n}$ can be improved to 2.*

Proof. Let Λ_0 be the cubic lattice so that B is the identity matrix. Then $R/r = \sqrt{n}$, $Q = A^{-1}$, $QAQ^{-1} = A$, and $D = \Lambda \cap Q^{-1}\mathbf{V}_{\Lambda_0} = \Lambda \cap AC$. The corollary now follows directly from Theorem 6. □

Corollary 5 can be applied directly to the square lattice in $\mathbb{R}^2$ to obtain the following result concerning radix representation in the Gaussian integers. If β is a Gaussian integer, not equal to 2 or $1 \pm i$, then there exists a fundamental domain D such that every Gaussian integer has a unique radix representation of the form $\sum_{i=0}^{m} s_i \beta^i$, where $s_i \in D$. Here D is a square Voronoi region centered at the origin.

A reasonable choice for Λ_0 in Theorem 6, besides the cubic lattice used in Corollary 5, is one having a small ratio R/r. One such lattice in all dimensions n is A_n^*, the dual to the root lattice A_n (generated by the roots of certain Lie algebra). A basis for A_n^* in $\mathbb{R}^n$ is any n of the $n+1$ vertices of an n-simplex centered at the origin. A particular choice for these $n+1$ vertices $b_0, b_1, \ldots, b_n$ is

$$b_i = \left(-\frac{c_0}{n}, -\frac{c_1}{n-1}, \ldots, -\frac{c_{i-1}}{n-i+1}, c_i, 0, \ldots, 0\right),$$

where $c_i = (((n-i)(n+1))/(n-i+1)n)^{\frac{1}{2}}$. Note that the b_i are unit vectors. Let $\mathbf{V}_n$ denote the Voronoi domain of the lattice A_n^*. The closure of $\mathbf{V}_2$ and $\mathbf{V}_3$ are a regular hexagon and truncated octahedron, respectively. In general, the n-dimensional Voronoi region is an n-dimensional permutahedron, congruent to a polytope with $(n+1)!$ vertices obtained by taking all permutations of the coordinates of the point $(-n/2, (-n+2)/2, (-n+4)/2, \ldots, (n-2)/2, n/2)$ in $\mathbb{R}^{n+1}$ [2]. It is known [2] that the packing radius of this lattice is $\frac{1}{2}$ and the covering radius is $\frac{1}{2}\sqrt{(n+2)/3}$. So, in applying Theorem 6, take

$$D = \Lambda \cap AB^{-1}(\mathbf{V}_n),$$
$$QAQ^{-1} = BAB^{-1},$$
$$3R/r = \sqrt{3(n+2)},$$

where B is the matrix whose columns are the basis vectors b_i.

Example. The matrix

$$A = \begin{pmatrix} 4 & -1 \\ -1 & 6 \end{pmatrix}$$

discussed earlier in this section satisfies the hypotheses of Corollary 5 and also the hypotheses of Theorem 6 when $\Lambda_0 = A_n^*$. Applying each of these results, two fundamental domains D_1 and D_2 are obtained, each of which induces a rep-tiling of $\mathbb{Z}^2$. These fundamental domains are indicated by circled dots in Figs. 6(b) and 6(c). Note that all three fundamental domains in Fig. 6 are slightly different.

In dimensions 1 and 2, Theorem 6 is best possible in the following sense. Consider the matrix $2I$ whose unique singular value is exactly 2. According to the remarks following Corollary 1 of Theorem 3, this matrix admits no fundamental domain S such that S induces a rep-tiling of $\mathbb{Z}^n$.

7. An algebraic construction. In this section, tessellations with a ring structure are constructed, allowing for multiplication, as well as addition, of lattice points. This construction generalizes radix representation, where the base is an algebraic integer, and the hexagonal tessellation used in image processing.

To construct the lattice, consider a monic polynomial $f(x) = x^n - a_{n-1}x^{n-1} + \ldots - a_0 \in \mathbb{Z}[x]$. In the quotient ring $\Lambda_f = \mathbb{Z}[x]/(f)$, let $\alpha = x + (f)$. Then Λ_f has the structure of a free abelian group Λ_f with basis $(1, \alpha, \alpha^2, \ldots, \alpha^{n-1})$. Λ_f can be realized (in many ways) as a lattice in $\mathbb{R}^n$ by embedding the n basis elements as n linearly independent vectors in $\mathbb{R}^n$. For example, the basis vectors can be identified with the standard unit vectors along the coordinate axes of $\mathbb{R}^n$. According to Proposition 3, questions about aggregate tessellation are independent of how Λ_f is realized. Now Λ_f is the basic lattice of our construction. Addition and multiplication of lattice points is just addition and multiplication in the ring $\Lambda_f = \mathbb{Z}[x]/(f)$.

In the special case that $f(x)$ is irreducible over $\mathbb{Z}$, then, as rings, $\Lambda_f = \mathbb{Z}[x]/(f) \cong \mathbb{Z}[\alpha]$, where α is any root of $f(x)$ in an approprpriate extension field of the rationals. For example, if $f(x) = x^2 + 1$, then the lattice Λ_f is the ring of Gaussian integers $\mathbb{Z}[i]$ with basis $(1, i)$ and can be realized as the square lattice in the complex plane. If $f(x) = x^2 + x + 1$, then the lattice Λ_f can be realized as the hexagonal lattice in the complex plane with basis $(1, -\frac{1}{2} + \sqrt{3}/2i)$. More generally, if $f(x)$ is any monic quadratic with complex roots $\alpha, \overline{\alpha}$, then $\Lambda_f = Z[\alpha] = \{a + b\alpha \ : \ a, b \in \mathbb{Z}\}$ can be considered a lattice in the complex plane. In this case, the addition and multiplication in the lattice $Z[\alpha]$ is the ordinary addition and multiplication of complex numbers.

To obtain a replicating tessellation, let β be any element of the lattice Λ_f and define the linear transformation

$$A_\beta : \Lambda \to \Lambda$$

by

$$A_\beta(x) = \beta x.$$

If S is a finite set of lattice points, then the address $s_0 s_1 \dots s_m$ denotes the lattice point

$$\sum_{i=0}^{m} s_i \beta^i = \sum_{i=0}^{m} A_\beta^i s_i,$$

where $s_i \in S$. In other words, (A_β, S) is a rep-tiling pair for Λ_f if and only if each element of Λ_f has a unique radix representation base β with coefficients in S. Proposition 2 applies directly to this situation.

COROLLARY 6. *If every element of Λ_f has a unique base β finite address with coefficients in S, then S is a complete set of residues of Λ_f modulo $\beta\Lambda_f$ and $|S|$ is the absolute value of the constant term in the characteristic polynomial of A_β.*

Proof. $|S| = |\det(A_\beta)| = |$ is the constant term in the characteristic polynomial of $A_\beta|$. □

Consider two special cases of the above construction.

Radix representation of algebraic numbers. Let β be an algebraic integer and S a finite set of elements in $\mathbb{Z}[\beta]$. The relevant question is: Does every element of $\mathbb{Z}[\beta]$ have a unique radix representation $\sum_{i=0}^{m} s_i \beta^i$, where $s_i \in S$? If $f(x) = x^n + a_{n-1}x^{n-1} + \dots + a_0 \in \mathbb{Z}[x]$ is the minimal monic polynomial for β, then $\Lambda_f = \mathbb{Z}[\beta]$ and with respect to the basis $(1, \beta, \dots, \beta^{n-1})$

$$A_\beta = \begin{pmatrix} 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & \cdots & 0 & -a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -a_{n-1} \end{pmatrix}$$

acts on the cubic lattice $\mathbb{Z}^n$. Now every element of $\mathbb{Z}[\beta]$ has a unique radix representation base β if and only if S induces a rep-tiling of $\mathbb{Z}^n$. By part 1 of Corollary 6 the cardinality of a fundamental domain S is $|N(\beta)|$, where $N(\beta) = (-1)^n a_0$. $N(\beta)$ is the *norm* of β and can alternatively be defined as the product of all conjugates of β.

Gilbert [8] asks about the case where $S = \{0, 1, \dots, N(\beta) - 1\}$. Consider the example $\beta = -1 + i$ and $S = \{0, 1\}$; then $\mathbb{Z}[\beta] = \mathbb{Z}[i]$, and this is exactly Example 2 of the Introduction concerning the Gaussian integers. Corollary 2 of Theorem 4 applies to this situation. Multiplication by the complex number β is a similarity (the composition of a $\pi/4$ rotation and a stretching by a factor of $\sqrt{2}$), and hence, to determine whether or not every element of $\mathbb{Z}[\beta]$ has a base β finite address, it is sufficient to check that each element in the ball of radius $\max\{|s| : s \in S\}/|\beta| - 1$ in the complex plane has a finite address. There are exactly 21 Gaussian integers within a ball of radius $1/\sqrt{2} - 1$. Testing with Algorithm A shows that all 21 have finite addresses (for example, $-1 = 10111$ and $-2 - i = 110010111$). Therefore, every Gaussian integer has a unique base β finite address. For β satisfying a quadratic polynomial $x^2 + cx + d$, Gilbert states [8] that every element of $\mathbb{Z}[\beta]$ has a unique radix representation with coefficients in $S = \{0, 1, \dots, |d| - 1\}$ if and only if $d \geq 2$ and $-1 \leq c \leq d$.

Generalized balanced ternary. The following example simultaneously generalizes the balanced ternary representation of integers in the first example of the Introduction and the two-dimensional hexagonal tessellation of the third example. Let $f(x) = x^n + x^{n-1} + \cdots + 1$ and denote by $\Lambda_n = \mathbb{Z}[x]/(f)$ the corresponding n-dimensional lattice. As previously mentioned, Λ_1 and Λ_2 can be realized as the integer and hexagon lattices in dimensions 1 and 2, respectively. Let $\omega = x + (f)$ denote the image of x in the quotient ring Λ_n and note that $\omega^{n+1} = 1$ in Λ_n. Let $\beta = 2 - \omega$. With respect to the basis $(1, \omega, \omega^2, \dots, \omega^{n-1})$

$$A_\beta = \begin{pmatrix} 2 & 0 & 0 & \cdots & 0 & 1 \\ -1 & 2 & 0 & \cdots & 0 & 1 \\ 0 & -1 & 2 & \cdots & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & 1 \\ 0 & 0 & 0 & \cdots & -1 & 3 \end{pmatrix}.$$

Define $S_n = \{\epsilon_0 + \epsilon_1\omega + \epsilon_2\omega^2 + \cdots + \epsilon_n\omega^n : \epsilon_i \in \{0, 1\}\}$. Note that $|S_n| = 2^{n+1} - 1$ because $1 + \omega + \cdots + \omega^n = 0$ and also $det(A_\beta) = 2^{n+1} - 1$. Therefore, S_n has the appropriate number of elements to serve as a fundamental domain for A_β. For $n = 1$, we have

$$\begin{aligned} \Lambda_1 &= \mathbb{Z}, \\ \beta &= 3, \\ S &= \{-1, 0, 1\}, \\ A_\beta &= (3), \end{aligned}$$

which leads to the balanced ternary representation of the integers. For $n = 2$, with respect to the standard basis, we have

$$\begin{aligned} \Lambda = {} & \text{the hexagonal lattice}, \\ \beta = {} & \tfrac{5}{2} - \tfrac{\sqrt{3}}{2} i, \end{aligned}$$

$$S = \{0, 1, \omega, \dots, \omega^5 \ : \ \omega \text{ is a 6th root of unity}\},$$
$$A_\beta = \begin{pmatrix} \frac{5}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{5}{2} \end{pmatrix},$$

which leads to the hexagonal rep-tiling of the third example in the Introduction.

COROLLARY 7. *The generalized balanced ternary pair* (A_β, S_n) *yields a rep-tiling of* Λ_n.

The proof of Corollary 7 will follow from some properties of the addition and multiplication of addresses in the generalized balanced ternary. An element $s = \epsilon_0 + \epsilon_1\omega + \epsilon_2\omega^2 + \cdots + \epsilon_n\omega^n \in S_n$ can be encoded by a corresponding binary string $b_s = \epsilon_n\epsilon_{n-1}\dots\epsilon_0$. Note that, as in ordinary integer notation, the order of the digits is reversed. In the generalized balanced ternary, addition and multiplication can be carried out by simple and fast bit string routines. Define three operations on such binary strings as follows. First, $b \oplus b'$ is circular base 2 addition; a carry from the ith column goes to the $(i+1)$st column $\operatorname{mod}(n+1)$. Note that the column numbers increase to the left. This first operation is equivalent to ordinary addition $\operatorname{mod}(2^{n+1}-1)$. For example, $1011 \oplus 1110 = 1010$. Second, $b \boxplus b'$ is base 2 addition with no carries. For example, $1011 \boxplus 1010 = 0001$. Third, $T(s)$ is the shift one position to the right $\operatorname{mod}(n+1)$. For example, $T(1011) = 1101$. Using the facts that $\omega^{n+1} = 1$ and $2 = \omega + \beta$ it can be routinely checked that if $s, s' \in S_n$, then in Λ_n we have

$$s + s' = s_0 + s_1\beta,$$

where

$$b_{s_0} = b_s \oplus b_{s'},$$
$$b_{s_1} = T[b_s \boxplus b_{s'} \boxplus (b_s \oplus b_{s'})].$$

(The latter expression for b_{s_1} yields a 1 or 0 at those positions where a carry in $b_s \oplus b_{s'}$ has or has not, respectively, occurred.) Addition of addresses is accomplished by using the carry rule above (sum $= s_0$; carry $= s_1$). Addition corresponds to vector addition in $\mathbb{R}^n$. Multiplication also uses the rule for addition and $0 \cdot 0 = 0 \cdot 1 = 1 \cdot 0 = 0$ and $1 \cdot 1 = 1$. For example with $n = 2$, let $x = (110) + (010)\beta$ and $y = (101) + (110)\beta$. Then $x + y = (100) + (001)\beta + (110)\beta^2$ and $xy = (010) + (100)\beta + (001)\beta^2$. (We have used the fact that $111 = 000$.)

Proof of Corollary 7. Note (1) S contains a basis $1, \omega, \dots, \omega^{n-1}$ for the lattice Λ_n. Also, the comments above concerning bit string operations imply that and (2) $S \pm S \subseteq S + \beta S$. The corollary then follows immediately from Theorem 4. □

Remark. The eigenvalues of A_β for the generalized balanced ternary are $\{2 - \omega \ : \ \omega \text{ is an (n+1)st root of unity}, \omega \neq 1\}$. Therefore, as $n \to \infty$, the minimum modulus of an eigenvalue tends to 1, but (A_β, S_n) is a rep-tiling pair for all n. This gives the example mentioned at the end of §3.

Appealing geometric properties of the generalized balanced ternary tessellation can be obtained by embedding the generator vectors $1, \omega, \dots, \omega^n$ for the lattice Λ_n at the points $b_0, \dots b_n$ that generate the dual root lattice A_n^* as descibed in the previous section. Then the Voronoi regions in dimensions 2 and 3, as previously mentioned, are regular hexagons and truncated octahedra, respectively.

Acknowledgments. The author thanks both referees for their valuable suggestions, in particular, for drawing our attention to recent references and for the proof of the last part of Proposition 4.

REFERENCES

[1] C. Bandt, *Self-similar sets 5. Integer matrices and and fractal tilings of* $\mathbb{R}^n$, Proc. Amer. Math Soc., 112 (1991), pp. 549–562.

[2] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, Grundlehren der mathematischen Wissenschaften 290, Springer-Verlag, New York, 1988.

[3] K. Culik II and A. Salomaas, *Ambiguity and descision problems concerning number systems*, Inform. and Control, 56 (1983), pp. 139–153.

[4] F. M. Dekking, *Recurrent sets*, Adv. Math., 44 (1982), pp. 78–104.

[5] ———, *Replicating superfigures and endomorphisms of free groups*, J. Combin. Theory Ser. A, 32 (1982), pp. 315–320.

[6] L. Gibson and D. Lucas, *Spatial data processing using generalized balanced ternary*, in Proceedings of the IEEE Computer Society Conference on Pattern Recognition and Image Processing, IEEE Computer Society, 1982, pp. 566–571.

[7] W. J. Gilbert, *Fractal geometry derived from complex bases*, Math. Intelligencer, 4 (1982), pp. 78–86.

[8] ———, *Geometry of radix representations*, The Geometric Vein: The Coxeter festscrift (1981), pp. 129–139.

[9] ———, *Radix representations of quadratic fields*, J. Math. Anal. Appl., 83 (1981), pp. 264–274.

[10] J. Giles, *Construction of replicating superfigures*, J. Combin. Theory Ser. A, 26 (1979), pp. 328–334.

[11] S. W. Golomb, *Replicating figures in the plane*, Math. Gaz., 48 (1964), pp. 403–412.

[12] K. Gröchenig and W. R. Madyeh, *Multiresolution analysis, Haar bases and self-similar tilings of* $\mathbb{R}^n$, IEEE Trans. Inform. Theory, 38 (1992), pp. 556–568.

[13] B. Grunbaum and G. C. Shephard, *Tilings and Patterns*, W. H. Freeman, New York, 1987.

[14] J. Honkala, *Bases and ambiguity of number systems*, Theoret. Comput. Sci., 31 (1984), pp. 61–71.

[15] T. W. Hungerford, *Algebra*, Springer-Verlag, New York, 1987.

[16] I. Katai and J. Szabo, *Canonical number systems for complex integers*, Acta Sci. Math. (Szeged), 37 (1975), pp. 255–260.

[17] R. Kenyon, *Self-replicating tilings*, preprint.

[18] W. Kitto, A. Vince, and D. Wilson, *An isomorphism between the p-adic integers and a ring associated with a tiling of n-space by permutohedra*, Discrete Appl. Math., to appear.

[19] L. Kronecher, *Zwei Sätze üper Gleichungen mit ganzzahlingen Koeffizienten*, J. Reine Angew., 53 (1857), pp. 173–175.

[20] D. E. Knuth, *The Art of Computer Programming*, Vol. 2, Seminumerical Algorithms, Addison-Wesley, Reading, MA, 1981.

[21] D. W. Matula, *Basic digit sets for radix representations*, J. Assoc. Comput. Mach., 4 (1982), pp. 1131–1143.

[22] A. M. Odlyzko, *Non-negative digit sets in positional number systems*, Proc. London Math. Soc., 37 (1978), pp. 213–229.

[23] M. Petekovsek, *Ambiguous numbers are dense*, Amer. Math. Monthly, 97 (1990), pp. 408–411.

[24] Polya and Szego, *Problems and Theorems in Analysis*, Springer-Verlag, New York, 1976.

[25] J. W. van Roessel, *Conversion of Cartesian coordinates from and to generalized balanced ternary addresses*, Photogrammetric Eng. and Remote Sensing, 54 (1988), pp. 1565–1570.

SIAM J. DISC. MATH.
Vol. 6, No. 4, pp. 523-532, November 1993

FAST PARALLEL RECOGNITION OF ULTRAMETRICS AND TREE METRICS*

ELIAS DAHLHAUS†

Abstract. A fast parallel algorithm for the recognition of ultrametrics is presented. Its time-processor product is of the same order as the time bound of the known sequential algorithm of Culberson and Rudnicki [*Inform. Process. Lett.*, 30 (1990), pp. 215-220] (compare also [*SIAM J. Disc. Math.*, 3 (1990), pp. 1-6] and [*Quart. Appl. Math.*, 26 (1968), pp. 607-609]. By the same way, tree metrics also can be recognized.

Key words. ultrametric, tree metric, parallel algorithms

Introduction. An *ultrametric* is a metric d on a set Ω of objects, where, for all $i, j, k \in \Omega$, the following extension of the triangle inequality is valid:

$$d(i,j) \leqq \max \{d(i,k), d(j,k)\}.$$

A *tree metric* is the distance function on a tree restricted to a subset of the vertices of the tree. This is equivalent to the statement that, for all i, j, k, l of the domain of the metric, the following inequality is valid [6]:

$$d(i,j) + d(k,l) \leqq \max \{d(i,k) + d(j,l), d(i,l) + d(j,k)\}.$$

Note that from this inequality the triangle inequality and the symmetry follows.

Tree metrics are highly interesting in the view of evolutionary problems—for example, in biology and archeology. There is an extensive literature on the topic of tree metrics [3], [5], [6], [9], [15], [16], [18], [19]. Each ultrametric is also a tree metric. Ultrametrics have their use in hierarchical classifications (see [12]).

Here we present a fast parallel algorithm to recognize ultrametrics and tree metrics. The time bound is logarithmic, and the processor bound is $O(n^2/\log n)$, where n is the number of objects. The main part is the parallel recognition of ultrametrics. To prove the same time and processor bound of the recognition of tree metrics, we consider the reduction of a tree metric to an ultrametric, as in [2]. We see that this reduction can be done in constant time and a processor bound of n^2, and therefore also in logarithmic time and a processor bound of $n^2/\log n$.

The time-processor product of $O(n^2)$ is optimal. Sequential algorithms of a time bound of $O(n^2)$ for the recognition of tree metrics are known (see [4], [8]). Ultrametrics can be recognized in sequential $O(n^2 \log n)$ time [2].

Section 2 introduces the fundamental notation and concepts. Section 3 presents the announced parallel algorithm to recognize an ultrametric. Section 4 discusses the parallel recognition of tree metrics and also the construction of the corresponding tree.

1. Notation and fundamental definitions.

1.1. Notions from metrics.

A *distance function* is a binary, symmetric, positively real-valued function d on a domain Ω. Moreover, we assume that, for $x \in \Omega$, the equation $d(x, x) = 0$ is valid. A *metric* is a distance function satisfying the triangle inequality.

Here we assume that Ω is a finite domain. Moreover, we let Ω be a set of the form $\{1, \ldots, n\}$. The distance function d is implemented as an $n \times n$ matrix.

* Received by the editors August 27, 1990; accepted for publication (in revised form) September 11, 1992.

† Department of Computer Science, University of Bonn, Römerstrasse 164, D-5300 Bonn 1, Germany. Present address, Basser Department of Computer Science, University of Sydney, NSW 2006, Australia.

A distance function d is called an *ultrametric* if and only if the following *extended triangle inequality* is valid:

$$d(i,j) \leqq \max\{d(i,k), d(j,k)\}.$$

To introduce a tree metric, we first introduce the notion of a *tree*. By a tree T, we mean a cycle-free connected graph consisting of a vertex set V_T and an edge set E_T. By a *rooted* tree, we mean a directed graph, whose underlying undirected graph is a tree, with the additional property that there is a vertex r, called the *root*, such that each vertex x has a directed path to r. For a rooted tree $T = (V_T, E_T)$ with root r and each $x \in V_T \setminus \{r\}$, the *parent* Par (x) of x is the y, such that $(x, y) \in E_T$. x is also called a *child* of y. Vertices without children are called *leaves*. $y \in V_T$ is called an *ancestor* of $x \in V_T$ if and only if there is a directed path (possibly of length 0) from x to y in T. x is also called a *descendent* of y if y is an ancestor of x. The set of descendents of t in T including t is denoted by T_t. We identify T_t and its induced subtree. For $x, y \in V_T$, the *least common ancestor* of x and y, denoted by LCA (x, y), is the common ancestor z of x and y, such that no child of z is an ancestor of x and y. For a distance function d on Ω, a tree T' with vertex set Ω is called a *minimum spanning tree* for d if and only if T' is a tree with vertex set Ω such that $\sum_{xy \text{ is an edge of } T'} d(x, y)$ is minimal. A distance function d with domain Ω is called a *tree metric* if and only if there is a tree T and a labeling l of the edges of T with positive real numbers, such that

1. Ω is a subset of the vertex set V_T of T, and
2. For each i and $j \in \Omega$, the distance $d(i, j)$ is the sum $\sum_{k_1k_2 \in P(i,j)} l(k_1k_2)$ of the labelings of the edges on the unique path $P(i,j)$ from i to j in T.

Tree metrics are exactly those metrics that satisfy the following *four-point inequality* [6]:

$$d(i,j) + d(k,l) \leqq \max\{d(i,k) + d(j,l), d(i,l) + d(j,k)\}.$$

We continue with a tree characterization of ultrametrics. A *dendrogram* is a rooted tree T together with a positively real-valued labeling h of the vertices with a *height function*, which means $h(v) < h(w)$ if w is an ancestor of v.

PROPOSITION (see, for example, [12]). *A distance function d on Ω is an ultrametric if and only if there is a dendrogram (T, h) such that*

1. *Ω is the set of leaves of T, and*
2. *For all $u, v \in \Omega$, the distance $d(u, v)$ is the labeling $h(\text{LCA}(u, v))$ of the least common ancestor $\text{LCA}(u, v)$ of u and v with respect to T.*

1.2. Notions from complexity theory. The computation model is the concurrent read exclusive write parallel random access machine (CREW-PRAM) [11]. Since we only compare numbers and add at most three numbers of the given matrix, we may measure arithmetic operations by one time and one processor unit. The logarithmic cost of the processor number would grow only by the factor of the length of the largest number, and the logarithmic cost of the time would grow only by a factor of the logarithm of the length of the maximal appearing number of the input.

2. Parallel recognition of ultrametrics.

The main result of this section is the following theorem.

THEOREM 1. *Ultrametrics can be recognized in $O(\log n)$ parallel time using $O(n^2/\log n)$ processors. Moreover, its corresponding dendrogram can also be computed in a processor bound of $O(n^2/\log n)$ and a time bound of $O(\log n)$.*

Proof. Let d be a distance function on the domain $\Omega = \{1, \ldots, n\}$. The problem is to check whether d is an ultrametric. For the case where d is an ultrametric, a dendrogram (T, h) for d will be constructed.

Our strategy is as follows. We construct a tree T', which is a minimum spanning tree in the case where d is an ultrametric. We see that this tree T' can be constructed in logarithmic time and an optimal processor time product. We use this tree T' for an efficient parallel ultrametric recognition algorithm. For the case where d is an ultrametric, this tree T' has additional properties that be used to construct the dendrogram (T, h).

T' is the tree defined by the following parent function P (that means T' consists of the edges $(t, P(t))$ with $t \in \Omega \setminus \{n\}$), which is constructed by the following algorithm:

1. **For each** $x \in \Omega$, **let** $H(x) = \{y > x : d(x, y)$ **is minimal**$\}$;
2. **For each** $x \in \Omega \setminus \{n\}$, **let** $P(x) = \max_{y \in H(x)} y$.

Using Brent's scheduling technique, we can compute $\min_{y>x} d(x, y)$ in a time bound of $\log n$ and a processor bound of $n^2/\log n$. Therefore $H(x)$, $P(x)$, and T' can be constructed in $O(\log n)$ time using $O(n^2/\log n)$ processors.

We note that this algorithm is a simplified version of a minimum spanning tree algorithm for the special case where the distance function is an ultrametric. The progress is that we need only $O(\log n)$ time in the simplified version, while the general minimum spanning tree computation needs $O(\log^2 n)$ time (on a CREW-PRAM) (see, for example, [1], [14]).

To test whether d is an ultrametric, we introduce the following relation $\prec$: We say $x \prec y$ if and only if $d(x, P(x)) \leqq d(y, P(y))$ or $x \neq n$ (and therefore $P(x)$ is defined) and $y = n$ (and therefore $P(y)$ is not defined).

Note that $\prec$ is transitive, and, for each $x \neq y$, $x \prec y$ or $y \prec x$. Note that $\prec$ is not an ordering. For $x \prec y$ such that $P(x) \neq y$, we test whether $d(x, y) = d(P(x), y)$. This can be tested in constant time using $O(n^2)$ processors.

PROPOSITION 1. 1. *The following statements are equivalent*:

(a) *d is an ultrametric*,

(b) *For all $x \prec y$ such that $P(x) \neq y$ and $x \neq y$, the equality $d(x, y) = d(P(x), y)$ is valid.*

2. *If T' is known, then it can be checked in constant time using $O(n^2)$ processors whether statement* 1(b) *is satisfied.*

Proof. The three following lemmas are significant for the proof of both directions of the first part of Proposition 1.

LEMMA 1. *Suppose that d satisfies one of the conditions* 1(a) *or* 1(b) *of Proposition* 1. *Then, for each $x \in \Omega$ for which $P(x)$ and $P(P(x))$ are defined, the inequality $d(x, P(x)) < d(P(x), P(P(x)))$ is valid.*

Proof. Suppose that 1(a) is satisfied (that means d is an ultrametric). Clearly, $x < P(x) < P(P(x))$. Since $P(x)$ is the greatest $y \in H(x)$ and therefore the greatest y with $d(x, y) = d(x, P(x))$, $d(x, P(P(x))) > d(x, P(x))$. Since d is an ultrametric,

$$d(x, P(P(x))) \leqq \max \{d(x, P(x)), d(P(x), P(P(x)))$$

and therefore

$$d(x, P(x)) < d(x, P(P(x))) \leqq d(P(x), P(P(x))).$$

Suppose that d satisfies 1(b). Assume that $d(x, P(x)) \geqq d(P(x), P(P(x)))$. Then $P(x) \prec x$. By 1(b), $d(x, P(x)) = d(x, P(P(x)))$. Since $P(x) < P(P(x))$, $P(x)$ is not the $<$-largest y such that $d(x, y) = d(x, P(x))$. This is a contradiction to the definition of $P(x)$ as the largest $y > x$ with minimal distance. □

LEMMA 2. *If $x \prec y$ and $x \neq y$, then $d(x, P(x)) \leqq d(x, y)$.*

Proof. Suppose that $x < y$. Then $d(x, P(x)) \leqq d(x, y)$ because $P(x)$ is some $z > x$ such that $d(x, z)$ is minimal.

Suppose that $y < x$. $d(x, P(x)) \leqq d(y, P(y))$ because $x \prec y$. The inequality $d(y, P(y)) \leqq d(x, y)$ because $P(y)$ is a $z > y$ such that $d(y, z)$ is minimal. Therefore $d(x, P(x) \leqq d(x, y)$. □

LEMMA 3 (see [2]). *Suppose that $(x, y) \in \Omega$ is a pair such that $d(x, y)$ is minimal. Then d is an ultrametric if and only if for all $v \in \Omega \setminus \{x, y\}$, $d(x, v) = d(y, v)$ and d restricted to $\Omega \setminus \{x\}$ is an ultrametric.*

For the proof of both directions of the first part of Proposition 1, we consider any enumeration $(x_1, \ldots, x_n)$ of Ω such that, for $i = 1, \ldots, n-1$, $x_i \prec x_{i+1}$.

Define $\Omega_i = \{x_i, \ldots, x_n\}$.

In the following corollaries, we assume that 1(a) *or* 1(b) *of Proposition* 1 *are satisfied.* Then, by Lemma 1, we have the following corollary.

COROLLARY 1. *For $y \in \Omega_i \setminus \{n\}$, $P(y) \in \Omega_i$.*

Since $x_i \in \Omega_i$, also $P(x_i) \in \Omega_i$. Since x_i and $P(x_i)$ are not equal, $P(x_i) \in \Omega_{i+1}$. Therefore, by Lemma 3, we have the following corollary.

COROLLARY 2. *d restricted to Ω_i is an ultrametric if and only if d restricted to Ω_{i+1} is an ultrametric and, for all $y \in \Omega_{i+1} \setminus \{P(x_i)\}$, $d(x_i, y) = d(P(x_i, y))$.*

We continue with the proof of Proposition 1.

Suppose that d satisfies 1(b). We show that d restricted to Ω_i is an ultrametric, for every $i = 1, \ldots, n$, by backward induction on i. Clearly, d restricted to Ω_n is an ultrametric. We assume now that d restricted to Ω_{i+1} is an ultrametric. By the assumption that 1(b) is true, for each $x_i \neq x_n$ and each y with $x_i \prec y$ with $y \notin \{x_i, P(x_i)\}$ and therefore for each $y \in \Omega_{i+1} \setminus \{P(x_i)\}$, we have $d(x_i, y) = d(P(x_i), y)$. By Corollary 2, d restricted to Ω_i is an ultrametric. Since d is nothing else than d restricted to $\Omega = \Omega_1$, d is an ultrametric, and therefore 1(a) is satisfied.

Suppose that d satisfies 1(a). Let x and y be any elements of Ω such that $x \prec y$, $x \neq y$ and $P(x) \neq y$. To prove 1(b), we must prove $d(x, y) = d(P(x), y)$. We find an enumeration $(x_1, \ldots, x_n)$ such that $x_i \prec x_{i+1}$, for $i = 1, \ldots, n-1$ with the additional property that x appears before y. Let $x = x_i$ and $y = x_j$. Clearly, $y \in \Omega_{i+1}$. Since 1(a) is satisfied, d is an ultrametric, and therefore also d restricted to Ω_i is an ultrametric. Therefore, by Corollary 2, $d(x, y) = d(x_i, y) = d(P(x_i), y) = d(P(x), y)$.

Herewith, the first part of Proposition 1 has been proved. The second part of Proposition 1 is obvious. Therefore, Proposition 1 is proved. □

COROLLARY. *Ultrametrics can be recognized in $O(\log n)$* CREW*-time using $O(n^2/\log n)$ processors.*

It remains to construct the dendrogram for a given ultrametric d. We use the tree T'.

PROPOSITION 2. *Suppose that d is an ultrametric. Then T' is, in fact, a minimum spanning tree.*

Proof. Let Ω_i be defined as in the proof of Proposition 1. We know that T' restricted to Ω_i is still a tree. Moreover, we know that $(x_i, P(x_i))$ is a pair of elements in Ω_i with minimal distance. By backward induction on i, we show that T' restricted to Ω_i is a minimum spanning tree for d restricted to Ω_i.

Suppose that $i = n$. Then we are done.

Assume that T' restricted to Ω_{i+1} is a minimum spanning tree for d restricted to Ω_{i+1}. Suppose that T'' is a minimum spanning tree for d restricted to Ω_i. Our aim is to show that the distance sum of the edges of T' restricted to Ω_i is at most the distance sum of the edges of T''. First, we construct a tree T''' that has $x_i P(x_i)$ as an edge. Let $x_i z$ be that edge of T'' incident with x_i, which is on the unique path from x_i to $P(x_i)$ in T''. T''' arises from T'' by replacing $x_i z$ by $x_i P(x_i)$. Clearly, T''' is a tree and $d(x_i P(x_i)) \leqq d(x_i, z)$. Therefore the sum of distances of the edges of T''' is bounded by the sum of distances of the edges of T''. Now we can replace all edges $x_i y$ with $y \neq P(x_i)$ by $P(x_i y)$, and the resulting graph $T^{(4)}$ remains a tree. Moreover, $d(x_i, y) = d(P(x_i), y)$, because d is an ultrametric. Therefore the sum of distances of the edges of $T^{(4)}$ is the same as the sum of distances of T'''. $T^{(4)}$ consists of an edge $x_i P(x_i)$ and a spanning tree of Ω_{i+1}. That means $x_i P(x_i)$ and T' restricted to Ω_{i+1} form a spanning tree with the property that the sum of distances of edges is at most the sum of distances of edges of T''. That means T' restricted to Ω_i has a sum of edge distances that is bounded by the sum of edge distances of T''. Therefore, since we assume that T'' is a minimum spanning tree for d restricted to Ω_i, T' restricted to Ω_i is a minimum spanning tree for d restricted to Ω_i.

Since for each i, T' restricted to Ω_i is a minimum spanning tree for d restricted to Ω_i, T' is a minimum spanning tree for d. □

Next, we show that each ultrametric can be reconstructed from any minimum spanning tree.

LEMMA 4. *Suppose that d is an ultrametric on Ω and T is a minimum spanning tree for d. Then, for any pair $x, y \in \Omega$, $d(x, y)$ is the maximum distance $d(x', y')$ of an edge $x'y'$ on the unique path from x to y in T.*

Proof. Suppose that $(x = x_1, \ldots, x_k = y)$ is the unique path from x to y in T. Then, by induction on j, $d(x, x_j) \leqq \max\{d(x_i, x_{i+1}) | i = 1, \ldots, j-1\}$. Therefore $d(x, y) \leqq \max\{d(x_i, x_{i+1}) | i = 1, \ldots, k-1\}$.

Suppose that $d(x, y) < \max\{d(x_i, x_{i+1}) | i = 1, \ldots, k-1\}$. Then we consider tree T_1, which arises from T by erasing the edge $x_i x_{i+1}$ with the distance $\max\{d(x_i, x_{i+1}) | i = 1, \ldots, k\}$ and adding the edge xy. Clearly, this tree has a smaller sum of distances of the edges. This is a contradiction to the assumption that T is a minimum spanning tree. □

The next statement explains how d can be reconstructed from d restricted to the edge set of T'.

LEMMA 5. 1. *Let y be an ancestor of x in T' and $y \neq x$. Let x' be the child of y, which is an ancestor of x in T'. Then $d(x, y) = d(x', y)$.*

2. *Let $x, y \in \Omega$ and suppose that x is not an ancestor or descendent of y in T'. Let z be the least common ancestor of x and y in T', let x' be the child of z on the path from x to z, and let y' be the child of z, which is an ancestor of y in T'.*

Then $d(x, y) = \max\{d(x', z), d(y', z)\}$.

Proof. Suppose that $(x = x_1, \ldots, x_k = z)$ is the unique path from x to the least common ancestor z of x and y in T' and that $(y = y_1, \ldots, y_l = z)$ is the unique path from y to z in T'. Note that $x_{i+1} = P(x_i)$ and $y_{i+1} = P(y_i)$. Therefore, for $i = 1, \ldots, k-2$, $d(x_i, x_{i+1}) < d(x_{i+1}, x_{i+2})$. By the same reason, $d(y_i, y_{i+1}) < d(y_{i+1}, y_{i+2})$ for $i = 1, \ldots, l-2$. Therefore $x_{k-1}x_k$ is an edge on the unique path from $x = x_1$ to $z = x_k$ in T' of maximal distance, and $y_{l-1}y_l$ is an edge on the unique path from $y = y_1$ to $z = y_l$ of maximal distance.

If y is an ancestor of x, then $z = y$ and $x_{k-1}x_k = x'y$ is the edge on the unique path from x to y with maximal distance. By Lemma 4, this $d(x, y)$ and $d(x', y) = d(x_{k-1}, x_k)$ coincide, and statement 1 of the lemma has been proved.

To prove statement 2, we need only observe that one of the edges $x_{k-1}x_k$ and $y_{l-1}y_l$ is an edge uv on the unique path from x to y in T' such that $d(u, v)$ is maximal. Using Lemma 4, the second statement of the lemma follows immediately. □

Now we are able to construct the dendrogram $D = (T, h)$ of (Ω, d). We compute it by the following algorithm:

1. For each $x \in \Omega$, let $C(x)$ be the set of children of x with respect to T', say

$$C(x) = \{y : P(y) = x\}.$$

2. **For each $x \in \Omega$, sort $C(x)$ by $d(y, x)$;**
   for $y_1, y_2 \in C(x)$, let $y_1 \equiv y_2$ if and only if $d(y_1, x) = d(y_2, x)$, (or set $y_1 \equiv y_2$ if and only if $P(y_1) = P(y_2)$ and $d(y_1, P(y_1)) = d(y_2, P(y_2))$);
   for each $x \in \Omega$, let $[x]$ be the $\equiv$-equivalence class to which x belongs.
3. **The vertex set of T consists of all $\equiv$-equivalence classes and all elements of Ω:**

$$V_T = \Omega \cup \{[x] : x \in \Omega \textbf{ and } P(x) \textbf{ is defined}\}.$$

4. **For each $u \in \Omega$, the height $h(u)$ of u is 0;**
   for each $\equiv$-equivalence class $[x]$, the height $h([x])$ is set $d(x, P(x))$.
5. **Directed edge set of T is defined by the following parent function** Par:
   - **If $u \in \Omega$, then** Par (u) **is $[x]$, where x is chosen from the set $\{u\} \cup C(u)$ such that $h([x]) = d(x, P(x))$ is minimum** (We consider all T'-edges, such that u is incident. From these T'-edges, we select one of minimal distance and let Par (u) be the equivalence class to which the source of the selected edge belongs).
   - **If $u = [x]$ and $x \in C(y)$ such that $d(x, y)$ is maximal, then** Par $([x]) = [y]$; **otherwise if $d(x, y)$ is not maximal, let** Par $([x])$ **be the $[z]$ such that $z \in C(y)$ $(P(z) = P(x))$, $d(z, y) > d(x, y)$, and $d(z, y)$ is minimal under these conditions.**

First, we note that the definitions of $h([x])$ and of Par $([x])$ depend only on the equivalence class but not on the special x. This follows directly from the definition of $\equiv$.

We illustrate the behavior of this algorithm by the following example. To do so, we consider the minimum spanning tree in Fig. 1. Then the corresponding dendrogram is as in Fig. 2.

The construction algorithm for Par can be executed in $O(\log n)$ time and $O(n)$ processors: The dominating step is the sorting procedure that can be executed in $O(\log n)$ time and $O(n)$ processors [7]. The equivalence classes $[x]$ are represented by the pair (min $[x]$, max $[x]$), where min $[x]$ is the smallest $v \in [x]$ and max $[x]$ is the largest $v \in [x]$ with respect to the sorting of the sets $C(y)$, which contain x.

Then Par $([x])$ and $h([x])$ can be computed in constant time using n processors, for all x simultaneously. We can set Par $([x]) := [z]$, where z is the immediate successor of max $[x]$ with respect to the sorted list of $C(y)$. If max $[x]$ is maximal in $C(y)$, we set Par $([x]) = [y]$. Obviously, this can be executed in constant time. If $u \in \Omega$, then we can compute Par (u) obviously in constant time. It remains to check that we really computed a dendrogram. For this purpose, we need the following result.

LEMMA 6. *In T the equivalence class $[y]$ with the representative y is an ancestor of $x \in \Omega$ if and only if either $x = P(y)$ or $P(y)$ is an ancestor of x in T' and $d(y, P(y)) \geq d(x', P(y))$, where x' is the unique child of $P(y)$, which is an ancestor of x in T'.*

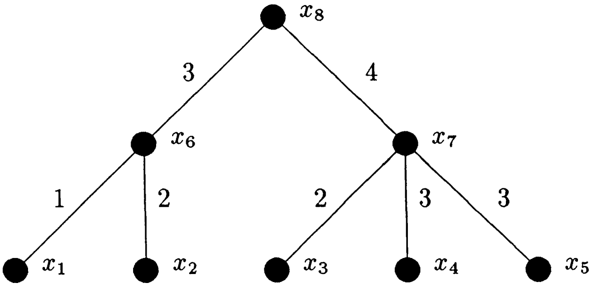


FIG. 1. *The minimum spanning tree T' for an ultrametric d, edges of T are labeled by their distances.*

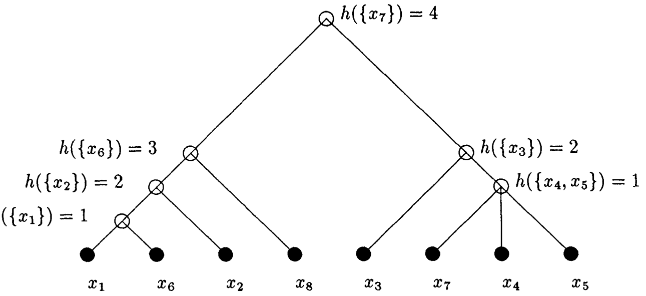


FIG. 2. *The dendrogram* (T, h) *resulting from* T' *and* d.

Proof. $\Rightarrow$: Note that $[y]$ is an ancestor of x in T if and only if $[y]$ arises from x by iterated application of the parent function Par. We prove this direction by induction on the application of Par to x. Since $[y]$ cannot be x, Par must be applied at least once to obtain $[y]$.

We first assume that $[y] = \text{Par }(x)$ *(i.e.,* $[y]$ *arises from* x *by one application of* Par).

First, we assume that $[y]$ is a subset of $C(x)$. Then $x = P(y)$.

If $[y]$ is not a subset of $C(x)$ then x has no children, $[x] = [y]$, and therefore $P(x) = P(y)$. x is the child x' of $P(y) = P(x)$, which is an ancestor of x. Therefore $d(y, P(y)) = d(x, P(y) = d(x', P(y))$, and therefore $d(x', P(y)) \leqq d(y, P(y))$.

Assume that we proved the $\Rightarrow$*-direction for* $[y] = \text{Par}^i\,(x)$.

To prove the induction step, we must prove that $[y'] = \text{Par }([y])$ satisfies the conditions as claimed. There are two cases.

First case. $P(y) = P(y')$.

If $x = P(y)$, then trivially also $x = P(y')$. We may assume that $x \neq P(y)$. Then trivially $P(y')$ is an ancestor of x. Moreover, the unique child of $P(y')$, which is an ancestor of x in T', remains x'. By construction of Par, $d(y, P(y')) < d(y', P(y'))$, and therefore also $d(x', P(y')) < d(y', P(y'))$.

Second case. $P(y) \neq P(y')$.

In that case, $\text{Par }([y]) = [P(y)]$, i.e., $P(y) \equiv y'$. Moreover, $P(y)$ is the unique child x'' of $P(y') = P(P(y))$, which is an ancestor of x in T'. Since $P(y) \equiv y'$, we also have $d(P(y), P(y')) = d(y', P(y'))$. Therefore y' satisfies the conditions as claimed in $\Rightarrow$.

$\Leftarrow$: It is easily checked that, for $P(x) = P(y)$, the equivalence class $[y]$ is an ancestor of $[x]$ in T if and only if $d(x, P(x)) \leqq d(y, P(y))$. From this, we also can follow that $[P(x)]$ is an ancestor of $[x]$ in T, and therefore for each ancestor y of x in T', $[y]$ also is an ancestor of $[x]$ in T.

Suppose that $P(y)$ is an ancestor of x in T'. If $x = P(y)$, then trivially x is an ancestor of $[y]$ in T. Otherwise, let x' be the unique child of $P(y)$, which is an ancestor of x in T'. Then $[x']$ is an ancestor of $[x]$ in T and therefore also an ancestor of x in T. Then, however, $[y]$ also is an ancestor of $[x']$ and therefore of x in T if $d(x', P(y)) \leqq d(y, P(y))$. □

From the last lemma, we can follow immediately with the following one.

LEMMA 7. *Let* x *and* y *be in* Ω. *Let* z *be the least common ancestor of* x *and* y *in* T'. *Let* x' *be the child of* z, *which is an ancestor of* x *in* T' *if* $z \neq x$, *and* y' *be the child of* z, *which is an ancestor of* y *in* T' *if* $z \neq y$. *If* $d(x', z) \leqq d(y', z)$ *or* $x = z$ *and* $y \neq z$, *then the least common ancestor of* x *and* y *in* T *is* $[y']$ *(the* $\equiv$*-equivalence class to which* y' *belongs).*

Combining Lemmas 5 and 7, and the fact that $h([y']) = d(y', P(y'))$, we obtain the following proposition.

PROPOSITION 3. *(T, h) is the dendrogram of (Ω, d).*

Remember that the earlier algorithm to construct the dendrogram (T, h) needs $O(\log n)$ time and $O(n)$ processors if T' is known. Since T' can be computed in $O(\log n)$ time using $O(n^2/\log n)$ processors, we obtain the following result.

PROPOSITION 4. *The dendrogram (T, h) is constructible in $O(\log n)$ time using $O(n^2/\log n)$ processors.*

This proves Theorem 1. □

3. Recognizing tree metrics. The problem we deal with in this section is the following: Given a distance function d on Ω. The problem is to decide whether d is a tree metric and if d is a tree metric, and to construct an edge-labeled tree T with edge labeling h, which represents the tree metric d, i.e., a tree T and a real-valued function h with E_T as its domain such that

1. $\Omega \subseteq V_T$, and
2. For $x, y \in \Omega$, $d(x, y)$ is the sum of the labelings $h(e)$ of all edges e on the unique path from x to y in T.

We reduce the recognition problem of tree metrics to the recognition problem of ultrametrics [2], and we reduce the construction of an edge-labeled tree belonging to a tree metric to the construction of a dendrogram of an ultrametric.

We pick up an $r \in \Omega$ and choose some $c > 2 \max_{x,y\in\Omega} d(x, y)$. We define a new distance function δ on $\Omega\setminus\{r\}$ with $\delta(x, y) = c - d(x, r) - d(y, r) + d(x, y)$ if $x \neq y$ and $\delta(x, y) = 0$ if $x = y$.

Bandelt [2] proved the following.

PROPOSITION 5. *Let d and δ be defined as above. Then d is a tree metric if and only if δ is an ultrametric.*

Therefore, to recognize tree metrics, we need only do the following:

1. **Compute** $c = 3 \max \{d(x, y) | x, y \in \Omega\}$.
2. **Select some** $r \in \Omega$ **and compute for all** $x, y \in \Omega\setminus\{r\}$,

$$\delta(x, y) = c - d(x, r) - d(y, r) + d(x, y).$$

3. **Check whether** δ **is an ultrametric.**

The first step can be done in $O(\log n)$ time using $O(n^2/\log n)$ processors. The second step can be done in constant time using $O(n^2)$ processors and therefore in $O(\log n)$ time using $O(n^2/\log n)$ processors. The third step can be done in $O(\log n)$ time using $O(n^2/\log n)$ processors. Therefore we get the following theorem.

THEOREM 2. *Tree metrics can be recognized in $O(\log n)$ time using $O(n^2/\log n)$ processors.*

It remains to construct the underlying tree of a tree metric.

1. **We construct the dendrogram** (T_δ, h_δ) **for the ultrametric** δ **as defined above. We define a tree** T_d **consisting of the vertices and edges of** T_δ**, the additional vertex** r**, and an additional edge** e**, which joins the root** r' **of** T_δ **with** r**. For each nonroot vertex** x **of** T_δ**, let** par (x) **be the parent of** x **with respect to** T_δ **and let** par $(r') = r$ (par **defines a parent function for** T_d).
2. **For each nonleaf vertex** x **of** T_δ**, let** $h_d(x) = h_\delta(x)$. **Define** $h_d(r) = c$. **For each leaf** $x \in \Omega\setminus\{r\}$, **let** $h_d(x) = c - 2d(x, r)$.
3. **We set** $d'(x, \text{par}(x)) = (h_d(\text{par}(x)) - h_d(x))/2$.

It is easily checked that all these steps can be done in $O(\log n)$ time using $O(n)$ processors if the minimum spanning tree for δ constructed in the ultrametric recognition algorithm for δ is known. It remains to prove that (T_d, d') is an underlying edge labeled tree for the tree metric d.

Let $x, y \in \Omega$. Let $(x = x_1, \dots, x_k = y)$ be the unique path from x to y in T_d. We must prove that $d(x, y) = \sum_{i=1}^{k-1} d'(x_i, x_{i+1})$. First, we suppose that $y = r$. Then $x_{i+1} = \operatorname{par}(x_i)$. It is easily checked that

$$\sum_{i=1}^{k-1} d'(x_i, x_{i+1}) = \sum_{i=1}^{k-1} (h_d(x_{i+1}) - h_d(x_i))/2$$

(by definition of $d'(x_i, x_{i+1})$)

$$= (h_d(x_k) - h_d(x_1))/2$$

$$= (h_d(r) - h_d(x))/2$$

(note that $x_k = r$ and $x_1 = x$)

$$= (c - (c - 2d(x, r))/2$$

(note that $h_d(r) = c$ and $h_d(x) = c - 2d(x, r)$)

(1) $$= d(x, r) = d(x, y).$$

Suppose now that x and y are different from r. Then we find a least common ancestor x_l of x and y in T_d, which is also the least common ancestor of x and y in T_δ. Let $(x_l = z_1, \dots, z_p = r)$ be the unique path from x_l to r in T_d. Then, since $\sum_{i=1}^{l-1} d'(x_i, x_{i+1}) + \sum_{j=1}^{p-1} d'(z_j, z_{j+1}) = d(x, r)$ and $\sum_{i=l+1}^{k} d'(x_i, x_{i-1}) + \sum_{j=1}^{p-1} d'(z_j, z_{j+1}) = d(y, r)$, we can derive the following equalities:

$$\sum_{i=1}^{k} d'(x_i, x_{i+1}) = \sum_{i=1}^{l-1} d'(x_i, x_{i+1}) + \sum_{i=l+1}^{k} d'(x_i, x_{i-1})$$

$$= \sum_{i=1}^{l-1} d'(x_i, x_{i+1}) + \sum_{j=1}^{p-1} d'(z_j, z_{j+1}) + \sum_{i=k}^{l+1} d'(x_i, x_{i-1})$$

$$+ \sum_{j=1}^{p-1} d'(z_j, z_{j+1}) - 2 \sum_{j=1}^{p-1} d'(z_j, z_{j+1})$$

(we add and subtract $2 \sum_{j=1}^{p-1} d'(z_j, z_{j+1})$)

$$= d(x, r) + d(y, r) - 2 \sum_{j=1}^{p-1} d'(z_j, z_{j+1}) = (*)$$

(the unique path from x to r is $(x_1, \dots, x_l = z_1, \dots, z_p = r)$, and, by (1), $d(x, r) = \sum_{i=1}^{l-1} d'(x_i, x_{i+1}) + \sum_{j=1}^{p-1} d'(z_j, z_{j+1})$; analogously, $d(y, r) = \sum_{i=k}^{l+1} d'(x_i, x_{i-1}) + \sum_{j=1}^{p-1} d'(z_j, z_{j+1})$)

$$(*) = d(x, r) + d(y, r) - 2 \sum_{j=1}^{p-1} (h_d(z_{j+1}) - h_d(z_j))/2$$

$$= d(x, r) + d(y, r) - h_d(z_p) + h_d(z_1) = d(x, r) + d(y, r) - h_d(r) + h_d(x_l)$$

(note that $z_1 = x_l$ and $z_p = r$)

$$= d(x, r) + d(y, r) - c + h_\delta(x_l)$$

$$= d(x, r) + d(y, r) - c + \delta(x, y) = (**)$$

(h_δ is the height function of δ and x_l is the least common ancestor of x and y in T_δ);

$$(**) = d(x,r) + d(y,r) - c + c - d(x,r) - d(y,r) + d(x,y)$$

(note that $\delta(x,y) = c - d(x,r) - d(y,r) + d(x,y)$)

$$= d(x,y).$$

Therefore (T_d, d') is an underlying edge labeled tree for the tree metric d. We can conclude the following result.

THEOREM 3. *Tree metrics can be recognized in $O(\log n)$ time using $O(n^2/\log n)$ processors. Moreover, its corresponding tree structure can be computed by the same time and processor bound.*

4. Conclusion. This paper discussed ultrametrics and tree metrics. In general, the given distance function is not an ultrametric. The general problem of hierarchical clustering is to find some ultrametric approximation and the corresponding dendrogram for a given distance function.

There are several hierarchical clustering heuristics (see, for example, [13]). The remaining problem is to parallelize those heuristics. Still, it is only possible to parallelize the so-called Single Linkage heuristics [10]. It remains an open problem to parallelize other hierarchical clustering heuristics.

REFERENCES

[1] B. AWERBUCH AND Y. SHILOACH, *New connectivity and* MSF *algorithms for shuttle-exchange network and* PRAM, IEEE Trans. Comput., 36 (1987), pp. 1258–1263.

[2] H. BANDELT, *Recognition tree metrics*, SIAM J. Discrete Math., 3 (1990), pp. 1–6.

[3] H. BANDELT AND A. DRESS, *Reconstructing the shape of a tree from observed similarity data*, Advances Appl. Math., 7 (1986), pp. 309–343.

[4] F. BOESCH, *Properties of the distance matrix of a tree*, Quart. Appl. Math., 26 (1968), pp. 607–609.

[5] P. BUNEMAN, *The recovery of trees from measures of dissimilarity*, in Mathematics in the Archaeological and Historical Sciences, F. R. Hodson et al., eds., Edinburgh University Press, Edinburgh, Scotland, 1971, pp. 387–395.

[6] P. BUNEMAN, *A note on the metric properties of trees*, J. Combin. Theory Ser. B, 17 (1974), pp. 48–50.

[7] R. COLE, *Parallel Merge Sort*, IEEE-FOCS, 27 (1986), pp. 511–516.

[8] J. CULBERSON AND P. RUDNICKI, *A fast algorithm for constructing trees from distance matrices*, Inform. Process. Lett., 30 (1990), pp. 215–220.

[9] J. CUNNINGHAM, *Free trees and bidirectional trees as representations of psychological distance*, J. Math. Psych., 17 (1978), pp. 165–188.

[10] E. DAHLHAUS, *Fast parallel algorithm for the single link heuristics of hierarchical clustering*, in Proc. 4th IEEE Sympos. on Parallel and Distributed Processing, Arlington, TX, 1992, pp. 184–186.

[11] S. FORTUNE AND J. WYLLIE, *Parallelism in random access machines*, ACM-STOC, 10 (1978), pp. 114–118.

[12] A. GORDON, *A review of hierarchical classification*, J. Roy. Statist. Soc. A, 150 (1987), pp. 119–137.

[13] A. JAIN AND R. DUBES, *Algorithms for Clustering Data*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[14] C. KRUSKAL, L. RUDOLPH, AND M. SNIR, *Efficient parallel algorithms for graph problems*, Algorithmica, 5 (1990), pp. 43–46.

[15] A. PATRINOS AND S. HAKIMI, *The distance matrix of a graph and its tree realization*, Quart. Appl. Math., 30 (1972), pp. 255–269.

[16] S. SATTAH AND A. TVERSKY, *Additive similarity trees*, Psychometrica, 42 (1977), pp. 319–345.

[17] B. SCHIEBER AND U. VISHKIN, *On finding lowest common ancestors: Simplification and parallelization*, SIAM J. Comput., 17 (1988), pp. 1253–1262.

[18] J. SIMÕES-PEREIRA, *A note on the tree realizability of a distance matrix*, J. Combin. Theory, 6 (1969), pp. 303–310.

[19] ———, *A note on optimal and suboptimal digraph realizations of quasidistance matrices*, SIAM J. Algebraic Discrete Meth., 5 (1984), pp. 117–132.

SIAM J. DISC. MATH.
Vol. 6, No. 4, pp. 533–547, November 1993

MONOTONE OPTIMAL MULTIPARTITIONS USING SCHUR CONVEXITY WITH RESPECT TO PARTIAL ORDERS*

FRANK K. HWANG†, URIEL G. ROTHBLUM‡, AND LARRY SHEPP†

Abstract. In a (t, n, m)-multipartitioning problem, t lists of nm ordered numbers are partitioned into n sets, where each set contains m numbers from each list. The goal is to maximize some objective function that depends on the sum of the elements in each set and is called the *partition function*. The authors use the recently developed theory of majorization and Schur convexity with respect to partially ordered sets to study optimal multipartitions for the above problem. In particular, they apply the results to construct a class of counterexamples to a recent conjecture of Du and Hwang, which asserts that (classic) Schur convex functions can be characterized as the partition functions for $(1, n, m)$-multipartitioning problems having monotone optimal solutions.

Key words. partitions, monotonicity, optimal partitions, Schur convexity, partial order

AMS subject classifications. 90B25, 26A51

1. Introduction.

For a vector $x \in R^n$ and $k = 1, \ldots, n$, let $x_{[k]}$ be the kth largest coordinate of x. We say that a vector $a \in R^n$ *majorizes* a vector $b \in R^n$, written $a \gg b$, if

$$\sum_{i=1}^{k} a_{[i]} \geqq \sum_{i=1}^{k} b_{[i]} \quad \text{for all } k = 1, \ldots, n-1 \tag{1.1}$$

and

$$\sum_{i=1}^{n} a_{[i]} = \sum_{i=1}^{n} b_{[i]}. \tag{1.2}$$

A function $g: R^n \rightarrow R$ is called *Schur convex* if

$$g(a) \geqq g(b) \quad \text{for all } a, b \in R^n \text{ satisfying } a \gg b. \tag{1.3}$$

The richness and usefulness of the theory associated with majorization and Schur convexity is well documented; see Marshall and Olkin [1979]. In particular, it is an important tool for studying optimization problems and establishing inequalities for stochastic systems. One example for the use of this theory concerns the problem of optimal assembly of parts of t different types into modules that compose a coherent system, where the goal is to maximize the system reliability. A survey of this problem when the modules are series modules, i.e., a module works if and only if all of its parts work, can be found in Hwang and Rothblum [1993a]. An assembly is called *monotone* if the modules can be labeled such that $i < j$ implies that for each type, each part in module i is not worse than any part of module j. Schur convexity has been used to establish optimality of monotone assemblies when the reliability function is symmetric and each module has the same number of parts of each type. This special case of the assembly problem has been abstracted into so-called (t, n, m)*-multipartitioning problems* in which t lists of nm ordered numbers are to be partitioned into n sets where each set contains m numbers from each list. The goal in these problems is to maximize an objective that depends on

* Received by the editors September 6, 1991; accepted for publication (in revised form) August 12, 1992.

† AT&T Bell Laboratories, Murray Hill, New Jersey 07974.

‡ Faculty of Industrial Engineering and Management, Technion–Israel Institute of Technology, Haifa 32000, Israel and Rutgers Center for Operations Research (RUTCOR), Rutgers University, P.O. Box 5062, New Brunswick, New Jersey 08903.

the sum of the elements in each set and is called the *partition function*. A monotone assembly corresponds to a *monotone multipartition* in which $i < j$ implies that

(1.4) for each list, every number in set i is ordered above any number in set j.

In the current paper, we use the theory of majorization and Schur convexity with respect to partial orders developed recently by Hwang and Rothblum [1993b] to study multipartitioning problems having optimal multipartitions for which (1.4) is exhibited for certain pairs of indices i and j, where i precedes j in an underlying partial order.

The (t, m, n)-multipartitioning problem is of particular interest because a question was raised of whether the optimality of monotone multipartitions characterizes Schur convexity (of the partition function). The question has been answered in the affirmative for $t \geqq 2$ by Hollander, Proschan, and Sethuraman [1977] and also by Du and Hwang [1990]. It is conjectured in the latter reference that the answer to the above question is also affirmative when $t = 1$. In the current paper, we give a class of counterexamples to this conjecture by constructing problems having monotone optimal multipartitions but whose partition functions are not Schur convex. We establish optimality of monotone multipartitions for these examples by using the results about optimal multipartitions described at the end of the above paragraph.

The organization of the paper is as follows. We summarize definitions and results from Hwang and Rothblum [1993b] about majorization and Schur convexity with respect to partial orders. We then formally introduce the multipartitioning problems in § 3 and establish the results about monotonicity properties of optimal multipartitions in § 4. The counterexamples to the conjecture of Du and Hwang [1990] are constructed in § 5, and some conclusions are discussed in § 6.

2. Majorization and Schur convexity with respect to partial orders. Throughout this paper, let n be a fixed positive integer and let $\Rightarrow$ be a given *partial order* on the integers $\{1, \ldots, n\}$ that is consistent with the linear order $>$ on the integers; i.e., $\Rightarrow$ is a transitive, asymmetric relation on $\{1, \ldots, n\}$ and if $j \Rightarrow \mathrm{i}$, then $j > i$. We call $\Rightarrow$ the *domination relation* and we say that *j dominates i* if $j \Rightarrow i$. When $j \Rightarrow i$, we sometimes write $i \Leftarrow j$. Also, we say that *j directly dominates i*, written $j \underline{\Rightarrow} i$, if j dominates i and there exists no integer k that dominates i and is dominated by j.

A subset K of $\{1, \ldots, n\}$ is called *closed under* $\Rightarrow$, abbreviated $\Rightarrow$*-closed*, if K contains all integers that are dominated by any integer that is in K. As we assume that the partial order $\Rightarrow$ is consistent with the linear order $>$, for every integer $k = 1, \ldots, n$, the set $\{1, \ldots, k\}$ is $\Rightarrow$-closed. Furthermore, these sets are the only $\Rightarrow$-closed sets if and only if the partial order $\Rightarrow$ is the linear order $>$.

Let a and b be vectors in R^n. We say that *a majorizes b with respect to* the partial order $\Rightarrow$, written $a \gg^{\Rightarrow} b$, if

(2.1) $\sum_{k \in K} a_k \geqq \sum_{k \in K} b_k$ for each subset K of $\{1, \ldots, n\}$ that is $\Rightarrow$-closed

and

$$\sum_{k=1}^{n} a_k = \sum_{k=1}^{n} b_k. \tag{2.2}$$

Of course, $\Rightarrow$-majorization is a transitive relation on R^n. Also, as the sets $\{1, \ldots, k\}$ for $k = 1, \ldots, n$ are $\Rightarrow$-closed, $\Rightarrow$-majorization implies $>$-majorization.

A vector $x \in R^n$ is called *decreasing* if $x_1 \geqq x_2 \geqq \cdots \geqq x_n$, and the set of all decreasing vectors in R^n will be denoted by $R^n_{\downarrow}$. The *decreasing rearrangement* of a vector $x \in R^n$, denoted $x_{\downarrow}$, is defined as the unique vector in $R^n_{\downarrow}$ obtained from x by rearranging its coordinates, i.e., for $k = 1, \dots, n$, $(x_{\downarrow})_k = x_{[k]}$, where $x_{[k]}$ is the kth largest coordinate of x. We observe that the standard definition of majorization (given in the Introduction) can be expressed by $>$-majorization on $R^n_{\downarrow}$; i.e., $a \in R^n$ majorizes $b \in R^n$ if and only if $a_{\downarrow} \gg^{>} b_{\downarrow}$.

The following lemma, given in Hwang and Rothblum [1993b, Lem. 2.2], shows that certain perturbations of an arbitrary vector result in majorizing vectors.

LEMMA 2.1. *Suppose that a is a vector in R^n and $i, j \in \{1, \dots, n\}$, where $j \Rightarrow i$. Then, for every scalar $\gamma \geqq 0$, $a + \gamma(e^i - e^j) \gg^{\Rightarrow} a$.*

We say that a subset S of R^n is *$\Rightarrow$-pairwise connected* if, for every a and b in S satisfying $a \gg^{\Rightarrow} b$, there exist vectors $u^0 = b, u^1, \dots, u^q = a$ in S, positive real numbers $\gamma_1, \dots, \gamma_q$ and integers $i(1), \dots, i(q)$ and $j(1), \dots, j(q)$ in $\{1, \dots, n\}$ such that, for $t = 1, \dots, q$,

$$j(t) \underline{\Rightarrow} i(t) \tag{2.3}$$

and

$$u^t = u^{t-1} + \gamma_t[e^{i(t)} - e^{j(t)}]; \tag{2.4}$$

in particular, Lemma 2.1 shows that (2.4) implies that $u^t \gg^{\Rightarrow} u^{t-1}$. We say that S is *weakly $\Rightarrow$-pairwise connected* if in the above definition for each $t = 1, \dots, q$, $j(t)$ dominates (rather than directly dominates) $i(t)$. Of course, $\Rightarrow$-pairwise connectedness implies weak $\Rightarrow$-pairwise connectedness, but the reverse implication need not hold; see an example in Hwang and Rothblum [1993b].

Hwang [1979, Thm. 3.2] proved that R^n is weakly $\Rightarrow$-pairwise connected. His original proof relied on the nonemptiness of the core of a convex game. Simpler proofs of Hwang's result that rely on the max-flow min-cut theorem and on a standard characterization of feasible transportation problems were given, respectively, in Lih [1982] and Rothblum [1993]. We note that both Hwang's and Lih's proofs actually show the stronger result that R^n is $\Rightarrow$-pairwise connected; see Hwang and Rothblum [1993b, Prop. 2.4]. The following generalization of this variant of Hwang's result was established in Hwang and Rothblum [1993b, Thm. 2.5].

THEOREM 2.2 *Let S be a convex, open subset of R^n. Then S is pairwise $\Rightarrow$-connected.*

The next theorem, established in Hwang and Rothblum [1993b, Thm. 2.6], is needed for our development. We will need another definition to formally state the result. We say that $j \in \{1, \dots, n\}$ *completely dominates* $i \in \{1, \dots, n\}$, written $j \Rightarrow\Rightarrow i$, if, for every $p \in \{1, \dots, i\}$ and $q \in \{j, \dots, n\}$, $q \Rightarrow p$. Of course, complete domination is a partial order on $\{1, \dots, n\}$; i.e., it is transitive and asymmetric. Also, complete domination and (regular) domination coincide when the partial order $\Rightarrow$ is the linear order $>$.

THEOREM 2.3. *Let $a \in R^n_{\downarrow}$ and $i, j \in \{1, \dots, n\}$, where $j \Rightarrow\Rightarrow i$. Then, for every scalar $\gamma \geqq 0$,*

$$[a + \gamma(e^i - e^j)]_{\downarrow} \gg^{\Rightarrow} a. \tag{2.5}$$

Let S be a subset of R^n. A function $h: S \to R$ is called *Schur convex on S with respect to* the partial order $\Rightarrow$, abbreviated *$\Rightarrow$-Schur convex on S*, if

$$h(a) \geqq h(b) \quad \text{for all } a, b \in S \text{ satisfying } a \gg^{\Rightarrow} b. \tag{2.6}$$

As $\Rightarrow$-majorization implies $>$-majorization, $>$-Schur convexity on a subset S of R^n implies $\Rightarrow$-Schur convexity on that subset. However, there are functions that are $\Rightarrow$-Schur convex for a particular partial order $\Rightarrow$, while they are not $>$-Schur convex; see Example B in § 5.

When the partial order $\Rightarrow$ is the linear order $>$ and $S = R^n$, the above definition of $\Rightarrow$-Schur convexity does not specialize to the standard definition of Schur convexity given in the Introduction, as we do not assume that the underlying function is symmetric and we do not require that (2.6) holds when the coordinates of the vectors a and b are rearranged. However, a symmetric function $g:R^n \to R$ satisfies the standard definition of Schur convexity if and only if its restriction to $R^n_\downarrow$ is $>$-Schur convex. Furthermore, given a function $h:R^n_\downarrow \to R$, let $h^\wedge:R^n \to R$ be defined for $x \in R^n$ by $h^\wedge(x) = h(x_\downarrow)$. Then $h^\wedge$ is the unique real-valued symmetric function on R^n that coincides with h on $R^n_\downarrow$ (see Marshall and Olkin [1979, p. 92]), and h is $>$-Schur convex if and only if $h^\wedge$ satisfies the standard definition of Schur convexity.

It is observed in the above paragraph that we can study symmetric functions on R^n (whether or not Schur convex) by considering arbitrary functions of $R^n_\downarrow$. When considering Schur-convex functions, this approach bypasses the need to refer to the decreasing rearrangement $x_\downarrow$ of a given vector x. Furthermore, the approach has another advantage. Given a symmetric function $g:R^n \to R$, differentiability of the restriction of g to $R^n_\downarrow$ is a weaker condition than differentiability of g on all of R^n (though continuity of the restriction of g to $R^n_\downarrow$ is equivalent to continuity of g on all of R^n). Indeed, a characterization of $>$-Schur convexity via differentiability that yields an extension of the Schur–Ostrowski characterization of differentiable Schur-convex functions was obtained in Hwang and Rothblum [1993b].

Let S be a subset of R^n. As usual, we say that S is *convex* if, for every a and b in S and scalar $0 \leqq \alpha \leqq 1$, $(1 - \alpha)a + \alpha b \in S$. The *interior* of a subset S of R^n will be denoted int (S), in particular, int $(R^n_\downarrow) = \{x \in R^n : x_1 > x_2 \cdots > x_n\}$.

The following result provides local characterizations of Schur convexity with respect to partial orders. It was established in Hwang and Rothblum [1993b, Thm. 3.2, Lem. 3.1].

THEOREM 2.4. *Let S be a subset of R^n and let $h:S \to R$. Then*

(a) *If S is weakly $\Rightarrow$-pairwise connected, the function h is $\Rightarrow$-Schur convex on S if and only if, for all $x \in S$ and $i, j \in \{1, \ldots, n\}$ satisfying $j \Rightarrow i$*

$$(2.7) \qquad h[x + \gamma(e^i - e^j)] \text{ is increasing in } \gamma \text{ on } \{\gamma \geqq 0 : x + \gamma(e^i - e^j) \in S\};$$

(b) *If S is $\Rightarrow$-pairwise connected, the function h is $\Rightarrow$-Schur convex on S if and only if (2.7) holds for all $x \in S$ and $i, j \in \{1, \ldots, n\}$ where $j \underline{\Rightarrow} i$; and*

(c) *If S is open and convex and the function h is continuously differentiable on S, then h is $\Rightarrow$-Schur convex on S if and only if*

$$(2.8) \qquad h_{x_i}(x) \geqq h_{x_j}(x) \quad \text{for all } x \in S \text{ and } i, j \in \{1, \ldots, n\} \text{ satisfying } j \underline{\Rightarrow} i;$$

(d) *If S is a convex set having dimension n and h is continuous, then h is $\Rightarrow$-Schur convex on S if and only if h is $\Rightarrow$-Schur convex on* int (S).

The following corollary specializes Theorem 2.4 to the case where $S = R^n_\downarrow$. Its formal proof is given in Hwang and Rothblum [1993b, Cor. 3.5].

COROLLARY 2.5. *Let h: $R^n_\downarrow \to R$ be a continuous function. Then*

(a) *The following are equivalent*:

(a1) *h is $\Rightarrow$-Schur convex on $R^n_\downarrow$,*

(a2) *h is $\Rightarrow$-Schur convex on* int $(R^n_\downarrow)$,

(a3) *for all $x \in$ int $(R^n_\downarrow)$ and $i, j \in \{1, \ldots, n\}$, where $j \Rightarrow i$, $h[x + \gamma(e^i - e^j)]$ is increasing in γ when $0 \leqq \gamma < \min_i \{x_{i-1} - x_i, x_j - x_{j+1}\}$, and* (a4) *for all* $x \in \operatorname{int}(R^n_\downarrow)$ *and* $i, j \in \{1, \ldots, n\}$, *where* $j \underline{\Rightarrow} i$, $h[x + \gamma(e^i - e^j)]$ *is increasing in* γ *when* $0 \leqq \gamma < \min_i \{x_{i-1} - x_i, x_j - x_{j+1}\}$;

(b) *If the function h is continuously differentiable on the* $\operatorname{int}(R^n_\downarrow)$, *then h is* $\Rightarrow$*-Schur convex on* $R^n_\downarrow$ *if and only if*

(2.9) $h_{x_i}(x) \geqq h_{x_j}(x)$ *for all* $x \in \operatorname{int}(R^n_\downarrow)$ *and* $i, j \in \{1, \ldots, n\}$ *satisfying* $j \underline{\Rightarrow} i$.

We note that condition (2.9) is equivalent to the (apparently more demanding) variant where $j \underline{\Rightarrow} i$ is replaced by $j \Rightarrow i$.

Lemma 3.A.2 of Marshall and Olkin [1979, p. 55] and the discussion in Arnold [1987] state that in the case where the partial order $\Rightarrow$ is the linear order $>$, the equivalence of parts (a1) and (a4) in Corollary 2.5 holds without any assumptions about continuity of the function h. However, an example provided in Hwang and Rothblum [1993b] shows that this assertion is false (though it is known that $R^n_\downarrow$ is weakly $>$-pairwise connected). Also, an application of Corollary 2.5 to symmetric functions on R^n provides an extension of the Schur–Ostrowski characterization of Schur convexity without the requirement that the function is differentiable on all of R^n; see Hwang and Rothblum [1993b, Cor. 3.6].

3. Formulation of the multipartitioning problem. One application of the classical theory on majorization and Schur convexity concerns the development of a framework for establishing optimality of monotone multipartitions, e.g., Derman, Lieberman, and Ross [1972], Du and Hwang [1990], El-Neweihi, Proschan, and Sethuraman [1987], Malon [1990], Hwang and Rothblum [1993a], and references therein.

We next give a formal formulation of multipartitioning problems. These problems are classified by a triplet (t, n, m) of positive integers, and we refer to a (t, n, m)-*multipartitioning problem*. The data for such a problem consists of a symmetric function $g: R^n \to R$ and t lists of nm real numbers

$$(3.1) \qquad \{a^u_k : k = 1, \ldots, nm\}, \qquad u = 1, \ldots, t,$$

where

$$(3.2) \qquad a^u_1 \geqq a^u_2 \geqq \cdots \geqq a^u_{nm}.$$

In particular, the function g is called the *partition function* of the problem. Throughout the remainder of this paper, we assume that t, m, and n are given positive integers and we consider (t, n, m)-multipartitioning problems with data given as above.

A *multipartition* for our (t, n, m)-multipartitioning problem is a family of sets $\sigma = \{\sigma_{ui} : u = 1, \ldots, t \text{ and } i = 1, \ldots, n\}$ such that

(3.3) $\{\sigma_{ui} : i = 1, \ldots, n\}$ is a partition of $\{1, \ldots, nm\}$ for each $u = 1, \ldots, t$

and

(3.4) the number of elements in σ_{ui} is m for each $u = 1, \ldots, t$ and $i = 1, \ldots, n$.

In this case, we define the *vector associated with* σ, denoted a^σ, by

$$(3.5) \qquad a^\sigma \equiv \left(\sum_{u=1}^{t} \sum_{k \in \sigma_{u1}} a^u_k, \sum_{u=1}^{t} \sum_{k \in \sigma_{u2}} a^u_k, \ldots, \sum_{u=1}^{t} \sum_{k \in \sigma_{un}} a^u_k \right) \in R^n.$$

Also, the *objective associated with* σ, is then defined by $R(\sigma) \equiv g(a^\sigma)$. We call a multipartition σ *optimal* if $R(\sigma) \geqq R(\sigma')$ for all multipartitions σ'.

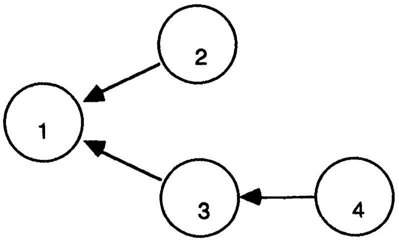


FIG. 1

Example A. Let $n = 4$ and let the partial order $\Rightarrow$ be given by Fig. 1, where arrows are used to represent direct domination in the obvious way. Also, consider the $(1, 4, 2)$-multipartitioning problem with data given by

k	1	2	3	4	5	6	7	8
a_k^1	9	9	8	6	5	3	2	1

and whose associated (symmetric) function g is defined for $x \in R_\downarrow^n$ by

$$g(x) = \begin{cases} 1 & \text{if } x \gg^{\Rightarrow} (18, 10, 9, 6), \\ 0 & \text{otherwise.} \end{cases}$$

We observe that the multipartition σ^* with $\sigma_{11}^* = \{1, 2\}$, $\sigma_{12}^* = \{3, 7\}$, $\sigma_{13}^* = \{4, 6\}$, $\sigma_{14}^* = \{5, 8\}$ has $a^{\sigma^*} = (18, 10, 9, 6)$ and $R(\sigma^*) = g(a^{\sigma^*}) = 1$; hence it is optimal.

Our definition of multipartitions focuses on the partitioning of the indices rather than the associated real numbers. This is particularly convenient when the numbers have ties. Also, this definition is independent of the data, and the sets of multipartitions for all (t, n, m)-multipartitioning problems coincide.

Multipartitioning problems have been studied in the literature of optimally assembling parts of t different types into a system consisting of n symmetric modules with the objective of maximizing reliability. When each module consists of m parts of each type, we have a (t, n, m)-multipartitioning problem. Starting with the original paper of Derman, Lieberman, and Ross [1972], several papers that establish the existence of monotone optimal assemblies rely on the theory of majorization and Schur convexity. Here, monotone means that one module gets the best m parts of each type, another gets the second best parts of each type, and so on. We now extend these results to the partial order case.

For a subset K of $\{1, \ldots, n\}$, we use the notation K^c to denote the *complement of K within* $\{1, \ldots, n\}$, i.e., $K^c \equiv \{1, \ldots, n\} \backslash K$. A subset K of $\{1, \ldots, n\}$ is called a $\Rightarrow$-*cut* if, for every $i \in K$ and $j \in K^c$, $j \Rightarrow i$. As the partial order $\Rightarrow$ is consistent with the linear order $>$, it follows that a nonempty $\Rightarrow$-cut K must have the form $\{1, \ldots, |K|\}$, where $|K|$ is the cardinality of K. Of course, not every set of this form is a $\Rightarrow$-cut. Recalling the definition of complete domination in § 2, we further note that, if K is a $\Rightarrow$-cut, then each $j \in K^c$ completely dominates each $i \in K$.

Let K be a subset of $\{1, \ldots, n\}$. Given a multipartition σ and $u \in \{1, \ldots, t\}$, we define σ_{uK} to be the set $\cup_{j \in K} \sigma_{uj}$. We say that σ is *monotone with respect to K*, or briefly that σ is *K-monotone*, if, for all $u = 1, \ldots, t$,

$$k \in \sigma_{uK} \text{ and } p \in \sigma_{uK^c} \quad \text{then } k < p. \tag{3.6}$$

We observe that (3.6) is equivalent to the assertion that

$$(3.6') \qquad \sigma_{uK} = \{1, \dots, m|K|\} \quad \text{for every } u = 1, \dots, t.$$

In the context of assembling parts into the modules of a system, K-monotonicity can be interpreted as the assertion that any part that is assigned to a module that corresponds to the index set K is better than any part that is assigned to a module that is not indexed by K. We say that a vector $a \in R^n$ is *decreasing with respect to K*, or briefly that a is *K-decreasing*, if

$$(3.7) \qquad a_i \geqq a_j \quad \text{for every } i \in K \text{ and } j \in K^c.$$

Of course, if σ is a multipartition that is K-monotone, then a^σ is K-decreasing. Also, $a \in R^n$ is decreasing if and only if a is K-decreasing for every set K having the form $\{1, \dots, s\}$ with $s \in \{1, \dots, n\}$; in particular, if $a \in R^n_\downarrow$, then a is K-decreasing for every $\Rightarrow$-cut K.

A one-to-one function π: $\{1, \dots, n\} \rightarrow \{1, \dots, n\}$ is called a *permutation* of $\{1, \dots, n\}$. Given a permutation π and a multipartition σ, we define σ^π to be the multipartition with

$$(3.8) \qquad \sigma^\pi_{ui} = \sigma_{u\pi(i)} \quad \text{for each } u = 1, \dots, t \text{ and } i = 1, \dots, n.$$

In particular, we then have that

$$(3.9) \qquad (a^{\sigma^\pi})_i = \sum_{u=1}^{t} \sum_{k \in \sigma^\pi_{ui}} a^u_k = \sum_{u=1}^{t} \sum_{k \in \sigma_{u\pi(i)}} a^u_k = (a^\sigma)_{\pi(i)} \quad \text{for } i = 1, \dots, n.$$

Thus, a^{σ^π} is obtained from a^σ by coordinate-reshuffling. As the function g is symmetric, we conclude that

$$(3.10) \qquad R(\sigma^\pi) = g(a^{\sigma^\pi}) = g(a^\sigma) = R(\sigma).$$

Finally, given a permutation π on $\{1, \dots, n\}$ and a subset K of $\{1, \dots, n\}$, we denote the set $\{\pi(i): i \in K\}$ by $\pi(K)$.

LEMMA 3.1. *Consider a (t, n, m)-multipartitioning problem and let σ be a corresponding multipartition. Then there exists a permutation π such that*

(a) $a^{\sigma^\pi} = (a^\sigma)_\downarrow$,

(b) *If K is a $\Rightarrow$-cut for which $a^\sigma$is K-decreasing, then $\pi(K) = K$, in particular, in such cases a^{σ^π} is also K-decreasing, and*

(c) *If K is a $\Rightarrow$-cut for which σ is K-monotone, then $\pi(K) = K$, in particular, in such cases σ^π is also K-monotone.*

Proof. We can easily construct a permutation π such that, for each $i = 1, \dots, m$, $\pi(i)$ is the index of the ith largest element of a^σ and $\pi(K) = K$ for every $\Rightarrow$-cut K with respect to which a^σ is K-decreasing (the latter imposes an extra requirement only when there are ties among $(a^\sigma)_1, \dots, (a^\sigma)_n$). In particular, $(a^\sigma)_{\pi(i)} = (a^\sigma)_{[i]}$ for each $i = 1, \dots, n$. Combining this observation with (3.9) shows that $a^{\sigma^\pi} = (a^\sigma)_\downarrow$. Also, if K is a subset of $\{1, \dots, n\}$ where a^σ is K-decreasing, then the assumption $\pi(K) = K$ and (3.9) imply that $\{(a^\sigma)_i: i \in K\} = \{(a^{\sigma^\pi})_i: i \in K\}$ and $\{(a^\sigma)_i: i \in K^c\} = \{(a^{\sigma^\pi})_i: i \in K^c\}$; hence, the assertion that a^σ is K-decreasing immediately implies that a^{σ^π} is K-decreasing as well. Finally, if σ is K-monotone, then σ is K-decreasing, and therefore $\pi(K) = K$, immediately implying that the σ^π is also K-monotone.

Lemma 3.1 implies that, for every multipartition σ, there is a multipartition σ' with $R(\sigma') = R(\sigma)$ and $a^{\sigma'} \in R^n_\downarrow$. In particular, when searching for an optimal multipartition, it suffices to restrict attention to those multipartitions σ for which a^σ is K-decreasing.

Example A (continued). Consider Example A introduced earlier in this section. We have seen that the multipartition σ^* with $\sigma^*_{11} = \{1, 2\}$, $\sigma^*_{12} = \{3, 7\}$, $\sigma^*_{13} = \{4, 6\}$, $\sigma^*_{14} = \{5, 8\}$ has $a^{\sigma^*} = (18, 10, 9, 6)$, $R(\sigma^*) = g(a^{\sigma^*}) = 1$, and is optimal. We will show that up to a permutation of the sets, σ^* is the only optimal multipartition. Let σ be an optimal multipartition. By possibly permuting the sets of σ, we will assume that a^σ is decreasing; see Lemma 3.1. Then, $a^\sigma \gg^{\Rightarrow} (18, 10, 9, 6)$; hence, a^σ satisfies

$$(a^\sigma)_1 \geqq 18,$$

$$(a^\sigma)_1 + (a^\sigma)_2 \geqq 28,$$

$$(a^\sigma)_1 + (a^\sigma)_3 \geqq 27,$$

$$(a^\sigma)_1 + (a^\sigma)_2 + (a^\sigma)_3 \geqq 37,$$

$$(a^\sigma)_1 + (a^\sigma)_3 + (a^\sigma)_4 \geqq 33,$$

$$(a^\sigma)_1 + (a^\sigma)_2 + (a^\sigma)_3 + (a^\sigma)_4 = 43.$$

We first note that the inequality $(a^\sigma)_1 \geqq 18$ can be satisfied with our data only if $\sigma_{11} = \{1, 2\}$, in which case $(a^\sigma)_1 = 18$. Substituting $(a^\sigma)_1 = 18$ into the remaining inequalities, we see that necessarily $(a^\sigma)_2 \geqq 10$, $(a^\sigma)_3 \geqq 9$, $(a^\sigma)_2 + (a^\sigma)_3 \geqq 19$, $(a^\sigma)_3 + (a^\sigma)_4 \geqq 15$, and $(a^\sigma)_2 + (a^\sigma)_3 + (a^\sigma)_4 = 25$, implying that $(a^\sigma)_2 = 10$, $(a^\sigma)_3 + (a^\sigma)_4 = 15$. In particular, as $(a^\sigma)_2 = 10$, our data implies that $\sigma_{12} = \{3, 7\}$. It now follows that $\sigma_{13} \cup \sigma_{14} = \{4, 5, 6, 8\}$, and therefore the requirement $(a^\sigma)_3 \geqq 9$ can be satisfied with our data only if either $\sigma_{13} = \{4, 5\}$ or $\sigma_{13} = \{4, 6\}$; but, if $\sigma_{13} = \{4, 6\}$, we have that $(a^\sigma)_3 = 11 > 10 = (a^\sigma)_2$, contradicting the assertion that σ is decreasing. So, the only remaining alternative is that $\sigma_{13} = \{4, 6\}$, in which case, $\sigma = \sigma^*$.

4. Existence of monotone optimal multipartitions. In this section, we show that Schur convexity with respect to partial orders can be used to establish existence of an optimal multipartition that is monotone with respect to each $\Rightarrow$-cut.

THEOREM 4.1. *Consider a (t, n, m)-multipartitioning problem where the restriction of the partition function to $R^n_\downarrow$ is $\Rightarrow$-Schur convex. Then there exists an optimal multipartition σ that is K-monotone for every $\Rightarrow$-cut K and for which a^σ is decreasing.*

Proof. Let K be a $\Rightarrow$-cut and let σ be a multipartition. A *K-violation* of σ is defined as a quintuple (i, j, u, k, p) such that $i \in K$, $j \in K^c$, $u \in \{1, \dots, t\}$, $k \in \sigma_{ui}$, and $p \in \sigma_{uj}$, where $k < p$. We denote by V^σ_K the set of pairs (i, j) in $\{1, \dots, n\} \times \{1, \dots, n\}$ such that, for some u, k, and p, (i, j, u, k, p) is a K-violation of σ. In particular, a multipartition σ is K-monotone if and only if V^σ_K is empty.

We first consider any fixed $\Rightarrow$-cut K^*. Let σ^* be an optimal multipartition for which a^{σ^*} is decreasing such that the cardinality of $V^{\sigma^*}_{K^*}$ is smallest among all optimal multipartitions σ for which a^σ is decreasing. We will show that σ^* is K^*-monotone by showing that $V^{\sigma^*}_K$ is empty. Suppose that this is not the case and that $V^{\sigma^*}_{K^*}$ contains a pair, say (i^*, j^*). To establish a contradiction, consider the multipartition σ' obtained from σ^* by repartitioning the elements in $\sigma_{ui^*} \cup \sigma_{uj^*}$ for each $u = 1, \dots, t$ and assigning the m elements with the higher indices to i^* and the remaining m elements to j^*. In particular, we have that

$$\sigma'_{ui} = \sigma^*_{ui} \quad \text{for } u = 1, \dots, t \quad \text{and} \quad i \in \{1, \dots, n\} \setminus \{i^*, j^*\}, \tag{4.1}$$

and using the fact that $(i^*, j^*) \in V^{\sigma^*}_{K^*}$, we also have that

$$(a^{\sigma'})_{i^*} > (a^{\sigma^*})_{i^*} \quad \text{and} \quad (a^{\sigma^*})_{j^*} < (a^{\sigma'})_{j^*}. \tag{4.2}$$

Furthermore, as

$$\sum_{k\in\sigma'_{ui}} a_k^u = \sum_{k\in\sigma^*_{ui}} a_k^u \quad \text{for all } u = 1, \ldots, t \text{ and } i \in \{1, \ldots, n\} \setminus \{i^*, j^*\}$$

and

$$\sum_{u=1}^{t} \sum_{i=1}^{n} \sum_{k\in\sigma'_{ui}} a_k^u = \sum_{u=1}^{t} \sum_{k=1}^{nm} a_k^u = \sum_{u=1}^{t} \sum_{i=1}^{n} \sum_{k\in\sigma^*_{ui}} a_k^u \quad \text{for all } u = 1, \ldots, t,$$

it follows that, for $\gamma \equiv (a^{\sigma'})_{i^*} - (a^{\sigma^*})_{i^*} = (a^{\sigma^*})_{j^*} - (a^{\sigma'})_{j^*} > 0$,

$$a^{\sigma'} = a^{\sigma^*} + \gamma(e^{i^*} - e^{j^*}). \tag{4.3}$$

Now, as K^* is a $\Rightarrow$-cut, $i^* \in K^*$, and $j^* \in K^{*c}$, we have that j^* completely dominates i^*. Furthermore, as $a^{\sigma^*} \in R^n_{\downarrow}$, Theorem 2.3 implies that $[a^{\sigma'}]_{\downarrow} = [a^{\alpha^*} + \gamma(e^{i^*} - e^{j^*})]_{\downarrow} \gg^{\Rightarrow} a^{\sigma^*}$. Thus, by the assumptions that g is symmetric and its restriction to $R^n_{\downarrow}$ is $\Rightarrow$-Schur convex,

$$R(\sigma') = g(a^{\sigma'}) = g[(a^{\sigma'})_{\downarrow}] = g\{[a^{\sigma^*} + \gamma(e^{i^*} - e^{j^*})]_{\downarrow}\} \geqq g(a^{\sigma^*}) = R(\sigma^*).$$

As σ^* is optimal, it follows that σ' is also optimal.

We will next argue that $V^{\sigma'}_{K^*}$ is a proper subset of $V^{\sigma^*}_{K^*}$. Of course, $(i^*, j^*) \in V^{\sigma^*}_{K^*}$, and the construction of σ' assures that $(i^*, j^*) \notin V^{\sigma'}_{K^*}$. To see that $V^{\sigma^*}_{K^*} \supseteq V^{\sigma'}_{K^*}$, assume that $(i', j') \in V^{\sigma'}_{K^*}$. So, σ' has a K^*-violation of the form (i', j', u', k', p'), i.e., $i' \in K^*$, $j' \in K^{*c}$, $u' \in \{1, \ldots, \tau\}$, $k' \in \sigma'_{u'i'}$, $p' \in \sigma'_{u'j'}$, and $k' < p'$. We first observe that, if $\{i', j'\} \cap \{i^*, j^*\} = \varnothing$, then (4.1) implies that $\{i', j'\} \in V^{\sigma^*}_{K^*}$. So, assume that $\{i', j'\} \cap \{i^*, j^*\} \neq \varnothing$. As $(i^*, j^*) \notin V^{\sigma'}_{K^*}$, $i^* \in K^*$, $i' \in K^*$, $j^* \in K^{*c}$, and $j' \in K^{*c}$, we conclude that either $i' = i^*$ and $j' \notin \{i^*, j^*\}$ or $j' = j^*$ and $i' \notin \{i^*, j^*\}$. We next examine the following distinct alternatives:

(a) $i' = i^*$ and $j' \notin \{i^*, j^*\}$. The construction of σ' assures that $\sigma'_{u'j'} = \sigma^*_{u'j'}$ and that $\sigma^*_{u'i^*}$ contains an element $k^\wedge \leqq k'$. Then, $k^\wedge \leqq k' < p'$, $k^\wedge \in \sigma^*_{u'i^*} = \sigma^*_{u'i'}$, and $p' \in \sigma'_{u'j'} = \sigma^*_{u'j'}$. It follows that $(i', j', u', k^\wedge, p')$ is a K^*-violation of σ^*; hence, $\{i', j'\} \in V^{\sigma^*}$;

(b) $j' = j^*$ and $i' \notin \{i^*, j^*\}$. The construction of σ' assures that $\sigma'_{u'i'} = \sigma^*_{u'i'}$ and that $\sigma^*_{u'j^*}$ contains an element $p^\wedge \geqq p'$. Then, $k' < p' \leqq p^\wedge$, $k' \in \sigma'_{u'i'} = \sigma^*_{u'i'}$, and $p^\wedge \in \sigma^*_{u'j^*} = \sigma^*_{u'j'}$. It follows that $(i', j', u', k', p^\wedge)$ is a K^*-violation of σ^*; hence, $\{i', j'\} \in V^{\sigma^*}$.

As a^{σ^*} is decreasing, $a^{\sigma'} = a^{\sigma^*} + \gamma(e^{i^*} - e^{j^*})$ for $\gamma > 0$, $i^* \in K^*$, and $j^* \in K^{*c}$, it follows that $a^{\sigma'}$ is K^*-decreasing. Hence, Lemma 3.1 implies that there exists a permutation π such that $a^{\sigma'\pi} = (a^{\sigma'})_{\downarrow} \in R^n_{\downarrow}$ and $\pi(K^*) = K^*$. We note that the set of K^*-violations of σ' and σ'^{π} do not coincide, but, as $\pi(K^*) = K^*$, (i, j, u, k, p) is a K^*-violation of σ' if and only if $(\pi(i), \pi(j), u, k, p)$ is a K^*-violation of σ'^{π}; in particular, the number of K^*-violations of σ' and σ'^{π} coincide, and the cardinality of $V^{\sigma'}_{K^*}$ and $V^{\sigma'\pi}_{K^*}$ are equal. As $V^{\sigma'}_{K^*}$ is a proper subset of $V^{\sigma^*}_{K^*}$, we conclude that $V^{\sigma'\pi}_{K^*}$ has a smaller cardinality than does $V^{\sigma^*}_{K^*}$. Furthermore, as $a^{\sigma'\pi}$ is obtained by coordinate-reshuffling of $a^{\sigma'}$, we have that $R(\sigma'^{\pi}) = R(\sigma')$, and therefore σ'^{π} is optimal. So, $a^{\sigma'\pi} \in R^n_{\downarrow}$, $V^{\sigma'\pi}_{K^*}$ has a smaller cardinality than does $V^{\sigma^*}_{K^*}$, and σ'^{π} is optimal, resulting in a contradiction to the selection of the σ^*. This contradiction proves that $V^{\sigma^*}_{K^*}$ is empty; i.e., σ^* is K^*-monotone.

It remains to show that we can achieve K-monotonicity simultaneously for all $\Rightarrow$-cuts K. We will sketch an (inductive) modification of the above construction that proves the general result. The key idea is to observe that it is possible to guarantee that the above construction will preserve K-monotonicity for $\Rightarrow$-cuts K.

Specifically, let $\mathscr{K}$ be a maximal set of $\Rightarrow$-cuts such that there exists an optimal multipartition σ that is K-monotone for every $K \in \mathscr{K}$ and where $a^{\sigma} \in R^n_{\downarrow}$. We will show that $\mathscr{K}$ is the set of all $\Rightarrow$-cuts. Suppose that this is not the case and that K^* is a $\Rightarrow$-cut that is not in $\mathscr{K}$. Let σ^*, i^*, j^*, σ', and π be defined in terms of K^* as in the earlier part of our proof, except that the permutation π is selected such that $\pi(K) = K$ for every $K \in \mathscr{K}$. The latter is possible because σ is K-monotone for each $K \in \mathscr{K}$, and therefore the permutation π selected by applying Lemma 3.1 has $\pi(K) = K$. We then have that σ'^{π} is an optimal multipartition that is K^*-monotone and for which $a^{\sigma'^{\pi}}$ is decreasing. Now, let $K \in \mathscr{K}$. Then K is a $\Rightarrow$-cut, and σ^* is K-monotone; in particular, σ^* has no K-violation, implying that either both i^* and j^* are in K or both are in K^c. This conclusion combines with the assertion that $\pi(K) = K$ to show that

$$\sigma^*_{uK} = \sigma'_{uK} = \sigma'^{\pi}_{uK} \quad \text{for all } u = 1, \dots, t.$$

As σ^* is K-monotone, we now conclude that so is σ'^{π}. Thus, σ'^{π} is an optimal multipartition that is K-monotone for every $\mathscr{K} \in K \cup \{K^*\}$ and for which $a^{\sigma'^{\pi}}$ is decreasing. This contradicts the maximality in the selection of $\mathscr{K}$, and thereby completes our proof.

Theorem 4.1 asserts the existence of optimal multipartitions having the property that, for certain pairs $(i, j) \in \{1, \dots, n\} \times \{1, \dots, n\}$, where $j \Rightarrow i$, we have that σ_{ui} gets uniformly lower elements than does σ_{uj}; i.e.,

$$\text{(4.4)} \qquad \text{if } u \in \{1, \dots, t\},\ k \in \sigma_{ui}, \text{ and } p \in \sigma_{uj}, \text{ then } k < p.$$

A natural direction for extending Theorem 4.1 is to study the existence of optimal multipartitions σ for which (4.4) is satisfied for a broader class of pairs (i, j) satisfying $j \Rightarrow i$, hopefully all such pairs. In fact, it is well known that such an extension is valid when the partial order is the linear order $>$, and in this case there exists an optimal multipartition for which (4.4) is satisfied for all pairs (i, j) for which $j \Rightarrow i$; see Corollary 4.4. However, the following example demonstrates that, in general, (4.4) cannot be achieved simultaneously for all pairs (i, j) satisfying $j \Rightarrow i$.

Example A (continued). Reconsider Example A discussed in § 3. It was shown that the multipartition σ^* with $\sigma^*_{11} = \{1, 2\}$, $\sigma^*_{12} = \{3, 7\}$, $\sigma^*_{13} = \{4, 6\}$, $\sigma^*_{14} = \{5, 8\}$ has $a^{\sigma^*} = (18, 10, 9, 6)$, $R(\sigma^*) = g(a^{\sigma^*}) = 1$ and is optimal. It was further shown that the only optimal multipartition σ for which a^{σ} is decreasing is $\sigma = \sigma^*$, so, up to permutation of the sets, σ^* is the only optimal multipartition. However, σ^* does not satisfy (4.4) for the pair $(3, 4)$, though $4 \Rightarrow 3$. Of course, there are additional optimal multipartitions for which a^{σ} is not decreasing, and the above discussion shows that these are necessarily obtained by applying a permutation to the sets of σ^*. We will next argue that no such optimal multipartition satisfies (4.4) for all pairs (i, j) for which $j \Rightarrow i$. Observe that under σ^* the only pairs $(i, j) \in \{1, 2, 3, 4\} \times \{1, 2, 3, 4\}$ for which (4.4) is satisfied are those with $i = 1$. Also, the pairs $(i, j) \in \{1, 2, 3, 4\} \times \{2, 3, 4\}$ satisfying $j \Rightarrow i$ are $(1, 2)$, $(1, 3)$, $(1, 4)$, and $(3, 4)$. Now, if σ is an optimal partition, then $\sigma = \sigma^{*\pi}$ for some permutation π; hence, if σ satisfies (4.4) for the pairs $(i, j) \in \{(1, 2), (1, 3), (1, 4)\}$, we have that $\pi(1) = 1$. However, in this case, $\sigma = \sigma^{*\pi}$ cannot satisfy (4.14) for the pair $(i, j) = (3, 4)$.

We next explain the idea underlying the above example. We first recall from Lemma 2.1 that, for all positive γ, $(18, 10, 9, 6) + \gamma(e^3 - e^4) \gg^{\Rightarrow} (18, 10, 9, 6)$. So, if we exchange 5 and 6 in σ^*_{u3} and σ^*_{u4}, we obtain the vector $(18, 10, 11, 4)$, which $\Rightarrow$-majorizes $(18, 10, 9, 6)$. However, the new vector is not decreasing. As the function g is symmetric, its definition is uniquely determined by the increasing permutation of the underlying vector. So, the objective value of the permuted multipartition is $g(18, 11, 10, 4)$. However, the $(18, 11, 10, 4)$ does not $\Rightarrow$-majorize $(18, 10, 9, 6)$; hence, the objective is reduced as $g(18, 10, 11, 4) = 0 < 1 = g(18, 10, 9, 6)$.

We observe that the proof of Theorem 4.1 can be used to show that, if $j \Rightarrow\Rightarrow i$, then there exists an optimal multipartition $\sigma = \{\sigma_{ux}\colon u = 1, \dots, t \text{ and } x = 1, \dots, n\}$, where, for each u, all the indices in σ_{ui} are smaller than any of the indices in σ_{uj}, but the proof cannot be used to show that this can be accomplished simultaneously over all pairs (i, j), where $j \Rightarrow\Rightarrow i$. We next determine sufficient conditions that "simultaneous monotonicity" can be achieved.

Call an integer $i \in \{1, \dots, n\}$, *well ordered with respect to* the partial order $\Rightarrow$, abbreviated $\Rightarrow$*-well ordered* if, for every integer $j \in \{1, \dots, n\} \setminus \{i\}$, either $j \Rightarrow i$ or $i \Rightarrow j$. As the partial order $\Rightarrow$ is consistent with the linear order $>$, we have that, if i is $\Rightarrow$-well ordered, $j \Rightarrow i$ whenever $j > i$ and $i \Rightarrow j$ whenever $i > j$.

The following lemma relates the property of being well ordered to $\Rightarrow$-cuts.

LEMMA 4.2. *An integer $i \in \{1, \dots, n\}$ is $\Rightarrow$-well ordered if and only if $\{1, \dots, i-1\}$ and $\{1, \dots, i\}$ are $\Rightarrow$-cuts.*

Proof. First, assume that i is well ordered. Then, for every $p, q \in \{1, \dots, n\}$ satisfying $p < i < q$, $q \Rightarrow i \Rightarrow p$. As we also have that $i \Rightarrow p$ for every $p < i$ and $q \Rightarrow i$ for every $q > i$, it immediately follows that both $\{1, \dots, i-1\}$ and $\{1, \dots, i\}$ are $\Rightarrow$-cuts. Alternatively, assume that both $\{1, \dots, i-1\}$ and $\{1, \dots, i\}$ are $\Rightarrow$-cuts. Then, for $j \in K \equiv \{1, \dots, i-1\}$, the fact that K is a $\Rightarrow$-cut and $i \in K^c$ implies that $i \Rightarrow j$. Also, for $j \in K' \equiv \{i+1, \dots, n\}$, the fact that K'^c is a $\Rightarrow$-cut that contains i and does not contain j implies that $j \Rightarrow i$.

The next result, obtained by combining Theorem 4.1 with the above lemma, shows that when the associated function is $\Rightarrow$-Schur convex on $R^n_\downarrow$, it is possible to determine the composition of sets corresponding to $\Rightarrow$-well-ordered integers in some optimal multipartitions.

THEOREM 4.3. *Consider a (t, n, m)-multipartitioning problem where the restriction of the partition function to $R^n_\downarrow$ is $\Rightarrow$-Schur convex. Then there exists an optimal multipartition σ such that a^σ is decreasing, and, for every $i \in \{1, \dots, n\}$ that is $\Rightarrow$-well ordered,*

(a) *For $j \in \{1, \dots, i-1\}$ and $u = 1, \dots, t$, σ_{uj} is a subset of $\{1, \dots, (i-1)m\}$;*

(b) *For $u = 1, \dots, t$, $\sigma_{ui} = \{(i-1)m + 1, \dots, im\}$; and*

(c) *For $j \in \{i+1, \dots, n\}$ and $u = 1, \dots, t$, σ_{uj} is a subset of $\{im + 1, \dots, nm\}$.*

Proof. Consider the optimal multipartition σ whose existence was established in Theorem 4.1. Select $i \in \{1, \dots, n\}$, which is $\Rightarrow$-well ordered, and let $u \in \{1, \dots, t\}$. By Lemma 4.2, $K \equiv \{1, \dots, i-1\}$ is a $\Rightarrow$-cut; thus, the construction of σ implies that σ is K-monotone. It follows that

$$(4.5) \qquad \sigma_{uK} = \{1, \dots, m|K|\} = \{1, \dots, im\},$$

establishing (a). Also, Lemma 4.3 shows that $K' \equiv \{1, \dots, i\}$ is a $\Rightarrow$-cut, and therefore σ is K'-monotone. Thus,

$$(4.6) \qquad \sigma_{uK'} = \{1, \dots, m|K'|\} = \{1, \dots, im\},$$

implying that

$$(4.7) \qquad \sigma_{uK'^c} = \{im + 1, \dots, nm\},$$

establishing (c). Finally, (4.6) and (4.7) combine to show that

$$(4.8) \qquad \sigma_{ui} = \sigma_{uK} \backslash \sigma_{uK'} = \sigma_{uK} = \{(i-1)n + 1, \dots, in\},$$

establishing (b).

COROLLARY 4.4. *Consider a (t, n, m)-multipartitioning problem where the restriction to the partition function to $R^n_\downarrow$ is Schur convex. Then the multipartition σ with*

$$(4.9) \qquad \sigma_{ui} = \{(i-1)m + 1, (i-1)m + 2, \dots, im\} \quad \textit{for all } u = 1, \dots, t$$

is optimal.

Proof. We first observe that every integer $i \in \{1, \ldots, n\}$ is $>$-well ordered. Thus the conclusion of our corollary follows directly from Theorem 4.3 and the fact that a function $g \colon R^n \to R$ is Schur convex if and only if its restriction of $R^n_{\downarrow}$ is $>$-Schur convex.

We note that Corollary 4.4 is well known, e.g., El-Neweihi, Proschan, and Sethuraman [1987] have a direct proof that relies on the definition of Schur convexity and the observation that the vector associated with the multipartition σ defined by (4.9) majorizes all vectors associated with the other multipartitions. Thus, the Schur convexity of the associated function g implies that, for every multipartition σ', $R(\sigma') = g(a^{\sigma'}) \geqq g(a^{\sigma}) = R(\sigma)$. This direct approach does not extend to prove Theorems 4.1 and 4.3.

5. An example contradicting the Du–Hwang conjecture. A multipartition σ for a (t, n, m)-multipartitioning problem is called *monotone* if, for all $u = 1, \ldots, t$ and i, $j = 1, \ldots, n$,

$$\text{if } j > i,\ k \in \sigma_{ui}, \text{ and } p \in \sigma_{uj}, \text{ then } k < p;$$

in particular, there is only one monotone multipartition σ^*, and for this partition

$$\sigma^*_{ui} = \{(i-1)m + 1, \ldots, im\} \quad \text{for all } u = 1, \ldots, t \text{ and } i = 1, \ldots, n.$$

Call a function $g \colon R^n \to R$ (t, n, m)*-partition-monotone* if, for every (t, n, m)-multipartitioning problem whose associated function is g, the monotone multipartition is optimal. It is well known, e.g., Corollary 4.4, that every Schur convex function on R^n is (t, n, m)-partition-monotone for all positive integers t and m. Here we construct an example of a continuous function that is $(1, n, m)$-partition-monotone for $t = 1$, arbitrary n, and $m = 2$, but is not Schur convex, thereby providing a counterexample to a conjecture of Du and Hwang [1990].

Example B. Let $n \geqq 4$ and consider the function $h \colon R^n_{\downarrow} \to R$ defined for $x \in R^n_{\downarrow}$ by

$$h(x_1, x_2, \ldots, x_n) = 3 \sum_{i=1}^{n-4} (x_i)^2 + (x_{n-3})^2 + 2x_{n-3}x_{n-2} + x_{n-3}x_{n-1}$$
$$+ 2x_{n-2}x_{n-1} + x_{n-2}x_n + 2x_{n-1}x_n.$$

Note that h is continuous on $R^n_{\downarrow}$ and continuously differentiable on int $(R^n_{\downarrow})$ with

$$h_{x_i}(x) = \begin{cases} 6x_i & \text{if } 1 \leqq i \leqq n-4, \\ 2x_{n-3} + 2x_{n-2} + x_{n-1} & \text{if } i = n-3, \\ 2x_{n-2} + 2x_{n-1} + x_n & \text{if } i = n-2, \\ x_{n-3} + 2x_{n-2} + 2x_n & \text{if } i = n-1, \\ x_{n-2} + 2x_{n-1} & \text{if } i = n. \end{cases}$$

Let the partial order $\Rightarrow$ be represented by Fig. 2. It is easy to verify that

$$h_{x_i}(x) \geqq h_{x_j}(x) \quad \text{for all } x \in \operatorname{int}(R^n_{\downarrow}) \text{ and } i, j = 1, \ldots, n \text{ satisfying } j \Rightarrow i.$$

Thus, Corollary 2.5 implies that the function h is $\Rightarrow$-Schur convex on $R^n_{\downarrow}$. Furthermore, the inequality $h_{x_{n-2}}(x) \geqq h_{x_{n-1}}(x)$ does not hold for all x in $R^n_{\downarrow}$, e.g., it fails when $x_{n-3} = 6$, $x_{n-2} = 5$, $x_{n-1} = 2$, and $x_n = 1$, in which case $h_{x_{n-1}}(x) - h_{x_{n-2}}(x) = 1$. Thus, Corollary 3.5 also implies that the function h is not $>$-Schur convex. In particular, we conclude that the symmetric extension $h^{\wedge}$ of h to R^n is not Schur convex.

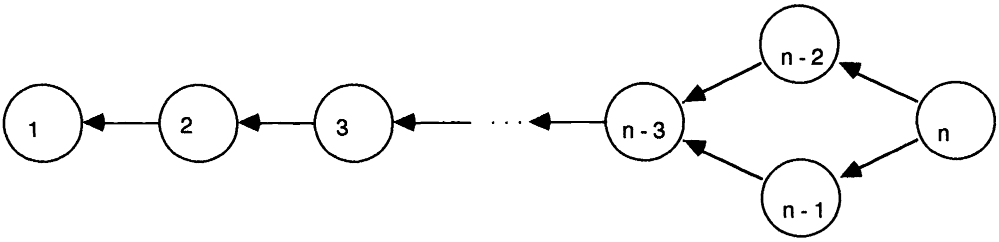


FIG. 2

We will next show that $h^\wedge$ is $(1, n, 2)$-partition-monotone for all $n \geqq 4$. Consider a $(1, n, 2)$-multipartitioning problem with data given by $\{a_k^1 \colon k = 1, \dots, 2n\}$, where

$$a_1^1 \geqq a_2^1 \geqq \cdots \geqq a_{2n}^1$$

and associated function $h^\wedge$. To simplify notation, we will suppress the superscript 1 of this data and refer to $a_1, a_2, \dots, a_{2n}$.

We observe that $1, 2, \dots, n-3$ and n are all $\Rightarrow$-well-ordered integers. Hence, Theorem 4.3 implies that there exists an optimal partition σ with $a^{\sigma'}$ decreasing and

$$\sigma_{1i} = \begin{cases} \{2i-1, 2i\} & \text{if } i = 1, \dots, n-3, \\ \{2n-1, 2n\} & \text{if } i = n. \end{cases}$$

The only elements whose assignment under σ is not uniquely determined are a_{2n-5}, a_{2n-4}, a_{2n-3}, and a_{2n-2}, which are assigned to σ_{n-2} and σ_{n-1}, respectively. Now assume that σ is not monotone and let σ^* be the unique monotone partition. We will show that $R(\sigma^*) \geqq R(\sigma)$. This conclusion is trite if $a_{2n-5} = a_{2n-4} = a_{2n-3} = a_{2n-2}$. Hence, assume that $a_{2n-5} < a_{2n-2}$.

Let $\sigma_{n-2} = \{u, v\}$ and $\sigma_{n-1} = \{w, t\}$. As a^σ is decreasing, we have that $u + v \geqq w + t$. Furthermore, as $a_{2n-5} < a_{2n-2}$ and σ is not monotone, $\{u, v\} = \sigma_{n-2} \neq \{a_{2n-5}, a_{2n-4}\}$. It follows that $\{u, v\}$ and $\{w, t\}$; each contains exactly one element from $\{a_{2n-5}, a_{2n-4}\}$ and exactly one element from $\{a_{2n-3}, a_{2n-2}\}$, respectively. By interchanging the roles of u and v and/or w and t, we may assume that $\{a_{2n-5}, a_{2n-4}\} = \{u, w\} = \sigma_{n-2}$ and $\{a_{2n-3}, a_{2n-2}\} = \{v, t\} = \sigma_{n-1}$. Then

$$\begin{aligned} R(\sigma^*) - R(\sigma) = {} & [2(a_{2n-7} + a_{2n-6})(u+v) + (a_{2n-7} + a_{2n-6})(w+t) \\ & + 2(u+v)(w+t) + (u+v)(a_{2n-1} + a_{2n}) + 2(w+t)(a_{2n-1} + a_{2n})] \\ & - [2(a_{2n-7} + a_{2n-6})(u+w) + (a_{2n-7} + a_{2n-6})(v+t) \\ & + 2(u+w)(v+t) + (u+w)(a_{2n-1} + a_{2n}) + 2(v+t)(a_{2n-1} + a_{2n})] \\ = {} & (v-w)[a_{2n-7} + a_{2n-6} - 2u + 2t - a_{2n-1} - a_{2n}]. \end{aligned}$$

As $a_{2n-7} + a_{2n-6} \geqq 2u$ and $2t \geqq a_{2n-1} + a_{2n}$, we conclude that $R(\sigma^*) \geqq R(\sigma)$. Hence, σ^* is optimal, thereby establishing that the function $h^\wedge$ is $(1, n, 2)$-partition-monotone.

We next show that the example we constructed was minimal in n; i.e., we show that $(1, n, m)$-partition-monotonicity for a continuous function implies Schur convexity when $n < 4$. This is trivial when $n = 1$; hence, we consider only the cases where $n = 2$ and $n = 3$.

Let $n = 2$. Suppose that g is $(1, 2, 2)$-partition-monotone, and we will show that g is Schur convex, or, equivalently, that the restriction of g to $R_\downarrow^2$ is $>$-Schur convex. Let $b, c \in R_\downarrow^2$, where $b \gg^> c$. Set

$$a_1^1 \equiv a_2^1 \equiv b_1/2, \quad a_3^1 \equiv c_1 - b_1/2, \quad \text{and} \quad a_4^1 = c_2 - b_1/2.$$

As $b_1 \geqq c_1 \geqq c_2$, it immediately follows that $a_1^1 \geqq a_2^1 \geqq a_3^1 \geqq a_4^1$. Thus, the $(1, 2, 2)$-partition-monotonicity of g implies that, if the a_k^1's are data for a multipartitioning problem whose associated function is g, then the (unique) monotone partition has a higher objective value than any other partition. Thus,

$$\begin{aligned} g(b) &= g(a_1^1 + a_2^1, a_3^1 + a_4^1) = R(\{1, 2\}, \{3, 4\}) \\ &\geqq R(\{1, 3\}, \{2, 4\}) = g(a_1^1 + a_3^1, a_2^1 + a_4^1) = g(c). \end{aligned}$$

So, the restriction of g to $R_\downarrow^2$ is $>$-Schur convex, and therefore g is Schur convex.

We next consider the case where $n = 3$. Suppose that g is a $(1, 3, 2)$-partition-monotone, and we will demonstrate that g is Schur convex, or, equivalently, that the restriction of g to $R_\downarrow^3$ is $>$-Schur convex. Let $b, c \in R_\downarrow^3$, where $b \gg^> c$, and we will show that $g(b) \geqq g(c)$. By condition (a4) of Corollary 2.5, it suffices to consider the case where $b_1 > b_2 > b_3$, $c_1 > c_2 > c_3$, and either $b_1 = c_1$ or $b_3 = c_3$.

First, assume that $b_3 = c_3$, in which case, $b_1 + b_2 = c_1 + c_2$. Let

$$\begin{aligned} &a_1^1 \equiv c_1 - b_2/2, \qquad a_2^1 \equiv b_1 - c_1 + b_2/2, \\ &a_3^1 \equiv a_4^1 \equiv b_2/2, \quad \text{and} \quad a_5^1 \equiv a_6^1 \equiv b_3/2. \end{aligned}$$

As $2c_1 \geqq c_1 + c_2 = b_1 + b_2$, $b_1 \geqq c_1$, and $b_2 \geqq b_3$, it immediately follows that $a_1^1 \geqq a_2^1 \geqq a_3^1 \geqq a_4^1 \geqq a_5^1 \geqq a_6^1$. Thus, the $(1, 3, 2)$-partition-monotonicity of g implies that, if the a_k^1's are data for a multipartitioning problem whose associated function is g, then the (unique) monotone partition has a higher objective value than any other partition. As $c_2 = b_1 + b_2 - c_1 = a_5^1 + a_6^1$, we conclude that

$$\begin{aligned} g(b) &= g(a_1^1 + a_2^1, a_3^1 + a_4^1, a_5^1 + a_6^1) = R(\{1, 2\}, \{3, 4\}, \{5, 6\}) \\ &\geqq R(\{1, 3\}, \{2, 4\}, \{5, 6\}) = g(a_1^1 + a_3^1, a_2^1 + a_4^1, a_5^1 + a_6^1) = g(c). \end{aligned}$$

Alternatively, assume that $b_1 = c_1$, in which case, $b_2 + b_3 = c_2 + c_3$. Let

$$\begin{aligned} &a_1^1 \equiv a_2^1 \equiv b_1/2, \qquad a_3^1 \equiv a_4^1 \equiv b_2/2, \\ &a_5^1 = c_2 - b_2/2, \quad \text{and} \quad a_6^1 \equiv c_3 - b_2/2. \end{aligned}$$

Again, it easily verified that $a_1^1 \geqq a_2^1 \geqq a_3^1 \geqq a_4^1 \geqq a_5^1 \geqq a_6^1$, and that, by the $(1, 3, 2)$-partition-monotonicity of g,

$$\begin{aligned} g(b) &= g(a_1^1 + a_2^1, a_3^1 + a_4^1, a_5^1 + a_6^1) = R(\{1, 2\}, \{3, 4\}, \{5, 6\}) \\ &\geqq R(\{1, 2\}, \{3, 5\}, \{4, 6\}) = g(a_1^1 + a_2^1, a_3^1 + a_5^1, a_4^1 + a_6^1) = g(c). \end{aligned}$$

We have established that the restriction of g to $R_\downarrow^n$ is $>$-Schur convex, and therefore g is Schur convex.

6. Conclusions. We identified monotonicity properties of optimal multipartitions for (t, n, m)-multipartitioning problems whose partition functions are Schur convex with respect to partial orders. This was accomplished by using the recent theory of majorization and Schur convexity with respect to partial orders. We then used the results to construct a class of counterexamples to a conjecture of Du and Hwang [1990], which asserts that Schur convex functions can be characterized by the existence of a monotone optimal partitions for $(1, n, m)$-multipartitioning problems. We hope that the methods we developed in this paper can be useful to more general assembly problems, for example, for problems where the number of parts for each type in a module does not have to be constant.

REFERENCES

B. C. ARNOLD (1987), *Majorization and the Lorenz Order: A Brief Introduction*, Springer-Verlag, Berlin.

C. DERMAN, G. J. LIEBERMAN, AND S. M. ROSS (1972), *On optimal assembly of systems*, Naval Res. Logist. Quart., 19, pp. 564–574.

D. Z. DU AND F. K. HWANG (1990), *Optimal assembly of an s-stage k-out-of-n system*, SIAM J. Discrete Math., 3, pp. 349–354.

E. EL-NEWEIHI, F. PROSCHAN, AND J. SETHURAMAN (1987), *Optimal allocation assembly of systems using Schur functions and majorization*, Naval Res. Logist. Quart., 34, pp. 705–712.

M. HOLLANDER, F. PROSCHAN, AND J. SETHURAMAN (1977), *Functions decreasing in transposition and their applications in ranking problem*, Ann. Statist., 5, pp. 722–733.

F. K. HWANG, (1979), *Majorization on a partially ordered set*, in Proc. of the American Mathematical Society, 76, pp. 199–204.

F. K. HWANG AND U. G. ROTHBLUM (1993a), *Optimality of monotone assemblies for coherent systems composed of series modules*, Oper. Res., to appear.

——— (1993b), *Majorization and Schur convexity with respect to partial orders*, Mathematics Oper. Res., to appear.

K. W. LIH (1982), *Majorization on finite partially ordered sets*, SIAM J. Algebraic and Discrete Mathematics, 3, pp. 495–503.

D. M. MALON (1990), *When is greedy module assembly optimal*, Naval Res. Logist. Quart., 37, pp. 847–854.

A. W. Marshall and I. Olkin (1979), *Inequalities, Theory of Majorization and Its Applications*, Academic Press, New York.

U. G. ROTHBLUM (1993), *Using a characterization of feasibility of transportation problems to establish the pairwise connectedness of R^n with respect to partial orders*, Bull. Inst. Math. Acad. Sinica, to appear.

SIAM J. DISC. MATH.
Vol. 6, No. 4, pp. 548-564, November 1993

003

INTERACTIVE COMMUNICATION OF BALANCED DISTRIBUTIONS AND OF CORRELATED FILES*

ALON ORLITSKY†

Abstract. (X, Y) is a pair of random variables distributed over a support set S. Person P_X knows X, person P_Y knows Y, and both know S. Using a predetermined protocol, they exchange binary messages for P_Y to learn X. P_X may or may not learn Y. The m-message complexity $\hat{C}_m$ is the number of information bits that must be transmitted (by both persons) in the worst case if only m messages are allowed. $\hat{C}_\infty$ is the number of bits required when there is no restriction on the number of messages exchanged.

A natural class of random pairs is considered. $\hat{\mu}$ is the maximum number of X values possible with a given Y value. $\hat{\eta}$ is the maximum number of Y values possible with a given X value. The random pair (X, Y) is *balanced* if $\hat{\mu} = \hat{\eta}$. The following hold for *all* balanced random pairs. One-way communication requires at most twice the minimum number of bits: $\hat{C}_1 \leqq 2\hat{C}_\infty + 1$. This bound is almost tight: For every α, there is a balanced random pair for which $\hat{C}_1 \geqq 2\hat{C}_\infty - 6 \geqq \alpha$. Three-message communication is asymptotically optimum, $\hat{C}_3 \leqq \hat{C}_\infty + 3 \log \hat{C}_\infty + 11$. More importantly, the number of bits required is only negligibly larger than the number needed when P_X knows Y in advance, $\hat{C}_\infty \leqq \hat{C}_3 \leqq \log \hat{\mu} + 3 \log \log \hat{\mu} + 11$.

These results are applied to the following *correlated files* problem. X and Y are binary strings (files) within a small edit distance from each other. P_X knows X, while P_Y knows Y and wants to learn X. The above results imply efficient three-message protocols for conveying X to P_Y. Efficient one-way protocols are provided for certain restricted cases and their possible generalizations are discussed.

Key words. communication complexity, data compression, interactive communication

AMS subject classifications. 94A15, 94A29, 94A99

1. Introduction. The Introduction is partitioned into three parts. Section 1.1 provides the necessary background. Section 1.2 describes the problem investigated and some of its possible applications. Section 1.3 reviews the results obtained.

1.1. Definitions and previous results. Consider two *communicators*: an *informant* P_X having a random variable X and a *recipient* P_Y having a random variable Y. The *random pair* (X, Y) is distributed according to a probability distribution that is known to both communicators. P_X and P_Y want the recipient P_Y to learn X with zero probability of error. The informant P_X may or may not learn Y. How many bits must be transmitted?

This *interactive communication* problem can be viewed as a variation on communication complexity [1], [2]. The function computed by P_X and P_Y is trivial: $f(X, Y) = X$, and difficulty arises because the inputs X and Y are correlated.

To further specify the problem, we assume that the communicators alternate in transmitting *messages*, finite sequences of bits. Messages are transmitted over an error-free channel and are determined by an agreed-upon, deterministic protocol. For every *input*—a possible value assignment for X and Y—the protocol determines a finite sequence of transmitted messages. The protocol is *m-message* if, for all inputs, the number of messages transmitted is at most m.

The (*worst-case*) *complexity* of a protocol is the number of bits it requires both communicators to transmit, maximized over all inputs. $\hat{C}_m$, the *m-message complexity*[1] of (X, Y), is the minimum complexity of an m-message protocol for (X, Y). For example,

* Received by the editors April 29, 1991; accepted for publication (in revised form) October 16, 1992. A preliminary version of this paper appeared in the 32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, October 1991.

† AT&T Bell Laboratories, 600 Mountain Avenue, Murray Hill, New Jersey 07974.

[1] The pair (X, Y) is implicit in the notation and will be implied by the context.

$\hat{C}_1$, the *one-way complexity* of (X, Y), is the number of bits required in the worst case when P_Y cannot transmit to P_X, and $\hat{C}_2$ is the number of bits required in the worst case when at most two messages are permitted: P_Y transmits a message reflecting Y, then P_X responds with a message from which P_Y must infer X. Since empty messages are allowed, $\hat{C}_m$ is a nonincreasing function of m bounded below by 0. We can therefore define $\hat{C}_\infty$, the *unbounded-message complexity* of (X, Y), to be the limit of $\hat{C}_m$ as $m \to \infty$. It is the *minimum* number of bits that must be transmitted for P_Y to know X, even if no restrictions are placed on the number of messages exchanged. Clearly,

$$\hat{C}_1 \geqq \hat{C}_2 \geqq \hat{C}_3 \geqq \dots \geqq \hat{C}_\infty.$$

The following example and results, taken from [3], relate these complexity measures. For other aspects of interactive communication, see [4]-[6].

Example 1. A league has t teams. P_Y knows two teams that played in a game, and P_X knows the team that won the game. They communicate for P_Y to learn the winning team.

If only one message is allowed, necessarily from P_X to P_Y, it must be based solely on the winner (for that is all P_X knows). If the message transmitted when team i wins is the same as (or a prefix of) the message transmitted when team j wins, then, in the event of a match between teams i and j, P_Y cannot discern the winner (or when the message ends). Therefore, there must be t different, prefix-free messages, and at least one of them must be of length $\geqq \lceil \log t \rceil$. This bound is clearly achievable; hence, $\hat{C}_1 = \lceil \log t \rceil$. If two messages are allowed, P_Y considers the binary representations of the two teams that played and transmits $\lceil \log \log t \rceil$ bits describing the location of the first bit where they differ. P_X responds by transmitting a single bit describing the bit value of the winning team in that location. Therefore, $\hat{C}_2 \leqq \lceil \log \log t \rceil + 1$. It can be shown that, for this example,

$$\hat{C}_2 = \dots = \hat{C}_\infty = \lceil \log \log t \rceil + 1.$$

The example shows that for some random pairs, one-message complexity is exponentially higher than the minimum $\hat{C}_1 = 2^{\hat{C}_\infty - 1}$. Yet [3] shows that two messages always suffice to reduce communication to almost the minimum: For all random pairs,

$$\hat{C}_2 \leqq 4\hat{C}_\infty + 3.$$

This contrasts with communication complexity, where a succession of papers [7]-[9] showed that, for every m, there is a function whose m-message complexity is almost exponentially higher than its $(m+1)$-message complexity.

It is not known whether there is an m such that m-messages are asymptotically optimum, namely, for all random pairs

$$\hat{C}_m \leqq \hat{C}_\infty + o(\hat{C}_\infty).$$

In this paper, we consider a natural class of random pairs for which stronger statements hold true.

1.2. Balanced pairs and correlated files. Let (X, Y) be a random pair. Its *support set* is the set S of possible inputs. The support set is of interest as it determines the m-message complexity $\hat{C}_m$ for all m. This evident property of worst-case complexities is formally proved in [3].

P_Y's *ambiguity* when he has the value y is

$$(1) \qquad \mu(y) \overset{\text{def}}{=} |\{x : (x, y) \in S\}|,$$

the number of possible X values when $Y = y$. P_Y's *maximum ambiguity* is

$$(2) \qquad \hat{\mu} \overset{\text{def}}{=} \max_{y} \{\mu(y)\},$$

the maximum number of X values possible with any given Y value. P_X's ambiguity $\eta(x)$ when he has the value x, and his maximum ambiguity $\hat{\eta}$, are similarly defined. In the league problem of Example 1, $\hat{\mu} = 2$ as, for every game known to P_Y, there are two possible winners known to P_X. Similarly, $\hat{\eta}$ in that case is $t - 1$, corresponding to the number of possible losing teams.

We consider the class of *balanced* random pairs, pairs satisfying $\hat{\mu} = \hat{\eta}$. Balanced pairs arise naturally whenever there is no distinction between the two communicators or when X and Y are known to be within some "distance" from each other. For example,[2]

1) X and Y, inaccurate measurements of the same quantity, are integers within a bounded absolute difference from each other, and

2) X and Y, obtained from a noisy binary transmission or from a faulty memory, are n-bit strings within a bounded Hamming distance from each other.

Of these and other examples of balanced pairs, the following *correlated files*, or *edit-distance problem*, shows the most promise of being practically useful. The *edit distance* between two binary strings X and Y is the minimum number of deletions and insertions to X needed to derive Y. For example, the edit distance between the empty string and any n-bit string is n, and the edit distance between 01010 and 10101 is 2. In the correlated-files problem, X and Y are binary strings within a small edit distance from each other. P_X knows X, while P_Y knows Y and wants to learn X.

This problem can arise in many situations: (1) P_X and P_Y write a joint book, and each updates his version individually; (2) X is the new digital image taken by a satellite P_X, and Y is the previous frame, known to P_Y (successive images are likely to be within a small edit distance); (3) X and Y are different versions of the same program or file; (4) X and Y were received from the same binary transmission with erroneous insertions, deletions, and reversal of bits.

In all those cases, the edit distance between X and Y is much smaller than the number of bits in each. We are looking for a way to communicate X to P_Y without transmitting all of it. Of course, in cases (1) and (2), if P_X keeps the original versions of the file (or image), he can efficiently transmit the locations of the insertions/deletions. In cases (3) and (4), however, there is no such reference sequence. Surprisingly, there is almost no difference between the number of bits required in the two cases. We show that, even when P_X knows only X (as we assume), X can be communicated to P_Y using only negligibly more bits than the number needed if P_X knew Y in advance.

The protocols achieving this near-minimum number of bits are interactive and require P_X and P_Y to exchange three messages. For general one-way protocols, we can only prove that they require twice as many bits as needed if P_X knows Y. However, efficient one-way protocols for this problem would have the following additional applications: (1) simultaneous updates of many *different* files (without the need to respond

[2] The pairs below are, in fact, *symmetric*: $(x, y) \in S$ if and only if $(y, x) \in S$. Clearly, every symmetric pair is also balanced. Our results hold for all balanced pairs and hence are presented that way.

individually to each one); (2) transmission of images to many recipients, each with a different (possibly erroneous) prior image; (3) efficient backup of files without keeping track of individual edit operations; (4) compression of long related sequences without keeping a reference sequence.

For these reasons, we consider the possibility of efficient one-way protocols for the correlated-files problem. Unfortunately, we can prove only partial results, and the main question remains unsolved.

1.3. Results. For general distributions, one-way communication may require exponentially more bits than the minimum necessary (e.g., $\hat{C}_1 = \lceil \log t \rceil$ versus $\hat{C}_\infty = \lceil \log \log t \rceil + 1$ for the league problem with t teams). Yet, Corollary 1 in § 2 shows that, for all balanced random pairs, one-way communication requires at most twice the minimum necessary,

$$\hat{C}_1 \leqq 2\hat{C}_\infty + 1.$$

This bound is almost tight. Lemma 2 shows that, for all $\alpha \geqq 0$, there is a balanced random pair such that

$$\hat{C}_1 \geqq 2\hat{C}_\infty - 6 \geqq \alpha.$$

In § 3 we show that three-messages communication is asymptotically optimum. More importantly, we prove that, although the informant P_X does not know Y, the number of bits needed to convey X to P_Y is only negligibly larger than would be required if P_X knew Y in advance.

Specifically, (1) and (2) defined P_Y's ambiguity $\mu(y)$ when he has the value y and his maximum ambiguity $\hat{\mu}$. Clearly, at least $\lceil \log \mu(y) \rceil$ bits must be transmitted in the worst case when P_Y's value is y. Hence,

$$\hat{C}_\infty \geqq \lceil \log \hat{\mu} \rceil. \tag{3}$$

Had P_X known Y in advance, this bound would be tight, $\hat{C}^* = \lceil \log \hat{\mu} \rceil$, where $\hat{C}^*$ denotes the number of bits needed *if* P_X knows Y in advance (note that in this case interaction cannot help; hence we omitted the subscript representing the number of messages). However, P_X does not know Y; hence $\lceil \log \hat{\mu} \rceil$ bits cannot always be achieved. In the league problem, for example, the maximum ambiguity $\hat{\mu}$ is two, and if P_X knew Y (the game) he would need to transmit only one bit (say, whether the winning team is lexicographically first). Yet, P_X does not know the game, and we saw that many more than $\lceil \log \hat{\mu} \rceil = 1$ bits must be transmitted: $\hat{C}_\infty = \lceil \log \log t \rceil + 1$ bits.

However, § 3 shows that, *for all balanced pairs, there is almost no increase in communication due to* P_X *not knowing* Y,

$$\hat{C}_3 \leqq \log \hat{\mu} + 3 \log \log \hat{\mu} + 11. \tag{4}$$

Furthermore, for general distributions it is not known whether there is an m such that m messages are asymptotically optimum. As a corollary of the last result, we get that, for all balanced distributions, three messages are asymptotically optimum,

$$\hat{C}_3 \leqq \hat{C}_\infty + 3 \log \hat{C}_\infty + 11.$$

The bound of inequality (4) is refined in § 4 for balanced distributions where P_Y's ambiguity $\mu(y)$ varies widely with y. We show (see Theorem 2 for the precise formulation) that, for all random pairs, there is a four-message protocol that, for all y's with a given $\mu(y)$, requires an average of at most $\log \mu(y) + 4 \log \log \mu(y)$ transmitted bits.

In § 5.1 we apply these results to the correlated-files (edit-distance) problem. We derive three-message protocols that are bitwise-efficient both asymptotically and for moderately-sized files. We show that similar protocols can be constructed also for more realistic edit models (e.g., where an edit operation can move a segment).

For one-way communication, the general results imply only protocols that require at most twice the number of bits necessary when P_X knows Y. In view of their significance, we consider asymptotically-optimal, one-way protocols for the correlated files problem in § 5.2. Unfortunately, we cannot prove that they generally exist. Instead, we (1) reduce the problem of conveying edited files to that of conveying files with insertions alone; this simplifies the task of designing protocols and analyzing them; (2) describe an optimal one-way protocol for a single insertion/deletion, showing that for the case $\hat{C}_1 = \ldots = \hat{C}_\infty = \lceil \log (n+2) \rceil$, where n is the length of the string Y; (3) analyze a possible reduction of insertion protocols to protocols identifying inversions—a problem for which efficient protocols exist (cf. Example 4). We then use this approach to construct efficient protocols for a very restricted case of insertions and deletions.

2. One-way complexity is at most twice the minimum. For general pairs, one-way communication may require exponentially more bits than the minimum necessary (e.g., $\lceil \log t \rceil$ versus $\lceil \log \log t \rceil + 1$ for the league problem). However, for all balanced pairs, one message requires at most twice the minimum number of bits. This will become apparent once we delineate some preliminary definitions and results.

Let (X, Y) be a random pair with support set S. The *support set* of X is the set

$$S_X \overset{\text{def}}{=} \{x : (x, y) \in S \text{ for some } y\}$$

of possible values of X. The *support set* S_Y of Y is similarly defined. Instrumental in determining $\hat{C}_1$ is G, the *characteristic hypergraph* of (X, Y). Its vertex set is S_X, and, for every $y \in S_Y$, it contains the hyperedge

$$E(y) \overset{\text{def}}{=} \{x : (x, y) \in S\}. \tag{4}$$

The characteristic hypergraph is equivalent to a graph defined by Witsenhausen [10] who considered the one-message version of this problem. For the league problem with t teams, for example, G has t vertices, one corresponding to each (team) value of X. It has $\binom{t}{2}$ edges, one corresponding to each possible (game) value of Y. Each edge contains two vertices (the possible winning teams in the game). In other words, G is K_t, the complete graph on t vertices.

A coloring of a hypergraph G is an assignment of a color to every vertex of G such that no two vertices sharing a hyperedge are assigned the same color. The chromatic number $\chi(G)$ of G is the minimum number of colors required to color G. It can be shown that the following holds.

RESULT 1 (see Lemma 2 in [3]). *For all random pairs*, $\hat{C}_1 = \lceil \log \chi(G) \rceil$.

This proves again that, for the league problem with t teams, $\hat{C}_1 = \lceil \log \chi(K_t) \rceil = \lceil \log t \rceil$, as was shown in Example 1.

LEMMA 1. *For all random pairs*,

$$\hat{C}_1 \leqq \log \hat{\mu} + \log \hat{\eta} + 1.$$

Proof. According to Result 1, the one-way complexity of (X, Y) is determined by the chromatic number of the characteristic hypergraph: $\hat{C}_1 = \lceil \log \chi(G) \rceil$. Each vertex in G belongs to at most $\hat{\eta}$ edges, and each edge contains at most $\hat{\mu}$ vertices. Hence,

$$\chi(G) \leqq \hat{\eta} \cdot (\hat{\mu} - 1) + 1 \leqq \hat{\mu} \cdot \hat{\eta}. \qquad \square$$

COROLLARY 1. *For all balanced random pairs*,

$$\hat{C}_1 \leqq 2 \log \hat{\mu} + 1 \leqq 2\hat{C}_\infty + 1.$$

Proof. The proof is immediate from the previous lemma and (3). □

The next lemma proves this bound almost tight. Example 2 in § 3 provides a slightly weaker result for another, more elementary family of balanced random pairs.

LEMMA 2 (with Jeff Kahn). *For every $\alpha \geqq 0$, there is a balanced random pair satisfying*

$$\hat{C}_1 \geqq 2\hat{C}_\infty - 6 \geqq \alpha.$$

Proof. Let n be a prime power. Then there is a projective plane π of order n. π consists of $n^2 + n + 1$ points and the same number of lines. Every line is incident upon $n + 1$ points, and every point is on $n + 1$ lines. Every two points are connected by exactly one line, and every two lines intersect at exactly one point.

P_Y knows a line Y of π, and P_X knows a point X on Y. The pair (X, Y) is balanced as

$$\hat{\mu} = \hat{\eta} = n + 1.$$

Since every two points are connected by a line, in a one-message protocol, each of the $n^2 + n + 1$ points of π must be assigned a different message; therefore

$$\hat{C}_1 = \lceil \log (n^2 + n + 1) \rceil.$$

We now construct a two-message protocol showing that

$$\hat{C}_2 \leqq \lceil \log (n + 1) \rceil + 2.$$

Let p_1, p_2, and p_3 be distinct points of π, not all on a single line. For each of the three points, P_X and P_Y agree in advance on a $\lceil \log (n + 1) \rceil$-bit encoding of the $n + 1$ lines incident upon it.

When P_Y is given a line Y and P_X is given a point X on Y, P_Y transmits the index i of a point among p_1, p_2, and p_3 that is not on Y (there must be one by the choice of the three points). P_X responds with the encoding of the (unique) line incident upon p_i and connecting it to X.

P_Y now knows two lines, both incident upon X. As every two lines intersect at a unique point, P_Y can determine X. □

We have proved that a single message may require up to twice the minimum number of bits. The next section shows that three messages require at most negligibly more bits than the minimum necessary.

3. Three-message communication is asymptotically optimum. In § 1.3 we noted that, for general distributions, (1) it is not known whether there is an m such that m-message communication is asymptotically optimum, and (2) there may be a large discrepancy between the number $\lceil \log \hat{\mu} \rceil$ of bits needed when P_X knows Y in advance and $\hat{C}_\infty$, required when P_X does not know Y.

Balanced pairs are different. We show that three messages require only negligibly more than $\lceil \log \hat{\mu} \rceil$ bits. In particular, three-message communication is asymptotically optimum.

First, we describe a two-message protocol that improves on Lemma 1 for random pairs with $\hat{\mu} \ll \hat{\eta}$. Recall that the support set S of a random pair (X, Y) is the set of all possible inputs, that the support set of X is the set S_X of all possible values of X, and that S_Y is similarly defined. P_Y's *ambiguity set* when his random variable attains the value $y \in S_Y$ is the set $E(y)$, defined in (4), of possible X values in that case. Denote the collection of P_Y's ambiguity sets by

$$\mathcal{E} \overset{\text{def}}{=} \{E(y) : y \in S_Y\}.$$

A collection of functions, each defined over S_X, *perfectly hashes* $\mathscr{E}$ if, for every $y \in S_Y$, there is a function in the collection that is one-to-one over (or *hashes*) $E(y)$. For various results on, and applications of, perfect-hash functions, see [11]–[14].

Let b be an integer and let $\mathscr{F}$ be a collection of functions from S_X to $\{1, \ldots, b\}$ that perfectly hashes $\mathscr{E}$. We show that

$$(5) \qquad \hat{C}_2 \leqq \lceil \log |\mathscr{F}| \rceil + \lceil \log b \rceil.$$

P_X and P_Y agree in advance on a $\lceil \log |\mathscr{F}| \rceil$-bit encoding of the functions in $\mathscr{F}$ and on a $\lceil \log b \rceil$-bit encoding of $\{1, \ldots, b\}$. When P_X is given X and P_Y is given Y, they execute the following protocol. P_Y finds a function $f_Y \in \mathscr{F}$ that perfectly hashes $E(Y)$. Using $\lceil \log |\mathscr{F}| \rceil$ bits, he transmits the encoding of f_Y to P_X. Now P_X knows X and f_Y. He computes $f_Y(X)$ and transmits it to P_Y using $\lceil \log b \rceil$ bits. Since f_Y is one-to-one over $E(Y)$, P_Y can recover X from $f_Y(X)$.

LEMMA 3. *For all nontrivial*[3] (X, Y) *pairs*,

$$\hat{C}_2 \leqq 2 \log \hat{\mu} + \log \log \max \{\hat{\mu}, \hat{\eta}\} + 4.$$

Proof. In view of the above, we show the existence of a small collection $\mathscr{F}$ of functions with a small range $\{1, \ldots, b\}$ that perfectly hashes $\mathscr{E}$. Let $b \overset{\text{def}}{=} \hat{\mu}^2$. Pick a random function $F: S_X \rightarrow \{1, \ldots, b\}$ by assigning, uniformly at random, a value in $\{1, \ldots, b\}$ to each element of S_X. For all $y \in S_Y$, the probability that $E(y)$ is hashed by F is

$$\frac{b^{\underline{\mu(y)}}}{b^{\mu(y)}} \geqq \frac{b^{\underline{\hat{\mu}}}}{b^{\hat{\mu}}} \geqq \left(1 - \frac{1}{\hat{\mu}}\right)^{\hat{\mu}} \geqq \frac{1}{4},$$

where $x^{\underline{i}} \overset{\text{def}}{=} x \cdot (x-1) \cdot \ldots \cdot (x - i + 1)$ denotes the ith falling power of x.

Independently pick m such random functions. Let $B(y)$ be the "bad" event that $E(y)$ is not hashed by any of these functions. For each y,

$$\Pr(B(y)) \leqq (\tfrac{3}{4})^m.$$

Furthermore, each $E(y)$ intersects $E(y')$ for at most $\hat{\mu} \cdot (\hat{\eta} - 1)$ values of $y' \in S_Y$. Therefore, the event $B(y)$ is independent of the events $B(y')$ for all but at most $\hat{\mu} \cdot (\hat{\eta} - 1)$ values of $y' \in S_Y$. By the Lovász local lemma (e.g., Chapter 8 in [15]), if

$$(6) \qquad 4 \cdot (\tfrac{3}{4})^m \hat{\mu} \cdot (\hat{\eta} - 1) < 1,$$

then $\Pr(\bigcup_{y \in S_Y} B(y)) < 1$; namely, there is a collection of m functions that perfectly hashes $\mathscr{E}$. To conclude, we note that (6) holds for

$$m = \left\lceil \frac{\log \hat{\mu} + \log \hat{\eta} + 2}{\log \frac{4}{3}} \right\rceil \leqq 8 \log \max \{\hat{\mu}, \hat{\eta}\}. \qquad \square$$

If $\log \hat{\mu}$ is larger than $2 \log \log \max \{\hat{\mu}, \hat{\eta}\} + 7$, we can reduce the total number of transmitted bits. First, we use a $\log \hat{\mu}$-bit message to convert the pair (X, Y) into a pair (X', Y) with $\hat{\eta}' = \hat{\eta}$ and $\hat{\mu}' \leqq \log \max \{\hat{\mu}, \hat{\eta}\} + 4$ and then apply the lemma.[4]

THEOREM 1. *For all nontrivial random pairs*,

$$\hat{C}_3 \leqq \log \hat{\mu} + 3 \log \log \max \{\hat{\mu}, \hat{\eta}\} + 11.$$

[3] A pair (X, Y) is *trivial* if $\hat{\mu} = 1$. For trivial pairs, $\hat{C}_1 = \ldots = \hat{C}_\infty = 0$.

[4] We use $\hat{\eta}'$ and $\hat{\mu}'$ to denote P_X's and P_Y's maximum ambiguities for the pair (X', Y).

Proof. Pick a random function $F: S_X \rightarrow \{1, \ldots, \hat{\mu}\}$ by assigning, uniformly at random, a value in $\{1, \ldots, \hat{\mu}\}$ to each element of S_X. For an integer k, let $B(y)$ be the "bad" event that F assigns some value in $\{1, \ldots, \hat{\mu}\}$ to more than k elements in $E(y)$. For all y,

$$\Pr(B(y)) \leqq \hat{\mu} \cdot \binom{\mu(y)}{k} \cdot \frac{1}{\hat{\mu}^k} \leqq \frac{\hat{\mu}}{k!}.$$

Furthermore, the event $B(y)$ is independent of the events $B(y')$ for all but at most $\hat{\mu} \cdot (\hat{\eta} - 1)$ values of $y' \in S_Y$. By the Lovász local lemma, if

$$(7) \qquad \frac{4 \cdot \hat{\mu} \cdot (\hat{\eta} - 1) \cdot \hat{\mu}}{k!} < 1,$$

then there is a function $f: S_X \rightarrow \{1, \ldots, \hat{\mu}\}$ that is not "bad" for any y; namely, for all $y \in S_Y$ and all $i \in \{1, \ldots, \hat{\mu}\}$,

$$|\{x \in E(y): f(x) = i\}| \leqq k.$$

We are interested in a small k satisfying (7). Using the inequality $k! \geqq (k/e)^k$, it suffices to require that

$$(8) \qquad 2 \log \hat{\mu} + \log \hat{\eta} + 2 \leqq k \log \left(\frac{k}{e}\right).$$

It is easy to verify that

$$k = \lceil \log \max \{\hat{\mu}, \hat{\eta}\} + 4 \rceil$$

satisfies (8) for sufficiently large max $\{\hat{\mu}, \hat{\eta}\}$'s and satisfies (7) directly for smaller max $\{\hat{\mu}, \hat{\eta}\}$'s.

Therefore, if P_X's first message consists of $\lceil \log \hat{\mu} \rceil$ bits describing $f(X)$, then P_X and P_Y can restrict their attention to a random pair (X', Y), where X' is the random variable X restricted to the domain $\{x: f(x) = f(X)\}$. By choice of f, we have $\hat{\mu}' \leqq \lceil \log \max \{\hat{\mu}, \hat{\eta}\} + 4 \rceil$ and $\hat{\eta}' \leqq \hat{\eta}$. Thereafter, P_X and P_Y can use the two-message protocol of the previous lemma to convey X' to P_Y. The total number of bits transmitted is at most

$$\begin{aligned} &\lceil \log \hat{\mu} \rceil + 2 \log \lceil \log \max \{\hat{\mu}, \hat{\eta}\} + 4 \rceil + \log \log (\max \{\hat{\mu}, \hat{\eta}\} + 3) + 4 \\ &\leqq \log \hat{\mu} + 3 \log \log \max \{\hat{\mu}, \hat{\eta}\} + 11. \end{aligned}$$

With more care, the constant 11 can be reduced. □

For balanced pairs, $\hat{\mu} = \hat{\eta}$, and we readily obtain the following corollary.

COROLLARY 2. *For all balanced random pairs,*

$$\hat{C}_3 \leqq \log \hat{\mu} + 3 \log \log \hat{\mu} + 11.$$

The techniques used to prove Lemma 3 and Theorem 1 work extremely efficiently in the following example.

Example 2. Consider an n-by-n chessboard with two rooks in mutually capturing positions (i.e., in the same row, or the same column, or both (same position)). P_X knows the position X of one rook. P_Y knows the position Y of the other rook and wants to learn X. Formally,

$$S \overset{\text{def}}{=} \{((x_1, x_2), (y_1, y_2)) : 0 \leqq x_1, x_2, y_1, y_2 \leqq n - 1 \text{ and either } x_1 = y_1 \text{ or } x_2 = y_2\}.$$

Consider one-way complexity first. If P_X assigns the same message to two different rook positions (x_1, x_2) and (x'_1, x'_2), then, in the case that P_Y's rook position is (x_1, x'_2) (or (x'_1, x_2)), he will not know X. Hence $\hat{C}_1 = \lceil 2 \log n \rceil$. For every rook position Y known to P_Y, there are $2n - 1$ possible X positions; hence $\hat{\mu} = 2n - 1$. Therefore,

$$\hat{C}_\infty \geqq \lceil \log(2n-1) \rceil,$$

and this many bits are needed even if P_X knew Y in advance.

A simple three-message protocol based on the league problem shows that

$$\hat{C}_\infty \leqq \hat{C}_3 \leqq \lceil \log n \rceil + \lceil \log \log n \rceil + 1.$$

Using $\lceil \log n \rceil$ bits, P_X transmits the diagonal $d = (x_1 + x_2) \bmod n$ that his rook is on. Now, P_Y knows that (x_1, x_2) is either $(y_1, (d - y_1)_n)$ or $((d - y_2)_n, y_2)$. They use the two-message league-problem protocol to determine (x_1, x_2) while exchanging $\lceil \log \log n \rceil + 1$ bits. Thus, for this example,

$$\hat{C}_3 \leqq \log \hat{\mu} + \log \log \hat{\mu} \leqq \hat{C}_\infty + \log \hat{C}_\infty,$$

providing another example of a balanced random pair where $\hat{C}_1 \approx 2\hat{C}_\infty$.

We do not know of a two-message protocol for this example whose worst-case complexity is $\log n + o(\log n)$ bits. □

4. Distributions with varying ambiguities. For some balanced pairs, $\mu(y)$ varies widely with y. For these pairs, the bound provided by Corollary 2 may not be very useful. We describe an efficient refinement for such cases. Let $l_\phi(x, y)$ denote the number of bits transmitted under the protocol ϕ for the input (x, y). Consider a (not necessarily balanced) random pair (X, Y), possibly with widely varying $\mu(y)$. For every y, there are $\mu(y)$ possible x values. Although $l_\phi(x, y)$ may be low for some x's, when averaged over the x's, it must be at least $\log \mu(y)$. Namely, for all y,

$$\frac{1}{\mu(y)} \sum_{x:(x,y)\in S} l_\phi(x, y) \geqq \log \mu(y).$$

Note that this holds true even if P_X knows that $Y = y$ in advance. For $\mu \in \mathscr{X}$, let

$$S_Y(\mu) \overset{\text{def}}{=} \{y : \mu(y) = \mu\}$$

be the set of Y values with ambiguity μ. Clearly, for all feasible[5] μ,

$$\frac{1}{|S_Y(\mu)|} \sum_{y \in S_Y(\mu)} \frac{1}{\mu} \sum_{x:(x,y)\in S} l_\phi(x, y) \geqq \log \mu.$$

The next theorem says that this average lower bound can almost be met. Its proof resembles that of Theorem 3 in [6]; hence it is omitted.

THEOREM 2. *Let (X, Y) be a random pair. There is a four-message protocol ϕ such that, for all feasible μ,*

$$\frac{1}{|S_Y(\mu)|} \sum_{y \in S_Y(\mu)} \frac{1}{\mu} \sum_{x:(x,y)\in S} l_\phi(x, y) \leqq \log \mu + 4 \log \log \mu.$$

Two progressively stronger statements are false in general.

(i) For every (balanced) (X, Y) pair, there is a protocol ϕ such that, for all y,

$$\frac{1}{\mu(y)} \sum_{x:(x,y)\in S} l_\phi(x, y) \leqq \log \mu(y) + o(\log \mu(y)).$$

[5] μ is *feasible* if $S_Y(\mu)$ is nonempty.

(ii) For every (balanced) (X, Y) pair, there is a protocol ϕ such that, for all y,

$$\max \{l_\phi(x, y) : (x, y) \in S\} \leqq \log \mu(y) + o(\log \mu(y)).$$

To see some cases where these statements are false, consider a generalization of the league problem (Example 1) with t teams, μ of them playing in every game. (Again, P_Y knows the game and wants to learn the winner, known to P_X.) If statement (i) were true, there would have been a protocol for that problem with worst-case complexity of at most $\mu \log \mu + o(\mu \log \mu)$. Results in [3] imply that every protocol for that problem must have worst-case complexity of at least $\log \log t + 1$ bits. For $t \gg \mu$, these two conclusions are contradictory. An even larger discrepancy is derived when statement (ii) is assumed to hold. Note that this problem is not balanced, but can be balanced easily.

5. Correlated files. Section 1.2 described the correlated-files problem and some of its potential applications. Essentially, X and Y are binary strings within edit distance of at most α from each other; P_X knows X, while P_Y knows Y and wants to learn X. When we want to emphasize the maximum distance between X and Y, we refer to the problem as the *α-edits problem*.

In the next section, we apply results proved in the previous sections to obtain efficient interactive protocols for this problem. In § 5.2 we discuss one-way protocols.

5.1. Interactive communication. First, we evaluate $\mu(y)$—P_Y's ambiguity when his string is y. P_X's string x can be obtained from y by i insertions and d deletions, where $i + d \leqq \alpha$. In the special case where x can be obtained from y by insertions alone, we say that x is a *superstring* of y (and that y is a *substring* of x). The number of $(n - \alpha)$-bit substrings of an n-bit string y depends on y. For example, 010 has three 2-bit substrings (01, 00, 10), whereas 000 has only one. Yet it is known that the number of $(n + \alpha)$-bit superstrings of y is the same for every n-bit string y.

LEMMA 4. *The number of $(n + \alpha)$-bit superstrings of an n-bit string is* $\sum_{i=0}^{\alpha} \binom{n+\alpha}{i}$.

We use the lemma to estimate P_Y's ambiguity in the α-edits problem. Let $|y|$ denote the number of bits in a string y; then

$$(9)\quad \binom{|y|+\alpha}{\alpha} \leqq \sum_{k=0}^{\alpha} \sum_{i=0}^{k} \binom{|y|+k}{i} \leqq \mu(y) \leqq \sum_{k=0}^{\alpha} k \cdot \sum_{i=0}^{k} \binom{|y|+k}{i} \leqq \alpha^3 \binom{|y|+\alpha}{\alpha}.$$

Therefore, the number of bits P_X must transmit *if he knows Y in advance* is bounded by

$$(10)\qquad \log \binom{|y|+\alpha}{\alpha} \leqq \lceil \log \mu(y) \rceil \leqq \log \binom{|y|+\alpha}{\alpha} + 3 \log \alpha.$$

In cases of interest, $|y|$ is (much) larger than α; hence this bound is tight. A simple protocol almost achieves this number: For each edit operation needed to convert y to x, P_X describes its nature (delete, insert 0, or insert 1) and location. The total number of bits transmitted under this protocol is at most $\alpha(\lceil \log (|Y| + \alpha) \rceil + 2)$. Note that the protocol assumes that P_X knows Y in advance.

The maximum ambiguity of the α-edits problem is infinite: There is no upper bound on the length of y, hence on $\mu(y)$. Therefore, all worst-case complexities are infinite, too. For a more sensitive measure, let ϕ be a protocol for a random pair (X, Y) and define $l_\phi(x, y)$ to be the number of bits exchanged under ϕ for the input (x, y). For every $y \in S_Y$, let

$$\hat{l}_\phi(y) \overset{\text{def}}{=} \max \{l_\phi(x, y) : (x, y) \in S\}$$

be the number of bits transmitted under ϕ in the worst case when P_Y's value is y. Clearly,

$$\hat{l}_\phi(y) \geqq \lceil \log \mu(y) \rceil,$$

and equality holds *if P_X knows Y in advance*. ϕ is *asymptotically efficient for every y*, or *efficient* for short, if, for all $y \in S_Y$,

$$\hat{l}_\phi(y) \leqq \log \mu(y) + o(\log \mu(y)),$$

namely, if ϕ requires only marginally more bits than needed when P_X knows y in advance.

The discussion at the end of the last section implies that many (X, Y) pairs, even balanced ones, do not have efficient protocols: $\hat{l}_\phi(y)$ may have to be much larger than $\log \mu(y)$ for some y's. However, the next corollary shows that the correlated-files problem has an efficient three-message protocol (one-way protocols are discussed in the next section).

COROLLARY 3. *For every α, the α-edits problem has an efficient three-message protocol ϕ satisfying*

$$\hat{l}_\phi(y) \leqq \log \mu(y) + 3 \log \log \mu(y) + \log \alpha + 13.$$

Proof. The lengths $|X|$ and $|Y|$ of the strings X and Y are within α from each other. Hence, if P_X transmits $|X| \bmod (2\alpha + 1)$, then P_Y can infer $|X|$. Thereafter, P_X and P_Y can use the three-message protocol of Corollary 2 for strings of length at most $|X| + \alpha$. □

The next numerical example shows that the protocol implied by the corollary is efficient not only asymptotically, but also for moderately sized files.

Example 3. Suppose that Y is a 2^{23}-bit file (roughly 1,000,000 characters) and that X can be derived from Y via 8,000 insertions and deletions of bits (say, 1,000 character changes). If P_X knew Y in advance, he would need to transmit $\lceil \log \mu(Y) \rceil$ bits in the worst case, which (10) evaluates as

$$\begin{aligned} 91{,}800 \leqq \log \binom{2^{23} + 8{,}000}{8{,}000} &\leqq \lceil \log \mu(Y) \rceil \\ \leqq \log \binom{2^{23} + 8{,}000}{8{,}000} &+ 3 \log (8{,}000) + 1 \leqq 91{,}845. \end{aligned}$$

The corollary shows that, even when P_X does not know Y in advance, using three messages, he can convey X to P_Y using at most

$$\log \mu(Y) + 3 \log \log \mu(Y) + \log (8{,}000) + 13 \leqq \log \mu(Y) + 75 \text{ bits.}$$

Generalized edit distance. We can assume a more general setting than before. For example, in addition to inserting and deleting bits, a single edit operation (a mouse or a key click) may move or complement contiguous segments of arbitrary length. Still, the number of operations needed to derive a string from another can be easily seen to be a "distance" between the two.

Assume that X and Y are derived from a string Z using, respectively, α_1 and α_2 edit operations. As all the above operations are reversible, X can be derived from Y using at most $\alpha = \alpha_1 + \alpha_2$ edit operations. Every edit operation is characterized by a location in the string and a small number of bits, say four, identifying the operation (delete, insert 0, insert 1, move, complement, and so forth). If Y is of length $n \gg \alpha$, then the number of possible X sequences within α edit operations from Y is at most $2^{\alpha(\log(n+\alpha)+4)}$. This bound is not far from the actual number as, *for every Y*, there are at least $\binom{n+\alpha}{\alpha}$ possible X string derived by α insertions to Y. We can therefore apply Corollary 2 to derive a near-optimum protocol for this problem.

The theorem still holds if we allow the editor to delete arbitrary long strings. However, now $\mu(Y)$ can be arbitrarily larger than the number of bits in Y (say, Y was derived by erasing all of X), and so all complexity measures are unbounded. Note that, in this case, even if P_X knew Y in advance, he would still need to transmit arbitrarily many (in the length of Y) bits to describe X.

5.2. One-way communication. Section 1.2 mentioned several reasons rendering one-way protocols more useful than interactive ones. However, the general results proved in § 2 only imply one-way protocols requiring at most twice the number of bits needed when P_X knows Y in advance. We therefore investigate one-way protocols for correlated files in more detail.

A one-way protocol for the α-edits problem is a function $\phi : \{0, 1\}^* \rightarrow \{0, 1\}^*$ such that, if x and x' are distinct strings obtained from the same y via at most α edits (insertions and/or deletions), then neither $\phi(x)$ nor $\phi(x')$ is a prefix[6] of the other. This guarantees that P_Y can tell when P_X's message ends and then deduce the value of X.

A special case of the α-edits problem is the *α-insertions problem*, where X is obtained from Y by *exactly* α insertions. A one-way protocol for the α-insertions problem, or an *α-insertions protocol*, is therefore a mapping $\phi : \{0, 1\}^* \rightarrow \{0, 1\}^*$ such that, if x and x' are distinct strings obtained from the same y by exactly α insertions, then neither $\phi(x)$ nor $\phi(x')$ is a prefix of the other. To simplify the statement of the next lemma, we place an additional demand on the behavior of the protocol for strings of length less than α. We require that, if $\beta < \alpha$, then $\phi(x)$ is not a prefix of $\phi(x')$ for all distinct $x, x' \in \{0, 1\}^\beta$. Intuitively, this corresponds to allowing P_Y to have a string of "negative length" $\beta - \alpha$. As all β-bit strings can be obtained by α insertions to this string, the protocol must distinguish between them.

LEMMA 5. *Let α be a nonnegative integer. Any α-insertions protocol is also a β-insertions protocol for all $0 \leqq \beta \leqq \alpha$.*

Proof. The proof follows by definition. Let ϕ be an α-insertions protocol. If $|x| = |x'| < \beta$, then neither $\phi(x)$ nor $\phi(x')$ is a prefix of the other by the additional requirement above. If x and x' can be obtained from a string y by β insertions, let y' be a string derived from y by deleting any $\alpha - \beta$ bits. Then x and x' are obtained from y' by α insertions, and again neither $\phi(x)$ nor $\phi(x')$ is a prefix of the other. □

Conceptually, insertion protocols are simpler to construct than edit protocols. They need to handle only insertions and only a fixed number of them (unlike insertions, different number of edits may result in strings of the same length). The α-insertions problem is also more appealing analytically. Whereas (9) only bounds $\hat{\mu}$ for the α-edits problem, Lemma 4 says that, for the α-insertions problem, all strings y have $\mu(y) = \sum_{i=0}^{\alpha} \binom{|y|+\alpha}{i}$.

The next theorem shows that, in a sense, the two problems are equally difficult. It reduces the problem of constructing a one-way protocol that handles insertions and deletions to that of constructing a one-way protocol that handles only insertions.

THEOREM 3. *If the α-insertions problem has an efficient one-message protocol, then so does the α-edits problem.*

Proof. Let ϕ be a one-message α-insertions protocol. Define

$$\psi(x) = (|x| \bmod (2\alpha + 1), \phi(x))$$

to be the protocol that, for every string x, transmits the binary representation of the length of x mod $(2\alpha + 1)$, follows by $\phi(x)$. We show that, if x and x' are distinct strings that can be derived from a string y by at most α edits, then $\psi(x)$ is not a prefix of $\psi(x')$.

If $|x| \neq |x'|$, then $|x| \neq |x'| \bmod (2\alpha + 1)$; hence $\psi(x)$ and $\psi(x')$ have different beginnings. If $|x| = |x'|$, then there are i, i', d, d' such that

[6] The string 011 is a *prefix* of 011 and of 01101; hence, if a string is not a prefix of another, they cannot, in particular, be equal.

1) $i + d \leqq \alpha$, and x can be derived from y by i insertions and d deletions;
2) $i' + d' \leqq \alpha$, and x' can be derived from y by i' insertions and d' deletions;
3) $i - d = i' - d'$ (because $|x| = |x'|$).

From 1) and 2), there is a sequence of d insertions and i deletions taking x into y and a sequence of i' insertions and d' deletions taking y into x'.

We can change the order of a sequence of insertions and deletions applied to a string without affecting the resulting string. We must only ensure that if, in the original sequence of insertions and deletions, a bit is inserted and then deleted, and this order is reversed in the new sequence; then, in the new sequence, we neither delete this (yet nonexistent) bit nor insert it later. Therefore, there is $\beta \leqq i + d'$ such that a sequence of β deletions followed by a sequence of β insertions takes x to x'. Let y be the sequence obtained from x after the β deletions. Both x and x' can be obtained from y by β insertions. From 3), $i + d' = i' + d$; hence $\beta \leqq i + d' = \frac{1}{2}(i + d' + i' + d) \leqq \alpha$. By Lemma 5, $\phi(x)$ is not a prefix of $\phi(x')$.

Regarding efficiency, ψ transmits only $\lceil \log (2\alpha + 1) \rceil = o(\log \binom{|y| + \alpha}{\alpha})$ bits more than ϕ. The maximum ambiguity of the α-edits problem is higher than that of the α-insertions problem. Hence, if ϕ is efficient, so is ψ. □

In view of the theorem, we consider only insertions protocols. First, we discuss their efficiency. Let ϕ be an α-insertions protocol. Define

$$l_\phi(n) \overset{\text{def}}{=} \max \{ |\phi(x)| : x \in \{0, 1\}^{n+\alpha} \}$$

to be the maximum number of bits P_X transmits according to ϕ when y is n-bit long. Lemma 4 said that every n-bit string has $\sum_{i=0}^{\alpha} \binom{n+\alpha}{i}$ superstrings of length $n + \alpha$. Hence,

$$l_\phi(n) \geqq \log \sum_{i=0}^{\alpha} \binom{n + \alpha}{i},$$

and ϕ is efficient if

$$\lim_{n \to \infty} \frac{l_\phi(n)}{\log \sum_{i=0}^{\alpha} \binom{n + \alpha}{i}} = 1,$$

namely, if the number of bits transmitted for every y is asymptotically the same as that needed when P_X knows Y in advance.

We now consider efficient insertion protocols for some restricted cases. We begin with the most elementary of these protocols, one that identifies a single insertion (and, by arguments similar to the ones used to prove Theorem 3, a single deletion, too).

Suppose that Y is an n-bit string and that X is an $(n + 1)$-bit string obtained from y by a single insertion. According to Lemma 4, for every possible value of Y, there are $n + 2$ possible X's. Therefore, even if P_X knew Y in advance, he would have to transmit $\lceil \log (\hat{\mu}) \rceil = \lceil \log (n + 2) \rceil$ bits in the worst case. We describe a protocol that transmits exactly that many bits and does not require P_X to know Y in advance.

LEMMA 6. *Let $X \in \{0, 1\}^{n+1}$ be a superstring of $Y \in \{0, 1\}^n$. Then*

$$\hat{C}_1 = \ldots = \hat{C}_\infty = \lceil \log \hat{\mu} \rceil = \lceil \log (n + 2) \rceil.$$

Proof. View a string z as an integer sequence $z_1, \ldots, z_{|z|}$. Using $\lceil \log (n + 2) \rceil$ bits, P_X transmits

$$\phi(x) \overset{\text{def}}{=} \sum_{i=1}^{n+1} i x_i \bmod (n + 2)$$

(here $\phi(x)$ is a number rather than its binary representation). To verify that P_Y can deduce X from $\phi(X)$, we show that ϕ maps all superstrings of a given string into different numbers. For example, if Y is 010, then there are $3 + 2 = 5$ possible X sequences, and ϕ maps each one into a different integer mod 5: $0100 \rightarrow 2$, $0010 \rightarrow 3$, $1010 \rightarrow 4$, $0110 \rightarrow 0$, $0101 \rightarrow 1$.

In the general case, assume that y has k "ones" (hence $n - k$ "zeros") and consider the effect of a single insertion on the value of the function $f : \{0, 1\}^* \rightarrow \mathscr{Z}^+$ defined by $f(z) = \sum_{i=1}^{|z|} iz_i$.

Assume first that x is derived from y by inserting a "zero" between the ith and $(i + 1)$th *rightmost* "ones" in y, where $0 \leqq i \leqq k$. Each of the i "ones" to the right of the inserted bit is shifted one position to the right, and so $f(x) = f(y) + i$.

Next, assume that x is derived from y by inserting a "one" between the ith and $(i + 1)$th *leftmost* "zeros" in y, where $0 \leqq i \leqq n - k$, and assume further that the "one" was inserted between the jth and $(j + 1)$th leftmost bits of y. It is easy to verify that, in that case, $f(x) = f(y) + (j + 1) + ((n - j) - (n - k - i)) = f(y) + k + 1 + i$.

Therefore, if x is derived from y by inserting a "zero," $f(y)$ increases by an integer ranging from 0 to k, while, if x is derived from y by inserting a "one," $f(y)$ increases by an integer ranging from $k + 1$ to $n + 1$. As $\phi(x)$ is $f(x)$ mod $(n + 2)$, it too is different for every supersequence x of y.

This one-insertion protocol is related to an insertion/deletion code of Varshamov and Tenengolt [16]. This code consists of all n-bit codewords x with a fixed $\sum_{i=1}^{n} ix_i \bmod (n + 2)$. We note, however, that an efficient code does not imply an efficient one-way protocol. An efficient protocol corresponds to a collection of disjoint efficient codes that cover $\{0, 1\}^*$. □

To derive a protocol for identifying more than a single insertion, we could try to generalize the above protocol. In the following, we suggest an alternative approach that may prove more tractable. It attempts to reduce the problem to one of exchanging strings to within small Hamming distance from each other. We show that this approach works in a restricted case of insertions and deletions.

The *Hamming distance* $d_H(x, y)$ between two equal-length strings x and y is the number of bit locations where they differ. In the *Hamming-distance problem*, X and Y are n-bit sequences within a small Hamming distance from each other. P_X knows X, while P_Y knows Y and wants to learn X. Witsenhausen and Wyner [17] considered the problem in the context of video compression. The (independent, but essentially the same) solution described here is taken from [18] and was done by El Gamal and Brandman, extending Example 5 in [4].

Example 4. P_X has an n-bit sequence X, and P_Y has an n-bit sequence Y. The Hamming distance between X and Y is at most t. How many bits must be transmitted in the worst case for P_Y to learn X?

P_Y's ambiguity is the same for every $y \in \{0, 1\}^n$:

$$\mu(y) = \sum_{i=1}^{t} \binom{n}{i}.$$

Therefore,

$$\hat{C}_1 \geqq \hat{C}_\infty \geqq \left[\log \sum_{i=1}^{t} \binom{n}{i}\right] \geqq t(\log n - \log t).$$

Using error-correcting codes, we show that

$$\hat{C}_1 \leqq t\lceil \log (n + 1)\rceil. \tag{11}$$

We are mostly interested in cases where $t \ll n$. For these cases, the two bounds are quite close. We assume that the reader is familiar with basic results concerning linear error-correcting codes. The protocol is based on an (n, k) linear t-error correcting code C (we will determine k later). P_X and P_Y agree a priori on a parity-check matrix H for C. When

P_X is given X and P_Y is given Y such that $d_H(X, Y) \leqq t$, they execute the following protocol.

P_X transmits XH^T, the $(n - k)$-bit syndrome of X. P_Y computes $XH^T - YH^T$, the syndrome of $(X - Y)$. He finds an n-bit sequence Z with Hamming weight $\leqq t$ such that $ZH^T = (X - Y)H^T$ and decides that X is $Z + Y$.

To prove that P_Y always determines X correctly, we now show that $Z = (X - Y)$. Namely, (1) $(X - Y)$ has Hamming weight $\leqq t$, and (2) no other n-bit sequence with Hamming weight $\leqq t$ has the syndrome $(X - Y)H^T$.

(1) holds as $d_H(X, Y) \leqq t$. To prove (2), note that, if an n-bit sequence s has Hamming weight $\leqq t$ and syndrome $(X - Y)H^T$, then $(X - Y) - s$ has Hamming weight $\leqq 2t$ and syndrome 0; that is, $(X - Y) - s$ is a codeword in C with Hamming weight $\leqq 2t$. Since C is t-error correcting, it has minimum distance $\geqq 2t + 1$; hence $s = X - Y$.

The number of bits transmitted under this protocol is $n - k$. To prove a low one-message complexity, we need a t-error correcting code with low redundancy. The BCH bound (e.g., [19] or Theorem 9.2 in [20]) guarantees that, if $n + 1$ is a power of 2, then there is a linear t-error correcting code with $n - k \leqq t \log (n + 1)$ implying (11).

To reduce the edit-distance problem to the Hamming-distance problem, we need the following two mappings: $\xi : \{0, 1\}^n \rightarrow \{0, 1\}^m$ and $\zeta : \{0, 1\}^{n+\alpha} \rightarrow \{0, 1\}^m$ (the value of m will be discussed later) such that, for all $y \in \{0, 1\}^n$ and for all $x \neq x' \in \{0, 1\}^{n+\alpha}$ that are superstrings of y,

(i) $\zeta(x) \neq \zeta(x')$,

(ii) $d_H(\zeta(x), \xi(y)) \leqq \alpha$.

If ξ and ζ exist, we can construct a one-message protocol for identifying α insertions as follows. P_X computes $\zeta(x)$, and P_Y computes $\xi(y)$. Condition (ii), above, guarantees that the Hamming distance between $\zeta(x)$ and $\xi(y)$ is at most α. By the last example, P_X can convey $\zeta(x)$ to P_Y while transmitting $\alpha\lceil \log m \rceil$ bits. Condition (i), above, ensures that x is the only superstring of y mapped by ζ into $\zeta(x)$; hence P_Y can determine x.

The number of bits transmitted by the protocol is $\alpha\lceil \log m \rceil$, whereas the number of bits needed if P_X knows Y in advance is $\log \mu(y) = \log (\sum_{i=0}^{\alpha} \binom{n+\alpha}{i})$. For the protocol to be efficient, m must grow as $(n + \alpha)^{1+o(1)}$ (say, $m = (n + \alpha) \log (n + \alpha)$).

The smallest value m can attain is $n + \alpha$: The number of $(n + \alpha)$-bit superstrings of y is $\sum_{i=0}^{\alpha} \binom{n+\alpha}{i}$, and, according to above conditions, this number must be smaller than $\sum_{i=0}^{\alpha} \binom{m}{i}$, the number of $(n + \alpha)$-bit strings within Hamming distance α from the m-bit string $\xi(y)$. Hence, at least in terms of "first-order" cardinalities, $m = n + \alpha$ may be possible. So far, ζ and ξ with $m = n + \alpha$ are known only in very special cases: all n and $\alpha = 1$; all α and $n = 1$; and, due to J. Reeds, $n = \alpha = 2$.

We do not know whether ξ and ζ exist in general (even with $m > n + \alpha$). However, following is a construction for a restricted case where x is obtained from y by α edits that *do not introduce new runs or destroy existing ones*. This restriction severely limits the sequences that can be obtained from y. For example, the string 0 cannot be transformed into the string 1 without destroying a run, and 010000 can be transformed into 000010 by four insertions and deletions, but six are needed if no runs are to be eliminated or introduced.

We demonstrate ξ and ζ for this *restricted-edit problem* by converting the strings x and y known to P_X and P_Y into integer sequences within L_1 distance of at most α from each other. In turn, this *L_1-distance problem* can be reduced via a Gray code to the Hamming-distance problem, thus exhibiting ξ and ζ.

Represent a string z with r runs[7] as an r-element integer sequence whose ith element is the length of the ith run in z. For example, 010000 is represented as 1, 1, 4. If x can

[7] A *run* in a string is a contiguous sequence of "zeros" or "ones."

be obtained from y by insertions and deletions that do not introduce or destroy runs, then x and y have the same number of runs and both start with the same bit. Therefore, given y, there is a one-to-one correspondence between x and its representation above. It follows that the restricted-edit problem is almost equivalent to the following L_1-distance problem.

Example 5. Let r and α be positive integers and let $X_1, \ldots, X_r$ and $Y_1, \ldots, Y_r$ be integers such that

$$\sum_{i=1}^{r} |X_i - Y_i| \leqq \alpha.$$

P_X knows $X = (X_1, \ldots, X_r)$, while P_Y knows $Y = (Y_1, \ldots, Y_r)$ and wants to learn X. How many bits must be transmitted in the worst case?

It is easy to verify that, for all $y = (y_1, \ldots, y_r)$,

$$\mu(y) = 1 + \sum_{i=1}^{\alpha} \sum_{j=1}^{\min\{i,r\}} 2^j \binom{r}{j}\binom{i-1}{j-1}.$$

Using Gray and error-correcting codes, we show that

$$\hat{C}_1 \leqq \alpha \lceil \log(1 + r \log(2\alpha + 1)) \rceil.$$

The bound is asymptotically tight, and the implied protocol is efficient whenever $\alpha / r \to 0$.

Let

$$\beta \overset{\text{def}}{=} \lceil \log(2\alpha + 1) \rceil.$$

A β-bit Gray code is a sequence $g_0, \ldots, g_{2^\beta - 1}$ of β-bit strings such that, for[8] $i = 0, \ldots, 2^\beta - 1$,

$$d_H(g_i, g_{i+1}) = 1.$$

Simple constructions of Gray codes are known for every positive β. Map the sequence $x_1, \ldots, x_r$ into the $(r\beta)$-bit string $g_{x_1} g_{x_2} \ldots g_{x_r}$ by concatenating the corresponding Gray-code strings (again, indices are interpreted modulo 2^β). Map $y_1, \ldots, y_r$ into $g_{y_1} g_{y_2} \ldots g_{y_r}$ similarly. Simple properties of Gray codes guarantee that

$$d_H(g_{x_i}, g_{y_i}) \leqq |x_i - y_i|.$$

Hence,

$$d_H(g_{x_1} g_{x_2} \ldots g_{x_r}, g_{y_1} g_{y_2} \ldots g_{y_r}) \leqq \text{sum}_{i=1}^{r} |x_i - y_i| \leqq \alpha.$$

Example 4 implies that P_X can transmit $\alpha \lceil \log(r\beta) \rceil$ bits, enabling P_Y to learn $g_{x_1} g_{x_2} \ldots g_{x_r}$. By choice of β, two integers that are mapped into the same Gray sequence must be at least $2\alpha + 1$ apart; hence P_X can infer $x_1, \ldots, x_r$.

We apply the example to prove the following corollary.

COROLLARY 4. *The α-restricted-edits problem has an efficient one-way protocol.*

Proof. The length of X can be transmitted using $\lceil \log(2\alpha + 1) \rceil$ bits. Thereafter, P_X and P_Y can use the protocol given in the example. The total number of bits, transmitted for X, is

$$\alpha \lceil \log(|X| \log(2\alpha + 1)) \rceil + \lceil \log(2\alpha + 1) \rceil = \alpha \log |X| + o(\alpha \log |X|).$$

For fixed α,

$$\log(\mu(y)) = \alpha \log |X| - o(\alpha \log |X|);$$

[8] In this equation and below, Gray-sequence indices are taken modulo 2^β; hence g_{2^β} denotes g_0.

hence the protocol is efficient. Note that, by invoking the example, we implicitly used the fact that the restricted-edits problem has ξ and ζ with range-dimension $m = |X| \lceil \log(2\alpha + 1) \rceil$. □

6. Open problem. The main open problem suggested by this paper is the construction of efficient one-way insertion protocols. As shown in Theorem 3, such protocols imply efficient one-way protocols for the edit-distance problem, hence may have practical applications. Two possible directions towards such a construction are (1) extension of the single-insertion protocol of Lemma 6, and (2) construction of the functions ξ and ζ discussed in the last section for additional values of n and α.

Acknowledgment. I thank Jeff Kahn for his help in formulating and proving Lemma 2.

REFERENCES

[1] H. Abelson, *Lower bounds on information transfer in distributed computations*, in Proc. of the 19th Ann. Sympos. on Foundations of Computer Science, 1978.

[2] A. C. Yao, *Some complexity questions related to distributive computing*, in Proc. of the 11th Ann. ACM Sympos. on Theory of Computing, 1979, pp. 209–213.

[3] A. Orlitsky, *Worst-case interactive communication* I: *Two messages are almost optimal*, IEEE Trans. Inform. Theory, 36 (1990), pp. 1111–1126.

[4] A. El Gamal and A. Orlitsky, *Interactive data compression*, in Proc. of the 25th Ann. Sympos. on Foundations of Computer Science, Singer Island, FL, 1984, pp. 100–108.

[5] A. Orlitsky, *Worst-case interactive communication* II: *Two messages are not optimal*, IEEE Trans. Inform. Theory, 37 (1991), pp. 995–1005.

[6] ———, *Average-case interactive communication*, Technical Memorandum, AT&T Bell Laboratories; IEEE Trans. Inform. Theory, 38 (1992), pp. 1534–1547.

[7] C. H. Papadimitriou and M. Sipser, *Communication complexity*, in Proc. of the 14th Ann. ACM Sympos. on Theory of Computing, San Francisco, CA, 1982, pp. 196–200.

[8] P. Duris, Z. Galil, and G. Schnitger, *Lower bounds on communication complexity*, in Proc. of the 16th Ann. ACM Sympos. on Theory of Computing, Washington, DC, 1984, pp. 81–91.

[9] N. Nisan and A. Wigderson, *Rounds in communication complexity revisited*, in Proc. of the 23rd Ann. ACM Sympos. on Theory of Computing, New Orleans, LA, 1991, pp. 419–429.

[10] H. Witsenhausen, *The zero-error side information problem and chromatic numbers*, IEEE Trans. Inform. Theory, 22 (1976), pp. 592–593.

[11] M. L. Fredman, J. Komlós, and E. Szemeredi, *Storing a sparse table with* $o(1)$ *worst case access time*, J. Assoc. Comput. Mach., 31 (1984), pp. 538–544.

[12] M. L. Fredman and J. Komlós, *On the size of separating systems and families of perfect hash functions*, SIAM J. Algebraic Discrete Meth., 5 (1984), pp. 61–68.

[13] J. Friedman, *Constructing* $o(n \log n)$ *size monotone formula for the kth threshold function on n variables*, SIAM J. Comput., 15 (1986), pp. 641–654.

[14] J. Körner, *Fredman-Komlós bounds and information theory*, SIAM J. Algebraic Discrete Meth., 7 (1986), pp. 560–570.

[15] J. Spencer, *Ten Lectures on the Probabilistic Method*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1987.

[16] V. I. Levenshtein, *Binary codes capable of correcting spurious insertions and deletions of ones*, Prob. Inform. Trans., 1 (1965), pp. 8–17.

[17] H. Witsenhausen and A. D. Wyner, *Interframe Coder for Video Signals*, United States Patent Number 4,191,970, 1980.

[18] A. Orlitsky, *Communication Issues in Distributed Communication*, Ph.D. thesis, Stanford University, Stanford, CA, 1986.

[19] R. E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley, Reading, MA, 1984.

[20] W. W. Peterson and E. J. Weldon, *Error-Correcting Codes*, 2nd ed., MIT Press, Cambridge, MA, 1972.

SIAM J. DISC. MATH.
Vol. 6, No. 4, pp. 565-574, November 1993

ON MINIMUM FAULT-TOLERANT NETWORKS*

S. UENO†, A. BAGCHI‡, S. L. HAKIMI§, AND E. F. SCHMEICHEL¶

Abstract. This paper considers the following problem: Given a positive integer t and graph H, construct a graph G from H by adding a minimum number $\Delta(t, H)$ (respectively, $\Delta'(t, H)$) of edges and an appropriate number of vertices, such that after removing any t vertices (respectively, t edges) from G the remaining graph contains H as a subgraph. This problem was motivated by the design of fault-tolerant interconnection networks for multiprocessor systems. The authors estimate $\Delta(t, H)$ and $\Delta'(t, H)$ for the cycle, path, complete binary tree, grid, torus, and hypercube on n vertices.

Key words. graph theory, fault-tolerant networks, network architecture preservation

AMS(MOS) subject classifications. 05C10, 05C60, 68R10

1. Introduction and preliminaries. This paper considers the following problem: Given a positive integer t and graph H, construct a graph G with the minimum number of edges, such that even after removing t vertices or edges from G, the remaining graph contains H as a subgraph. This problem was motivated by the design of fault-tolerant multiprocessor interconnection networks. We construct such graphs with small number of edges for the cycle, path, complete binary tree, grid, torus, and hypercube on n vertices. We give lower bounds for the number of edges in G and show that our constructions for the cycle, path, and complete binary tree are optimal in the sense of the "order" of the number of edges with respect to n. Many related results can be found in the literature [1]-[14].

Let G be a graph and let $V(G)$ and $E(G)$ denote the vertex set and edge set of G, respectively. If $S \subseteq V(G)$, $G - S$ is the graph obtained from G by deleting S together with the edges incident to the vertices in S. If $S \subseteq E(G)$, $G \backslash S$ is the graph obtained from G by deleting the edges of S. We use $d_G(v)$ to denote the degree of a vertex v of G. If H is a subgraph of G, we define $\Delta(G, H) = |E(G)| - |E(H)|$. Let C_n, P_n, S_n, B_n, R_n, D_n, and Q_n denote the cycle, path, star, complete binary tree, grid, torus, and hypercube on n vertices, respectively. Note that $n = 2^p - 1$ for some p in the case of B_n, and we assume that $n = r^2$ for some r in the case of R_n and D_n, and, of course, $n = 2^d$ for some d in the case of Q_n.

Let t be a nonnegative integer. A graph G is called a t-FT (t-fault-tolerant) graph with respect to H if $G - S$ contains H as a subgraph for every $S \subseteq V(G)$, with $|S| \leqq t$. Define $\Delta(t, H) = \min \{\Delta(G, H) | G \text{ is a } t\text{-FT graph with respect to } H\}$. That is, $\Delta(t, H)$ is the minimum number of edges added to H to construct a t-FT graph with respect to H. A graph G is called a t-EFT (t-edge-fault-tolerant) graph with respect to a graph H if $G \backslash S$ contains H as a subgraph for every $S \subseteq E(G)$, with $|S| \leqq t$. Define $\Delta'(t, H) =$

* Received by the editors December 3, 1990; accepted for publication (in revised form) September 11, 1992. This work was supported in part by National Science Foundation grants NCR-91-02524 and DMS-92-06991.

† Department of Electrical and Electronic Engineering, Tokyo Institute of Technology, Tokyo 152, Japan.

‡ Bellcore, Red Bank, New Jersey 07701.

§ Department of Electrical Engineering and Computer Science, University of California, Davis, California 95616.

¶ Department of Mathematics and Computer Science, San Jose State University, San Jose, California 05192.

min $\{\Delta(G, H) | G$ is a t-EFT graph with respect to $H\}$. Since a t-FT graph with respect to H is also a t-EFT graph with respect to H, we have the following inequality.

PROPERTY 1. $\Delta(t, H) \geqq \Delta'(t, H)$ *for any graph* H.

It should be noted that $|V(G)| \geqq |V(H)| + t$ if G is a t-FT graph with respect to H, and $|V(G)| \geqq |V(H)|$ if G is a t-EFT graph with respect to H. We investigate the following problems.

PROBLEM 1. *Given a graph H, construct a t-*FT *graph with respect to H by adding $\Delta(t, H)$ edges and an appropriate number of vertices.*

PROBLEM 2. *Given a graph H, construct a t-*EFT *graph with respect to H by adding $\Delta'(t, H)$ edges and an appropriate number of vertices.*

We estimate $\Delta(t, H)$ and $\Delta'(t, H)$ for cycles, paths, complete binary trees, grids, tori, and hypercubes.

Some related problems are considered in the literature. Erdös, Graham, and Szemerédi [5] consider the directed analogue of Problem 1 for dipaths, in connection with the complexity of Boolean functions, and give a good estimate for the number of edges in an n-FT acyclic digraph with respect to the dipath of length n. Alon and Chung [1] construct a graph G with $O(n/\varepsilon)$ vertices and maximum degree $O(1/\varepsilon^2)$, such that, even after deleting $(1-\varepsilon)|V(G)|$ vertices or $(1-\varepsilon)|E(G)|$ edges, the remaining graph still contains P_n as a subgraph, for every $\varepsilon > 0$ and every positive integer n. This result settled a problem raised by Rosenberg [12] in connection with fault-tolerant linear arrays. Friedman and Pippenger [8] generalize the result in [1] from P_n to the trees with n vertices and maximum degree at most d. For some special cases, the existence of such graphs is proved by Beck [2] nonconstructively.

The following problem, which is a variant of Problem 1, is considered in the literature [3], [6], [7], [9].

PROBLEM 3. *Given a graph H, construct a t-*FT *graph with respect to H on $|V(H)| + t$ vertices by adding a minimum number of edges.*

Chartrand and Kapoor [3] and Hayes [9] show that the minimum number of edges in Problem 3 for C_n and P_n are $\frac{1}{2}tn + \frac{1}{2}t(t+2)$ and $\frac{1}{2}(t-1)n + \frac{1}{2}t(t+1) + 1$, respectively. Farrag and Dawson [6] solve Problem 3 for S_n; in particular, they show that the minimum number of added edges is $tn + \frac{1}{2}(t-1)t$ if $t \leqq n$. Note that this is equal to the following trivial upper bound for $\Delta(t, H)$. Let $G_1 * G_2$ denote the join of graphs G_1 and G_2; that is, $V(G_1 * G_2) = V(G_1) \cup V(G_2)$ and $E(G_1 * G_2) = E(G_1) \cup E(G_2) \cup \{(u, v) | u \in V(G_1), v \in V(G_2)\}$. It is easy to see that $H * K_t$ is a t-FT graph with respect to any graph H. Thus we have the following upper bound for $\Delta(t, H)$.

PROPERTY 2. $\Delta(t, H) \leqq \Delta(H * K_t, H) = t|V(H)| + \frac{1}{2}t(t-1)$ *for any* H.

Note also that, if H is connected and G is a t-FT graph with respect to H on $|V(H)| + t$ vertices, then $d_G(v) \geqq t + 1$ for any $v \in V(G)$. It follows that $|E(G)| \geqq \frac{1}{2}(t+1)(|V(H)| + t)$. Thus we have the following lower bound for $\Delta(G, T_n)$, where T_n is a tree on n vertices.

PROPERTY 3. *If G is a t-*FT *graph with respect to T_n on $n + t$ vertices, then* $\Delta(G, T_n) \geqq \frac{1}{2}(t-1)n + \frac{1}{2}t(t+1) + 1$.

It is interesting to note that $\Delta(t, S_n) = tn - t$, which we can easily see, since S_n has a vertex of degree $n - 1$. This shows that the result in [6] mentioned above is quite close to the solution of Problem 1 for S_n. In fact, the difference between the solutions of Problems 1 and 3 for S_n depends only on t. However, we will show that the "orders" of $\Delta(t, B_n)$ and $\Delta(t, P_n)$ with respect to n are smaller than that of the lower bound in Property 3. We also show that the "order" of $\Delta(t, C_n)$ with respect to n is smaller than that of the result in [3] and [9] mentioned above. On the other hand, Paoli, Wong, and

Wong [11], and Wong and Wong [13] consider the following problem, which is a variant of Problem 2.

PROBLEM 4. *Given a graph H, construct a t-*EFT *graph with respect to H on $|V(H)|$ vertices by adding a minimum number of edges.*

They show that the minimum numbers of edges in Problem 4 for C_n and P_n are $\frac{1}{2}tn$, if $t \leqq n - 3$ and $\frac{1}{2}(t-1)n + 1$, if $t \leqq n - 2$, respectively. Again, we will show that the "orders" of $\Delta'(t, C_n)$ and $\Delta'(t, P_n)$ with respect to n are smaller than those of the numbers above. We will also examine Problem 4 for B_n and Q_n when $t = 1$.

2. t-FT graph with respect to B_n. Let p be the number of levels of B_n. The root of B_n is at level 0, and the leaves are at level $p - 1$. The number of vertices at level i is 2^i, and $n = 2^p - 1$.

THEOREM 1. (i) $\Delta(t, B_n) \leqq \frac{1}{2}t(t+4)\sqrt{n+1} - 2t$, *if p is even.*

(ii) $\Delta(t, B_n) \leqq (\sqrt{2}/4)t(t+6)\sqrt{n+1} - 2t$, *if p is odd.*

Proof. Let $q = \lceil p/2 \rceil$, and $X^1, X^2, \dots, X^t$ be vertex disjoint q-level complete binary trees. Let Y be the complete q-level binary subtree of B_n consisting of the vertices at levels 0 to $q - 1$ of B_n. Note that Y is isomorphic to X^i $(i = 1, 2, \dots, t)$. Suppose that the leaves $u_1, u_2, \dots, u_{2^{q-1}}$ of Y correspond to the leaves $v_{i1}, v_{i2}, \dots, v_{i2^{q-1}}$ of X^i $(i = 1, 2, \dots, t)$ by an isomorphism, respectively.

Proof of (i). We construct a graph G from $B_n, X^1, X^2, \dots, X^t$ by the following procedure:

1. Join the root of X^i $(i = 1, 2, \dots, t)$ to the vertices at level $q - 1$ of B_n;

2. Join the root of X^i $(i = 1, 2, \dots, t)$ to the leaves of X^j $(j = 1, 2, \dots, i-1, i+1, \dots, t)$;

3. For each j $(j = 1, 2, \dots, 2^{q-1})$, join v_{ij} $(i = 1, 2, \dots, t)$ to the children of u_j in B_n.

We show that G is a t-FT graph with respect to B_n. Let S be a subset of $V(G)$ with $|S| \leqq t$. We show that $G - S$ contains B_n as a subgraph. From the construction of G, we may assume that $S \subseteq V(B_n)$. Each connected component of $B_n - V(Y)$ is a q-level complete binary subtree of B_n. Let Q be the set of all such subtrees of B_n including Y. Let $Z^1, Z^2, \dots, Z^k$ $(k \leqq t)$ be the subtrees in Q that have a vertex of S. It is easy to see that, if we replace Z^i by X^i $(i = 1, 2, \dots, k)$, then B_n is in $G - S$. Thus G is a t-FT graph with respect to B_n. To construct G, we added $t2^{q-1}$ edges in step 1, $t(t-1)2^{q-1}$ edges in step 2, $t2^q$ edges in step 3, and the edges in X^i $(i = 1, 2, \dots, t)$. Thus $\Delta(G, B_n) = t2^{q-1} + t(t-1)2^{q-1} + t2^q + t((2^q - 1) - 1) = \frac{1}{2}t(t+4)\sqrt{n+1} - 2t$.

Proof of (ii). We construct a graph G from $B_n, X^1, X^2, \dots, X^t$ by the following procedure:

1. Join the root of X^i $(i = 1, 2, \dots, t)$ to the vertices at level $q - 2$ of B_n;

2. Join the root of X^i $(i = 1, 2, \dots, t)$ to the vertices at level $q - 2$ of X^j $(j = 1, 2, \dots, i-1, i+1, \dots, t)$;

3. Join u_j $(j = 1, 2, \dots, 2^{q-1})$ to the father of v_{ij} $(i = 1, 2, \dots, t)$.

We can easily see by similar arguments as in the proof of (i) that G is a t-FT graph with respect to B_n and $\Delta(G, B_n) = (\sqrt{2}/4)t(t+6)\sqrt{n+1} - 2t$. □

THEOREM 2. $\Delta(t, B_n) \geqq \frac{1}{4}\sqrt{t(n+1)} - \frac{1}{2}t$.

Proof. Let G be a t-FT graph with respect to B_n. G contains a subgraph H isomorphic to B_n. Define $A = V(G) - V(H)$, $\alpha = |A|$, $B = \{v \mid v \in V(H)$, there exists $v' \in V(G)$ such that $(v, v') \in E(G) - E(H)\}$, and $\beta = |B|$.

Case 1. $\beta \leqq t$. $G - B$ contains a subgraph isomorphic to B_n, which is vertex disjoint from H. Thus, $\Delta(G, H) \geqq n - 1 \geqq \frac{1}{4}\sqrt{t(n+1)} - \frac{1}{2}t$.

Case 2. $\beta > t$. We may assume that $\beta < 2^{p-2}$; otherwise $\Delta(G, H) \geqq \frac{1}{2}\beta \geqq \frac{1}{8}(n+1) \geqq \frac{1}{4}\sqrt{t(n+1)} - \frac{1}{2}t$. Let $\gamma = \lfloor \log \beta \rfloor$. Let K be the set of complete binary subtrees of H rooted at the vertices at level $\gamma + 2$ of H. Since $|K| = 2^{\gamma+2} > 2\beta$, there exist t subtrees $Y^1, Y^2, \ldots, Y^t$ in K that have no vertex in B. Let S be the set of fathers of roots of $Y^1, Y^2, \ldots, Y^t$ in H. $G - S$ contains a subgraph H' isomorphic to B_n as $|S| \leqq t$. Since $V(H') \cap V(Y^i) = \varnothing$ for $i = 1, 2, \ldots, t$, we have that $\alpha \geqq |A \cap V(H')| \geqq \sum_{i=1}^{t} |V(Y^i)| = t(2^{p-\gamma-2} - 1)$. It follows that $(\alpha + t)\beta \geqq \frac{1}{4}t(n+1)$, which means that $\max\{\alpha + t, \beta\} \geqq \frac{1}{2}\sqrt{t(n+1)}$. Thus $\Delta(G, H) \geqq \frac{1}{2}\max\{\alpha, \beta\} \geqq \frac{1}{4}\sqrt{t(n+1)} - \frac{1}{2}t$. □

3. t-EFT graph with respect to B_n.

THEOREM 3. (i) $\Delta'(t, B_n) \leqq 2t\sqrt{n+1} - 2t$, *if p is even.*

(ii) $\Delta'(t, B_n) \leqq (3\sqrt{2}/2)t\sqrt{n+1} - 2t$, *if p is odd.*

Proof. We slightly change the construction procedures in the proof of Theorem 1.

Proof of (i). We construct a graph G from $B_n, X^1, X^2, \ldots, X^t$ by the following procedure:

1. Join the root of X^i $(i = 1, 2, \ldots, t)$ to the vertices at level $q - 1$ of B_n;
2. Join u_j $(j = 1, 2, \ldots, 2^{q-1})$ to the father of v_{ij} $(i = 1, 2, \ldots, t)$.

We can see by a similar argument as in the proof of Theorem 1 that G is a t-EFT graph with respect to B_n. To construct G, we add $t2^{q-1}$ edges in step 1, $t2^{q-1}$ edges in step 2, and the edges in X^i $(i = 1, 2, \ldots, t)$. Thus $\Delta(G, B_n) = t2^{q-1} + t2^{q-1} + t((2^q - 1) - 1) = 2t\sqrt{n+1} - 2t$.

Proof of (ii). We construct a graph G from $B_n, X^1, X^2, \ldots, X^t$ by the following procedure:

1. Join the root of X^i $(i = 1, 2, \ldots, t)$ to the vertices at level $q - 2$ of B_n;
2. Join the father of u_j $(j = 1, 2, \ldots, 2^{q-1})$ to the grandfather of v_{ij} $(i = 1, 2, \ldots, t)$.

We can easily see that G is a t-EFT graph with respect to B_n, and $\Delta(G, B_n) = (3\sqrt{2}/2)t\sqrt{n+1} - 2t$. □

THEOREM 4. $\Delta'(t, B_n) \geqq (\sqrt{2}/8)\sqrt{t(n+1)} - \frac{1}{2}t$.

Proof. Let G be a t-EFT graph with respect to B_n. G contains a subgraph H isomorphic to B_n. Define A, α, B, and β as in the proof of Theorem 2.

Case 1. $3\beta \leqq t$. Let $S = \{(u, v) \mid v \in B, (u, v) \in E(H)\}$. $G \backslash S$ contains a subgraph H' isomorphic to B_n as $|S| \leqq t$. Since H' is edge disjoint from H, $\Delta(G, H) \geqq |E(H')| = n - 1 \geqq (\sqrt{2}/8)\sqrt{t(n+1)} - \frac{1}{2}t$.

Case 2. $3\beta > t$. We may assume that $\beta < 2^{p-3}$; otherwise $\Delta(G, H) \geqq \frac{1}{2}\beta \geqq \frac{1}{16}(n+1) \geqq (\sqrt{2}/8)\sqrt{t(n+1)} - \frac{1}{2}t$. Let $\gamma = \lfloor \log \beta \rfloor$. Let K be the set of complete binary subtrees of H rooted at the vertices at level $\gamma + 3$ of H. Since $|K| = 2^{\gamma+3} > 4\beta$, there exist t subtrees $Y^1, Y^2, \ldots, Y^t$ in K, which have no vertex in B. Let $S = \{(u_i, r_i) \mid r_i$ is the root of Y^i, u_i is the father of r_i in H, $i = 1, 2, \ldots, t\}$. $G \backslash S$ contains a subgraph H'' isomorphic to B_n. Since $V(H'') \cap V(Y^i) = \varnothing$ for $i = 1, 2, \ldots, t$, we have that $\alpha \geqq t(2^{p-\gamma-3} - 1)$. It follows that $(\alpha + t)\beta \geqq \frac{1}{8}t(n+1)$, which means that $\max\{\alpha + t, \beta\} \geqq (1/2\sqrt{2})\sqrt{(t(n+1))}$. Thus $\Delta(G, H) \geqq \frac{1}{2}\max\{\alpha, \beta\} \geqq (\sqrt{2}/8)\sqrt{t(n+1)} - \frac{1}{2}t$. □

4. t-FT and t-EFT graphs with respect to C_n.

THEOREM 5. $\Delta(t, C_n) < 6t\sqrt{n} + 12t^2 - 2t$.

Proof. Suppose that the vertices of C_n are labeled $0, 1, 2, \ldots, n-1$ such that $(i, i+1) \in E(C_n) \pmod{n}$ for $i = 0, 1, 2, \ldots, n-1$. Let $q = \lceil \sqrt{n} \rceil$, $m = \lfloor n/q \rfloor$, and

$r = n - qm$. Define $I = \{iq \,(\mathrm{mod}\, n) | i = 0, 1, 2, \ldots, m\} \subseteq V(C_n)$. Let $P^1, P^2, \ldots, P^{2t}$ be vertex disjoint paths with $q - 1$ vertices. Suppose that the vertices of P^i ($i = 1, 2, \ldots, 2t$) are labeled $i_1, i_2, \ldots, i_{q-1}$ such that $(i_j, i_{j+1}) \in E(P^i)$ for $j = 1, 2, \ldots, q - 2$. Define $J_1 = \{i_1 | i = 1, 2, \ldots, 2t\}$, $J_2 = \{i_{q-1} | i = 1, 2, \ldots, 2t\}$, and $J = J_1 \cup J_2$. We construct a graph G from $C_n, P^1, P^2, \ldots, P^{2t}$ by the following procedure:

1. Join each vertex of I to each vertex of J;
2. Construct a complete graph K_{2t} on J_i ($i = 1, 2$), and subdivide each edge of K_{2t} by placing a vertex in the middle of that edge;
3. If $r \neq 0$, join the vertex $0 \in V(C_n)$ to i_{r-1} ($i = 1, 2, \ldots, 2t$).
4. If $r \neq 0$, join the vertex i_{r-1} ($i = 1, 2, \ldots, 2t$) to the vertices added in step 2 and adjacent to i_{q-1}.

We show that G is a t-FT graph with respect to C_n. Let S be a subset of $V(G)$ with $|S| \leqq t$. We show that $G - S$ contains C_n as a subgraph. From the construction of G, we may assume that $S \subseteq V(C_n)$. Each connected component of $C_n - I$ is called an interval. We can denote each interval as the open interval $(iq, (i + 1)q)$ for some i. If $iq, (i + 1)q \notin S$ and $(iq, (i + 1)q)$ has a vertex in S, then we replace $(iq, (i + 1)q)$ by a path in $\{P^1, P^2, \ldots, P^{2t}\}$. If iq or $(i + 1)q$, say $(i + 1)q$, is in S, then we replace $(iq, (i + 1)q)$ together with $((i + 1)q, (i + 2)q)$ by the composition of paths in $\{P^1, P^2, \ldots, P^{2t}\}$. Since the number of intervals to be replaced is at most $2t$, we construct C_n in $G - S$. To construct G, we add at most $4t(m + 1)$ edges in step 1, $4t(2t - 1)$ edges in step 2, at most $2t$ edges in step 3, at most $2t(2t - 1)$ edges in step 4, and the edges in P^i ($i = 1, 2, \ldots, 2t$). Thus $\Delta(G, C_n) \leqq 4t(m + 1) + 4t(2t - 1) + 2t + 2t(2t - 1) + 2t(q - 2) < 4t(\sqrt{n} + 1) + 4t(2t - 1) + 2t + 2t(2t - 1) + 2t(\sqrt{n} - 1) = 6t\sqrt{n} + 12t^2 - 2t$. □

THEOREM 6. $\Delta'(t, C_n) < 2t\sqrt{n} + t$.

Proof. We follow the construction procedure used in the proof of Theorem 5. We will present the construction when $|I|$ is even, where I is defined as in the proof of Theorem 5. The construction when $|I|$ is odd is quite similar and is omitted. We construct a graph G from $C_n, P^1, P^2, \ldots, P^t$ by the following procedure:

1. Join i_1 ($i = 1, 2, \ldots, t$) to jq ($j = 1, 3, \ldots, m - 1$ if $r = 0$, m otherwise);
2. Join i_{q-1} ($i = 1, 2, \ldots, t$) to jq ($j = 0, 2, \ldots, m - 2$ if $r = 0$, $m - 1$ otherwise);
3. If $r \neq 0$, joint 0 to i_{r-1} ($i = 1, 2, \ldots, t$).

It is easy to verify that G is a t-EFT graph with respect to C_n, and $\Delta(G, C_n) < 2t\sqrt{n} + t$. □

It should be noted that the number of vertices added in the procedure above is less than $t\sqrt{n}$.

THEOREM 7. $\Delta'(t, C_n) \geqq \frac{1}{2}\sqrt{tn} - \frac{1}{2}t$.

Proof. Let G be a t-EFT graph with respect to C_n. G contains a subgraph H isomorphic to C_n. Suppose that the vertices of H are labeled $0, 1, 2, \ldots, n - 1$ such that $(i, i + 1) \in E(H)$ (mod n) for $i = 0, 1, 2, \ldots, n - 1$. Define $A = V(G) - V(H)$, $\alpha = |A|$, $B = \{v | v \in V(H), d_G(v) \geqq 3\}$, and $\beta = |B|$.

Case 1. $\beta \leqq t$. Let $S = \{(i, i + 1) \,(\mathrm{mod}\, n) | i \in B\} \subseteq E(G)$. $G \backslash S$ contains a subgraph H' isomorphic to C_n as $|S| \leqq t$. Since H' is edge disjoint from H, $\Delta(G, H) \geqq |E(H')| = n \geqq \frac{1}{2}\sqrt{tn} - \frac{1}{2}t$.

Case 2. $\beta > t$. Each connected component of $H - B$ is a path. Suppose that P^j ($j = 1, 2, \ldots, t$) is the jth longest path among these paths. Let e_j be an edge in P^j ($j = 1, 2, \ldots, t$) and $S = \{e_1, e_2, \ldots, e_t\}$. $G \backslash S$ contains a subgraph H'' isomorphic to C_n. Since $V(H'') \cap V(P^j) = \varnothing$ for $j = 1, 2, \ldots, t$, $\alpha \geqq |A \cap V(H'')| \geqq \sum_{i=1}^{t} |V(P^j)| = t(n - \beta)/\beta$. It follows that $(\alpha + t)\beta \geqq tn$, which means that $\max\{\alpha + t, \beta\} \geqq \sqrt{tn}$. Thus $\Delta(G, H) \geqq \frac{1}{2}\max\{\alpha, \beta\} \geqq \frac{1}{2}\sqrt{tn} - \frac{1}{2}t$. □

COROLLARY 1. $\Delta(t, C_n) \geqq \frac{1}{2}\sqrt{tn} - \frac{1}{2}t$.

Proof. It follows from Property 1 and Theorem 7. □

5. *t*-FT and *t*-EFT graphs with respect to P_n.

THEOREM 8. $\Delta(t, P_n) \leqq 6(t-1)\sqrt{n+1} + 12(t-1)^2 - 2(t-1) + 2$.

Proof. Since a $(t-1)$-FT graph with respect to C_{n+1} is a t-FT graph with respect to P_n, $\Delta(t, P_n) \leqq \Delta(t-1, C_{n+1}) + 2$. □

THEOREM 9. $\Delta'(t, P_n) \leqq 2(t-1)\sqrt{n} + (t-1) + 1$.

Proof. Since a $(t-1)$-EFT graph with respect to C_n is a t-EFT graph with respect to P_n, $\Delta'(t, P_n) \leqq \Delta'(t-1, C_n) + 1$. □

THEOREM 10. $\Delta'(t, P_n) \geqq (\sqrt{2}/4)\sqrt{(t-1)(n+1)} - \frac{1}{4}(t-1)$.

Proof. Let G be a t-EFT graph with respect to P_n. G contains a subgraph H isomorphic to P_n. Suppose that the vertices of H are labeled $0, 1, 2, \ldots, n-1$ such that $(i, i+1) \in E(H)$ for $i = 0, 1, \ldots, n-2$. Define $A = V(G) - V(H)$, $\alpha = |A|$, $B = \{v \mid v \in V(H), d_G(v) > d_H(v)\}$, and $\beta = |B|$.

Case 1. $\beta \leqq t/2$. Let $S = \{e \mid e \in E(H), e$ is incident to some vertex in $B\}$. $G\backslash S$ contains a subgraph H' isomorphic to P_n as $|S| \leqq t$. Since H' is edge disjoint from H, $\Delta(G, H) \geqq |E(H')| = n - 1 \geqq (\sqrt{2}/4)\sqrt{(t-1)(n+1)} - \frac{1}{4}(t-1)$.

Case 2. $\beta > t/2$. Each connected component of $H - B$ is a path. Suppose that P^j $(j = 1, 2, \ldots, \lfloor t/2 \rfloor)$ is the jth longest path among these paths. Let e_{j1} and e_{j2} be edges of H connecting a vertex in P^j and a vertex in B and let $S = \{e_{11}, e_{12}, e_{21}, e_{22}, \ldots, e_{q1}, e_{q2}\}$, where $q = \lfloor t/2 \rfloor$. $G\backslash S$ contains a subgraph H'' isomorphic to P_n. Since $V(H'') \cap V(P^j) = \varnothing$ for $j = 1, 2, \ldots, q$,

$$\alpha \geqq |A \cap V(H'')| \geqq \sum_{j=1}^{q} |V(P^j)| \geqq \frac{q(n-\beta)}{\beta+1} \geqq \frac{\frac{t-1}{2}(n-\beta)}{\beta+1}.$$

It follows that

$$\left(\alpha + \frac{t-1}{2}\right)(\beta+1) \geqq \frac{(t-1)(n+1)}{2},$$

which means that $\max\{\alpha + (t-1)/2, \beta + 1\} \geqq (1/\sqrt{2})\sqrt{(t-1)(n+1)}$. Thus $\Delta(G, H) \geqq \frac{1}{2}\max\{\alpha, \beta\} \geqq (\sqrt{2}/4)\sqrt{(t-1)(n+1)} - \frac{1}{4}(t-1)$. □

By This theorem and Property 1, we have the following corollary.

COROLLARY 2. $\Delta(t, P_n) \geqq (\sqrt{2}/4)\sqrt{(t-1)(n+1)} - \frac{1}{4}(t-1)$.

6. *t*-FT and *t*-EFT graphs with respect to R_n and D_n.

Let $G_1 \times G_2$ be the Cartesian product of graphs G_1 and G_2; that is, $V(G_1 \times G_2) = V(G_1) \times V(G_2)$ and $E(G_1 \times G_2) = \{(u_1v_1, u_2v_2) \mid u_1 = u_2$ and $(v_1, v_2) \in E(G_2)$, or $v_1 = v_2$ and $(u_1, u_2) \in E(G_1)\}$. $R_{r^2} = P_r \times P_r$ is called a (square) grid and $D_{r^2} = C_r \times C_r$ is called a torus. It is easy to verify the following lemma.

LEMMA 1. *If G is a t-FT (respectively, t-EFT) graph with respect to H, then $G \times G$ is a t-FT (respectively, t-EFT) graph with respect to $H \times H$.*

We show the "order" of $\Delta(t, H)$ and $\Delta'(t, H)$ with respect to $|V(H)|$ for R_n and D_n when t is fixed. Although we can give explicit upper bounds by means of t and n as in the previous sections, they are somewhat complicated.

THEOREM 11. *$\Delta(t, D_n)$ and $\Delta'(t, D_n)$ are $O(n^{3/4})$ if t is fixed.*

Proof. This follows from Theorems 5 and 6, the notes following their proofs, and Lemma 1. □

Similarly, we can prove the following.

THEOREM 12. (i) $\Delta(1, R_n)$ *and* $\Delta'(1, R_n)$ *are* $O(\sqrt{n})$.

(ii) $\Delta(t, R_n)$ *and* $\Delta'(t, R_n)$ *are* $O(n^{3/4})$ *if* $t \geqq 2$ *is fixed.*

We conclude this section with the following remarks:

1. It should be noted that the results for B_n can be generalized to the complete k-ary tree for any k;

2. Although we showed sublinear upper bounds for R_n and D_n, we do not know any nontrivial lower bound. We may expect that improvements on our results are possible. In fact, the result in [5] provides some evidence in support of this. Specifically, the result in [5] implies that $\Delta(n, P_n)$ and $\Delta'(n, P_n)$ are $O(n \log n)$, while our upper bounds imply that $\Delta(n, P_n)$ is $O(n^2)$ and $\Delta'(n, P_n)$ is $O(n\sqrt{n})$;

3. Finally, given a connected n-vertex graph H_n and a positive integer $t \geqq 2$, it appears that there may exist a universal lower bound for $\Delta(t, H_n)$. In fact, we conjecture that $\Delta(t, H_n) \geqq c\sqrt{tn} + f(t)$ for a constant $c > 0$. In this connection, it is worth noting that $\Delta(1, P_n) = 2$ and $\Delta(t, mK_2) = t$, where mK_2 is m vertex disjoint union of copies of K_2. Note also that $\Delta'(t, S_n) = t$.

7. 1-EFT graphs for $H = Q_n$ and $H = B_n$ on $|V(H)|$ vertices. Let us consider Problem 4 for $H = Q_n$ and $H = B_n$ when $t = 1$. We will denote the minimum number of edges that must be added to H to make it 1-EFT by $\Delta'(H)$.

(i) $H = Q_n$. We consider first $H = Q_n$, where $n = 2^d$. As usual, we will denote the vertices of Q_n by d-bit strings $(x_1, \dots, x_d)$, with two such strings adjacent if they differ in exactly one component. If v is a vertex of Q_n, the vertex *diametrically opposite* v is the one that differs from v in every component. We say that an edge in Q_n is of dimension k if the end vertices differ in the kth component. We now obtain the precise value of $\Delta'(Q_n)$.

THEOREM 13. $\Delta'(Q_n) = n/2$.

Proof. Certainly, $\Delta'(Q_n) \geqq n/2$, since each vertex must be incident to at least one of the additional edges.

For the reverse inequality, consider the graph G obtained from Q_n by adding the $n/2$ edges between each of the $n/2$ diametrically opposite pairs of vertices. To show that G is 1-EFT for Q_n, suppose that we remove some edge of dimension k in the original Q_n from G. Upon removing from G the remaining edges of dimension k, we obtain two copies $Q'_{n/2}$ and $Q''_{n/2}$ of the $(d-1)$-dimension hypercube $Q_{n/2}$ joined by the $n/2$ new edges. An isomorphism from this remaining subgraph of G to Q_n may be easily described as follows: Each vertex of $Q'_{n/2}$ is mapped to itself, and each vertex of $Q''_{n/2}$ is mapped to the diametrically opposite vertex in $Q''_{n/2}$. □

(ii) $H = B_n$. We now consider $H = B_n$, where $n = 2^p - 1$. Our goal is to establish lower and upper bounds for $\Delta'(B_n)$.

THEOREM 14. $\Delta'(B_n) \geqq \frac{3}{8}(n+1)$.

Proof. Suppose that we add edges to B_n to form a 1-EFT graph G with respect to B_n. We make the following trivial observations:

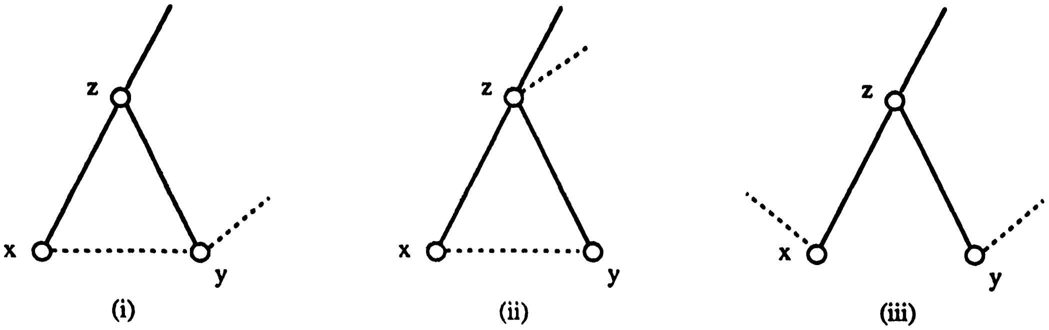
1. Every vertex degree in G must be at least 2;

2. Removing any edge of G must leave neither a vertex of degree 1 adjacent to a vertex of degree at most 2, nor a path of three vertices of degree 2.

Consider now two leaves x, y in the original B_n that are distance 2 apart in B_n and let z denote their father in B_n.

Claim. There must be three or more edges of $E(G) - E(B_n)$ that are incident in G to the vertices $\{x, y, z\}$.

Proof of claim. By observation 1, there must be an edge of $E(G) - E(B_n)$ incident to both x and y (this might be the single edge (x, y)). So if there are at most two edges



(i) (ii) (iii)

FIG. 1

in $E(G) - E(B_n)$ incident to $\{x, y, z\}$, we must have (up to symmetry) one of the situations depicted in Fig. 1, in which broken lines denote edges in $E(G) - E(B_n)$. In Fig. 1, (i) and (ii) (respectively, in (iii)), removing (y, z) (respectively, $(z, \text{father}(z))$) violates observation 2. This proves the claim.

To complete the proof of Theorem 14, note that there are $(n + 1)/4$ such triples $\{x, y, z\}$. Since an edge in $E(G) - E(B_n)$ could be incident to vertices in at most two triples, it follows by the claim that $\Delta'(B_n) \geqq \frac{3}{8}(n + 1)$. □

We now give a constructive upper bound for $\Delta'(B_n)$.

THEOREM 15. $\Delta'(B_n) \leqq \frac{11}{16}(n + 1) - 2$.

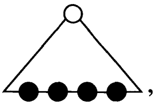
First, we prove the following result.

LEMMA 2. *The graph B in Fig.* 2 *is* 1-EFT *for* B_{15}.

We note that B contains just $6 = \frac{3}{8}(15 + 1)$ extra edges, which is optimal by the above theorem.

Proof. We indicate in Fig. 3 four other B_{15}'s in B, letting the darkened vertex denote the root. Noting that each edge of the original B_{15} fails to occur in at least one of the four, we have the result that B is 1-EFT for B_{15}. □

In the following, we will schematically denote B by

,

where the darkened vertices represent the four roots of the B_{15}'s in Fig. 3 (i)–(iv). We will also need the following easy lemma, whose proof we leave to the reader.

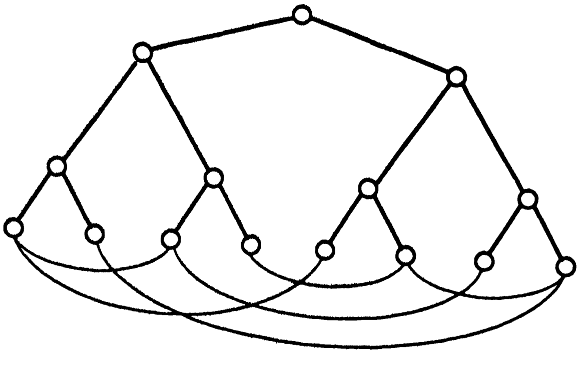
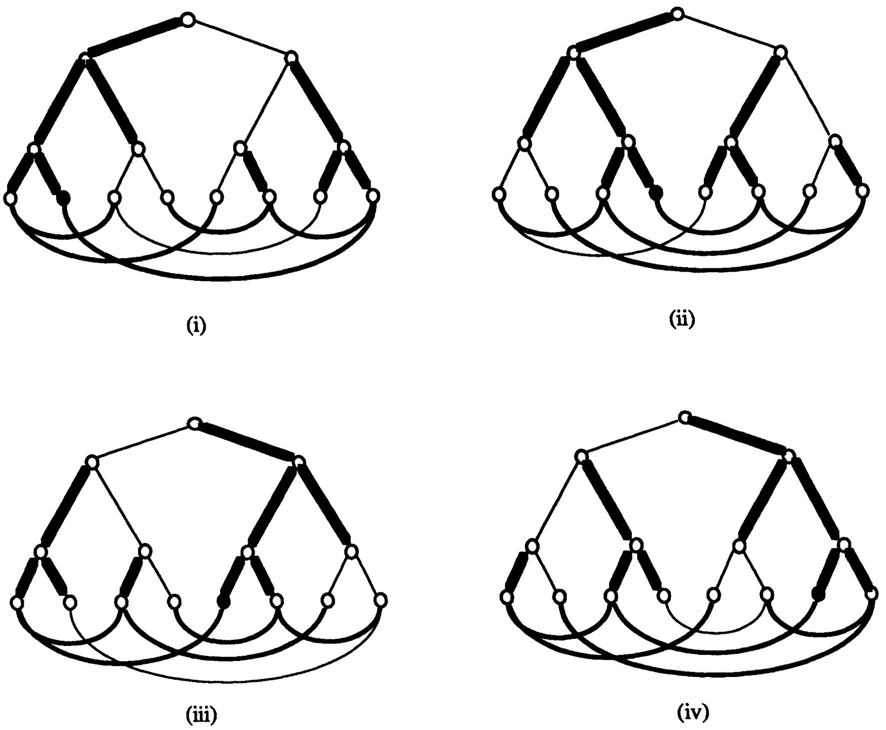


FIG. 2. *The graph B.*


(i)

(ii)

(iii)

(iv)

FIG. 3. *Copies of B_{15} in B.*

LEMMA 3. *For $n \geqq 15$, we can add $n - 1$ edges to B_n (i.e., double the number of edges), so that the resulting graph B'_n is* 1-EFT *for B_n and that regardless of which original edge in B_n is removed from B'_n, the reconfigured B_n has the same root and leaves as the original B_n.*

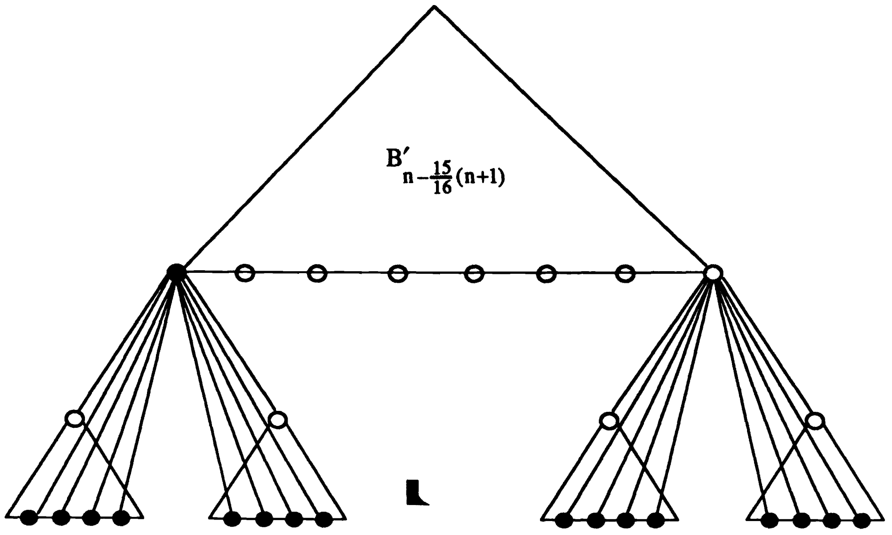


FIG. 4. *The graph G.*

Proof of Theorem 15. Let $n \geqq 2^8 - 1$ and consider the graph G as depicted in Fig. 4. It is easily seen that G is 1-EFT for B_n, and that the number of extra edges is $\frac{11}{16}(n+1) - 2$. □

Acknowledgment. The first author is grateful to Professor Y. Kajitani for his encouragement.

REFERENCES

[1] N. Alon and F. R. K. Chung, *Explicit construction of linear sized tolerant networks*, Discrete Math., 72 (1988), pp. 15–19.

[2] J. Beck, *On Size Ramsey number of paths, trees, and circuits*, I, J. Graph Theory, 7 (1983), pp. 115–129.

[3] G. Chartrand and S. F. Kapoor, *On Hamiltonian properties of powers of special Hamiltonian graphs*, Colloq. Math., 29 (1974), pp. 113–117.

[4] S. Dutt and J. P. Hayes, *Design and reconfiguration strategies for near-optimal k-fault-tolerant tree architectures*, IEEE International Sympos. Fault-Tolerant Computing, 1988, pp. 328–333.

[5] P. Erdös, R. L. Graham, and E. Szemerédi, *On sparse graphs with dense long paths*, Comp. Math. Appl., 1 (1975), pp. 365–369.

[6] A. A. Farrag and R. J. Dawson, *Designing optimal fault-tolerant star networks*, Networks, 19 (1989), pp. 707–716.

[7] ———, *Fault-tolerant extensions of complete multipartite networks*, Proc. 9th International Conference on Distributed Computing Systems, 1989, pp. 143–150.

[8] J. Friedman and N. Pippenger, *Expanding graphs contain all small trees*, Combinatorica, 7 (1987), pp. 71–76.

[9] J. P. Hayes, *A graph model for fault-tolerant computing systems*, IEEE Trans. on Comput., C-25 (1976), pp. 875–883.

[10] C. L. Kwan and S. Toida, *An optimal* 2-FT *realization of binary symmetric hierarchical tree systems*, Networks, 12 (1982), pp. 231–239.

[11] M. Paoli, W. W. Wong, and C. K. Wong, *Minimum k-Hamiltonian graphs* II, J. Graph Theory, 10 (1986), pp. 79–95.

[12] A. L. Rosenberg, *Fault-tolerant interconnection networks, a graph theoretic approach*, in Workshop on Graph-Theoretic Concepts in Computer Science, Trauner Verlag, Linz, 1983, pp. 286–297.

[13] W. W. Wong and C. K. Wong, *Minimum k-Hamiltonian graphs*, J. Graph Theory, 8 (1984), pp. 155–165.

[14] R. M. Yanney and J. P. Hayes, *Distributed recovery in fault-tolerant microprocessor networks*, IEEE Trans. on Comput., C-35 (1986), pp. 871–879.

SIAM J. DISC. MATH.
Vol. 6, No. 4, pp. 575–581, November 1993

IMPROVED SPACE FOR BOUNDED-SPACE, ON-LINE BIN-PACKING*

GERHARD WOEGINGER†

Abstract. The author presents a sequence of linear-time, bounded-space, on-line, bin-packing algorithms that are based on the "HARMONIC" algorithms H_k introduced by Lee and Lee [*J. Assoc. Comput. Mach.*, 32 (1985), pp. 562–572]. The algorithms in this paper guarantee the worst case performance of H_k, whereas they only use $O(\log \log k)$ instead of k active bins. For $k \geqq 6$, the algorithms in this paper outperform all known heuristics using k active bins. For example, the author gives an algorithm that has worst case ratio less than 17/10 and uses only six active bins.

Key words. combinatorial problems, on-line, bin-packing, suboptimal algorithms

AMS subject classifications. 90B35, 90C27

1. Introduction. In the classical one-dimensional bin-packing problem, we are given a list of items $L = (a_1, a_2, \dots, a_n)$, each item $a_i \in (0, 1]$, and we must find a packing of these items into a minimum number of unit-capacity bins. This problem arises in a wide variety of contexts and has been studied extensively since the early 1970s. Since the problem of finding an optimal packing is NP-hard, research has concentrated on approximation algorithms that find near-optimal packings.

Let $\mathrm{OPT}(L)$ and $A(L)$ denote, respectively, the number of bins used by an optimum algorithm and the number of bins used by a heuristic algorithm A to pack the input list L. Then the *worst case performance* of A, denoted by $r(A)$, is defined as

$$\lim_{\mathrm{OPT}(L)\to\infty} \sup_{L} A(L)/\mathrm{OPT}(L).$$

This ratio is customarily used to measure the performance of a heuristic bin-packing algorithm. A bin-packing algorithm is called *on-line* if it packs all items a_i solely on the basis of the sizes of the items a_j, $1 \leqq j \leqq i$ and without any information on subsequent items. A bin-packing algorithm uses *k-bounded space* if, for each item a_i, the choice of bins to pack it into is restricted to a set of k or fewer *active* bins, where each bin becomes active when it receives its first item, but, once it is declared inactive (or *closed*), it can never become active again.

The latter restrictions (on-line and bounded-space) arise in many applications, as in packing trucks at a loading dock or in communicating via channels with bounded buffer size. Essentially, only the following three types of bounded-space, on-line, bin-packing heuristics have been studied:

(i) The Next-k-Fit (NF_k, $k \geqq 2$) introduced in [6] simply puts an item a_i into the lowest indexed of k active bins into which it will fit. If no active bin has room for a_i, the lowest-indexed active bin is closed, and a_i is put into a new opened bin. Csirik and Imreh [2] and Mao [8] proved that $r(\mathrm{NF}_k) = 17/10 + 3/(10k - 10)$ holds;

(ii) The k-bounded Best Fit (BBF_k, $k \geqq 2$) introduced in [4] always places an item into the fullest active bin into which it will fit. If no active bin has enough room, a new bin is started, and the fullest active bin is closed. Csirik and Johnson [4] showed in a very sophisticated proof that, independently of the value of k, $r(\mathrm{BBF}_k) = 17/10$ holds;

* Received by the editors June 12, 1991; accepted for publication (in revised form) September 11, 1992. Part of this work was carried out while the author was visiting József Attila University. This work was partially supported by the Christian Doppler Laboratorium für Diskrete Optimierung.

† TU Graz, Institut für Theoretische Informatik, Klosterwiesgasse 32/II, A-8010 Graz, Austria, (gwoegi@figids02.tu-graz.ac.at).

(iii) The HARMONIC algorithm H_k [7] is based on a special nonuniform partition of the interval (0, 1] into k subintervals (where the partitioning points are $1/2$, $1/3$, $\dots$, $1/k$). To each of these subintervals, there corresponds one active bin, and only items belonging to this subinterval are packed into this bin. If some item does not fit into its assigned bin, this bin is closed, and a new bin is used. Lee and Lee [7] analysed the worst case ratio of H_k. They showed that, as k tends to infinity, $r(H_k)$ tends to $h_\infty \approx 1.69103$.

A summary of the worst case ratios of these heuristics for some small values of k is given in Table 1. Lee and Lee [7] also showed that for any k-bounded-space, on-line, bin-packing algorithm A, $r(A) \geqq h_\infty$ must hold. That means that, asymptotically, H_k is optimal.

In this paper, we present an on-line bin-packing algorithm, called SIMPLIFIED HARMONIC$_k$, or SH_k. By using a better partition of the interval (0, 1] than H_k does, we get a worst case performance of approximately $h_\infty + 10^{-5}$ while using only nine active bins! To reach this worst case performance, H_k had to use 43 active bins, whereas NF_k and BBF_k cannot even come beneath 17/10. Generally, SH_k has a worst case performance that the HARMONIC algorithm cannot reach by using a number of active bins less than doubly exponential in k.

Furthermore, our heuristic SH_6 has worst case ratio beneath 17/10 while using only six active bins. This contradicts a conjecture of Csirik [1].

The paper is organized as follows. Sections 2 and 3 present the results on SH_k for the case where $k = 3m$. Section 4 extends these results to the other values of k, and § 5 gives the discussion.

2. The simplified harmonic algorithm. The following sequence (introduced by Golomb [5]) is essential in the definition and in the analysis of our algorithm:

$$t_1 = 2,$$

$$t_{i+1} = t_i(t_i - 1) + 1 \quad \text{for } i \geqq 1.$$

We will define the algorithm SH_k only for $k = 3m$, $m \geqq 1$. We fix the value of m for this and the next section and consider the following partition $\mathcal{P}_k$ of the unit-interval into

TABLE 1
Asymptotic worst case ratios, rounded to five decimal places.

k	NF_k	BBF_k	H_k	SH_k	Minimum
2	2.00000	1.70000	2.00000	2.00000	1.70000
3	1.85000	1.70000	1.75000	1.75000	1.70000
4	1.80000	1.70000	1.72222	1.72222	1.70000
5	1.77500	1.70000	1.70833	1.70000	1.70000
6	1.76000	1.70000	1.70000	1.69444	1.69444
7	1.75000	1.70000	1.69444	1.69388	1.69388
8	1.74286	1.70000	1.69388	1.69106	1.69106
9	1.73750	1.70000	1.69345	1.69104	1.69104
42	1.70732	1.70000	1.69106	1.69103	1.69103
43	1.70714	1.70000	1.69103	1.69103	1.69103
∞	1.70000	1.70000	1.69103	1.69103	1.69103

$k = 3m$ subintervals:

$$\begin{aligned} A &= (1/2, 1], \\ B_i &= (1/t_i, 1/(t_i - 1)] \quad \text{for } i = 2 \dots m + 1, \\ C_i &= (1/(t_i + 1), 1/t_i] \quad \text{for } i = 2 \dots m, \\ D_i &= (1/(t_{i+1} - 1), 1/(t_i + 1)] \quad \text{for } i = 2 \dots m, \\ E &= (0, 1/t_{m+1}]. \end{aligned}$$

The algorithm SH_k simply proceeds as follows. For each of the k subintervals, it keeps a separate active bin. In this bin, only items belonging to the corresponding subinterval are packed. Now, if the algorithm receives a new item a_i to pack, it first classifies a_i according to the partition $\mathscr{P}_k$. Then it tries to pack a_i into its corresponding active bin. If there is not enough room in the bin, this bin is closed, a new active bin is opened, and a_i is put into it.

Since item classification can be done in $O(\log k)$ time and there are only k active bins at any time, the algorithm runs in $O(n \log k)$. Hence, if we take k to be constant, the time complexity is linear.

3. Worst case analysis of simplified harmonic. We define below a weighting function $W_k(x)$, which we prove has the following two properties. If $W_k(L)$ is the cumulative weight of the pieces in L, then (i) the length of the SH_k-packing of L cannot exceed $W_k(L) + k$ and (ii) $W_k(L)$ cannot exceed Γ_k times the length of an optimum packing, where (remember that $k = 3m$)

$$\Gamma_k = \sum_{i=1}^{m} \frac{1}{t_i - 1} + \frac{t_{m+1}}{(t_{m+1} - 1)^2}$$

holds. From this, it immediately follows that $\mathrm{SH}_k(L) - k \leqq W_k(L) \leqq \Gamma_k \mathrm{OPT}(L)$, and this yields $r(\mathrm{SH}_k) \leqq \Gamma_k$. Define the weighting function $W_k(x)$ as follows:

$$\begin{aligned} W_k(x) &= x + 1/2 && \text{for } 1/2 < x \\ &= x + \frac{1}{t_{i+1} - 1} && \text{for } 1/t_i < x \leqq 1/(t_i - 1) \text{ and } 2 \leqq i \leqq m + 1 \\ &= \frac{t_i + 1}{t_i} \cdot x && \text{for } 1/(t_{i+1} - 1) < x \leqq 1/t_i \text{ and } 2 \leqq i \leqq m \\ &= \frac{t_{m+1}}{t_{m+1} - 1} \cdot x && \text{for } x \leqq 1/t_{m+1}. \end{aligned}$$

An illustration for this weighting function and for the partition of the unit-interval in the case where $k = 6$ is given in Table 2. The following observation immediately follows from the definition of the weighting function.

OBSERVATION 1. *For $i \leqq m$ and $x \leqq 1/t_i$, $W_k(x)/x \leqq (t_i + 1)/t_i$ holds.*

CLAIM 1. $W_k(L) \geqq \mathrm{SH}_k(L) - k$.

Proof. We show that, in the SH_k-packing, every closed bin B has weight at least 1. Together with the k last active bins, this implies the claim. We distinguish the following five cases:

(i) The bin B corresponds to the A-interval. Then it contains a single item of weight greater than 1;

TABLE 2
Illustration for SH$_6$. *The two rightmost columns hold for* closed *bins.*

Type	Interval	Weight (x)	Contents	# Items
A	(1/2, 1]	$x + 1/2$	>1/2	=1
B_2	(1/3, 1/2]	$x + 1/6$	>2/3	=3
C_2	(1/4, 1/3]	$4x/3$	>3/4	=3
D_2	(1/6, 1/4]	$4x/3$	>3/4	4, 5
B_3	(1/7, 1/6]	$x + 1/42$	>6/7	=6
E	(0, 1/7]	$7x/6$	>6/7	>6

(ii) The bin B corresponds to some B_i-interval, $2 \leqq i \leqq m + 1$. Then it contains exactly $t_i - 1$ items, each of size greater than $1/t_i$. Consequently, the total weight of B is at least

$$(t_i - 1) \cdot W_k\left(\frac{1}{t_i} + \varepsilon\right) > (t_i - 1) \cdot \left(\frac{1}{t_i} + \frac{1}{t_{i+1} - 1}\right) = 1;$$

(iii) The bin B corresponds to some C_i-interval, $2 \leqq i \leqq m$. Then it contains exactly t_i items, each of size greater than $1/(t_i + 1)$. This gives a total weight of at least

$$t_i \cdot W_k\left(\frac{1}{t_i + 1}\right) = t_i \cdot \frac{t_i + 1}{t_i} \cdot \frac{1}{t_i + 1} = 1;$$

(iv) The bin B corresponds to some D_i-interval, $2 \leqq i \leqq m$. Since the bin was closed, some item in D_i did not fit into it. Therefore B is at least $t_i/(t_i + 1)$ full and as the weight function is linear on this interval, the total weight is at least

$$\frac{t_i}{t_i + 1} \cdot \frac{t_i + 1}{t_i} = 1;$$

(v) The bin B corrresponds to the E-interval. Analogously to (iv), we see that B is at least $(t_{m+1} - 1)/t_{m+1}$ full and that the total weight is at least 1. □

CLAIM 2. *In any packing of L, the weight of any bin is at most Γ_k. Hence, $W_k(L) \leqq \Gamma_k \mathrm{OPT}(L)$ holds.*

Proof. Consider some fixed bin B that contains items $q_1 \geqq q_2 \geqq \cdots \geqq q_n$. We distinguish two cases.

(i) $q_i \in (1/t_i, 1/(t_i - 1)]$ for $i = 1 \dots m$. We denote by Q the sum $\sum_{i=m+1}^{n} q_i$. Obviously, $Q < 1/(t_{m+1} - 1)$ holds. Now

$$W_k(B) = \sum_{i=1}^{n} W_k(q_i) = \sum_{i=1}^{m} \left(q_i + \frac{1}{t_{i+1} - 1}\right) + \sum_{i=m+1}^{n} W_k(q_i)$$
$$\leqq 1 - Q + \sum_{i=1}^{m} \frac{1}{t_{i+1} - 1} + \frac{t_{m+1}}{t_{m+1} - 1} \cdot Q.$$

It is easy to see that the latter expression becomes maximum when Q takes its maximum value $1/(t_{m+1} - 1)$; in this case, the expression is exactly Γ_k.

(ii) Suppose that $r \leqq m$ is the least i such that $q_i \notin (1/t_i, 1/(t_i - 1)]$, and hence $q_r \leqq 1/t_r$. We denote by Q the sum $\sum_{i=r}^{n} q_i$. Obviously, $Q < 1/(t_r - 1)$ holds, and, by Observation 1, the total weight of all elements $q_r \dots q_n$ is less than or equal to

$(t_r+1)Q/t_r$. Similarly as in (i), this yields

$$W_k(B) \leqq 1 - Q + \sum_{i=1}^{r-1} \frac{1}{t_{i+1}-1} + \frac{t_r+1}{t_r} \cdot Q$$

$$\leqq \sum_{i=1}^{r-1} \frac{1}{t_i - 1} + \frac{t_r+1}{t_r(t_r-1)} = \sum_{i=1}^{r+1} \frac{1}{t_i-1} < \Gamma_k,$$

and the proof of Claim 2 is complete. □

LEMMA 1. *For $k = 3m$, the asymptotic worst case ratio of the heuristic* SIMPLIFIED HARMONIC$_k$ *is* Γ_k.

Proof. Claims 1 and 2 imply that $r(\mathrm{SH}_k) \leqq \Gamma_k$ holds. To show that the bound is tight, we present a family of lists L_n. We define

$$\alpha_m = (t_{m+1} - 1)(t_{m+1} - 2).$$

Now let n be a multiple of α_m. The optimum packing of our list L_n will use $n + 1$ bins, and the SH$_k$-packing will use $n\Gamma_k$ bins. We choose two very small positive reals ε and δ such that

$$(m+1)\alpha_m \cdot \varepsilon + \delta \leqq \alpha_m / n.$$

We define L_n by giving its optimum packing. In this packing, we have bins of the following contents:

n/α_m times a bin that contains

$$1/t_i + \varepsilon \text{ (for } i = 1 \dots m), \quad 1/t_{m+1}, \quad 1/(t_{m+2} - 1) - (m+1)\varepsilon;$$

$(\alpha_m - 1)n/\alpha_m$ times a bin that contains

$$1/t_i + \varepsilon \text{ (for } i = 1 \dots m), \quad 1/t_{m+1} + \varepsilon, \quad 1/(t_{m+2} - 1) - (m+1)\varepsilon;$$

a single bin containing

$$(n/\alpha_m)\text{-times an item of size } (m+1)\alpha_m \cdot \varepsilon + \delta.$$

In the SH$_k$-packing, the packing of the items of size $1/t_i + \varepsilon$, $1 \leqq i \leqq m$ is easy to analyse. Independently of their ordering, they use exactly $n/(t_i - 1)$ bins. Analogously, we see that the items of size $1/t_{m+1} + \varepsilon$ are packed into $(\alpha_m - 1)/\alpha_m \cdot n/(t_{m+1} - 1)$ bins. Thus, the only interesting items are the items that are $\leqq 1/t_{m+1}$ (i.e., the E-items). These are given to SH$_k$ in $n\alpha_m$ "packages" of the following type:

$$\frac{1}{t_{m+1}}, \quad \underbrace{\frac{1}{t_{m+2}-1} - (m+1)\varepsilon,}_{\alpha_m\text{-times}} \quad (m+1)\alpha_m \cdot \varepsilon + \delta.$$

We show that SH$_k$ puts each package into a separate bin. This holds, since the total size of a package is exactly $(t_{m+1} - 1)/t_{m+1} + \delta$. Hence, when the first item of the next package arrives, it does not fit into the active bin. Consequently, the bin is closed, and the next package is treated in the same way. Summarizing, SH$_k$ uses a total number of

$$\sum_{i=1}^{m} \frac{n}{t_i - 1} + \frac{\alpha_m - 1}{\alpha_m} \cdot \frac{n}{t_{m+1} - 1} + \frac{n}{\alpha_m}$$

bins, and this number is equal to $n\Gamma_k$. □

4. Main results. In this section, we extend the results of the preceding sections to the cases where $k = 3m + 1$ and $k = 3m - 1$. The underlying set T of partitioning points of the unit-interval is given by

$$T = \bigcup_{i=1}^{\infty} \left\{ \frac{1}{t_i + 1}, \frac{1}{t_i}, \frac{1}{t_i - 1} \right\}.$$

To define the algorithm SH_k for general k, we take the $k - 1$ largest values in T. These values partition the interval $(0, 1]$ into k subintervals. SH_k keeps for each subinterval a separate active bin and proceeds exactly as in the preceding sections. It is easy to verify that for $k = 3m$ this indeed leads to our old algorithm. Before we can state our main theorem, we give the following definitions, for $m \geqq 1$:

$$\Gamma_{3m-1} = \sum_{i=1}^{m} \frac{1}{t_i - 1} + \frac{1}{t_{m+1} - 2},$$

$$\Gamma_{3m} = \sum_{i=1}^{m} \frac{1}{t_i - 1} + \frac{t_{m+1}}{(t_{m+1} - 1)^2},$$

$$\Gamma_{3m+1} = \sum_{i=1}^{m} \frac{1}{t_i - 1} + \frac{t_{m+1}^2 + t_{m+1} + 1}{t_{m+1}^2 (t_{m+1} - 1)}.$$

THEOREM 1. *For $k \geqq 2$, the asymptotic worst case ratio of the heuristic* SIMPLIFIED HARMONIC$_k$ *is Γ_k.*

The proofs are analogous to the proofs of Claims 1 and 2 and to the proof of Lemma 1. Essentially, we use the same weighting function again. The only modification concerns elements x in the smallest interval $(0, a]$; these elements always get weight $x/(1 - a)$. The details are left to the reader as an exercise. Some values of Γ_k for some small k are given in the fourth column of Table 1.

Finally, we compare the behaviour of the heuristics H_k and SH_k. Theorem 2 in [7] states that, for $k = t_{m+1} - 1$,

$$r(\mathrm{H}_k) = \sum_{i=1}^{m} \frac{1}{t_i - 1} + \frac{1}{t_{m+1} - 2}.$$

This value is equal to our Γ_{3m-1}. Consequently, HARMONIC using $t_{m+1} - 1$ active bins and SIMPLIFIED HARMONIC using $3m - 1$ active bins achieve the same asymptotic worst case ratio. As the t_i grow doubly exponentially, the following theorem holds.

THEOREM 2. *To achieve the worst case performance of heuristic* HARMONIC$_k$ *with k active bins, the heuristic* SIMPLIFIED HARMONIC *only has to use $O(\log \log k)$ active bins.*

5. Discussion. In this paper, we derived a sequence of new k-bounded-space, online, bin-packing algorithms called SIMPLIFIED HARMONIC$_k$. For $k \geqq 6$, the worst case behaviour of our algorithms outperforms all known heuristics using k active bins. For $k \leqq 4$, the best-known algorithms are the BBF_k due to Csirik and Johnson. For $k = 5$, BBF_5 and SH_5 both have the same worst case performance.

The *average* performance of SIMPLIFIED HARMONIC$_k$ suffers from the usual drawback of harmonic algorithms: For L_n, a random list of n items with sizes chosen independently from a uniform distribution, the average value $\mathrm{H}_k(L)/\mathrm{OPT}(L)$ approaches 1.28987 as k tends to ∞ (see [3]), whereas the average value of $\mathrm{NF}_k(L)/\mathrm{OPT}(L)$ empirically approaches 1. Computational experiments performed on large item lists indicate that, in the average case, SH_k performs as poorly as H_k does.

There remains a number of (seemingly hard) open questions.

(1) What is the best possible worst case performance of any on-line, bin-packing heuristic using 2-bounded space? (BBF_2 achieves a worst case ratio of 17/10.)

(2) What is the smallest k such that there exists an on-line, bin-packing heuristic using k-bounded space with asymptotic worst case ratio strictly less than 17/10? (SH_6 comes beneath 17/10 by using 6-bounded space.)

(3) If we only consider algorithms that pack the items by Next-Fit according to some fixed partition of $(0, 1]$ into k subintervals, which partition gives the best worst case ratio? (It is easy to see that, for $k = 1$ and $k = 2$, in this case the best possible worst case performance is 2, but for $k \geqq 3$ no tight bounds are known.)

Acknowledgments. I gratefully acknowledge the hospitality of József Attila. Furthermore, I thank János Csirik, Gábor Galambos, and Hannes Hassler for several helpful discussions.

REFERENCES

[1] J. Csirik, private communication, 1991.

[2] J. Csirik and B. Imreh, *On the worst-case performance of the* NkF *bin-packing heuristic*, Acta Cybernetica, 9 (1989), pp. 89–105.

[3] J. Csirik, J. B. G. Frenk, A. Frieze, G. Galambos, and A. H. G. Rinnooy Kan, *A probabilistic analysis of the next fit decreasing bin packing heuristic*, Oper. Res. Lett., 5 (1986), pp. 233–236.

[4] J. Csirik and D. S. Johnson, *Bounded space on-line bin packing: Best is better than first*, in Proc. 2nd Annual ACM-SIAM Sympos. on Discrete Algorithms, San Francisco, CA, January 1991.

[5] S. Golomb, *On certain non-linear recurring sequences*, Amer. Math. Monthly, 70 (1963), pp. 403–405.

[6] D. S. Johnson, *Fast algorithms for bin packing*, J. Comput. System Sci., 8 (1974), pp. 272–314.

[7] C. C. Lee and D. T. Lee, *A simple on-line bin-packing algorithm*, J. Assoc. Comput. Mach., 35 (1985), pp. 562–572.

[8] W. Mao, *Tight worst-case performance bounds for Next-k-Fit bin packing*, SIAM J. Comput., 22 (1993), pp. 46–56.

SIAM J. DISCRETE MATH.
Vol. 6, No. 4, pp. 582-611, November 1993

SURJECTIVE EXTENSIONS OF SLIDING-BLOCK CODES*

JONATHAN ASHLEY†, BRIAN MARCUS†, DOMINIQUE PERRIN‡, AND SELIM TUNCEL§

Abstract. Several constructions are presented for extending a bounded-to-one sliding-block code to a bounded-to-one surjection onto its range, while preserving nice properties of the original code.

Key words. sliding-block code, factor map, extension, shift of finite type

AMS subject classifications. primary 68R10, 94A24; secondary 05C20, 58F11

1. Introduction. Motivated by some results from the theory of codes, we investigate the following problem: Given a bounded-to-one k-block map $\phi : S \to T$ between symbolic shift spaces S and T, construct an enlarged domain $\bar{S} \supseteq S$ and a surjective bounded-to-one extension $\bar{\phi} : \bar{S} \to T$ of the given map ϕ. We present several constructions that apply under different hypotheses on the given k-block map $\phi : S \to T$; each construction preserves some desirable property of ϕ. In §2 we extend a general bounded-to-one block map between shifts of finite type (SFTs) to a bounded-to-one factor map between SFTs. This is linked to the imbedding of a code into a maximal code, given by Ehrenfeucht and Rozenberg [10]. In §3 we extend a right-closing block map between SFTs to a right-closing factor map between SFTs, and in §4 we extend a biclosing block map between SFTs to a biclosing factor map between SFTs. In §5 we prove a sofic version of the general bounded-to-one extension theorem. Our last two constructions deal with the simultaneous extension of two block maps $\phi : S \to T$ and $\psi : S \to T'$. In §6 we simultaneously extend two bounded-to-one maps between SFTs, and in §7 we simultaneously extend two right-closing maps between SFTs.

In each construction, with the exception of that of §5, the map ϕ is assumed one-block with shift of finite type domain $\Sigma = \Sigma_{\mathcal{G}}$ given by bi-infinite walks on some graph $\mathcal{G}$. In this setting, our extensions are constructed by imbedding the graph $\mathcal{G}$ into a graph $\bar{\mathcal{G}}$ and extending ϕ, regarded as a graph homomorphism, to all of $\bar{\mathcal{G}}$. We depart from this theme in §5, where the domain is sofic.

The constructions have roots both in the theory of codes and in symbolic dynamics. The first extension construction relies almost entirely on the theory of codes, while the following constructions become progressively more dependent on symbolic dynamics.

As we hope to interest both coding theorists and symbolic dynamicists in these results, we provide here the necessary definitions and background from both fields. Our primary reference for the theory of codes is [3]. For symbolic dynamics, they are [1] and [6].

Given a finite set $\mathcal{A}$ of *letters* or *symbols*, we define a *word* w over $\mathcal{A}$ to be a finite sequence $(a_1 \dots a_n)$ of symbols from $\mathcal{A}$ and we denote $w = a_1 \dots a_n$. We form the *concatenation* uv of two words $u = a_1 a_2 \dots a_n$ and $v = b_1 b_2 \dots b_m$ by juxtaposing them as follows: $uv = a_1 \dots a_n b_1 \dots b_m$. The empty sequence, denoted by ϵ, is the identity element for the operation of concatenation.

The set of all words over $\mathcal{A}$, denoted by $\mathcal{A}^\star$, is called the *free monoid* over $\mathcal{A}$. If $X \subset \mathcal{A}^\star$, we denote the set of all concatenations of elements of X (including the empty

[*] Received by the editors February 19, 1991; accepted for publication (in revised form) August 12, 1992.

[†] IBM Research Division, Almaden Research Center, 650 Harry Road, San Jose, California 95120.

[‡] LITP, 2 Place Jussieu 7.5251, Paris Cedex 05, France. This work was done while the author was visiting IBM Almaden.

[§] Department of Mathematics, University of Washington, Seattle, Washington 98195. This work was partially supported by National Science Foundation grant DMS-9004253.

concatenation, ϵ) by $X^\star$. We denote the set of all nonempty concatenations by X^+. A subset $X \subset \mathcal{A}^\star$ is a *code* over $\mathcal{A}$ if each element of X^+ has a unique factorization into words from X. As a particular case, a *prefix code* is one in which no code word is a prefix of another.

A *graph* $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ is defined by a finite set $\mathcal{S}$ of *states* and a finite set $\mathcal{E}$ of *edges*, where each edge has an *initial state* and a *terminal state* in $\mathcal{S}$. We allow multiple edges from one state to another in $\mathcal{G}$. An *n-path* of $\mathcal{G}$ is a sequence of edges $x_1 \dots x_n$ of $\mathcal{G}$, where the terminal state of edge x_i is the initial state of edge x_{i+1} for $1 \leq i \leq n-1$. The path $x_1 \dots x_n$ *begins* at the initial state of edge x_1 and *ends* at the terminal state of edge x_n.

We denote by ϵ_s the *empty path of $\mathcal{G}$ based at state* $s \in \mathcal{S}$. The terminal and initial states of ϵ_s are both state s, and $u\epsilon_s v$ is a path of $\mathcal{G}$ if and only if path u ends and path v begins at state s. The path ϵ_s is a 0-path.

We often regard the edges $\mathcal{E}$ of a graph $\mathcal{G}$ as symbols. In this connection, a nonempty path in $\mathcal{G}$ is an element of $\mathcal{E}^+$.

A *labeling* of a graph $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ is a function $\delta : \mathcal{E} \to \mathcal{B}$, where $\mathcal{B}$ is a set of symbols. We can regard δ as a function from the nonempty paths of $\mathcal{G}$ into $\mathcal{B}^+$: we say path $x_1 \dots x_n$ has *δ-label* $\delta(x_1 \dots x_n) = \delta(x_1) \dots \delta(x_n)$. A labeling δ of $\mathcal{G}$ is *deterministic* if, for each state s of $\mathcal{G}$, δ is one-to-one on the set of edges beginning at state s.

A *finite automaton* M is defined by $(\mathcal{S}, \mathcal{E}, \mathcal{B}, \delta, q_0, \mathcal{F})$, where $(\mathcal{S}, \mathcal{E})$ is a graph, $\mathcal{B}$ is a finite alphabet, $\delta : \mathcal{E} \to \mathcal{B}$ is a labeling, $q_0 \in \mathcal{S}$ is the *start state*, $\mathcal{F} \subset \mathcal{S}$ is the set of *accepting states*. A *deterministic* finite automaton (DFA) has a deterministic labeling.

We say the set $L \subset \mathcal{B}^\star$ is *recognized by* M if L is the set of labelings of paths in M starting at state q_0 and ending at an accepting state. By a construction of Rabin and Scott [16], any set $L \subset \mathcal{B}^\star$ accepted by finite automaton is also accepted by a *deterministic* finite automaton. Such a set L is called a *regular language* or a *rational language*. In particular, if a code is regular as a language, we say that the code is a regular code. It is well known that the set of regular languages is closed under Boolean operations (union, intersection, complement in $\mathcal{B}^\star$) and closed under concatenation and the Kleene-star operation (if L is regular, then so is $L^\star$). The set of regular languages over an alphabet $\mathcal{B}$ is the closure under the operations of union, concatenation, and Kleene star of the set of single symbol languages $\{b\}$, $b \in \mathcal{B}$ [12].

If $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ is a finite directed graph and $s_0 \in \mathcal{S}$, then it is easy to see that the set $L \subseteq \mathcal{E}^\star$ of nonempty paths starting and ending at state s_0 but not passing through state s_0 is a regular language and a code. We call L the set of *first returns* to state s_0 in $\mathcal{G}$.

We now give the necessary background from symbolic dynamics. We denote

$$\mathcal{A}^{\mathbb{Z}} = \{(\dots x_{-1} x_0 x_1 \dots) : x_i \in \mathcal{A}\}.$$

The elements of $\mathcal{A}^{\mathbb{Z}}$ are the bi-infinite sequences of symbols from $\mathcal{A}$. The map $\sigma : \mathcal{A}^{\mathbb{Z}} \to \mathcal{A}^{\mathbb{Z}}$ defined $(\sigma x)_i = x_{i+1}$ is called the *shift map* on $\mathcal{A}^{\mathbb{Z}}$. A shift-invariant subset $\Sigma \subseteq \mathcal{A}^{\mathbb{Z}}$ is an *n-step shift of finite type* if membership of $x \in \mathcal{A}^{\mathbb{Z}}$ in Σ can be determined by examining the words of length $n+1$ occurring in x: There is a set of words $\mathcal{W} \subset \mathcal{A}^{n+1}$ with

$$\Sigma = \{x \in \mathcal{A}^{\mathbb{Z}} : x_i \dots x_{i+n} \in \mathcal{W} \text{ for all } i \in \mathbb{Z}\}.$$

An example is $\mathcal{A}^{\mathbb{Z}}$ itself: it is a 0-step shift of finite type.

Of particular interest are the shifts of finite type constructed from square nonnegative integer matrices. Given such a matrix A indexed by a set $\mathcal{S}$, define a finite directed graph $\mathcal{G}_A = (\mathcal{S}, \mathcal{A})$ such that A_{ij} edges point from state i to state j. Define the (one-step) shift of finite type Σ_A as

$$\Sigma_A = \left\{x \in \mathcal{A}^{\mathbb{Z}} : \text{edge } x_{i+1} \text{ follows edge } x_i \text{ in the graph } \mathcal{G}_A\right\}.$$

It is sometimes convenient to define an SFT $\Sigma_{\mathcal{G}}$ directly in terms of a graph $\mathcal{G}$, below:

$$\Sigma_{\mathcal{G}} = \left\{x \in \mathcal{E}^{\mathbb{Z}} : \text{edge } x_{i+1} \text{ follows edge } x_i \text{ in the graph } \mathcal{G} = (\mathcal{S}, \mathcal{E})\right\}.$$

In a certain sense, it involves no loss of generality to confine our study to SFTs defined by matrices if we allow recoding by a *conjugacy*, a notion we now explain.

Given a shift-invariant subset $S \subseteq \mathcal{A}^{\mathbb{Z}}$, an *n-block map* $\phi : S \to \mathcal{B}^{\mathbb{Z}}$ is a function satisfying the following conditions: $\phi \circ \sigma = \sigma \circ \phi$ (shift commuting), and there is an integer l such that, if $x_l x_{l+1} \dots x_{l+n-1} = y_l y_{l+1} \dots y_{l+n-1}$, then $\phi(x)_0 = \phi(y)_0$. Thus each coordinate $\phi(x)_i$ of $\phi(x)$ can be determined by examining a window

$$x_{l+i} x_{l+i+1} \dots x_{l+i+n-1}$$

of x of length n. Such maps are called *block maps* or *sliding-block codes*.

Of particular interest are one-block maps $\phi : \Sigma_A \to \Sigma_B$. Such a one-block map ϕ defines a labeling from the edges of $\mathcal{G}_A$ to the edges of $\mathcal{G}_B$ that we again call ϕ. What is less obvious, ϕ also defines a function from the *states* of $\mathcal{G}_A$ to the *states* of $\mathcal{G}_B$ via

$$\phi(\text{initial state of edge } e) = \text{initial state of edge } \phi(e).$$

We must check that ϕ is well defined and that ϕ, as a function of edges to edges and states to states, is a *graph homomorphism*: ϕ respects the initial and final states of edges.

It is a fundamental observation [11] that block maps are exactly the shift-commuting continous maps when we regard $\mathcal{A}^{\mathbb{Z}}$ and $\mathcal{B}^{\mathbb{Z}}$ as metric spaces with distance formula

$$d(x, y) = \sum_{i \in \mathbb{Z}} \delta(x_i, y_i) \cdot 2^{-|i|},$$

where

$$\delta(a, b) = \begin{cases} 0 & \text{if } a = b, \\ 1 & \text{if } a \neq b. \end{cases}$$

All the shift-invariant subsets of the full shift $\mathcal{A}^{\mathbb{Z}}$ that we consider in this paper are *closed* subsets of $\mathcal{A}^{\mathbb{Z}}$, when $\mathcal{A}^{\mathbb{Z}}$ is regarded as a metric space. Such subsets are called *subshifts*.

If $\phi : S \to T$ is a one-to-one and onto block map between subshifts, it follows from a simple general topological argument that $\phi^{-1} : T \to S$ is also a block map. We call such a ϕ a *conjugacy*, because it conjugates the shift map on S to the shift map on T: $\sigma_T = \phi \circ \sigma_S \circ \phi^{-1}$.

It is in this regard that the study of shifts of finite type can be reduced to the study of those defined by nonnegative integer matrices. If Σ is an n-step SFT where $n > 1$, then we define a one-step SFT $\Sigma^{[n]}$ conjugate to Σ by letting the symbols of $\Sigma^{[n]}$ be the words of length n that occur in points of Σ and allowing symbol $b_1 \dots b_n$ to follow symbol $a_1 \dots a_n$ in a point of $\Sigma^{[n]}$ if and only if $a_2 \dots a_n = b_1 \dots b_{n-1}$. It is easy to see that $\Sigma^{[n]}$ is conjugate to Σ. Given a one-step SFT Λ, it is easy to define a 0-1 matrix A such that Λ is conjugate to Σ_A: index A by the symbols of Λ and set $A_{ij} = 1$ if and only if ij occurs in Λ. Thus, Σ is conjugate to Σ_A.

A *factor map* is a surjective block map, and a *factor* of a subshift is its image under a factor map. A *sofic system* [18] is a subshift that is a factor of an SFT.

A directed graph $\mathcal{G}$ is *irreducible* if, for each pair of states s, t in $\mathcal{G}$, there is a path in $\mathcal{G}$ from s to t. An SFT Σ is *irreducible* if, for each pair of words u, v occurring in Σ, there is a word w such that uwv occurs in Σ. A sofic system is irreducible if it is the factor of an irreducible SFT.

An SFT Σ is *aperiodic* if there is an $l > 0$ such that, for each pair of words u, v occurring in Σ, there is a word w such that $|w| = l$ and uwv occurs in Σ.

An *irreducible component* of an SFT Σ is an irreducible SFT $\Sigma_0 \subseteq \Sigma$ that is maximal with respect to inclusion among all irreducible SFTs contained in Σ. It is not hard to show that any SFT has a finite number of irreducible components.

The *entropy* $h(A)$ of a shift space A is the asymptotic growth rate of words of length n in A, below:

$$h(A) = \lim_{n \to \infty} \frac{1}{n} \log |\text{words of length } n \text{ in } A| .$$

For an SFT Λ, the entropy $h(\Lambda)$ is also the asymptotic growth rate of the number $\Pi_n(\Lambda)$ of periodic points of least period n in Λ.

We use the following two lemmas from [1].

LEMMA 1.1 (see [1, (3.24)]). *If Σ is an SFT and $\Sigma_0, \ldots \Sigma_n$ are its irreducible components, then*

$$h(\Sigma) = \max_{0 \le i \le n} h(\Sigma_i).$$

LEMMA 1.2 (see [8], [1, (3.21)]). *If Σ is an irreducible SFT and $\Lambda \subseteq \Sigma$ is any shift space with $h(\Lambda) = h(\Sigma)$, then $\Lambda = \Sigma$.*

A block map $\phi : A \to B$ is *bounded-to-one* if $\sup_{y \in B} \left|\phi^{-1}(y)\right|$ is finite.

The following theorem is due to Coven and Paul [9].

THEOREM 1.3. *A factor map $\phi : S \to T$ between irreducible sofic systems S and T is bounded-to-one if and only if $h(T) = h(S)$.*

A *diamond* for a k-block map $\phi : S \to T$ is a pair of distinct points $x, y \in S$ with a common left-infinite tail, a common right-infinite tail, and with $\phi(x) = \phi(y)$.

THEOREM 1.4 (see [13], [8]). *A block map $\phi : \Sigma \to T$ with irreducible SFT domain Σ is bounded-to-one if and only if ϕ has no diamonds.*

If the edges of a directed graph $\mathcal{G}_A = (\mathcal{S}, \mathcal{E})$ are labeled by a function $\phi : \mathcal{E} \to \mathcal{B}$, then ϕ defines a one-block map $\phi : \Sigma_A \to \mathcal{B}^{\mathbb{Z}}$. The function ϕ has no diamonds if and only if any two distinct paths in $\mathcal{G}_A$ with the same initial state and same final state have distinct labels.

THEOREM 1.5. *Let $\mathcal{G}_A$ be an irreducible graph. Let $\phi : \Sigma_A \to T$ be the one-block map defined by a labeling of $\mathcal{G}_A$. Let R be the set of first returns in $\mathcal{G}_A$ to a fixed state i_0. The following are equivalent*:

(i) *ϕ is bounded-to-one*;
(ii) *ϕ has no diamonds*;
(iii) *ϕ is one-to-one on $R^\star$*;
(iv) *ϕ is one-to-one on R and $\phi(R)$ is a code.*

The equivalence (i) $\Leftrightarrow$ (ii) is Theorem 1.4; (ii) $\Leftrightarrow$ (iii) and (iii) $\Leftrightarrow$ (iv) are easy to see. If any of the equivalent conditions of Theorem 1.5 hold, we call the labeling given by ϕ *unambiguous*.

2. Bounded-to-one maps between SFTs. In this section, we prove the following theorem. It can be viewed as a generalization in terms of symbolic dynamics of a result due to Ehrenfeucht and Rozenberg [10]. Our proof closely follows their proof recapitulated in [3, Chap. 1, Prop. 5.2].

THEOREM 2.1. *Let $\mathcal{G}_A$ and $\mathcal{G}_B$ be irreducible graphs and let the one-block map $\phi : \Sigma_A \to \Sigma_B$ defined by an edge-labeling of $\mathcal{G}_A$ be bounded-to-one. Then there is an irreducible graph $\mathcal{G}_{\bar{A}} \supseteq \mathcal{G}_A$ with a labeling extending that of $\mathcal{G}_A$ defining a bounded-to-one one-block factor map $\bar{\phi} : \Sigma_{\bar{A}} \to \Sigma_B$.*

COROLLARY 2.2. *If Σ and Λ are irreducible SFTs and $\phi : \Sigma \to \Lambda$ is a bounded-to-one block map, then there is an irreducible SFT $\bar{\Sigma} \supseteq \Sigma$ and a bounded-to-one factor map $\bar{\phi} : \bar{\Sigma} \to \Lambda$ extending ϕ.*

Before proving Theorem 2.1, we need a *marker* path y, as provided by the following lemma.

LEMMA 2.3. *Given a state i in a irreducible graph $\mathcal{G}_B$ and a proper subshift $\Lambda \subset \Sigma_B$, there is a path y in $\mathcal{G}_B$ starting and ending at state i such that*

(i) *y occurs in no point of Λ,*

(ii) *y has no nontrivial self-overlaps (we say y is unbordered).*

Proof. Since Λ is a *proper* subshift of Σ_B, there is a word u occurring in some point of Σ_B that occurs in no point of Λ. Now $\mathcal{G}_B$ is irreducible, so by prepending and appending words to u, we can assume that u starts and ends at state i. Now (uniquely) parse u as $u = w_1 w_2 \dots w_k$, where each w_j is a first return in $\mathcal{G}_B$ to state i. As Σ_B is irreducible and as Λ is a proper subshift of Σ_B, we can choose a first return w to state i that is distinct from w_1. Define $y = uw_1w^k$.

Suppose that t is nonempty, t is a prefix of y, and t is a suffix of y. To show that y is unbordered, we prove $t = y$. The path t begins and ends at state i, so t can be uniquely parsed into first returns to state i, below:

$$t = v_1 v_2 \dots v_l,$$

where $v_1 = w_1$ (since t is a prefix of y) and $l \leq 2k + 1$. Since t is a suffix of y, t is the concatenation of the *last* l first returns to i in y. If $l \leq k$, then $t = w^l$ giving the contradiction $w_1 = w \neq w_1$; so $l \geq k+1$ and $t = w_1 w_2 \dots w_k w_1 w^{l-k-1}$. If $l-k-1 < k$, then $w_1 = w$ (because t is a suffix of y), so $l-k-1 \geq k$. Thus $l \geq 2k+1$, giving $l = 2k+1$ and $t = y$. Thus y is unbordered. □

Proof of Theorem 2.1. We can assume that ϕ is not onto. Fix a state α in $\mathcal{G}_A$. Let $R \subseteq \mathcal{A}^\star$ be the set of first returns to state α in $\mathcal{G}_A$, let $S \subseteq \mathcal{B}^\star$ be the set of first returns to state $\phi(\alpha)$ in $\mathcal{G}_B$, and let $X = \phi(R) = \{\phi(x) : x \in R\}$. Note that $X^\star \subseteq S^\star$ and that $X^\star$ and $S^\star$ are regular languages. Using Lemma 2.3, choose a path $y \in S^\star$ such that y occurs in no point of $\phi(\Sigma_A)$ and y is unbordered. Define a regular language

$$U = S^\star \setminus (X^\star \cup \mathcal{B}^\star y \mathcal{B}^\star).$$

Let $M = (\mathcal{R}, \mathcal{E}, \mathcal{B}, \delta, q_0, \mathcal{F})$ be a DFA that recognizes U. Imbed the graph $\mathcal{G}_A$ into a graph $\mathcal{G}_{\bar{A}}$ and extend the labeling on $\mathcal{G}_A$ to a labeling on $\mathcal{G}_{\bar{A}}$ as follows (see Fig. 1):

(1) From state α, draw a new path $p_{\alpha\alpha}$ labeled y returning to state α;

(2) From state α, draw a new path $p_{\alpha\beta}$ labeled y terminating at a new state β;

(3) From state β, draw a copy $\mathcal{H}$ of the graph $(\mathcal{R}, \mathcal{E})$ with its labeling δ, with state β identified with state $q_0 \in \mathcal{R}$;

(4) For each accepting state $s \in \mathcal{F}$, draw a path $p_{s\beta}$ labeled y from its copy in $\mathcal{H}$ to the state β;

(5) For each accepting state $s \in \mathcal{F}$, draw a path $p_{s\alpha}$ labeled y from its copy in $\mathcal{H}$ to the state α.
Let $\Sigma_{\bar{A}}$ be the SFT defined by the graph $\mathcal{G}_{\bar{A}}$ and let $\bar{\phi} : \Sigma_{\bar{A}} \to \mathcal{B}^\star$ be the one-block map defined by the labeling of $\mathcal{G}_{\bar{A}}$.

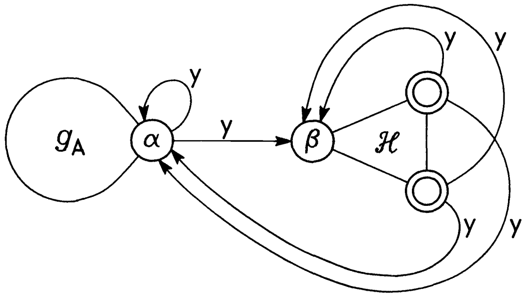


FIG. 1. $\mathcal{G}_{\bar{A}}$.

We first show that $\bar{\phi}$ really maps into Σ_B. Let $\bar{R}$ be the set of first returns to state α in $\mathcal{G}_{\bar{A}}$ and let $\bar{X} = \bar{\phi}(\bar{R})$. Then $\bar{X} = X \cup y(Uy)^\star$ because any first return to state α in $\mathcal{G}_{\bar{A}}$ either (i) remains in $\mathcal{G}_A$ and therefore has label in X, (ii) is $p_{\alpha\alpha}$ and therefore has label y, (iii) passes through state β $k \geq 1$ times and therefore has label in $y(Uy)^k$.
Since $y \in S^\star$, $X \subseteq S^\star$, and $U \subseteq S^\star$, it follows that $\bar{X}^\star \subseteq S^\star$. By the irreducibility of $\mathcal{G}_{\bar{A}}$, any word u occurring in $\Sigma_{\bar{A}}$ occurs in a word w in $\bar{R}^\star$. Now $\bar{\phi}(w) \in \bar{\phi}(\bar{R})^\star = \bar{X}^\star \subseteq S^\star$; so $\bar{\phi}(u)$ is a path in $\mathcal{G}_B$. Thus $\bar{\phi}(\Sigma_{\bar{A}})$ is contained in the closure of Σ_B as a subset of $\mathcal{B}^{\mathbb{Z}}$. As Σ_B is closed, $\bar{\phi}(\Sigma_{\bar{A}}) \subseteq \Sigma_B$.

We show that $\bar{\phi} : \Sigma_{\bar{A}} \to \Sigma_B$ is bounded-to-one by showing that $\bar{\phi}$ is one-to-one on $\bar{R}^\star$ and by applying Theorem 1.5. Suppose for a contradiction that there are distinct $x, \hat{x} \in \bar{R}^\star$ with $\bar{\phi}(x) = \bar{\phi}(\hat{x})$. Let x and $\hat{x}$ be shortest such paths. We make the following two observations.

$\bar{\phi}(x)$ begins and ends with y. That prefix w of $\bar{\phi}(x)$ preceding the first occurrence of y is in $X^\star$ and therefore labels a unique path p in $R^\star$. Thus $x = px'$ and $\hat{x} = p\hat{x}'$, where x' and $\hat{x}'$ are distinct elements of $\bar{R}^\star$. Using the minimality of $|x|$, we conclude that w is empty. Thus $\bar{\phi}(x)$ begins with y. A similar argument shows that $\bar{\phi}(x)$ ends with y.

$\bar{\phi}(x)$ contains no occurrence of ywy, where $w \in X^\star$. Any path p in $\mathcal{G}_{\bar{A}}$ with $\bar{\phi}(p) = ywy$, where $w \in X^\star$ is of the form $p = p'p''p'''$, where $\bar{\phi}(p') = y$, $\bar{\phi}(p''') = y$, and p'' is the unique path in $R^\star$ with $\phi(p'') = w$. Thus, if $\bar{\phi}(x)$ contains an occurrence of ywy, then $x = x'p'p''p'''x''$ and $\hat{x} = \hat{x}'\hat{p}'p''\hat{p}'''\hat{x}''$, where $x'p'$, $p'''x''$, $\hat{x}'\hat{p}'$, $\hat{p}'''\hat{x}'' \in \bar{R}^\star$ and either $x'p' \neq \hat{x}'\hat{p}'$ or $p'''x'' \neq \hat{p}'''\hat{x}''$. This contradicts the minimality of $|x|$.

Using these observations, we conclude that $\bar{\phi}(x) \in y(Uy)^\star$ and hence that x and $\hat{x}$ are paths in $(\mathcal{G}_{\bar{A}} \backslash \mathcal{G}_A) \cup \{\alpha\}$ beginning and ending at state α. However, $\bar{\phi}$ is right closing on $(\mathcal{G}_{\bar{A}} \backslash \mathcal{G}_A) \cup \{\alpha\}$, so $x = \hat{x}$. This contradiction shows that $\bar{\phi}$ is one-to-one on $\bar{R}^\star$.

Finally, we show that $\bar{\phi} : \Sigma_{\bar{A}} \to \Sigma_B$ is onto. Let w be any path in $\mathcal{G}_B$. By the irreducibility of $\mathcal{G}_B$, there are paths u and v such that $uwv \in S^\star$. Express

$$uwv = w_1 y w_2 y \dots w_{k-1} y w_k,$$

where y overlaps no w_i. Since y and uwv each begin and end at state $\phi(\alpha)$, so do each of $w_1, \dots, w_k$. Thus $w_i \in S^\star \setminus \mathcal{B}^\star y \mathcal{B}^\star = U \cup X^\star$. Hence

$$\begin{aligned} yuwvy &= yw_1yw_2y\dots yw_ky \\ &\in (y(Uy)^\star \cup X^\star)^\star \\ &= (y(Uy)^\star \cup X)^\star \\ &= \bar{X}^\star \\ &= \bar{\phi}(\bar{R})^\star. \end{aligned}$$

Thus w occurs in $\bar{\phi}(\Sigma_{\bar{A}})$. Because $\bar{\phi}(\Sigma_{\bar{A}})$ is closed, $\bar{\phi}(\Sigma_{\bar{A}}) = \Sigma_B$. □

The original result of Ehrenfeucht and Rozenberg can be expressed using the following notation: A code is said to be *complete* if any block of symbols occurs within some concatenation of code words. Their theorem states that any regular code can be imbedded into a complete code. It is known by a theorem of Schutzenberger [3, Chap. 1] that a regular code is complete if and only if it is maximal with respect to inclusion. This holds in particular for a finite code. It is not known, however, when a finite code can be imbedded into a *finite* maximal code.

Theorem 2.1, when reformulated in terms of codes, says that any regular code can be imbedded into a regular code that is complete with respect to a local condition; in other words, each block occurring in some fixed shift of finite type occurs in some product of words from the code. This is again equivalent to a maximality condition by a result of Restivo [17].

3. Right-closing maps between SFTs. A one-block map $\phi : S \to T$ is *right-resolving* if, whenever aa' and aa'' are words of length 2 occurring in S and $\phi(aa') = \phi(aa'')$, then $a' = a''$. It follows that a one-block map $\phi : \Sigma_A \to \Sigma_B$ is right-resolving if and only if ϕ defines a deterministic labeling of the graph $\mathcal{G}_A$. The labels of the first returns to a fixed state then form a prefix code. Conversely, each regular prefix code can be obtained in this way.

A block map $\phi : S \to T$ is *right-closing* if, whenever $x, y \in S$ have a common left-infinite tail and $\phi(x) = \phi(y)$, then $x = y$. We can similarly define *left-closing* by replacing *right* by *left*. It is immediate that, if ϕ is right-closing, then ϕ has no diamonds and therefore is bounded-to-one. A one-block map $\phi : \Sigma_A \to \Sigma_B$ is right closing if and only if ϕ defines a labeling of $\mathcal{G}_A$ that has finite *delay* d, where

$$d = \min \left\{ n : \begin{array}{c} \forall \text{ states } s \text{ of } \mathcal{G}_A, \\ \forall \text{ paths } x = x_0 \dots x_n \text{ and } y = y_0 \dots y_n \text{ beginning at state } s \\ \text{if } \phi(x) = \phi(y) \text{ then } x_0 = y_0 \end{array} \right\}.$$

This states that the present state of $\mathcal{G}_A$ and $d+1$ forthcoming edge labels together determine the next edge of $\mathcal{G}_A$ and therefore the next state. A deterministic labeling has delay 0, so a right-resolving map $\phi : \Sigma_A \to \Sigma_B$ is right-closing. We note that by precomposing with a conjugacy, we can replace a right-closing map with a right-resolving map [5].

In terms of codes, a slightly different definition of delay is in use: Let X be a code over an alphabet $\mathcal{A}$. Then X is said to have *deciphering delay* at most d if, for all $u, u' \in X$ and $v = v_1 \dots v_d \in X^d$ and for all $w \in \mathcal{A}^\star$, we have that $uvw \in u'X^\star$ only when $u = u'$. This means that all the initial parsings of a string into $d+1$ words of X share the same initial code word. Therefore prefix codes are exactly those with deciphering delay 0. The relation between the two notions of delay is the following. Let $\mathcal{G}_A$ be an irreducible graph with an unambiguous labeling defining a one-block map ϕ. Let R be the set of first returns in $\mathcal{G}_A$ to a fixed state i. Let X be the code $\phi(R)$. If ϕ has finite delay, then X has finite deciphering delay. The converse, however, is not true, as is shown by the following example.

Example 1. Let X be the code $a^\star b \cup ba^\star c$. It is not hard to show that this code has deciphering delay 1 and that there is no finite automaton having a labeling with finite delay presenting this code as the labels of the first returns to some fixed state.

However, it is easy to verify that, when X is a *finite* code, the labeling given by ϕ has finite delay if and only if the code X has finite deciphering delay.

In this section, we prove the following result.

THEOREM 3.1. *Let $\mathcal{G}_A$ and $\mathcal{G}_B$ be irreducible graphs and let the one-block map ϕ : $\Sigma_A \rightarrow \Sigma_B$ defined by an edge-labeling of $\mathcal{G}_A$ be right-closing with delay d. Then there is an irreducible graph $\mathcal{G}_{\bar{A}} \supseteq \mathcal{G}_A$ with a labeling extending that of $\mathcal{G}_A$ defining a right-closing one-block factor map $\bar{\phi} : \Sigma_{\bar{A}} \rightarrow \Sigma_B$ also with delay d.*

Before proving this theorem, we set up some notation and prove a lemma. If $\mathcal{G}$ is a graph, s is a state in $\mathcal{G}$, and $n \geq 0$, let

$$F_{\mathcal{G}}(s,n) = \{x : x = x_1 \dots x_n \text{ is a path in } \mathcal{G} \text{ starting at state } s\}.$$

Note that $F_{\mathcal{G}}(s,0) = \{\epsilon_s\}$. If e is an edge with terminal state s in $\mathcal{G}$, define $F_{\mathcal{G}}(e,n) = F_{\mathcal{G}}(s,n)$. If $\mathcal{G} = \mathcal{G}_A$, abbreviate $F_{\mathcal{G}_A} = F_A$.

LEMMA 3.2. *Let ϕ : $\Sigma_A \rightarrow \Sigma_B$ be a one-block map and let $d \geq 0$ be an integer. Suppose that $\mathcal{G}_B$ is irreducible. If, for each state s of $\mathcal{G}_A$ and for each d-path $y_1 \dots y_d$ in $\mathcal{G}_B$ either*

$$y_1 \dots y_d \notin \phi F_A(s,d)$$

or

$$y_1 \dots y_d F_B(y_d, 1) \subseteq \phi F_A(s, d+1),$$

then ϕ is surjective. Moreover, if, in addition, ϕ is right-closing, then the delay of ϕ is at most d.

Proof. We show by induction that, if $y_1 \dots y_d \in \phi F_A(s,d)$, then

$$y_1 \dots y_d F_B(y_d, l) \subseteq \phi F_A(s, d+l)$$

for $l \geq 1$. The case where $l = 1$ is the lemma assumption. Fix $l \geq 1$ and assume that the statement is true for l. Suppose that $y_1 \dots y_d \in \phi F_A(s,d)$. Let $y_{d+1} \dots y_{d+l+1} \in F_B(y_d, l+1)$. We must show that $y_1 \dots y_{d+l+1} \in \phi F_A(s, d+l+1)$. By the inductive hypothesis, $y_1 \dots y_d F_B(y_d, l) \subseteq \phi F_A(s, d+l)$. So there is a path $x_1 \dots x_{d+l}$ in $\mathcal{G}_A$ with label $y_1 \dots y_{d+l}$ starting at state s. Let t be the state along this path between the edges x_l and x_{l+1}. We have $y_{l+1} \dots y_{d+l} \in \phi F_A(t,d)$; so, by the lemma assumption, $y_{l+1} \dots y_{d+l} F_B(y_{d+l}, 1) \subseteq \phi F_A(t, d+1)$. So

$$\begin{aligned} y_1 \dots y_{d+l+1} &\in y_1 \dots y_l \phi F_A(t, d+1) \\ &\subseteq \phi F_A(s, d+l+1), \end{aligned}$$

where the inclusion follows from the existence of the labeled path $s \xrightarrow{y_1 \dots y_l} t$ in $\mathcal{G}_A$. So

$$y_1 \dots y_d F_B(y_d, l+1) \subseteq \phi F_A(s, d+l+1),$$

completing the induction. It follows from the irreducibility of $\mathcal{G}_B$ that any finite path y in $\mathcal{G}_B$ occurs in $\phi(\Sigma_A)$ and therefore that $\phi : \Sigma_A \rightarrow \Sigma_B$ is onto.

Now suppose, in addition, that ϕ is right-closing. Suppose for a contradiction that the delay of ϕ is greater than d, that is, there are two $(d+1)$-paths $x_0 \dots x_d$ and $x'_0 \dots x'_d$ beginning at the same state in $\mathcal{G}_A$, having the same label $\phi(x_0 \dots x_d) = \phi(x'_0 \dots x'_d)$, but having distinct initial edges $x_0 \neq x'_0$. By the above induction, we have for all $l \geq 0$ that

$$\phi(x_1 \dots x_d) F_B(\phi(x_d), l) \subseteq \phi F_A(i(x_1), d+l),$$
$$\phi(x'_1 \dots x'_d) F_B(\phi(x'_d), l) \subseteq \phi F_A(i(x'_1), d+l).$$

Since the left-hand sides of these inclusions are equal, ϕ has delay at least $d+l+1$. This contradicts the assumption that ϕ is right-closing. □

For completeness, we include the following partial converse to Lemma 3.2.

LEMMA 3.3. *Let $\phi : \Sigma_A \to \Sigma_B$ be a surjective one-block map with delay d and suppose that $\mathcal{G}_A$ and $\mathcal{G}_B$ are irreducible. Then, for each state s of $\mathcal{G}_A$ and for each d-path $y_1 \dots y_d$ in $\mathcal{G}_B$, either*

$$y_1 \dots y_d \notin \phi F_A(s, d)$$

or

$$y_1 \dots y_d F_B(y_d, 1) \subseteq \phi F_A(s, d+1).$$

Proof. Suppose for a contradiction that there is a state s_0 in $\mathcal{G}_A$ and a $(d+1)$-path $y_1 \dots y_d y_{d+1}$ in $\mathcal{G}_B$ with

$$y_1 \dots y_d \in \phi F_A(s_0, d),$$

but

$$y_1 \dots y_d y_{d+1} \notin \phi F_A(s_0, d+1).$$

For any path $z = z_1 \dots z_L$ in $\mathcal{G}_B$ with $L \geq d$, define $D(z)$ by

$$D(z) = \{i(x_{L-d+1}) : \exists x_1 \dots x_L \text{ such that } \phi(x_1 \dots x_L) = z\}.$$

Since ϕ has delay d, $\sharp D(z') \geq \sharp D(z)$ whenever z' is a prefix of z with $|z'| \geq d$. If x is a path in $\mathcal{G}_A$ with $|x| \geq d$ and x terminates at state s_0, then $\phi(x) y_1 \dots y_{d+1}$ is a path in $\mathcal{G}_B$ and

$$\sharp D(\phi(x)) > \sharp D(\phi(x) y_1 \dots y_{d+1}).$$

If the right-hand side is nonzero, we can use the irreducibility of $\mathcal{G}_A$ to append a path x' to some path u having $\phi(u) = \phi(x) y_1 \dots y_{d+1}$ to form a concatenation ux' terminating at state s_0. Now apply the above inequality with x replaced by ux' to further reduce the size of the set. In this way, we get a path w in $\mathcal{G}_B$ with $D(w) = \emptyset$. This shows that ϕ is not surjective. □

Proof of Theorem 3.1. We define the graph $\mathcal{G}_{\tilde{A}}$ in three stages. First, let $\mathcal{G}_0 \supseteq \mathcal{G}_A$ be a graph satisfying the following:

(1) $\mathcal{G}_0$ has a labeling extending ϕ as a right-closing graph homomorphism with delay d and with range $\mathcal{G}_B$ (that we again denote by ϕ),

(2) For each d-path v in $\mathcal{G}_B$, there is a state s_v of $\mathcal{G}_0$ with $\phi F_{\mathcal{G}_0}(s_v, d) = \{v\}$,

(3) For each d-path v of $\mathcal{G}_B$, each path leading from state s_v can be extended to a path terminating in the irreducible subgraph $\mathcal{G}_A$.

To construct $\mathcal{G}_0$, we could, for instance, choose a path w in $\mathcal{G}_B$ containing as a subpath each d-path of $\mathcal{G}_B$ and ending at a state $\phi(s)$ in the image of ϕ, then attach to $\mathcal{G}_A$ a path x labeled w terminating at state s. (See Fig. 2.) Alternatively, we could attach trees to $\mathcal{G}_A$ for more efficient use of new states.

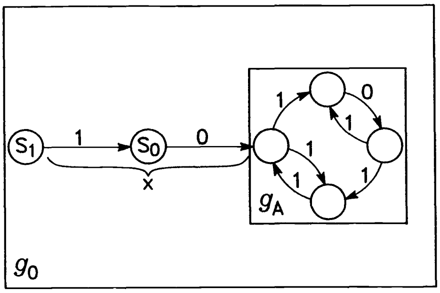


FIG. 2. $\mathcal{G}_0$. *The labeling defines a closing map into the two-shift having delay* 1.

We now add further edges to the graph $\mathcal{G}_0$ to create $\mathcal{G}_1$ as follows. (See Fig. 3.) For each state s of $\mathcal{G}_0$, for each path $y = y_1 \dots y_{d+1} \in F_{\mathcal{G}_B}(\phi(s), d+1)$ satisfying

$$y_1 \dots y_d \in \phi F_{\mathcal{G}_0}(s, d)$$

and

$$y_1 \dots y_d y_{d+1} \notin \phi F_{\mathcal{G}_0}(s, d+1),$$

attach an outgoing edge to state s labeled y_1 and terminating at state $s_{y_2 \dots y_{d+1}}$. (If $d = 0$, we are attaching an outgoing edge to state s labeled y_1 and terminating at state s_{ϵ_t}, where t is the terminal state of edge y_1 in $\mathcal{G}_B$.) Clearly, this labeling extends ϕ as a graph homomorphism with range $\mathcal{G}_B$. Note that we have not added any new states in constructing $\mathcal{G}_1$ from $\mathcal{G}_0$.

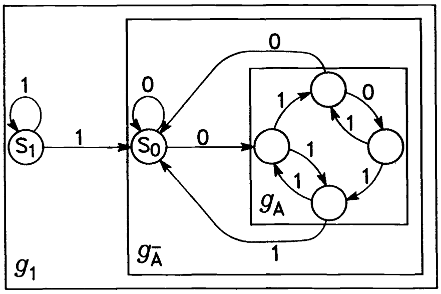


FIG. 3. *Mapping from* $\mathcal{G}_{\bar{A}}$ *onto the two-shift with delay* 1.

We prove that $\mathcal{G}_1$ satisfies the following:

(1) For each state s of $\mathcal{G}_0$ (equivalently, of $\mathcal{G}_1$),

$$\phi F_{\mathcal{G}_1}(s, d) = \phi F_{\mathcal{G}_0}(s, d),$$

(2) $\phi : \mathcal{G}_1 \to \mathcal{G}_B$ has delay d,

(3) For each state s of $\mathcal{G}_0$, for each path $y_1 \dots y_d$ in $\mathcal{G}_B$, either

$$y_1 \dots y_d \notin \phi F_{\mathcal{G}_1}(s, d)$$

or

$$y_1 \dots y_d F_B(y_d, 1) \subseteq \phi F_{\mathcal{G}_1}(s, d+1).$$

We show (1). Since $\mathcal{G}_1 \supseteq \mathcal{G}_0$, we have

$$\phi F_{\mathcal{G}_1}(s,d) \supseteq \phi F_{\mathcal{G}_0}(s,d).$$

If $z_1 \dots z_d \in \phi F_{\mathcal{G}_1}(s,d)$, then $z_1 \dots z_d$ is the label·of a path $u_1 \dots u_d \in F_{\mathcal{G}_1}(s,d)$. We can assume that $u_1 \dots u_d \notin F_{\mathcal{G}_0}(s,d)$. Let u_j be the last edge of $u_1 \dots u_d$ that is not in $\mathcal{G}_0$. Let t be the initial state of the edge u_j. By the construction of $\mathcal{G}_1$, $u_j \dots u_d$ is the prefix of a path $\bar{x}_1 \dots \bar{x}_{d+1}$ in $\mathcal{G}_1$ labeled $y_1 \dots y_d y_{d+1}$, where $y_1 \dots y_d$ also labels some path $x_1 \dots x_d$ in $\mathcal{G}_0$ with initial state t. Now $u_1 \dots u_{j-1} x_1 \dots x_{d-j+1}$ is a path in $\mathcal{G}_0$ with initial state s, with $\phi(u_1 \dots u_{j-1} x_1 \dots x_{d-j+1}) = z_1 \dots z_d$, and with a longer suffix in the subgraph $\mathcal{G}_0 \subseteq \mathcal{G}_1$ than $u_1 \dots u_d$ has. By induction, there is a path in $\mathcal{G}_0$ with initial state s that is labeled $z_1 \dots z_d$. Thus $z_1 \dots z_d \in \phi F_{\mathcal{G}_0}(s,d)$; so $\phi F_{\mathcal{G}_0}(s,d) = \phi F_{\mathcal{G}_1}(s,d)$, as claimed.

We show (2). Equivalently, we show that, for any state s of $\mathcal{G}_1$, the sets $\phi(e)\phi F_{\mathcal{G}_1}(e,d)$, $e \in F_{\mathcal{G}_1}(s,1)$ are pairwise disjoint. The sets $\phi(e)\phi F_{\mathcal{G}_1}(e,d)$, $e \in F_{\mathcal{G}_0}(s,1)$ are pairwise disjoint because $\phi F_{\mathcal{G}_1}(e,d) = \phi F_{\mathcal{G}_0}(e,d)$ and $\phi|_{\Sigma_{\mathcal{G}_0}}$ is right-closing with delay d. By the construction of $\mathcal{G}_1$ there is a one-to-one correspondence between the set of edges

$$F_{\mathcal{G}_1}(s,1) \backslash F_{\mathcal{G}_0}(s,1)$$

and the set of $(d+1)$-paths of $\mathcal{G}_B$

$$\{y_1 \dots y_{d+1} : y_1 \dots y_d \in \phi F_{\mathcal{G}_0}(s,d) \text{ but } y_1 \dots y_{d+1} \notin \phi F_{\mathcal{G}_0}(s,d+1)\}$$

such that edge e corresponds to path $y_1 \dots y_{d+1}$ if and only if

$$\phi(e)\phi F_{\mathcal{G}_1}(e,d) = \{y_1 \dots y_{d+1}\}.$$

Now

$$\bigcup \{\phi(e)\phi F_{\mathcal{G}_1}(e,d) : e \in F_{\mathcal{G}_0}(s,1)\} = \phi F_{\mathcal{G}_0}(s,d+1),$$

so the sets $\phi(e)\phi F_{\mathcal{G}_1}(e,d)$, $e \in F_{\mathcal{G}_1}(s,1)$, are pairwise disjoint as claimed.

We show (3). Suppose that $y_1 \dots y_d \in \phi F_{\mathcal{G}_1}(s,d)$. Then $y_1 \dots y_d \in \phi F_{\mathcal{G}_0}(s,d)$; so, by the construction of $\mathcal{G}_1$, $y_1 \dots y_d y_{d+1} \in \phi F_{\mathcal{G}_1}(s,d+1)$ for each $y_{d+1} \in F_{\mathcal{G}_B}(y_d,1)$. This is (3).

If $\mathcal{G}_1$ is not irreducible, let $\mathcal{G}_{\bar{A}} \subseteq \mathcal{G}_1$ be the subgraph consisting of all states and edges of $\mathcal{G}_1$ that are accessible by *forward* transitions from the graph $\mathcal{G}_A$. As $\mathcal{G}_A$ is irreducible and as any path in $\mathcal{G}_1$ can be extended to a path terminating in $\mathcal{G}_A$, we conclude that $\mathcal{G}_{\bar{A}}$ is irreducible.

Define the graph homomorphism $\bar{\phi} : \mathcal{G}_{\bar{A}} \to \mathcal{G}_B$ by restricting ϕ to $\mathcal{G}_{\bar{A}}$. It is easy to verify that $\bar{\phi}$ satisfies the hypotheses of Lemma 3.2; so the corresponding one-block map $\bar{\phi} : \Sigma_{\bar{A}} \to \Sigma_B$ is onto. This proves Theorem 3.1. □

Theorem 3.1 is related to a result of Bruyere, Wong, and Zhang [7]. They have proved that any regular code with finite deciphering delay can be imbedded into a code that is maximal with respect to inclusion, is regular, and has the same delay. This solves a question raised in [3]. The relationship with Theorem 3.1 is as follows. If we start with a *finite* code with finite deciphering delay, then we can realize the code as the labeling of the first returns in a labeled graph with finite delay. By applying Theorem 3.1, we can imbed the finite code into a maximal regular code having finite deciphering delay.

Except in the case where the given code is prefix, the maximal code that we obtain is not finite. In fact, by a theorem of Schutzenberger [3, Thm. 8.4], a code that is finite, maximal, and with finite deciphering delay is, in fact, prefix.

4. Biclosing maps between SFTs. In this section, we extend a right- and left-closing one-block map $\phi : \Sigma_{\mathcal{G}} \to \Sigma_{\mathcal{H}}$ between irreducible SFTs to a surjection $\bar{\phi} : \Sigma_{\bar{\mathcal{G}}} \to \Sigma_{\mathcal{H}}$ enjoying the same properties.

We say that a one-block map ϕ is *d-right-closing* if it is right-closing with delay d, and that ϕ is *$d^\star$-left-closing* if it is left-closing with delay $d^\star$. If ϕ is both d-right-closing and $d^\star$-left-closing, we say that ϕ is *$(d^\star, d)$-biclosing.*

This is related to the notion of a *biprefix* code; that is, a code that is prefix in both directions. The relation is the following: Any finite biprefix code can be obtained as the labels of the first returns in a graph with a biclosing labeling. Our next theorem is related to a result by Berstel and Perrin [3, Chap. 3] according to which any finite biprefix code can be imbedded into a regular maximal code. Contrasting with the case of *general* finite codes, it is relatively easy to give a necessary and sufficient criterion for a finite biprefix code to be imbeddable into a *finite* maximal code. However the question of imbedding a regular biprefix code into a maximal code is open.

THEOREM 4.1. *Let $\mathcal{G}_A$ and $\mathcal{G}_B$ be irreducible graphs and let the one-block map $\phi : \Sigma_A \to \Sigma_B$ defined by an edge-labeling of $\mathcal{G}_A$ be $(d^\star, d)$-biclosing. Then there is an irreducible graph $\mathcal{G}_{\bar{A}} \supseteq \mathcal{G}_A$ with a labeling extending that of $\mathcal{G}_A$ defining a $(d^\star, d)$-biclosing one-block factor map $\bar{\phi} : \Sigma_{\bar{A}} \to \Sigma_B$.*

Before proving this theorem, we set up some notation, make some definitions, and prove some preparatory lemmas.

A *graph with boundary* is a finite directed graph, except that some of the edges have an initial state but no terminal state; these are *forward boundary edges*. Some of the edges have a terminal state but no initial state; these are *backward boundary edges*.

For the remainder of this section, we fix an irreducible directed graph $\mathcal{H}$. For any set of paths $\mathcal{W}$ in $\mathcal{H}$, define

$$\begin{aligned} F_+\mathcal{W} &= \{wb : w \in \mathcal{W},\ \ b \text{ is an edge, and } wb \text{ is a path in } \mathcal{H}\},\\ P_+\mathcal{W} &= \{aw : w \in \mathcal{W},\ \ a \text{ is an edge, and } aw \text{ is a path in } \mathcal{H}\},\\ F_-\mathcal{W} &= \{u : ub \in \mathcal{W} \text{ for some edge } b\},\\ P_-\mathcal{W} &= \{v : av \in \mathcal{W} \text{ for some edge } a\}.\end{aligned}$$

Denote the paths of length $d^\star$ *preceding* a state s in a directed graph $\mathcal{H}$ by $P_{\mathcal{H}}(s, d^\star)$.

Given the directed graph $\mathcal{H}$, a *molecule $\mathcal{M}$ over $\mathcal{H}$ with delay* $(d^\star, d)$ is a graph with boundary together with *state data* and *edge data* satisfying the transition conditions below. For each state s of $\mathcal{M}$, there is (1) a state $\phi(s)$ of $\mathcal{H}$, (2) a set $\mathcal{U}_s \subseteq P_{\mathcal{H}}(\phi(s), d^\star)$, and (3) a set $\mathcal{V}_s \subseteq F_{\mathcal{H}}(\phi(s), d)$. We collect this data as a triple $(\mathcal{U}_s, \phi(s), \mathcal{V}_s)$. For each edge e of $\mathcal{M}$, there is (1) an edge $\phi(e)$ of $\mathcal{H}$, (2) a set $\mathcal{U}_e \subseteq P_{\mathcal{H}}(\phi(e), d^\star)$, and (3) a set $\mathcal{V}_e \subseteq F_{\mathcal{H}}(\phi(e), d)$. We collect this data as a triple $(\mathcal{U}_e, \phi(e), \mathcal{V}_e)$.

As we see in Lemma 4.5, the following *transition conditions* on the edge and state data of a molecule $\mathcal{M}$ ensure, among other things, that, if $\mathcal{M}$ has no boundary edges, then for each state s of $\mathcal{M}$ we have $\mathcal{U}_s = \phi P_{\mathcal{M}}(s, d^\star)$ and $\mathcal{V}_s = \phi F_{\mathcal{M}}(s, d)$.

For each state s of $\mathcal{M}$, denote the set of outgoing edges from s by $\mathcal{F}(s)$ and the set of incoming edges to s by $\mathcal{P}(s)$.

Note the following transition conditions:

(1) For each state s, $\{\phi(e)\mathcal{V}_e : e \in \mathcal{F}(s)\}$ is a partition of $F_+\mathcal{V}_s$. In particular, no two sets $\phi(e)\mathcal{V}_e$ and $\phi(e')\mathcal{V}_{e'}$ coincide unless $e = e'$;

(2) For each state s, for each edge $e \in \mathcal{F}(s), \mathcal{U}_e = \mathcal{U}_s$;

(3) For each state s, $\{\mathcal{U}_e\phi(e) : e \in \mathcal{P}(s)\}$ is a partition of $P_+\mathcal{U}_s$. In particular, no two sets $\mathcal{U}_e\phi(e)$ and $\mathcal{U}_{e'}\phi(e')$ coincide unless $e = e'$;

(4) For each state s, for each edge $e \in \mathcal{P}(s)$, $\mathcal{V}_e = \mathcal{V}_s$.

Transition conditions (1) and (3) alone or (2) and (4) alone ensure that ϕ is a graph homomorphism in the sense that $\phi(e) \in \mathcal{P}_{\mathcal{H}}\phi(s)$ for each $e \in \mathcal{P}_{\mathcal{M}}(s)$ and $\phi(e) \in \mathcal{F}_{\mathcal{H}}\phi(s)$ for each $e \in \mathcal{F}_{\mathcal{M}}(s)$.

We use the term *molecule* metaphorically: boundary edges are potential molecular bonding sites. Loosely speaking, a forward boundary edge can bond to a backward boundary edge having the same edge data.

Let $\mathcal{W}_k = \{w : w \text{ is a path in } \mathcal{H} \text{ of length } k\}$. Define a function ∂ from the states and edges of $\mathcal{M}$ to $\mathbb{Z}^{\mathcal{W}_{d^\star+d+1}}$ by

$$\partial(s) = \sum[\vec{e}_w : w \in \mathcal{U}_s F_+ \mathcal{V}_s] - \sum[\vec{e}_w : w \in P_+(\mathcal{U}_s)\mathcal{V}_s]$$

and

$$\partial(e) = \begin{cases} \sum[\vec{e}_w : w \in \mathcal{U}_e\phi(e)\mathcal{V}_e] & \text{if } e \text{ is a forward boundary edge,} \\ -\sum[\vec{e}_w : w \in \mathcal{U}_e\phi(e)\mathcal{V}_e] & \text{if } e \text{ is a backward boundary edge,} \\ 0 & \text{otherwise.} \end{cases}$$

For each word $x \in \mathcal{W}_{d^\star+d}$, define $\vec{f}_x \in \mathbb{Z}^{\mathcal{W}_{d^\star+d+1}}$ by

$$\vec{f}_x = \sum[\vec{e}_w : w \in F_+\{x\}] - \sum[\vec{e}_w : w \in P_+\{x\}].$$

Let $\mathcal{L} \subseteq \mathbb{Z}^{\mathcal{W}_{d^\star+d+1}}$ be the lattice generated by the set

$$\{\vec{f}_x : x \in \mathcal{W}_{d^\star+d}\}.$$

It is clear that $\partial(s) \in \mathcal{L}$ for each state s of $\mathcal{M}$.

CLAIM 1. *It holds that*

$$\sum[\partial(s) : s \textit{ is a state of } \mathcal{M}] = \sum[\partial(e) : e \textit{ is an edge of } \mathcal{M}].$$

Proof. Use the fact that

$$\{\phi(e)\mathcal{V}_e : e \in \mathcal{F}(s)\}$$

partitions $F_+\mathcal{V}_s$ and that

$$\{\mathcal{U}_e\phi(e) : e \in \mathcal{P}(s)\}$$

partitions $P_+\mathcal{U}_s$. □

Using the claim, we see that

$$\sum[\partial(e) : e \text{ is a boundary edge of } \mathcal{M}] \in \mathcal{L}.$$

Example 2. Define a molecule $\mathcal{M}_0$ as follows:

states: $(d^\star + d)$-paths uv of $\mathcal{H}$, where u has length $d^\star$, v has length d, and s is the terminal state of u, with $\mathcal{U}_{uv} = \{u\}, \phi(uv) = s$, and $\mathcal{V}_{uv} = \{v\}$,

edges: $(d^\star + 1 + d)$-paths uav of $\mathcal{H}$, where u has length $d^\star$, v has length d, and a is an edge, with $\mathcal{U}_{uav} = \{u\}$, $\phi(uav) = a$, and $\mathcal{V}_{uav} = \{v\}$,
incidence:

$$\mathcal{F}_{\mathcal{M}_0}(uv) = \{uvb : \text{edge } b \text{ follows path } uv \text{ in } \mathcal{H}\},$$
$$\mathcal{P}_{\mathcal{M}_0}(uv) = \{auv : \text{edge } a \text{ precedes path } uv \text{ in } \mathcal{H}\}.$$

It is easy to check that $\mathcal{M}_0$ satisfies the definition of a molecule. The molecule $\mathcal{M}_0$ has no boundary edges.

Example 3. Let x be a $(d^\star + d)$-path of $\mathcal{H}$. Define a molecule $\mathcal{M}_{-x}$ by deleting from $\mathcal{M}_0$ the single state x. Then $\mathcal{M}_{-x}$ has forward boundary edges

$$\{ax : \text{edge } a \text{ precedes path } x \text{ in } \mathcal{H}\}$$

and backward boundary edges

$$\{xb : \text{edge } b \text{ follows path } x \text{ in } \mathcal{H}\};$$

so

$$\begin{aligned}&\sum[\partial(e) : e \text{ is an edge of } \mathcal{M}_{-x}]\\ &= \sum[\vec{e}_w : w \in P_+\{x\}] - \sum[\vec{e}_w : w \in F_+\{x\}]\\ &= -\vec{f}_x.\end{aligned}$$

Example 4. Let x be a $(d^\star + d)$-path of $\mathcal{H}$. Define a molecule $\mathcal{M}_x$ by deleting from $\mathcal{M}_0$ all states except the single state x and deleting all edges except those incident to state x. The forward boundary edges of $\mathcal{M}_x$ are exactly the backward boundary edges of $\mathcal{M}_{-x}$, and vice versa. So

$$\sum[\partial(e) : e \text{ is an edge of } \mathcal{M}_x] = \vec{f}_x.$$

LEMMA 4.2. *If $\phi : \mathcal{G} \to \mathcal{H}$ is a graph homomorphism that as a one-block map is $(d^\star, d)$-biclosing, then $\mathcal{G}$ can be imbedded into a molecule $\mathcal{M}$ over $\mathcal{H}$ in such a way that $\phi : \mathcal{M} \to \mathcal{H}$ extends $\phi : \mathcal{G} \to \mathcal{H}$ as an edge labeling.*

Proof. For each state s of $\mathcal{G}$, define $\mathcal{U}_s = \phi P_\mathcal{G}(s, d^\star)$ and $\mathcal{V}_s = \phi F_\mathcal{G}(s, d)$. For each edge e of $\mathcal{G}$ from state s to t, define $\mathcal{U}_e = \mathcal{U}_s$ and $\mathcal{V}_e = \mathcal{V}_t$.

As $\mathcal{G}$ stands, it might not be the case that

$$\{\phi(e)\mathcal{V}_e : e \in \mathcal{F}(s)\}$$

is a partition of $F_+\mathcal{V}_s$; however, it *is* true that $\phi(e)\mathcal{V}_e \subseteq F_+\mathcal{V}_s$ and, as ϕ is right-closing with delay d, that the sets $\phi(e)\mathcal{V}_e$, $e \in \mathcal{F}(s)$, *are* disjoint. We augment the collection

$$\{\phi(e)\mathcal{V}_e : e \in \mathcal{F}(s)\}$$

by attaching forward boundary edges to the state s to complete a partition of $F_+\mathcal{V}_s$ as follows. For each word

$$w = w_1 \dots w_{d+1} \in F_+\mathcal{V}_s \setminus \bigcup_{e \in \mathcal{F}(s)} \phi(e)\mathcal{V}_e,$$

define an edge e_w with data

$$\begin{aligned} \mathcal{U}_{e_w} &= \mathcal{U}_s, \\ \phi(e_w) &= w_1, \\ \mathcal{V}_{e_w} &= P_-\{w\} = \{w_2 \ldots w_{d+1}\} \end{aligned}$$

and attach e_w to s as an element of $\mathcal{F}(s)$ and as a boundary edge of $\mathcal{M}$. Now,

$$\{\phi(e)\mathcal{V}_e : e \in \mathcal{F}(s)\}$$

is a partition of $F_+\mathcal{V}_s$. In a completely symmetric fashion, attach backward boundary edges to the state s to augment the collection

$$\{\mathcal{U}_e\phi(e) : e \in \mathcal{P}(s)\}$$

to create a partition of $P_+\mathcal{U}_s$. This construction is done at each state of $\mathcal{G}$ to produce the desired molecule $\mathcal{M}$. □

We say an edge e of a molecule is (k, l)*-ramified* if

$$k = d^\star - \max\{n : \text{all words of } \mathcal{U}_e \text{ have a common suffix of length } n\}$$

and

$$l = d - \max\{n : \text{all words of } \mathcal{V}_e \text{ have a common prefix of length } n\}.$$

Thus e is $(0, 0)$-ramified if and only if $\mathcal{U}_e = \{u\}$ and $\mathcal{V}_e = \{v\}$ for paths u and v of $\mathcal{H}$ with $|u| = d^\star$ and $|v| = d$. We say the edge e is *simple* in this case.

LEMMA 4.3. *Any molecule $\mathcal{M}$ can be imbedded in a molecule $\mathcal{N}$ whose boundary edges are all simple.*

Proof. Suppose that the forward boundary edge e of $\mathcal{M}$ with edge data $(\mathcal{U}_e, \phi(e), \mathcal{V}_e)$ is (k, l)-ramified. Let t be the terminal state of the edge $\phi(e)$ in $\mathcal{H}$. Attach to edge e a terminal state s with state data

$$(\mathcal{U}_s, \phi(s), \mathcal{V}_s) = (P_-(\mathcal{U}_e\phi(e)), t, \mathcal{V}_e).$$

For every path $w = w_1 \ldots w_{d+1} \in F_+\mathcal{V}_e$, attach an outgoing edge e_w to state s with edge data

$$(\mathcal{U}_{e_w}, \phi(e_w), \mathcal{V}_{e_w}) = (P_-(\mathcal{U}_e\phi(e)), w_1, P_-\{w\}).$$

Note that, if $k > 0$, then e_w is $(k-1, 0)$-ramified; if $k = 0$, then e_w is simple. For every path

$$w = w_1 \ldots w_{d^\star+1} \in P_+P_-(\mathcal{U}_e\phi(e)) \backslash \mathcal{U}_e\phi(e),$$

attach an incoming edge $e_w^\star$ to state s with edge data

$$(\mathcal{U}_{e_w^\star}, \phi(e_w^\star), \mathcal{V}_{e_w^\star}) = (F_-\{w\}, w_{d^\star+1}, \mathcal{V}_e).$$

Note that $e_w^\star$ is $(0, l)$-ramified.

It is easy to check that the transition conditions are satisfied at state s. Thus we have replaced the (k, l)-ramified forward boundary edge e in $\mathcal{M}$ by (1) $(k - 1, 0)$-ramified forward boundary edges and (2) $(0, l)$-ramified backward boundary edges.

In a completely symmetric fashion (reversing time), we can replace a (k, l)-ramified backward boundary edge of $\mathcal{M}$ by (1) $(0, l-1)$-ramified backward boundary edges, and (2) $(k, 0)$-ramified forward boundary edges.

After $\max(k+1, l+1)$ rounds, we can replace a (k, l)-ramified forward boundary edge of $\mathcal{M}$ by (perhaps many) simple forward and backward boundary edges. □

LEMMA 4.4. *Any molecule $\mathcal{M}$ can be imbedded in a molecule $\bar{\mathcal{M}}$ having no boundary edges.*

Proof. First, use Lemma 4.3 to imbed $\mathcal{M}$ in a molecule $\mathcal{N}$ with whose boundary edges are all simple. Define

$$\vec{b} = \sum [\partial(e) : e \text{ is a boundary edge of } \mathcal{N}].$$

Claim 1 gives $\vec{b} \in \mathcal{L}$; so we can express

$$\vec{b} = \sum_{x \in \mathcal{W}_{d^\star+d}} b_x \vec{f}_x,$$

where b_x, $x \in \mathcal{W}_{d^\star+d}$ are integers. Define a molecule $\bar{\mathcal{M}}$ as follows. For each $(d^\star + d)$-path x of $\mathcal{H}$ define $\mathcal{M}_{-x}$ as in Example 3 and $\mathcal{M}_x$ as in Example 4. Recall that

$$\sum [\partial(e) : e \text{ is a boundary edge of } \mathcal{M}_{-x}] = -\vec{f}_x$$

and

$$\sum [\partial(e) : e \text{ is a boundary edge of } \mathcal{M}_x] = \vec{f}_x.$$

If $b_x > 0$, let $\mathcal{N}_x^1, \ldots, \mathcal{N}_x^{b_x}$ be disjoint copies of $\mathcal{M}_{-x}$. If $b_x < 0$, let $\mathcal{N}_x^1, \ldots, \mathcal{N}_x^{-b_x}$ be disjoint copies of $\mathcal{M}_x$. Let $\bar{\mathcal{M}}$ be the disjoint union of $\mathcal{N}$ and all of the $\mathcal{N}_x^i$. Now

$$\sum [\partial(e) : e \text{ is a boundary edge of } \bar{\mathcal{M}}] = \vec{b} - \vec{b} = 0.$$

As all boundary edges of $\mathcal{N}$, $\mathcal{M}_{-w}$, and $\mathcal{M}_w$ are simple, the boundary edges of $\bar{\mathcal{M}}$ are simple, also. Any simple edge e has edge data of the form $(\mathcal{U}_e, \phi(e), \mathcal{V}_e) = (\{u\}, \phi(e), \{v\})$, where $u\phi(e)v = w$ is a $(d^\star + d + 1)$-path in $\mathcal{H}$; so any simple forward boundary edge e has $\partial(e) = \vec{e}_w$, where $\{w\} = \mathcal{U}_e \phi(e) \mathcal{V}_e$, and any simple backward boundary edge $e^\star$ has $\partial(e^\star) = -\vec{e}_w$, where $\{w\} = \mathcal{U}_{e^\star} \phi(e^\star) \mathcal{V}_{e^\star}$. As the w-component of $\sum \partial(e)$ for $\bar{\mathcal{M}}$ is 0,

$$|\{e : \partial(e) = \vec{e}_w\}| = |\{e : \partial(e) = -\vec{e}_w\}|.$$

For each w, define a bijection

$$\tau_w : \{e : \partial(e) = \vec{e}_w\} \to \{e : \partial(e) = -\vec{e}_w\}$$

and identify each forward boundary edge e of $\bar{\mathcal{M}}$ having $\partial(e) = \vec{e}_w$ with the backward boundary edge $\tau_w(e)$ of $\bar{\mathcal{M}}$. Since the edges e and $\tau_w(e)$ have the same data, this identification respects the transition conditions. Now $\bar{\mathcal{M}}$ (with the boundary edge identifications) has no boundary edges. □

We verify in Lemmas 4.5–4.7 that $\phi : \Sigma_{\mathcal{M}} \to \Sigma_{\mathcal{H}}$ is a $(d^\star, d)$-biclosing factor map if $\mathcal{M}$ is a molecule over $\mathcal{H}$ without boundary edges.

LEMMA 4.5. (i) *If s is a state of a molecule $\mathcal{M}$, then*

$$\phi F_{\mathcal{M}}(s, d+1) \subseteq F_+ \mathcal{V}_s.$$

Moreover,

$$\phi F_{\mathcal{M}}(s, d+1) = F_+ \mathcal{V}_s$$

if no forward boundary edge of $\mathcal{M}$ occurs on a path in $F_{\mathcal{M}}(s,k)$ for any $k \le d$.

(ii) *If s is a state of a molecule $\mathcal{M}$, then*

$$\phi P_{\mathcal{M}}(s, d^\star + 1) \subseteq P_+ \mathcal{U}_s.$$

Moreover,

$$\phi P_{\mathcal{M}}(s, d^\star + 1) = P_+ \mathcal{U}_s$$

if no backward boundary edge of $\mathcal{M}$ occurs on a path in $P_{\mathcal{M}}(s,k)$ for any $k \le d^\star$.

Proof. We prove only (i), the proof of (ii) being similar. We denote the d-fold iteration of the function F_- by F_-^d, below:

$$\begin{aligned}\phi F_{\mathcal{M}}(s,1) &= \{\phi(e) : e \in \mathcal{F}(s)\}\\ &= \bigcup_{e\in\mathcal{F}(s)} F_-^d(\phi(e)\mathcal{V}_e)\\ &= F_-^d F_+ \mathcal{V}_s.\end{aligned}$$

Make the inductive hypothesis for $1 \le l < d+1$ that

$$\phi F_{\mathcal{M}}(s,l) \subseteq F_-^{d+1-l} F_+ \mathcal{V}_s$$

for all states s of $\mathcal{M}$ and that equality holds if no forward boundary edge of $\mathcal{M}$ occurs on any path in $F_{\mathcal{M}}(s,k)$ for any $k < l$. For any edge e that has a terminal state, denote the terminal state by $t(e)$. We have

$$\begin{aligned}\phi F_{\mathcal{M}}(s,l+1) &= \bigcup\{\phi(e)\phi F_{\mathcal{M}}(t(e),l) : e \in \mathcal{F}(s) \text{ and } e \text{ is not a boundary edge}\}\\ &\subseteq \bigcup_{e\in\mathcal{F}(s)} \phi(e) F_-^{d+1-l} F_+ \mathcal{V}_e\\ &= F_-^{d+1-l} F_+^2 \mathcal{V}_s\\ &= F_-^{d+1-(l+1)} F_+ \mathcal{V}_s.\end{aligned}$$

We prove that the inclusion is actually an equality in case there are no boundary edges reached from state s before time $l+1$ as follows:

No boundary edge occurs on a path in $F_{\mathcal{M}}(s,k)$ for any $k < l+1$

$\Rightarrow$ No edge $e \in \mathcal{F}(s)$ is a boundary edge and no boundary edge occurs on a path in $F_{\mathcal{M}}(t(e),k)$ for any $k < l$

$\Rightarrow$ No edge $e \in \mathcal{F}(s)$ is a boundary edge and

$$\phi F_{\mathcal{M}}(t(e),l) = F_-^{d+1-l} F_+ \mathcal{V}_e$$

$\Rightarrow$

$$\begin{aligned}&\bigcup\{\phi(e)\phi F_{\mathcal{M}}(t(e),l) : e \in \mathcal{F}(s) \text{ and } e \text{ is not a boundary edge}\}\\ =\ &\bigcup_{e\in\mathcal{F}(s)} \phi(e) F_-^{d+1-l} F_+ \mathcal{V}_e. \quad \square\end{aligned}$$

LEMMA 4.6. *If $\mathcal{M}$ is a molecule, then the map $\phi : \Sigma_{\mathcal{M}} \to \Sigma_{\mathcal{H}}$ is $(d^\star, d)$-biclosing in the following sense*:

(i) *For any paths $x = x_1 \dots x_{d+1}$ and $y = y_1 \dots y_{d+1}$ beginning at a state s of $\mathcal{M}$, if $\phi(x) = \phi(y)$, then $x_1 = y_1$*;

(ii) *For any paths $x = x_1 \dots x_{d^\star+1}$ and $y = y_1 \dots y_{d^\star+1}$ ending at a state s of $\mathcal{M}$, if $\phi(x) = \phi(y)$, then $x_{d^\star+1} = y_{d^\star+1}$.*

Proof. Let $x = x_1 \dots x_{d+1}$ and $y = y_1 \dots y_{d+1}$ be two paths in $\mathcal{M}$ beginning at a state s with $\phi(x) = \phi(y)$. By the inclusion part of Lemma 4.5, we have $\phi(x) \in \phi(x_1)\mathcal{V}_{x_1}$ and $\phi(y) \in \phi(y_1)\mathcal{V}_{y_1}$. As

$$\{\phi(e)\mathcal{V}_e : e \in \mathcal{F}(s)\}$$

is a partition of $F_+\mathcal{V}_s$, we have $x_1 = y_1$. □

LEMMA 4.7. *If $\mathcal{M}$ is a molecule without boundary edges, then $\phi : \Sigma_{\mathcal{M}} \to \Sigma_{\mathcal{H}}$ is surjective as a one-block map.*

Proof. By Lemma 4.6, we have that $\phi : \Sigma_{\mathcal{M}} \to \Sigma_{\mathcal{H}}$ is d-right-closing. By Lemma 4.5, we have that $\phi F_{\mathcal{M}}(s, d+1) = F_+\mathcal{V}_s$, so ϕ satisfies the hypotheses of Lemma 3.2. Hence ϕ is surjective. □

Proof of Theorem 4.1. Set $\mathcal{G}_A = \mathcal{G}$ and $\mathcal{G}_B = \mathcal{H}$. Use Lemma 4.2 to imbed $\mathcal{G}$ in a molecule $\mathcal{M}$ in such a way that $\phi : \mathcal{M} \to \mathcal{H}$ extends $\phi : \mathcal{G} \to \mathcal{H}$ as an edge labeling. By Lemma 4.4, we can assume that $\mathcal{M}$ has no boundary edges.

Let $\mathcal{M}_0$ be the connected component of $\mathcal{M}$ containing $\mathcal{G}$ as a subgraph. By Lemma 4.6, $\phi : \Sigma_{\mathcal{M}_0} \to \Sigma_{\mathcal{H}}$ is $(d^\star, d)$-biclosing. As no connected component of $\mathcal{M}$ has a boundary edge, by Lemma 4.7, $\phi : \Sigma_{\mathcal{M}_0} \to \Sigma_{\mathcal{H}}$ is surjective. □

5. Bounded-to-one maps between sofic systems. In this section, we prove the following sofic version of Theorem 2.1.

THEOREM 5.1. *Let $\phi : S \to T$ be a bounded-to-one one-block map from an irreducible sofic system S into an irreducible sofic system T. Then there is an irreducible sofic system $\bar{S} \supseteq S$ and a bounded-to-one one-block factor map $\bar{\phi} : \bar{S} \to T$ extending ϕ.*

To extend $\phi : S \to T$ to a *sofic* domain, $\bar{S}$ is the best we can hope to do, in general: As the following example shows, even assuming that the domain of ϕ is SFT is not sufficient to give an extension $\bar{\phi}$ of ϕ with an SFT domain. Example 6 shows that assuming that the range of ϕ is an SFT is not sufficient to give an extension $\bar{\phi}$ with SFT domain.

Example 5. Let $\mathcal{G}_A$ be the graph shown in Fig. 4. The *even system* $S \subseteq \{0,1\}^{\mathbb{Z}}$ is the image of Σ_A under the one-block map $\pi : \Sigma_A \to S$ defined by the edge-labeling

$$\pi(a) = 1, \qquad \pi(b) = \pi(c) = 0.$$

The even system is that set of sequences in $\{0,1\}^{\mathbb{Z}}$ such that, between any two 1's, there occurs an even number of 0's.

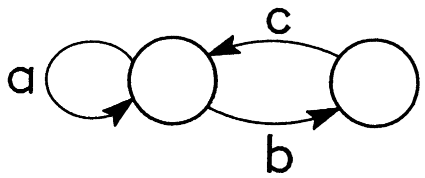


FIG. 4. $\mathcal{G}_A$.

Define a three-block map $\phi : S \to \{\alpha, \beta\}^{\mathbb{Z}}$ by Table 1. We can verify that ϕ is bounded-to-one by checking that the three-block composition

$$\Sigma_A \xrightarrow{\pi} S \xrightarrow{\phi} \{\alpha, \beta\}^{\mathbb{Z}}$$

has no diamonds and by applying Theorem 1.4.

TABLE 1

ϕ

00	α
010	α
011	β
10	β
11	α

There is no bounded-to-one extension of $\phi : S \to \{\alpha, \beta\}^{\mathbb{Z}}$ to an SFT domain because ϕ has a diamond. Specifically, we have

$$\phi(0^\infty \underline{1}0010^\infty) = \phi(0^\infty \underline{0}1110^\infty) = \alpha^\infty \underline{\beta}\alpha\alpha\beta\alpha^\infty,$$

where the underscored symbols occur at time 0.

Example 6. Let $\Sigma = \{0^\infty\}$ and let S be the even system as in Example 5. Define $\phi : \Sigma \to S$ by $\phi(0^\infty) = 0^\infty$. Suppose that there is an irreducible SFT $\bar{\Sigma} \supseteq \Sigma$ and a surjective extension $\bar{\phi} : \bar{\Sigma} \to S$ of ϕ. By replacing $\bar{\Sigma}$ by $\bar{\Sigma}^{[n]}$ if necessary, we can assume that $\bar{\phi}$ is a one-block map defined by an edge labeling of an irreducible graph $\mathcal{G}_A$ and that $\bar{\Sigma} = \Sigma_A$. As $0^\infty \in \Sigma_A$, and as $\phi(0^\infty) = 0^\infty$, the graph $\mathcal{G}_A$ has a self-loop at a state i_0 labeled 0. As $\mathcal{G}_A$ is irreducible and as $\bar{\phi} : \Sigma_A \to S$ is onto, there is a path in $\mathcal{G}_A$ starting at state i_0 labeled 0^m1 for some $m \geq 0$, and there is a path ending at state i_0 labeled 10^k for some $k \geq 0$. Thus all words of the form $10^k0^l0^m1$, where $l \geq 0$ occur in $\bar{\phi}(\Sigma_A)$, contradicting $\bar{\phi}(\Sigma_A) \subseteq S$. Thus $\phi : \{0^\infty\} \to S$ cannot be extended to an SFT domain.

To prove Theorem 5.1, we need more background from symbolic dynamics. The following lemma is an immediate consequence of a much stronger theorem in [9].

LEMMA 5.2. *If S is an irreducible sofic system, then there is an irreducible SFT Σ_R and a bounded-to-one one-block factor map $\pi_R : \Sigma_R \to S$.*

Given two block maps $\pi : S \to T$ and $\pi' : S' \to T$, we define the *fibered product* R of π and π' to be the shift space

$$R = \{(x, y) \in S \times S' : \pi(x) = \pi'(y)\}.$$

We have the one-block projections $\psi : R \to S$ and $\psi' : R \to S'$ defined by $\psi((x, y)) = x$ and $\psi'((x, y)) = y$.

The following is well known. (See [1].)

LEMMA 5.3. *Let R be the fibered product of $\pi : S \to T$ and $\pi' : S' \to T$ and let $\psi : R \to S$ and $\psi' : R \to S'$ be the corresponding projections. Then*

(i) *π is bounded-to-one $\Rightarrow$ ψ' is bounded-to-one,*

(ii) *π' is bounded-to-one $\Rightarrow$ ψ is bounded-to-one,*

(iii) *π is surjective $\Rightarrow$ ψ' is surjective,*

(iv) *π' is surjective $\Rightarrow$ ψ is surjective.*

Proof of lemma. We prove only (i). Let $y \in S'$. Then $(x,y) \in (\psi')^{-1}(y) \Leftrightarrow \pi(x) = \pi'(y) \Leftrightarrow x \in \pi^{-1}\pi'(y)$. So $\left|(\psi')^{-1}(y)\right| = \left|\pi^{-1}\pi'(y)\right|$ for each $y \in S'$. As π is bounded-to-one,

$$\sup_{y\in S'} \left|(\psi')^{-1}(y)\right| = \sup_{y\in S'} \left|\pi^{-1}\pi'(y)\right| < \infty.$$

So ψ' is bounded-to-one. □

Proof of Theorem 5.1. The following construction is illustrated in Fig. 5. Using Lemma 5.2, let $\pi_A : \Sigma_A \to S$ and $\pi_B : \Sigma_B \to T$ be bounded-to-one one-block factor maps from irreducible SFTs. Let Σ be the fibered product of $\phi \circ \pi_A : \Sigma_A \to T$ and $\pi_B : \Sigma_B \to T$ with projections $\psi_A : \Sigma \to \Sigma_A$ and $\psi_B : \Sigma \to \Sigma_B$. By Lemma 1.1, there is an irreducible component Σ_0 of Σ with $h(\Sigma_0) = h(\Sigma)$. As $\phi \circ \pi_A$ is bounded-to-one, Lemma 5.3 gives that $\psi_B : \Sigma_0 \to \Sigma_B$ is bounded-to-one. By Theorem 2.1, there is an irreducible SFT $\bar\Sigma \supseteq \Sigma_0$ and a bounded-to-one one-block factor map $\bar\psi : \bar\Sigma \to \Sigma_B$ extending $\psi_B : \Sigma_0 \to \Sigma_B$. We define an irreducible sofic system $\bar S$ by identifying (as elements of an equivalence class) those symbols that occur in $\Sigma_0 \subseteq \bar\Sigma$ that are mapped via the one-block map $\pi_A \circ \psi_A : \Sigma_0 \to S$ to the same symbol of S, and by making no identifications among the symbols of $\bar\Sigma$ that are not symbols of Σ_0. Let $\rho : \bar\Sigma \to \bar S$ be the one-block factor map making these identifications; in other words, ρ maps each symbol of $\bar\Sigma$ to the equivalence class containing it.

$$\begin{tikzcd}
 & & \Sigma \arrow[ddll, "\psi_A"'] \arrow[ddrr, "\psi_B"] & & \\
 & & \Sigma_0 \arrow[u, hook] \arrow[d, hook] & & \\
\Sigma_A \arrow[dr, "\pi_A"'] & & \bar\Sigma \arrow[rr, "\bar\psi"] \arrow[d, "\rho"] & & \Sigma_B \arrow[ddll, "\pi_B"] \\
 & S \arrow[r, "\iota"] \arrow[dr, "\varphi"'] & \bar S \arrow[d, "\bar\varphi"] & & \\
 & & T & &
\end{tikzcd}$$

FIG. 5. *Commutative diagram for extending $\phi : S \to T$.*

Since π_B is bounded-to-one and onto, so is ψ_A; thus $h(\psi_A(\Sigma_0)) = h(\Sigma_0) = h(\Sigma) = h(\Sigma_A)$, and, as Σ_A is irreducible, Lemma 1.2 gives that $\psi_A : \Sigma_0 \to \Sigma_A$ is onto. Hence $\pi_A \circ \psi_A : \Sigma_0 \to S$ is onto, also. Thus the sofic system S can naturally be identified with a subsystem of $\bar S$ via the one-block imbedding $\iota : S \to \bar S$ sending each symbol s of S to the *nonempty* equivalence class of those symbols t in $\Sigma_0 \subseteq \bar\Sigma$ with $\pi_A \circ \psi_A(t) = s$. That ι actually maps S to $\bar S$ follows from the fact that, for all symbols s that occur in Σ_0,

$$\iota \circ \pi_A \circ \psi_A(s) = \rho(s)$$

and the fact that $\pi_A \circ \psi_A : \Sigma_0 \to S$ is onto.

The bounded-to-one one-block factor map $\pi_B \circ \bar\psi : \bar\Sigma \to T$ decomposes into one-block maps $\bar\Sigma \xrightarrow{\rho} \bar S \xrightarrow{\bar\phi} T$ because, for all symbols s, t occurring in Σ_0, if $\pi_A \circ \psi_A(s) = \pi_A \circ \psi_A(t)$, then

$$\begin{aligned}\pi_B \circ \bar{\psi}(s) &= \pi_B \circ \psi_B(s)\\ &= \phi \circ \pi_A \circ \psi_A(s)\\ &= \phi \circ \pi_A \circ \psi_A(t)\\ &= \pi_B \circ \psi_B(t)\\ &= \pi_B \circ \bar{\psi}(t).\end{aligned}$$

In other words, each fiber of $\pi_A \circ \psi_A : \Sigma_0 \to S$ is contained in a fiber of $\pi_B \circ \bar{\psi} : \bar{\Sigma} \to T$. Since $\pi_B \circ \bar{\psi} : \bar{\Sigma} \to T$ is surjective, so is $\bar{\phi} : \bar{S} \to T$.

We now verify that $\bar{\phi} : \bar{S} \to T$ is an extension of $\phi : S \to T$ in the sense that $\bar{\phi} \circ \iota = \phi$. Let $y \in S$. Choose $x \in \Sigma_0$ such that $\pi_A \circ \psi_A(x) = y$. Then

$$\begin{aligned}\phi(y) &= \phi \circ \pi_A \circ \psi_A(x)\\ &= \pi_B \circ \psi_B(x)\\ &= \pi_B \circ \bar{\psi}(x)\\ &= \bar{\phi} \circ \rho(x)\\ &= \bar{\phi} \circ \iota \circ \pi_A \circ \psi_A(x)\\ &= \bar{\phi} \circ \iota(y). \qquad \square\end{aligned}$$

6. Simultaneous extension of bounded-to-one maps. We are given two bounded-to-one one-block maps $\phi_B : \Sigma_A \to \Sigma_B$ and $\phi_C : \Sigma_A \to \Sigma_C$ between irreducible SFTs. We want to construct a single irreducible SFT $\Sigma_{\bar{A}} \supseteq \Sigma_A$ and bounded-to-one one-block factor maps $\bar{\phi}_B : \Sigma_{\bar{A}} \to \Sigma_B$ and $\bar{\phi}_C : \Sigma_{\bar{A}} \to \Sigma_C$ extending ϕ_B and ϕ_C, respectively. If such an SFT $\Sigma_{\bar{A}}$ exists, then necessarily $h(\Sigma_B) = h(\Sigma_{\bar{A}}) = h(\Sigma_C)$ by Theorem 1.3. This necessary condition is also sufficient.

THEOREM 6.1. *Let $\mathcal{G}_A$, $\mathcal{G}_B$, and $\mathcal{G}_C$ be irreducible graphs with $h(\Sigma_B) = h(\Sigma_C)$. Let the one-block maps $\phi_B : \Sigma_A \to \Sigma_B$ and $\phi_C : \Sigma_A \to \Sigma_C$ defined by edge-labelings of $\mathcal{G}_A$ be bounded-to-one. Then there is an irreducible graph $\mathcal{G}_{\bar{A}} \supseteq \mathcal{G}_A$ with two labelings extending those of $\mathcal{G}_A$ defining bounded-to-one factor maps $\bar{\phi}_B : \Sigma_{\bar{A}} \to \Sigma_B$ and $\bar{\phi}_C : \Sigma_{\bar{A}} \to \Sigma_C$.*

The proof is almost entirely a recapitulation of the tableau method of [1]; the only new element is a particular choice of a way of "filling in the tableau" to suit our present needs. The method is based on the following theorem of Furstenberg [1].

THEOREM 6.2. *Let $\mathcal{G}_{A'}$ and $\mathcal{G}_{A''}$ be irreducible graphs (with transition matrices A' and A''). Then $h(\Sigma_{A'}) = h(\Sigma_{A''})$ if and only if there is a positive integral matrix F such that $A'F = FA''$.*

Proof of Theorem 6.1. Use Theorem 2.1 to construct an irreducible graph $\mathcal{G}_{A'} \supseteq \mathcal{G}_A$ with an edge-labeling defining a bounded-to-one one-block factor map $\bar{\phi}_B : \Sigma_{A'} \to \Sigma_B$ extending $\phi_B : \Sigma_A \to \Sigma_B$. Similarly, construct $\mathcal{G}_{A''} \supseteq \mathcal{G}_A$ with an edge-labeling defining a bounded-to-one one-block factor map $\bar{\phi}_C : \Sigma_{A''} \to \Sigma_C$ extending ϕ_C.

Regard $\mathcal{G}_{A'}$ and $\mathcal{G}_{A''}$ as disjoint graphs and let $\iota' : \mathcal{G}_A \to \mathcal{G}_{A'}$ and $\iota'' : \mathcal{G}_A \to \mathcal{G}_{A''}$ be the graph homomorphisms imbedding $\mathcal{G}_A$ into $\mathcal{G}_{A'}$ and $\mathcal{G}_{A''}$. In this regard, $\phi_B = \bar{\phi}_B \circ \iota'$ and $\phi_C = \bar{\phi}_C \circ \iota''$.

Theorem 1.3 gives $h(\Sigma_{A'}) = h(\Sigma_B)$ and $h(\Sigma_{A''}) = h(\Sigma_C)$. We are assuming that $h(\Sigma_B) = h(\Sigma_C)$; so, by Theorem 6.2, there is a positive integral matrix F such that $A'F = FA''$.

We define a directed graph $\mathcal{G}_F$ as follows. For each state s of $\mathcal{G}_{A'}$, for each state t of $\mathcal{G}_{A''}$, $\mathcal{G}_F$ has F_{st} directed edges $e_F(s,t,n)$, $1 \leq n \leq F_{st}$, each with initial state s and terminal state t. Thus all of the edges of $\mathcal{G}_F$ run from $\mathcal{G}_{A'}$ to $\mathcal{G}_{A''}$.

For each state r in $\mathcal{G}_{A'}$, for each state t in $\mathcal{G}_{A''}$, define two sets of two-paths in $\mathcal{G}_{A'} \cup \mathcal{G}_F \cup \mathcal{G}_{A''}$:

$$\mathcal{E}_{FA''}(r,t) = \left\{ fa'' : \begin{array}{c} \exists \text{ state } s \text{ in } \mathcal{G}_{A''} \text{ such that} \\ r \xrightarrow{f} s \text{ in } \mathcal{G}_F \text{ and } s \xrightarrow{a''} t \text{ in } \mathcal{G}_{A''} \end{array} \right\}$$

and

$$\mathcal{E}_{A'F}(r,t) = \left\{ a'f : \begin{array}{c} \exists \text{ state } s \text{ in } \mathcal{G}_{A'} \text{ such that} \\ r \xrightarrow{a'} s \text{ in } \mathcal{G}_{A'} \text{ and } s \xrightarrow{f} t \text{ in } \mathcal{G}_F \end{array} \right\}$$

CLAIM 2. *For all states r in $\mathcal{G}_{A'}$ and all states t in $\mathcal{G}_{A''}$,*

$$|\mathcal{E}_{FA''}(r,t)| = |\mathcal{E}_{A'F}(r,t)|.$$

Proof of claim. We have

$$|\mathcal{E}_{FA''}(r,t)| = \sum_s F_{rs}A''_{st} = (FA'')_{rt}$$

and

$$|\mathcal{E}_{A'F}(r,t)| = \sum_s A'_{rs}F_{st} = (A'F)_{rt};$$

so the claim follows from $A'F = FA''$.

Let $\tau_{rt} : \mathcal{E}_{FA''}(r,t) \to \mathcal{E}_{A'F}(r,t)$ be a bijection. If $r = \iota' r_0$ and $t = \iota'' t_0$, where r_0 and t_0 are states of $\mathcal{G}_A$ such that $A_{r_0 t_0} > 0$, we further require that

$$\tau_{rt}(e_F(\iota' r_0, \iota'' r_0, 1)\iota'' a) = (\iota' a) e_F(\iota' t_0, \iota'' t_0, 1)$$

for each edge a of $\mathcal{G}_A$ from state r_0 to state t_0.

The collection of bijections τ_{rt} defines a bijection

$$\tau : \cup_{r,t}\mathcal{E}_{FA''}(r,t) \to \cup_{r,t}\mathcal{E}_{A'F}(r,t)$$

because the unions are disjoint. We make this definition for notational convenience.

We define a directed graph $\mathcal{G}_D$ as follows. The *states* of $\mathcal{G}_D$ are the *edges* of $\mathcal{G}_F$. For each two-path $f_1 a'' \in \mathcal{E}_{FA''} = \cup_{r,t}\mathcal{E}_{FA''}(r,t)$, there is an edge $e_D(f_1 a'')$ of $\mathcal{G}_D$ from state f_1 to state f_2, where $\tau(f_1 a'') = a' f_2 \in \mathcal{E}_{A'F} = \cup_{r,t}\mathcal{E}_{A'F}(r,t)$.

As usual, denote $\mathcal{G}_F = (\mathcal{S}_F, \mathcal{E}_F)$ and $\mathcal{G}_D = (\mathcal{S}_D, \mathcal{E}_D)$.

Define a graph homomorphism $\pi_{A''} : \mathcal{G}_D \to \mathcal{G}_{A''}$ by

(1) $\pi_{A''} f = s$, where $s \in \mathcal{S}_{A''}$ is the terminal state of edge $f \in \mathcal{E}_F = \mathcal{S}_D$,

(2) $\pi_{A''} e_D(fa'') = a''$.

Similarly, define a graph homomorphism $\pi_{A'} : \mathcal{G}_D \to \mathcal{G}_{A'}$ by

(1) $\pi_{A'} f = s$, where $s \in \mathcal{S}_{A'}$ is the initial state of edge $f \in \mathcal{E}_F = \mathcal{S}_D$,

(2) $\pi_{A'} e_D(\tau^{-1}(a'f)) = a'$.

It is easy to verify that $\pi_{A''}$ and $\pi_{A'}$ are well-defined graph homomorphisms. Denote the corresponding one-block maps by $\pi_{A''} : \Sigma_D \to \Sigma_{A''}$ and $\pi_{A'} : \Sigma_D \to \Sigma_{A'}$.

A one-block map $\phi : S \to T$ is *left-resolving* if, whenever $a'a$ and $a''a$ occur in S and $\phi(a'a) = \phi(a''a)$, then $a' = a''$.

It is helpful to regard an edge $e_D(f_1a'') \in \mathcal{E}_D$ as a box with left and bottom sides given by the two-path f_1a'' and top and right sides given by the two-path $a'f_2 = \tau(f_1a'')$. See Fig. 6. Intuitively, $\pi_{A''}$ is right-resolving because the left side (initial state f_1) and bottom side ($\pi_{A''}$-label a'') of a box $e_D(f_1a'')$ determine the entire box. Similarly, $\pi_{A'}$ is left-resolving because the right side (terminal state f_2) and top side ($\pi_{A'}$-label a') of a box $e_D(\tau^{-1}(a'f_2))$ determine the entire box.

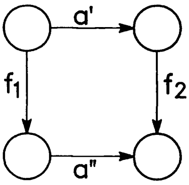


FIG. 6. *An edge of $\mathcal{G}_D$.*

Parts (iii) and (iv) of the following claim are due to B. Kitchens.

CLAIM 3. (i) $\pi_{A''} : \Sigma_D \to \Sigma_{A''}$ *is right-resolving.*

(ii) $\pi_{A'} : \Sigma_D \to \Sigma_{A'}$ *is left-resolving.*

(iii) $\pi_{A''}(\Sigma) = \Sigma_{A''}$ *for any irreducible component* Σ *of* Σ_D.

(iv) $\pi_{A'}(\Sigma) = \Sigma_{A'}$ *for any irreducible component* Σ *of* Σ_D.

Proof of claim. (i). The $\pi_{A''}$-image of the edge $e_D(fa'')$ starting at state f in $\mathcal{G}_D$ is a'', so the edges starting at state f are distinctly $\pi_{A''}$-labeled, so $\pi_{A''}$ is right-resolving.

(ii). The $\pi_{A'}$-image of the edge $e_D(\tau^{-1}(a'f))$ terminating at state f in $\mathcal{G}_D$ is a', so the edges terminating at state f are distinctly $\pi_{A'}$-labeled, so $\pi_{A'}$ is left-resolving.

(iii) and (iv). The irreducible components of Σ_D are $\Sigma_{D_1}, \ldots, \Sigma_{D_n}$, where $\mathcal{G}_{D_1}, \ldots, \mathcal{G}_{D_n}$ are the maximal irreducible subgraphs of $\mathcal{G}_D$. Let $\mathcal{H}$ be the directed graph with states $1, \ldots, n$ and an edge $i \to j$ if there is an edge in $\mathcal{G}_D$ from $\mathcal{G}_{D_i}$ to $\mathcal{G}_{D_j}$. Because the $\mathcal{G}_{D_i}$ are *maximal* irreducible subgraphs, $\mathcal{H}$ has no cycles.

A *sink* is a state with no outgoing edges, and a *source* is a state with no incoming edges. Assume that 1 is a sink state of $\mathcal{H}$. We verify that $\pi_{A''} : \mathcal{G}_{D_1} \to \mathcal{G}_{A''}$ satisfies the hypothesis of Lemma 3.2. For $d = 0$, the hypothesis reduces to the following: For each state s of $\mathcal{G}_{D_1}$,

$$F_{A''}(\pi_{A''}(s), 1) \subseteq \pi_{A''} F_{D_1}(s, 1).$$

As this hypothesis is satisfied by $\mathcal{G}_D$ and as $\mathcal{G}_D$ has no edges leaving $\mathcal{G}_{D_1}$, it is also satisfied by $\mathcal{G}_{D_1}$. We can conclude that $\pi_{A''}(\Sigma_{D_1}) = \Sigma_{A''}$. Theorem 1.3 gives $h(\Sigma_{D_1}) = h(\Sigma_{A''})$. As $h(\Sigma_{A'}) = h(\Sigma_{A''})$, Lemma 1.2 gives $\pi_{A'}(\Sigma_{D_1}) = \Sigma_{A'}$. By reversing the sense of time and applying Corollary 3.3 to the map $\pi_{A'} : \Sigma_{D_1} \to \Sigma_{A'}$, we see that 1 is a source state of $\mathcal{H}$. In short, every sink of $\mathcal{H}$ is a source. As $\mathcal{H}$ has no cycles, it follows that $\mathcal{H}$ has no edges. Using Lemma 3.2, we obtain $\pi_{A''}(\Sigma_{D_i}) = \Sigma_{A''}$ and $\pi_{A'}(\Sigma_{D_i}) = \Sigma_{A'}$ for each $1 \le i \le n$. This proves the claim.

Define a graph injection $\iota : \mathcal{G}_A \to \mathcal{G}_D$ by

(1) $\iota(r) = e_F(\iota'r, \iota''r, 1)$ for each state r of $\mathcal{G}_A$,

(2) $\iota(a) = e_D(e_F(\iota'r, \iota''r, 1)\iota''a)$ where r is the initial state of edge a in $\mathcal{G}_A$.

That ι is well defined follows from the fact that the terminal state of edge

$$e_D(e_F(\iota' r, \iota'' r, 1)\iota'' a) \text{ in } \mathcal{G}_D \text{ is } e_F(\iota' s, \iota'' s, 1),$$

where s is the terminal state of edge a in $\mathcal{G}_A$. Note that $\tau(e_F(\iota' r, \iota'' r, 1)\iota'' a) = (\iota' a)e_F(\iota' s, \iota'' s, 1)$.

CLAIM 4. (i) $\pi_{A''} \circ \iota = \iota''$ *as homomorphisms from* $\mathcal{G}_A$ *to* $\mathcal{G}_{A''}$.

(ii) $\pi_{A'} \circ \iota = \iota'$ *as homomorphisms from* $\mathcal{G}_A$ *to* $\mathcal{G}_{A'}$.

The proof is a simple calculation. Thus we have the commutative diagram of Fig. 7.

$$\begin{tikzcd}
 & \Sigma_D \arrow[dl, "\pi_{A'}"'] \arrow[dr, "\pi_{A''}"] & \\
\Sigma_{A'} \arrow[d, "\bar{\varphi}_B"'] & \Sigma_A \arrow[u, "\iota"'] \arrow[l, "\iota'"'] \arrow[r, "\iota''"] \arrow[dl, "\varphi_B"] \arrow[dr, "\varphi_C"'] & \Sigma_{A''} \arrow[d, "\bar{\varphi}_C"] \\
\Sigma_B & & \Sigma_C
\end{tikzcd}$$

FIG. 7. *Simultaneous extension of ϕ_B and ϕ_C.*

Let $\mathcal{G}_{\bar{A}}$ be the irreducible component of $\mathcal{G}_D$ containing the irreducible graph $\iota(\mathcal{G}_A)$. The restriction of $\bar{\phi}_C \circ \pi_{A''}$ to $\mathcal{G}_{\bar{A}}$ is a surjective extension of $\phi_C : \mathcal{G}_A \to \mathcal{G}_C$ in the sense that $\bar{\phi}_C \circ \pi_{A''} \circ \iota = \phi_C$ because $\bar{\phi}_C \circ \pi_{A''} \circ \iota = \bar{\phi}_C \circ \iota'' = \phi_C$. Similarly, $\bar{\phi}_B \circ \pi_{A'} \circ \iota = \bar{\phi}_B \circ \iota' = \phi_B$. Note that $\bar{\phi}_C \circ \pi_{A''}$ and $\bar{\phi}_B \circ \pi_{A'}$ are bounded-to-one because they are compositions of bounded-to-one maps. □

7. Simultaneous extension of right-closing maps. We are given two right-closing maps $\phi_B : \Sigma_A \to \Sigma_B$ and $\phi_C : \Sigma_A \to \Sigma_C$ between irreducible SFTs. We would like to construct a single irreducible SFT $\Sigma_{\bar{A}} \supseteq \Sigma_A$ and right-closing factor maps $\bar{\phi}_B : \Sigma_{\bar{A}} \to \Sigma_B$ and $\bar{\phi}_C : \Sigma_{\bar{A}} \to \Sigma_C$ extending ϕ_B and ϕ_C, respectively.

We cannot hope, in general, to preserve the delays of right-closing one-block maps ϕ_B and ϕ_C. For instance, if ϕ_B and ϕ_C are both right-resolving, and there is a state s of the domain graph $\mathcal{G}_A$ whose image states $\phi_B(s)$ in $\mathcal{G}_B$ and $\phi_C(s)$ in $\mathcal{G}_C$ have an unequal number of following edges in their respective graphs, then it is impossible to satisfy both

$$\bar{\phi}_B F_{\bar{A}}(s,1) = F_B(\bar{\phi}_B(s),1) \quad \text{and} \quad \bar{\phi}_C F_{\bar{A}}(s,1) = F_C(\bar{\phi}_C(s),1),$$

which, by Corollary 3.3, are necessary conditions for the surjectivity of $\bar{\phi}_B$ and $\bar{\phi}_C$, respectively.

We content ourselves with finding right-closing factor maps $\bar{\phi}_B$ and $\bar{\phi}_C$ with a common irreducible domain $\Sigma_{\bar{A}}$ extending ϕ_B and ϕ_C in the sense that there is an imbedding (one-to-one block map) $\iota : \Sigma_A \to \Sigma_{\bar{A}}$ with $\phi_B = \bar{\phi}_B \circ \iota$ and $\phi_C = \bar{\phi}_C \circ \iota$. Using the Masking Lemma of Nasu [15], we can actually take $\iota : \Sigma_A \to \Sigma_{\bar{A}}$ to be a one-block map given by a graph imbedding of $\mathcal{G}_A$ into $\mathcal{G}_{\bar{A}}$. Furthermore, as explained in Remark 1, below, if the given maps ϕ_B and ϕ_C are one-block maps, we can take their extensions $\bar{\phi}_B$ and $\bar{\phi}_C$ to be one-block maps as well.

As in the previous section, a necessary condition for the existence of $\Sigma_{\bar{A}}$ is that $h(\Sigma_B) = h(\Sigma_C)$. The requirement that $\bar{\phi}_B$ and $\bar{\phi}_C$ be right-closing imposes a second necessary condition on Σ_B and Σ_C. To explain this condition, we must give further background from symbolic dynamics.

We define (following [6]) the dimension group of an integral matrix A and collect its properties that we use in this paper. If A is an $\alpha \times \alpha$ integral matrix, let V_A be the eventual range of A regarded as a map $A : \mathbb{Q}^\alpha \to \mathbb{Q}^\alpha$ acting on row vectors. Thus $V_A = A^\alpha(\mathbb{Q}^\alpha)$. Define the dimension group

$$G_A = \{\vec{q} \in V_A : \vec{q}A^k \in \mathbb{Z}^\alpha \text{ for sufficiently large } k \in \mathbb{N}\}$$

and the automorphism $\hat{A}$ of G_A by $\hat{A} = A|_{G_A}$. The dimension group pair $(G_A, \hat{A})$ associated to the shift of finite type Σ_A is a conjugacy invariant; i.e., if Σ_A is conjugate to Σ_B, then there is a group isomorphism $\theta : G_A \to G_B$ with $\hat{B} \circ \theta = \theta \circ \hat{A}$.

We say a square matrix A is *eventually positive* if $A^n > 0$ for all sufficiently large powers n.

POPOSITION 7.1 (see [6, Prop. 2.12]). *Let A and B be integral eventually positive matrices with the same spectral radius λ. Then the following are equivalent*:

(i) *$(G_B, \hat{B})$ is a quotient of $(G_A, \hat{A})$*,

(ii) *There exists $L \geq 0$ and nonnegative integral matrices S, R such that*

$$AS = SB \quad \textit{and} \quad RS = B^L,$$

(iii) *There exists $L \geq 0$ and positive integral matrices S, R such that*

$$AS = SB \quad \textit{and} \quad RS = B^L.$$

The following is the easy direction of the Eventual Factors Theorem of [6].

THEOREM 7.2. *If there is a right-closing factor map $\phi : \Sigma_A \to \Sigma_B$ between aperiodic SFTs Σ_A and Σ_B, then $(G_B, \hat{B})$ is a quotient of $(G_A, \hat{A})$.*

We remark that Theorem 7.2 can be proved directly for a one-block map $\phi : \Sigma_A \to \Sigma_B$ with delay d by defining

$$S_{st} = |\{y \in \phi F_A(s, d) : \text{path } y \text{ ends at state } t \text{ in } \mathcal{G}_B\}|$$

and

$$R_{ts} = |\{x \in F_A(s_t, d) : \phi(x) = y_t \text{ and } x \text{ ends at state } s\}|,$$

where, for each state t of $\mathcal{G}_B$, state s_t of $\mathcal{G}_A$ is chosen so that there is a path $y_t \in \phi F_A(s_t, d)$ ending at state t. We can show that

$$AS = SB \quad \text{and} \quad RS = B^d$$

and therefore that $(G_B, \hat{B})$ is a quotient of $(G_A, \hat{A})$.

Following [6], we define the *ideal class* $\mathcal{I}(A)$ of an integral eventually positive matrix A. Denote the (unique) largest eigenvalue of A by λ. Let $\vec{r}$ be a right eigenvector for eigenvalue λ with entries in $\mathbb{Z}[\lambda]$. Since λ is an algebraic integer, we have $\lambda \in \mathbb{Z}[1/\lambda]$. Thus the entries of $\vec{r}$ generate an ideal $[\vec{r}]$ in the ring $\mathbb{Z}[1/\lambda]$. Let $\mathcal{I}(A)$ be the equivalence class of the ideal $[\vec{r}]$, where two ideals $\mathcal{I}$ and $\mathcal{J}$ in $\mathbb{Z}[1/\lambda]$ are equivalent if there are nonzero $x, y \in \mathbb{Z}[1/\lambda]$ with $y\mathcal{I} = x\mathcal{J}$. Although $\vec{r}$ and hence $[\vec{r}]$ are defined only up to scalar multiples, $\mathcal{I}(A)$ is independent of the particular choice of the eigenvector $\vec{r}$.

The following lemma is contained in Proposition 5.10 of [6].

LEMMA 7.3. *Let A and B be integral eventually positive matrices with the same spectral radius λ. If $(G_B, \hat{B})$ is a quotient of $(G_A, \hat{A})$, then $\mathcal{I}(A) = \mathcal{I}(B)$.*

Proof. By Proposition 7.1, there are integer matrices R, S and $l \geq 0$ with

$$AS = SB \quad \text{and} \quad RS = B^l.$$

Let $\vec{r}$ be a right eigenvector of B for eigenvalue λ with entries in $\mathbb{Z}[\lambda]$. Then $S\vec{r}$ is a right eigenvector for A and $[S\vec{r}] \subseteq [\vec{r}]$. Now $[RS\vec{r}] = [B^l\vec{r}] = \lambda^l[\vec{r}] = [\vec{r}]$, so $[\vec{r}] \subseteq [S\vec{r}]$, so $[\vec{r}] = [S\vec{r}]$. Hence $\mathcal{I}(A) = \mathcal{I}(B)$. □

Chaining together Theorem 7.2 with Lemma 7.3, we conclude that, for aperiodic Σ_B and Σ_C to both be right-closing factors of a single aperiodic SFT $\Sigma_{\bar{A}}$, it is necessary that $\mathcal{I}(A) = \mathcal{I}(B)$. This condition, together with $h(\Sigma_A) = h(\Sigma_B)$, is also sufficient even when we ask that the factor maps $\bar{\phi}_B : \Sigma_{\bar{A}} \to \Sigma_B$ and $\bar{\phi}_C : \Sigma_{\bar{A}} \to \Sigma_C$ extend given maps $\phi_B : \Sigma_A \to \Sigma_B$ and $\phi_C : \Sigma_A \to \Sigma_C$ with domain Σ_A imbedded in $\Sigma_{\bar{A}}$. We first prove the aperiodic case.

THEOREM 7.4. *Let $\phi_B : \Sigma_A \to \Sigma_B$ and $\phi_C : \Sigma_A \to \Sigma_C$ be right-closing maps between aperiodic SFTs. Suppose that $h(\Sigma_B) = h(\Sigma_C)$ and $\mathcal{I}(B) = \mathcal{I}(C)$. Then there is an aperiodic SFT $\Sigma_{\bar{A}}$, an imbedding $\iota : \Sigma_A \to \Sigma_{\bar{A}}$, and right-closing factor maps $\bar{\phi}_B : \Sigma_{\bar{A}} \to \Sigma_B$ and $\bar{\phi}_C : \Sigma_{\bar{A}} \to \Sigma_C$ such that $\bar{\phi}_B \circ \iota = \phi_B$ and $\bar{\phi}_C \circ \iota = \phi_C$. Moreover, the imbedding $\iota : \Sigma_A \to \Sigma_{\bar{A}}$ can be taken to be a one-block map defined by a graph imbedding of $\mathcal{G}_A$ into $\mathcal{G}_{\bar{A}}$.*

Proof of Theorem 7.4. We can assume that $h(\Sigma_A) < h(\Sigma_B)$ $(= h(\Sigma_C))$ and that ϕ_B and ϕ_C are one-block maps. Apply Theorem 3.1 to construct right-closing factor maps $\bar{\phi}_B : \Sigma_{A'} \to \Sigma_B$ and $\bar{\phi}_C : \Sigma_{A''} \to \Sigma_C$, where $\Sigma_{A'}$ and $\Sigma_{A''}$ are irreducible SFTs with one-block imbeddings $\iota' : \Sigma_A \to \Sigma_{A'}$ and $\iota'' : \Sigma_A \to \Sigma_{A''}$ satisfying $\bar{\phi}_B \circ \iota' = \phi_B$ and $\bar{\phi}_C \circ \iota'' = \phi_C$. Alternatively, $\iota' : \Sigma_A \to \Sigma_{A'}$ can be constructed by first applying a result of [5] to replace Σ_A by a conjugate SFT thereby reducing to the case where $\phi_B : \Sigma_A \to \Sigma_B$ is right-*resolving*. Now apply the resolving case ($d = 0$) of Theorem 3.1 to construct $\iota' : \Sigma_A \to \Sigma_{A'}$ as above. The attraction here is that the resolving case of Theorem 3.1 is conceptually simpler; furthermore, we are not applying the delay-preserving feature of Theorem 3.1 in the present construction. As Σ_A is aperiodic, $\Sigma_{A'}$ and $\Sigma_{A''}$ are necessarily aperiodic. By Theorem 7.2 and Lemma 7.3, we obtain $\mathcal{I}(A') = \mathcal{I}(B)$ and $\mathcal{I}(A'') = \mathcal{I}(C)$. As $\mathcal{I}(B) = \mathcal{I}(C)$, we conclude $\mathcal{I}(A') = \mathcal{I}(A'')$.

The algebraic part of the proof of Theorem 7.1 of [6] gives the following lemma.

LEMMA 7.5. *Let A' and A'' be aperiodic matrices with $h(\Sigma_{A'}) = h(\Sigma_{A''})$ and $\mathcal{I}(A') = \mathcal{I}(A'')$. Then there is an aperiodic matrix D such that* $\operatorname{tr}(D) > 0$, $h(\Sigma_D) = h(\Sigma_{A'}) = h(\Sigma_{A''})$, *and $(G_{A'}, \hat{A}')$ and $(G_{A''}, \hat{A}'')$ are both quotients of $(G_D, \hat{D})$.*

We remark that in [6] the matrix D used in the proof of Theorem 7.1 there (and constructed in Theorem 5.14 [6]) has $\operatorname{tr}(D) > 0$, although this fact is used there only to show that D is aperiodic.

To complete the present proof, we want to imbed Σ_A into Σ_D via an imbedding $\iota : \Sigma_A \to \Sigma_D$, then extend the right-closing maps $\iota' \circ \iota^{-1} : \iota(\Sigma_A) \to \Sigma_{A'}$ and $\iota'' \circ \iota^{-1} : \iota(\Sigma_A) \to \Sigma_{A''}$ to right-closing factor maps $\pi_{A'} : \Sigma_D \to \Sigma_{A'}$ and $\pi_{A''} : \Sigma_D \to \Sigma_{A''}$. We would use the following two Theorems (Theorem 7.6 to construct the imbedding ι and Theorem 7.7 to construct the extensions $\pi_{A'}$ and $\pi_{A''}$). In their statements, we denote by $\Pi_j(\Sigma)$ the number of periodic points of *least* period j in the shift space Σ. Note that $j | \Pi_j(\Sigma)$ for $j \geq 1$.

THEOREM 7.6 (see [14]). *Let Λ and Σ be SFTs with Σ irreducible. There exists an imbedding $\iota : \Lambda \to \Sigma$ if the following two conditions hold:*

(i) $h(\Lambda) < h(\Sigma)$,

(ii) $\Pi_j(\Lambda) \leq \Pi_j(\Sigma)$ *for all* $j \geq 1$. (*We denote this by* $\Lambda \overset{\text{per}}{\hookrightarrow} \Sigma$.)

THEOREM 7.7 (see [2]). *Let A and B be aperiodic matrices satisfying*

(i) $h(\Sigma_A) = h(\Sigma_B)$,

(ii) *The dimension group pair* $(G_B, \hat{B})$ *is a quotient of the dimension group pair* $(G_A, \hat{A})$,

(iii) *For all* $j \geq 1$, *if* $\Pi_j(\Sigma_A) > 0$, *then there is* $q \geq 1$ *such that* $q|j$ *and* $\Pi_q(\Sigma_B) > 0$.

(*We denote this by* $\Sigma_A \overset{\text{per}}{\twoheadrightarrow} \Sigma_B$.)

If $\Lambda \subseteq \Sigma_A$ *is a subshift of finite type and* $\phi : \Lambda \to \Sigma_B$ *is a right-closing map, then* ϕ *can be extended to a right-closing factor map* $\bar{\phi} : \Sigma_A \to \Sigma_B$.

However, Σ_D might not have enough periodic points of certain orders to accommodate an imbedding $\iota : \Sigma_A \to \Sigma_D$ (Σ_D might violate $\Sigma_A \overset{\text{per}}{\hookrightarrow} \Sigma_D$). Furthermore, Σ_D might have periodic points of certain orders precluding the factor map $\pi_{A'} : \Sigma_D \to \Sigma_{A'}$ or $\pi_{A''} : \Sigma_D \to \Sigma_{A''}$. ($\Sigma_D$ might violate the condition $\Sigma_D \overset{\text{per}}{\twoheadrightarrow} \Sigma_{A'}$ or the condition $\Sigma_D \overset{\text{per}}{\twoheadrightarrow} \Sigma_{A''}$.) The following lemma of Boyle [4] solves these periodic point problems of Σ_D.

LEMMA 7.8 (see [4, Lem. 2.1]). *Suppose that* Γ *is an irreducible shift of finite type of positive entropy and* $\Pi_q(\Gamma) > 0$, *and* $M_0, M_1, \ldots, M_k$ *are positive integers. Then there is an irreducible shift of finite type* $\hat{\Gamma}$ *such that*

(i) *There is a right-closing factor map* $\hat{\phi} : \hat{\Gamma} \to \Gamma$,

(ii) *If* Γ *is aperiodic, then* $\hat{\Gamma}$ *is aperiodic*,

(iii) $\Pi_j(\hat{\Gamma}) = \Pi_j(\Gamma)$ *if* j *is not among* $q, qM_0, \ldots, qM_k$,

(iv) $\Pi_q(\hat{\Gamma}) = \Pi_q(\Gamma) - q + q\,|\{i : M_i = 1\}|$,

(v) $\Pi_{qM_i}(\hat{\Gamma}) = \Pi_{qM_i}(\Gamma) + qM_i\,|\{j : M_j = M_i\}|$, *for those* i *with* $M_i > 1$.

As remarked in [4], $\hat{\phi} : \hat{\Gamma} \to \Gamma$ as constructed there is right-closing. We remark that the proof of Lemma 7.8 proceeds by "blowing up" a periodic orbit of Γ of least period q into $k + 1$ periodic orbits, the ith of which has least period qM_i, for $0 \leq i \leq k$.

We use Lemma 7.8 to construct an SFT $\Sigma_{\bar{A}}$ satisfying

(1) There is a right-closing factor map $\hat{\phi} : \Sigma_{\bar{A}} \to \Sigma_D$,

(2) $\Sigma_{\bar{A}}$ is aperiodic,

(3) $\Sigma_A \overset{\text{per}}{\hookrightarrow} \Sigma_{\bar{A}}$,

(4) $\Sigma_{\bar{A}} \overset{\text{per}}{\twoheadrightarrow} \Sigma_{A'}$ and $\Sigma_{\bar{A}} \overset{\text{per}}{\twoheadrightarrow} \Sigma_{A''}$,

as follows. As $h(\Sigma_D) > h(\Sigma_A)$, we can fix $N > 0$ such that $\Pi_j(\Sigma_D) \geq \Pi_j(\Sigma_A)$ for all $j \geq N$. Apply Lemma 7.8 to Σ_D with $q = 1$ (recall $\operatorname{tr}(D) > 0$) and with the set $\{M_0, \ldots M_k\}$ satisfying

$$|\{i : M_i = j\}| = \frac{1}{j}\Pi_j(\Sigma_A) \quad \text{for } 1 \leq j < N$$

to produce an SFT Γ_0 satisfying

(1′) There is a right-closing factor map $\phi_0 : \Gamma_0 \to \Sigma_D$,

(2′) Γ_0 is aperiodic,

(3′) $\Sigma_A \overset{\text{per}}{\hookrightarrow} \Gamma_0$.

We now "blow up" a finite number of periodic orbits of Γ_0 using Lemma 7.8 to construct an SFT $\Sigma_{\bar{A}}$ satisfying (4): $\Sigma_{\bar{A}} \overset{\text{per}}{\twoheadrightarrow} \Sigma_{A'}$ and $\Sigma_{\bar{A}} \overset{\text{per}}{\twoheadrightarrow} \Sigma_{A''}$. As $\Sigma_{A'}$ and $\Sigma_{A''}$ are aperiodic, we can fix $M > 0$ such that $\Pi_j(\Sigma_{A'}) > 0$ and $\Pi_j(\Sigma_{A''}) > 0$ for all $j \geq M$. Fix $m_0 > 0$ such that $\Pi_{m_0}(\Sigma_{A'}) > 0$ and $\Pi_{m_0}(\Sigma_{A''}) > 0$. There are a finite number of periodic orbits of Γ_0 whose least period j satisfies $\Pi_j(\Sigma_{A'}) = 0$ or $\Pi_j(\Sigma_{A''}) = 0$ because any such least period j has $j < M$. Iteratively apply Lemma 7.8 to each such orbit with $q = j$ and with a single $M_0 = m_0$ to blow up the orbit of least period j into an orbit of least period jm_0.

After applying Lemma 7.8 once for each such orbit, we will have an SFT $\Sigma_{\bar{A}}$ satisfying (1)–(4), above. That (3) holds requires an explanation. As Σ_A imbeds into $\Sigma_{A'}$ and into $\Sigma_{A''}$, for all $j \geq 1$, if $\Pi_j(\Sigma_{A'}) = 0$ or $\Pi_j(\Sigma_{A''}) = 0$, then $\Pi_j(\Sigma_A) = 0$. Therefore, in each application of Lemma 7.8 after our first, no point of period j in Γ_0, where $\Pi_j(\Sigma_A) > 0$ is lost. Thus the condition $\Sigma_A \overset{\text{per}}{\hookrightarrow} \Gamma_0$ remains undisturbed.

Now use Theorem 7.6 to construct an imbedding $\iota : \Sigma_A \to \Sigma_{\bar{A}}$. Because there is a right-closing factor map $\phi : \Sigma_{\bar{A}} \to \Sigma_D$, Theorem 7.2 tells us that $(G_D, \hat{D})$ is a quotient of $(G_{\bar{A}}, \hat{\bar{A}})$. As both $(G_{A'}, \hat{A}')$ and $(G_{A''}, \hat{A}'')$ are quotients of $(G_D, \hat{D})$, as $\Sigma_{\bar{A}} \overset{\text{per}}{\twoheadrightarrow} \Sigma_{A'}$ and $\Sigma_{\bar{A}} \overset{\text{per}}{\twoheadrightarrow} \Sigma_{A''}$, and as any imbedding is right-closing, by Theorem 7.7 there are right-closing factor maps $\pi_{A'} : \Sigma_{\bar{A}} \to \Sigma_{A'}$ and $\pi_{A''} : \Sigma_{\bar{A}} \to \Sigma_{A''}$ extending the imbeddings $\iota' \circ \iota^{-1} : \iota(\Sigma_A) \to \Sigma_{A'}$ and $\iota'' \circ \iota^{-1} : \iota(\Sigma_A) \to \Sigma_{A''}$, respectively. So $\bar{\phi}_B \circ \pi_{A'} : \Sigma_{\bar{A}} \to \Sigma_B$ is an extension of $\phi_B : \Sigma_A \to \Sigma_B$ in the sense that $\bar{\phi}_B \circ \pi_{A'} \circ \iota = \phi_B$ because

$$\bar{\phi}_B \circ \pi_{A'} \circ \iota = \bar{\phi}_B \circ \iota' \circ \iota^{-1} \circ \iota = \bar{\phi}_B \circ \iota' = \phi_B.$$

Similarly, $\bar{\phi}_C \circ \pi_{A''} \circ \iota = \phi_C$. See Fig. 8. That $\iota : \Sigma_A \to \Sigma_{\bar{A}}$ can be taken to be a one-block map given by a graph imbedding of $\mathcal{G}_A$ into $\mathcal{G}_{\bar{A}}$ follows immediately from Nasu's Masking Lemma [15, Lem. 3.18], a version of which we state here.

$$\begin{tikzcd}
 & \Sigma_{\bar{A}} \arrow[d] \arrow[ddl, "\pi_{A'}"'] \arrow[ddr, "\pi_{A''}"] & \\
 & \Sigma_D & \\
\Sigma_{A'} \arrow[d, "\bar{\varphi}_B"'] & \Sigma_A \arrow[l, "\iota'"'] \arrow[r, "\iota''"] \arrow[uu, bend left, "\iota"] \arrow[dl, "\varphi_B"] \arrow[dr, "\varphi_C"'] & \Sigma_{A''} \arrow[d, "\bar{\varphi}_C"] \\
\Sigma_B & & \Sigma_C
\end{tikzcd}
\qquad
\begin{tikzcd}
 & (G_{\bar{A}}, \hat{\bar{A}}) \arrow[d] & \\
 & (G_D, \hat{D}) \arrow[dl] \arrow[dr] & \\
(G_{A'}, \hat{A'}) & & (G_{A''}, \hat{A''})
\end{tikzcd}$$

FIG. 8. *Extending right-closing ϕ_B and ϕ_C simultaneously.*

LEMMA 7.9. *Let $\iota : \Sigma_A \to \Sigma_{\bar{A}}$ be an imbedding. Then there is a conjugacy $\phi : \Sigma_{\bar{A}} \to \Sigma_E$ such that the imbedding $\phi \circ \iota : \Sigma_A \to \Sigma_E$ is a one-block map given by a graph imbedding of $\mathcal{G}_A$ into $\mathcal{G}_E$.*

We use the lemma to replace $\Sigma_{\bar{A}}$ by Σ_E, ι by $\phi \circ \iota$, $\bar{\phi}_B$ by $\bar{\phi}_B \circ \phi^{-1}$, and $\bar{\phi}_C$ by $\bar{\phi}_C \circ \phi^{-1}$.

Remark 1. If the maps $\phi_B : \Sigma_A \to \Sigma_B$ and $\phi_C : \Sigma_A \to \Sigma_C$ given in the hypothesis of Theorem 7.4 are *one-block* right-closing maps, we can ensure simultaneously that $\iota : \Sigma_A \to \Sigma_{\bar{A}}$ is a one-block imbedding given by a graph imbedding, that $\bar{\phi}_B : \Sigma_{\bar{A}} \to \Sigma_B$ and $\bar{\phi}_C : \Sigma_{\bar{A}} \to \Sigma_C$ are *one-block* right-closing factor maps and that $\phi_B = \bar{\phi}_B \circ \iota$ and $\phi_C = \bar{\phi}_C \circ \iota$.

Theorem 7.4 as stated ensures all of this, except that $\bar{\phi}_B$ and $\bar{\phi}_C$ are one-block maps. To achieve this as well, we can replace $\Sigma_{\bar{A}}$ by a conjugate SFT $\Sigma_{\mathcal{H}}$, where we define the graph $\mathcal{H}$ as follows.

Choose an integer N large enough so that both $\bar{\phi}_B(x)_0$ and $\bar{\phi}_C(x)_0$ are determined by $x_{-N}\dots x_N$. The states of $\mathcal{H}$ are the equivalence classes of $(2N+1)$-paths in $\mathcal{G}_{\bar{A}}$ defined by the equivalence relation

$$x_{-N}\cdots x_N \sim y_{-N}\cdots y_N$$

if and only if

(1) The edges x_0 and y_0 in $\mathcal{G}_{\bar{A}}$ have the same initial state,

(2) The edges $\bar{\phi}_B(x_{-N}\dots x_N)_0$ and $\bar{\phi}_B(y_{-N}\dots y_N)_0$ in $\mathcal{G}_B$ have the same initial state, and

(3) The edges $\bar{\phi}_C(x_{-N}\dots x_N)_0$ and $\bar{\phi}_C(y_{-N}\dots y_N)_0$ in $\mathcal{G}_C$ have the same initial state.

The edges from state s to state t in $\mathcal{H}$ are defined to be the triples (s, e, t) such that there is $x_{-N}\dots x_N \in s$ and $y_{-N}\dots y_N \in t$ with $x_{-N+1}\dots x_N = y_{-N}\dots y_{N-1}$ and $e = x_0 = y_{-1}$.

We define an edge mapping ψ from the edges of $\mathcal{H}$ to the edges of $\mathcal{G}_{\bar{A}}$ by $\psi((s,e,t)) = e$. It is easy to show that ψ defines a one-block conjugacy $\psi : \Sigma_{\mathcal{H}} \to \Sigma_{\bar{A}}$ with a $(2N+1)$-block inverse and that $\bar{\phi}_B \circ \psi$ and $\bar{\phi}_C \circ \psi$ are one-block maps.

It only remains to show that $\psi^{-1}|_{\iota(\Sigma_A)}$ is a one-block map given by a graph injection of $\iota(\mathcal{G}_A)$ (regarded as a subgraph of $\mathcal{G}_{\bar{A}}$) into the graph $\mathcal{H}$. To this end, we define a graph homomorphism $\rho : \iota(\mathcal{G}_A) \to \mathcal{H}$ as follows. For each state r of $\iota(\mathcal{G}_A)$, define $\rho(r)$ to be that state of $\mathcal{H}$ that (as an equivalence class) contains all of the $(2N+1)$-paths $x_{-N}\dots x_N$ of $\iota(\mathcal{G}_A)$ having state r as the initial state of the edge x_0. Note that these $(2N+1)$-paths are indeed all in a single equivalence class because $\bar{\phi}_B|_{\iota(\mathcal{G}_A)}$ and $\bar{\phi}_C|_{\iota(\mathcal{G}_A)}$ are one-block maps. For each edge e of $\iota(\mathcal{G}_A)$, define $\rho(e) = (\rho(s), e, \rho(t))$, where s and t are the initial and terminal states of the edge e. Regarding the one-block map ψ as a graph homomorphism from $\mathcal{H}$ to $\mathcal{G}_{\bar{A}}$, we have that $\psi \circ \rho$ is the identity map on the graph $\iota(\mathcal{G}_A)$. Thus ρ is a graph injection; furthermore, regarding ρ now as a one-block map $\rho : \iota(\Sigma_A) \to \Sigma_{\mathcal{H}}$, we have $\rho = \psi^{-1}|_{\iota(\Sigma_A)}$. Thus, $\psi^{-1}|_{\iota(\Sigma_A)}$ is a one-block map given by a graph injection of $\iota(\mathcal{G}_A)$ into the graph $\mathcal{H}$, as claimed.

To complete the argument, we replace $\Sigma_{\bar{A}}$ by $\Sigma_{\mathcal{H}}$, ι by $\psi^{-1} \circ \iota$ (given by a graph imbedding), $\bar{\phi}_B$ by $\bar{\phi}_B \circ \psi$ (a one-block map), and $\bar{\phi}_C$ by $\bar{\phi}_C \circ \psi$ (also a one-block map).

The *period* of an SFT Σ is the greatest common divisor of the set of periods of all periodic points in Σ. Note that if Σ_A has period p, then A^p is eventually positive. We state without proof the version of Theorem 7.4 for periodic SFTs. The proof is a reduction to the aperiodic case (Theorem 7.4).

THEOREM 7.10. *Let $\phi_B : \Sigma_A \to \Sigma_B$ and $\phi_C : \Sigma_A \to \Sigma_C$ be right-closing maps between irreducible SFTs. Let p be the least common multiple of the periods of Σ_B and Σ_C. Suppose that $h(\Sigma_B) = h(\Sigma_C)$ and $\mathcal{I}(B^p) = \mathcal{I}(C^p)$. Then there is an SFT $\Sigma_{\bar{A}}$ with period p, an imbedding $\iota : \Sigma_A \to \Sigma_{\bar{A}}$, and right-closing factor maps $\bar{\phi}_B : \Sigma_{\bar{A}} \to \Sigma_B$ and $\bar{\phi}_C : \Sigma_{\bar{A}} \to \Sigma_C$ such that $\bar{\phi}_B \circ \iota = \phi_B$ and $\bar{\phi}_C \circ \iota = \phi_C$. Moreover, the imbedding $\iota : \Sigma_A \to \Sigma_{\bar{A}}$ can be taken to be a one-block map defined by a graph imbedding of $\mathcal{G}_A$ into $\mathcal{G}_{\bar{A}}$.*

We remark that $\mathcal{I}(B^p) = \mathcal{I}(C^p)$ is a necessary condition for there to exist simultaneous right-closing extensions $\bar{\phi}_B$ and $\bar{\phi}_C$ as in the theorem. The proof of this is a reduction to the aperiodic case (Lemma 7.3).

Acknowledgment. We acknowledge the anonymous referee for his very thorough suggestions that clarified and straightened out some of these constructions. In particular, the proof of Lemma 3.3 is entirely his.

REFERENCES

[1] R. Adler and B. Marcus, *Topological Entropy and Equivalence of Dynamical Systems*, Mem. Amer. Math. Soc., Vol. 219, 1979.

[2] J. Ashley, *An extension theorem for closing maps of shifts of finite type*, Trans. Amer. Math. Soc., 336 (1993).

[3] J. Berstel and D. Perrin, *Theory of Codes*, Pure Appl. Math., Vol. 117, Academic Press, New York, 1985.

[4] M. Boyle, *Lower entropy factors of sofic systems*, Ergodic Theory Dynamical Systems, 4 (1984), pp. 541–557.

[5] M. Boyle, B. Kitchens, and B. Marcus, *A note on minimal covers for sofic systems*, Proc. Amer. Math. Soc., 95 (1985), pp. 403–411.

[6] M. Boyle, B. Marcus, and P. Trow, *Resolving Maps and the Dimension Group for Shifts of Finite Type*, Mem. Amer. Math. Soc., 1987.

[7] V. Bruyere, L. Wong, and L. Zhang, *On completion of codes with finite deciphering delay*, European J. Combin., 11 (1990), pp. 513–521.

[8] E. Coven and M. Paul, *Endomorphisms of irreducible shifts of finite type*, Math. Systems Theory, 8 (1974), pp. 167–175.

[9] ———, *Sofic systems*, Israel J. Math., 20 (1975), pp. 165–177.

[10] A. Ehrenfeucht and G. Rozenberg, *Each regular code is included in a regular maximal code*, RAIRO Inform. Théor. App., 20 (1983), pp. 89–96.

[11] G. Hedlund, *Endomorphisms and automorphisms of the shift dynamical systems*, Math. Systems Theory, 3 (1969), pp. 320–375.

[12] J. Hopcroft and J. Ullman, *Introuction to Automata Theory, Languages, and Computation*, Addison-Wesley, Reading, MA, 1979.

[13] B. Kitchens, B. Marcus, and P. Trow, *Eventual factor maps and compositions of closing maps*, Ergodic Theory Dynamical Systems, 11 (1991), pp. 85–113.

[14] W. Krieger, *On the subsystems of topological Markov chains*, Ergodic Theory Dynamical Systems, 2 (1982), pp. 195–202.

[15] M. Nasu, *Topological conjugacy of sofic systems and extensions of automorphisms of finite subsystems of topological Markov shifts*, in Proc. of the Special Year in Dynamical Systems of the University of Maryland, College Park, MD, 1986–87, Springer-Verlag, Berlin, 1987.

[16] M. Rabin and D. Scott, *Finite automata and their decision problems*, IBM J. Res., 3 (1959), pp. 115–125.

[17] A. Restivo, *Codes and local constraints*, Theoret. Comput. Sci., 72 (1990), pp. 55–64.

[18] B. Weiss, *Subshifts of finite type and sofic systems*, Monatsh. Math., 77 (1973), pp. 462–474.

SIAM J. DISCRETE MATH.
Vol. 6, No. 4, pp. 612-630, November 1993

007

AN INTEGER POLYTOPE RELATED TO THE DESIGN OF SURVIVABLE COMMUNICATION NETWORKS*

SYLVIA C. BOYD† AND TIANBAO HAO†

Abstract. The problem of designing communication networks that can survive the loss of any single link is studied. Such problems can be formulated as minimum cost 2-edge connected subgraph problems in a complete graph. The linear programming cutting plane approach has been used effectively for related problems in [*Schwerpunktprogramm der Deutschen Forschungsgemeinschaft, Anwendungsbezogene Optimierung und Steuerung*, Report No. 188, 1989], where problem-specific cutting planes that define facets of the underlying integer polyhedra are used. This paper introduces a new class of valid inequalities for the polytope associated with the minimum cost 2-edge connected subgraph problem, and necessary and sufficient conditions for these inequalities to be facet-inducing for this polytope are given.

Key words. network design, connectivity, polyhedra, facets, cutting planes

AMS subject classifications. 05C40, 90C27, 90C50

1. Introduction.

Communication networks have become more and more pervasive today, thanks to advances in computer technology and in transmission technology. They range from local area networks to cross-continent networks, and in many cases their role is vital.

There are two main issues in network design—economy and survivability. Economy refers to the construction cost. Survivability refers to the restoration of services in the event of catastrophic failures, such as the loss of a link or failure of a facility switch. The aim of network design is to minimize the construction cost while satisfying given survivability requirements. This leads naturally to the problem of designing certain k-connected networks (see [20]).

In this paper, we study networks that survive the loss of any single link and, for simplicity, we assume the construction cost to be the sum of each link cost (typically proportional to its length). This problem is known as the *2-edge connected spanning subgraph problem* (the TECSP, for short) and is of interest not only practically, but theoretically as well. Mathematically, the TECSP can be formulated as follows: Given a graph $G = (V, E)$ and vector $c \in \mathbb{R}^E$ of edge costs, find a 2-edge connected spanning subgraph having the minimum total edge cost. As is often the case, we restrict ourselves to problems where G is the complete graph K_n, since any edge not existing in G can be included with a sufficiently large cost without affecting the optimal solution. For convenience, we will denote such a problem on n nodes by TECSP(n).

The TECSP is closely related to the widely studied (symmetric) Traveling Salesman Problem (the TSP, for short), in which the aim is to find a minimum cost Hamiltonian cycle in a given weighted complete graph. Like the TSP, the TECSP is NP-hard, since the problem of determining if a graph contains a Hamiltonian cycle can be reduced to the TECSP (see [9]).

In this paper, we study the TECSP(n) using a polyhedral approach. In this approach, we first associate a polytope Q^n_{2E} with the TESCP(n), which is the convex hull of all 0-1 incidence vectors of edge sets of 2-edge connected subgraphs of K_n. We next try to find some of the necessary or "facet-inducing" linear inequalities that describe Q^n_{2E}. Some classes of these facets are already known (see [4], [13], [19]). Note that it is unlikely

*Received by the editors April 30, 1991; accepted for publication (in revised form) September 14, 1992. This research was partially supported by grants from the Natural Sciences and Engineering Research Council of Canada.

†Department of Computer Science, University of Ottawa, Ottawa, Ontario, Canada K1N 6N5.

that we will find a complete linear description of Q^n_{2E}, since the TECSP(n) is NP-hard (see [18]). However, a partial linear description can be used in a linear programming cutting plane approach to this problem. This approach, first introduced in [7], proved to be quite successful in solving large-scale TSPs to optimality; cf. [6], [10], [23]–[25]. It was also used effectively for other NP-hard combinatorial optimization problems (cf. [1], [11]) and, in particular, for the problem of designing communication networks with low connectivity constraints (see [14]).

In this paper, we introduce a large new class of facet-inducing inequalities for Q^n_{2E}, thus extending the currently known partial linear description of Q^n_{2E} and enlarging the set of TECSPs, which can potentially be solved to optimality using a cutting plane approach. This class of constraints, called the *complemented comb inequalities*, is closely related to the *comb inequalities*, a class of facet-inducing inequalities for the TSP polytope, which are used extensively in the cutting plane approach for solving TSPs. Note that the successful incorporation of the complemented comb inequalities into a cutting plane algorithm for TECSPs would require a separation algorithm (see [22]). Although we do not address the problem of separation for the constraints in this paper, we feel that their close relationship with the comb inequalities is an indication that they will be useful in cutting plane approaches for TESCPs.

The remainder of this section is devoted to some definitions and notation. In §2 we describe the complemented comb inequalities, as well as the polyhedral relationship between the Traveling Salesman polytope (Q^n_T) and Q^n_{2E}. We also describe other known classes of facet-inducing inequalities for Q^n_{2E}, which are discussed in [4], [13], and [19]. In §3 we show the validity of the complemented comb inequalities for Q^n_{2E}, and in §4 we show that most of this new class are facet-inducing, using a new proof technique described in [4], which takes advantage of the relationship between Q^n_T and Q^n_{2E}. In §5 we show that, in the general case, the complemented comb inequalities are not equivalent to any other known facet-inducing inequalities for Q^n_{2E} and characterize exactly when two complemented comb inequalities are equivalent. Finally, in §6 we make some concluding remarks.

For any finite set E, we let $\mathbb{R}^E$ denote the set of all real vectors indexed by E. For any $J \subseteq E$ and $x \in \mathbb{R}^E$, we let $x(J)$ denote $\Sigma(x_j : j \in E)$. For any subset F of E, the *incidence vector of* F is denoted by x^F and defined by

$$x^F_e = \begin{cases} 1 & \text{if } e \in F, \\ 0 & \text{otherwise.} \end{cases}$$

Given a matrix $A \in \mathbb{R}^{L \times E}$ and subset $J \subseteq E$, we let A_J represent the $(|L| \times |J|)$-submatrix of A consisting of those columns of A indexed by J. We abbreviate $A_{\{j\}}$ by A_j. The linear column rank of A is denoted by $r_\ell(A)$.

We assume that the reader is familiar with standard graph theoretical terms; accordingly, we summarize our notation and conventions here. Refer to [3] for the necessary background.

The graph $G = (V, E)$ has node set V and edge set E, where each edge has two distinct ends, belonging to V. If there is a unique edge with ends u, v, then we may denote it by uv.

Given a graph G, we let $V(G)$ denote the node set of G. For any $S \subseteq V(G)$, we let $\delta(S)$ denote the set of edges with exactly one end in S and we let $E(S)$ denote the set of edges with both ends in S. We abbreviate $\delta(\{v\})$ by $\delta(v)$. If X and Y are two subsets of $V(G)$ (not necessarily distinct), then we let $[X : Y]$ denote the set of edges in G with one end in X and the other in Y. A *Hamiltonian cycle* of G is the edge set of a connected spanning subgraph of G in which each node has degree 2. A *Hamiltonian path* of G is a Hamiltonian cycle with one edge removed. A *2-edge connected (spanning) subgraph of* G is a subgraph that remains connected after the removal of any single edge.

We assume that the reader is familiar with the basic definitions and concepts of polyhedral combinatorics; here we summarize our notation and specialized definitions. Refer to [2], [21], or [26] for the necessary background.

For any polyhedron $P \subseteq \mathbb{R}^E$, we let $\dim(P)$ represent the dimension of P. For any finite $X \subseteq \mathbb{R}^E$, we denote the convex hull of X by $\operatorname{conv}(X)$.

Given a linear system defining a polyhedron P, the set of constraints that are satisfied with equality by all $x \in P$ is called an *equation system for* P. Given a polyhedron P with equation system $Ax = b$ and valid inequalities $ax \leq a_0$ and $\bar{a}x \leq \bar{a}_0$, we say that these inequalities are *equivalent* (*with respect to* P) if $\{x \in P : ax = a_0\} = \{x \in P : \bar{a}x = \bar{a}_0\}$; i.e., they induce the same face of P. If $ax \leq a_0$ and $\bar{a}x \leq \bar{a}_0$ are facet-inducing for P, these inequalities are equivalent if and only if there exists $\gamma > 0$ and vector λ such that $a = \gamma\bar{a} + \lambda A$ and $a_0 = \gamma\bar{a}_0 + \lambda b$.

2. The complemented comb constraints. Let $K_n = (V, E)$ be the complete graph on n nodes. The *Traveling Salesman polytope*, denoted by Q_T^n, is the convex hull of all incidence vectors of Hamiltonian cycles of K_n, i.e.,

$$Q_T^n := \operatorname{conv}\left\{x^H \in \mathbb{R}^E : H \text{ is a Hamiltonian cycle of } K_n\right\}.$$

The *2-edge connected spanning subgraph polytope*, denoted by Q_{2E}^n, is the convex hull of all incidence vectors of the edge sets of 2-edge connected spanning subgraphs of K_n, i.e.,

$$Q_{2E}^n := \operatorname{conv}\left\{x^F \in \mathbb{R}^E : (V, F) \text{ is a 2-edge connected spanning subgraph of } K_n\right\}.$$

We begin this section with a brief discussion of the currently known facets for Q_T^n and Q_{2E}^n.

There has been extensive research on Q_T^n and its facets. Here we only mention the results that are essential for later sections.

THEOREM 2.1 (see [15]). *The degree constraints* $x(\delta(v)) = 2$ *for all* $v \in V$ *form a minimal equation system for* Q_T^n.

Note that the set of degree constraints for Q_T^n can be written as $Ax = \mathbf{2}$, where A is the node-edge incidence matrix of K_n. A consequence of Theorem 2.1 is the following theorem.

THEOREM 2.2 (see [15]). *The dimension of* Q_T^n *is* $|E| - n$ *for* $n \geq 3$.

Since Q_T^n is not full-dimensional, we may have inequalities $ax \leq a_0$ and $bx \leq b_0$, which induce the same facet of Q_T^n and yet look quite different. An exact description of the equivalence relationship for Q_T^n is the following (see §1).

THEOREM 2.3. *Two facet-inducing inequalities* $ax \leq a_0$ *and* $bx \leq b_0$ *are equivalent for* Q_T^n *if and only if there exists* $\gamma > 0$ *and vector* λ *such that* $b = \gamma a + \lambda A$ *and* $b_0 = \gamma a_0 + \lambda\mathbf{2}$, *where* A *is the node-edge incidence matrix for* K_n.

There are several well-known classes of facets for Q_T^n.

THEOREM 2.4 (see [15]). *For all* $n \geq 5$ *and all* $e \in E$, *the nonnegativity constraints* $x_e \geq 0$ *induce distinct facets of* Q_T^n.

THEOREM 2.5 (see [16]). *For all* $n \geq 4$ *and* $S \subseteq V$ *satisfying* $2 \leq |S| \leq \lfloor n/2 \rfloor$, *the subtour elimination constraints* $x(\delta(S)) \geq 2$ *induce distinct facets of* Q_T^n.

Another well-known class of facet-inducing inequalities for Q_T^n are the *comb constraints*, which were first introduced by Chvátal [5] and later generalized by Grötschel and Padberg in [15]. A *comb* consists of a *handle* $H \subseteq V$ and mutually disjoint *teeth* $T_1, T_2, \ldots, T_k \subseteq V$ ($k \geq 3$ and odd) such that

$$T_j \cap H \neq \emptyset \neq T_j \backslash H, \qquad 1 \leq j \leq k.$$

The associated *comb inequality* is

$$x(E(H)) + \sum_{i=1}^{k} x(E(T_i)) \leq |H| + \sum_{i=1}^{k} (|T_i| - 1) - \frac{k+1}{2}. \tag{2.1}$$

Note that the coefficients on the left-hand side of this inequality are 0,1,2. Figure 2.1 illustrates the left-hand side coefficients for the comb constraint $ax \leq 8$. Edges with coefficient 0 are not shown.

THEOREM 2.6 (see [15], [17]). *If* $\{H, T_1, T_2, \ldots, T_k\}$ *is a comb, then so too is* $\{V\backslash H, T_1, T_2, \ldots, T_k\}$, *and these induce the same facet of* Q^n_T. *In all other cases, each comb induces a facet distinct from all other combs, subtour elmination constraints, and nonnegativity constraints.*

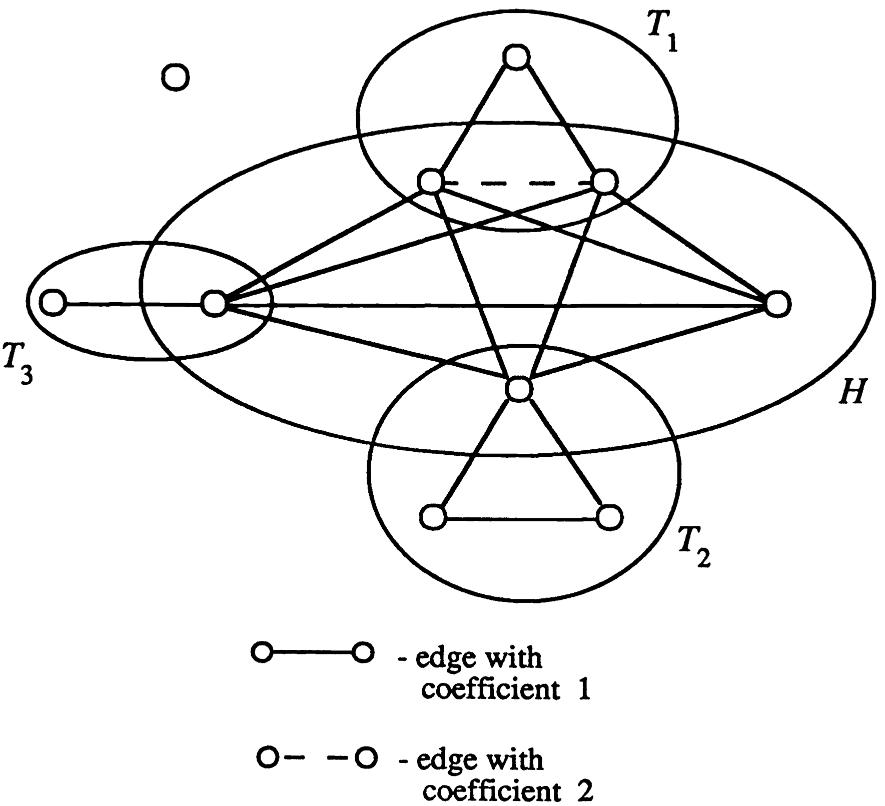


FIG. 2.1. *Edge coefficients of a comb inequality.*

In the case where $|T_i| = 2$ for all $1 \leq i \leq k$, the comb inequality (2.1) simplifies to

$$x(E(H)) + \sum_{i=1}^{k} x(E(T_i)) \leq |H| + \frac{k-1}{2}. \tag{2.2}$$

These inequalities are called the *2-matching constraints* and were first introduced in [8].

There has been some research done on the polytope Q_{2E}^n. In a much more general form, Q_{2E}^n is studied by Grötschel and Monma in [12] and by Grötschel, Monma, and Stoer in [13]. It is also studied by Mahjoub in [19] for general graphs G. Below, we describe all the results pertaining to the facets of Q_{2E}^n from [12], [13], and [19].

THEOREM 2.7. *The dimension of Q_{2E}^n is $|E|$ for $n \geq 4$.*

Note that, by Theorem 2.7, the equivalence problem for Q_{2E}^n is much simpler than for Q_T^n. Two facet-inducing inequalities for Q_{2E}^n are equivalent if and only if one is a positive multiple of the other.

THEOREM 2.8. *For $e \in E$, $x_e \leq 1$ is facet-inducing for Q_{2E}^n if $n \geq 4$.*

THEOREM 2.9. *For $e \in E$, $x_e \geq 0$ is facet-inducing for Q_{2E}^n if $n \geq 5$.*

THEOREM 2.10. *For all $S \subseteq V$, $\emptyset \neq S \neq V$, $x(\delta(S)) \geq 2$ is facet-inducing for Q_{2E}^n.*

Another class of inequalities called the lifted 2-cover inequalities are introduced in [13] for a more general polytope (of which Q_{2E}^n is a special case), and these are shown to be facet-inducing under certain conditions. These inequalities can be described as follows: Let $H \subseteq V$ be a node set, let $T \subseteq \delta(H)$ be an edge set such that $|T| \geq 3$ and odd, and let $H_1, H_2, \ldots, H_p$, $p \geq 3$ be a partition of H into nonempty disjoint node sets such that no more than two edges in T intersect any H_i. Then the lifted 2-cover inequality is given by

$$x(E(H)) - \sum_{i=1}^{p} x(E(H_i)) + x(\delta(H)) - x(T) \geq p - \frac{(|T|-1)}{2}. \tag{2.3}$$

In [19] Mahjoub found the same class of inequalities, which he calls the "odd wheel inequalities," for the 2-edge connected spanning subgraph polytope for general graphs.

The polytope Q_{2E}^n is closely related to the polytope Q_T^n. It is easy to see, from a graph theoretic point of view, that a 2-edge connected spanning subgraph forms a Hamiltonian cycle if and only if each node in the subgraph is of degree 2. So $Q_T^n = \{x \in Q_{2E}^n | Ax = \mathbf{2}\}$, where A is the node-edge incidence matrix for K_n, i.e., Q_T^n is a face of Q_{2E}^n. Therefore, for every facet-inducing inequality for Q_T^n, there exists an equivalent form of that inequality (with respect to Q_T^n), which is also facet-inducing for Q_{2E}^n. It would be desirable to exploit this relationship and somehow transform a facet-inducing inequality for Q_T^n into an equivalent one that is facet-inducing for Q_{2E}^n. In [4] Boyd and Pulleyblank discuss how to do this for general polyhedra through a process called dimension augmentation. Using this process, they convert the 2-matching constraints (2.2) for Q_T^n into an equivalent form that is facet-inducing for Q_{2E}^n. These inequalities are called the *complemented 2-matching constraints* and have the form

$$ax \geq |\overset{o}{V}| + (k+1)/2, \tag{2.4}$$

where $\overset{o}{V} \subset V$ is the set of nodes not contained in H or any T_i, $1 \leq i \leq k$, and a is defined by

$$a_e = \begin{cases} 0 & \text{for } e \in E(H) \text{ or } e \in E(T_i),\ 1 \leq i \leq k, \\ 1 & \text{otherwise.} \end{cases}$$

We now convert the comb inequalities (2.1) into a set of equivalent inequalities (with respect to Q_T^n) that are valid for Q_{2E}^n and contain the complemented 2-matching constraints (2.4). Using some of the methods discussed in [4], we later show that these inequalities are facet-inducing for Q_{2E}^n in most cases.

Given a comb, we define a *2-matching tooth* to be a tooth consisting of only two nodes and divide the nodes of a comb into the following six different classes:

Class 1. Nodes in the handle but not in any tooth;

Class 2. Nodes in the handle as well as in a 2-matching tooth;

Class 3. Nodes in a 2-matching tooth but not in the handle;

Class 4. Nodes in a non-2-matching tooth but not in the handle;

Class 5. Nodes in a non-2-matching tooth as well as in the handle;

Class 6. Nodes not in the handle or in any tooth.

We denote the node set for each class by C_i, $i = 1, 2, 3, 4, 5, 6$. Figure 2.2 shows a comb with the different classes of nodes indicated.

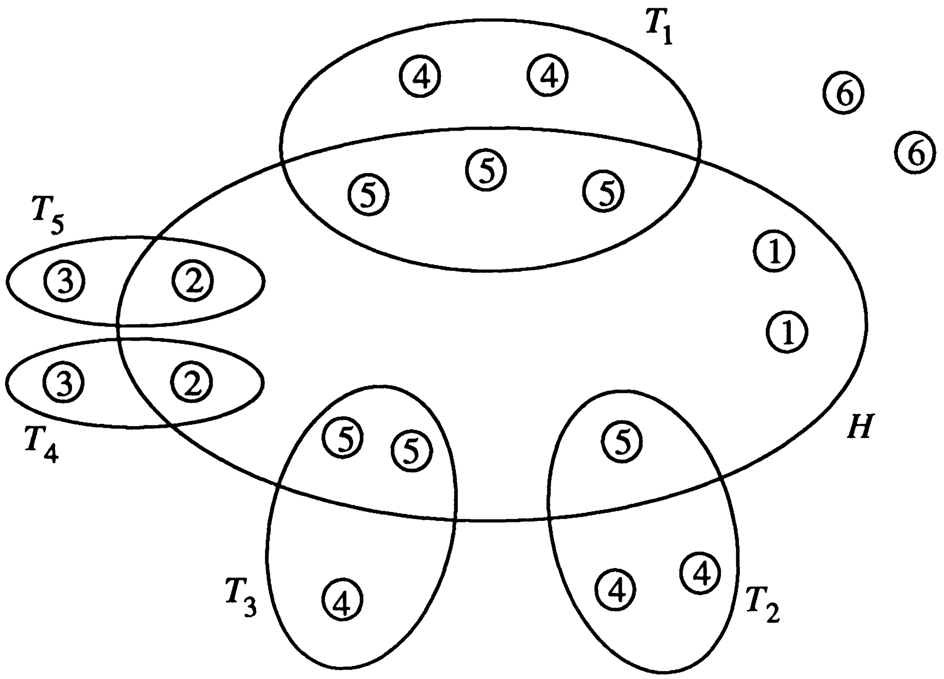


FIG. 2.2. *A comb with the different classes of nodes labeled.*

Given a comb C, we let $s = |C_1|$, $r = |C_6|$, p be the number of non-2-matching teeth, and q be the number of 2-matching teeth. Furthermore, unless otherwise stated, we assume that $T_1, T_2, \ldots, T_p$ are the non-2-matching teeth and that $T_{p+1}, T_{p+2}, \ldots, T_k$ are the 2-matching teeth and we let $\underset{o}{T_i} = T_i \cap H$ and $\overset{o}{T_i} = T_i \backslash H$ for $1 \leq i \leq k$.

We now convert the comb inequality (2.1) into an equivalent form (with respect to Q_T^n) as follows:

1. Negate the inequality;
2. Multiply by 2;
3. Add two times the degree constraint for the nodes in C_5 and one time the degree constraint for each other node.

By using the relation $\sum_{v \in S} x(\delta(v)) = 2x(E(S)) + x(\delta(S))$ for each subset S of V, the resulting inequality is

$$(2.5)\qquad x(\delta(H))+\sum_{i=1}^{k}x(\delta(T_i))-\sum_{v\in C_2}x(\delta(v))+\sum_{w\in C_6}x(\delta(w))\geq 3p+2r+q+1.$$

We call the inequalities (2.5) the *complemented comb constraints* and use the notation $bx \geq b_0$ to represent such a constraint for convenience, where

$$b_0 = 3p+2r+q+1$$

and where b is defined by

$$b_e=\begin{cases} 0 & \text{if } e\in E(C_1\cup C_2)\cup E(\overset{o}{T_i})\cup E(\underset{o}{T_i})\cup E(T_j) \quad \text{for } 1\leq i\leq p,\ p+1\leq j\leq k;\\ 1 & \text{if } e\in [\underset{o}{T_i}: C_1\cup C_2\cup \overset{o}{T_i}] \quad \text{for } 1\leq i\leq p;\\ 3 & e\in [\underset{o}{T_i}: C_6\cup \overset{o}{T_j}] \quad \text{for } 1\leq i\leq p,\ 1\leq j\leq k, \text{ and } i\neq j;\\ 2 & \text{elsewhere.}\end{cases}$$

Figure 2.3 shows the edge coefficients b for the complemented comb inequality $bx \geq 10$ corresponding to the comb shown in Fig. 2.1. Here, edges e for which $b_e = 2$ are not shown.

Note that, in the case where $|T_i| = 2$ for all teeth T_i of comb C, the corresponding complemented comb inequality is simply a complemented 2-matching inequality multiplied by 2, and thus these are equivalent for Q^n_{2E}. Thus the complemented 2-matching constraints (2.4) are a subset of the complemented comb constraints (2.5).

3. Validity of the complemented comb constraints. In this section, we give a proof of the validity of the complemented comb inequalities (2.5) for Q^n_{2E}. First, we introduce the following lemma.

LEMMA 3.1. *For any $x \in Q^n_{2E}$ and comb C, we have*

(i) $x(\delta(v)) \geq 2$ *for all* $v \in C_6$;

(ii) $x(\delta(\overset{o}{T_i})) - x(E(T_i)) \geq 1$ *for* $p+1 \leq i \leq k$;

(iii) $\frac{1}{2}x(\delta(\overset{o}{T_i})) \geq 1$ *for* $1 \leq i \leq p$;

(iv) $\frac{1}{2}x(\delta(\underset{o}{T_i})) \geq 1$ *for* $1 \leq i \leq p$;

(v) $\frac{1}{2}x(\delta(T_i)) \geq 1$ *for* $1 \leq i \leq p$;

(vi) $x_e \geq 0$ *for* $e \in [C_6 : H]$;

(vii) $x_e \geq 0$ *for* $e \in [C_3 : H]\setminus \bigcup_{i=p+1}^{k} E(T_i)$;

(viii) $x_e \geq 0$ *for* $e \in [C_4 : H]\setminus \bigcup_{i=1}^{p} E(T_i)$.

Furthermore, not all of the above inequalities can hold with equality simultaneously.

Proof. Let $K_n = (V, E)$. For any $x \in Q^n_{2E}$, we clearly have

$$0 \leq x_e \leq 1 \quad \text{for all } e \in E, \qquad \text{and}$$

$$x(\delta(S)) \geq 2 \quad \text{for all } S \subset V,\ \emptyset \neq S \neq V.$$

Thus (i), (iii)–(viii) are all valid for Q^n_{2E}. Inequality (ii) can be obtained by adding the two valid constraints $-x_e \geq -1$ and $x(\delta(\overset{o}{T_i})) \geq 2$ for $p+1 \leq i \leq k$ and $\{e\} = E(T_i)$, and thus (ii) is valid for Q^n_{2E} as well.

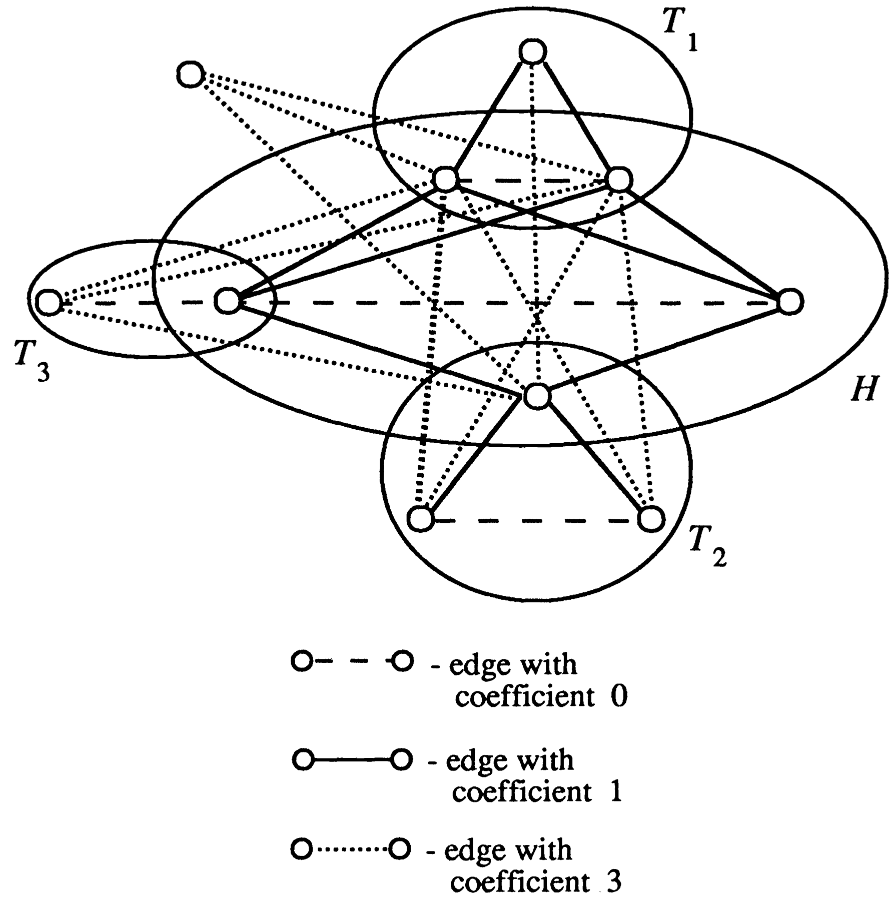

T_1
T_3
H
T_2
- edge with coefficient 0
- edge with coefficient 1
- edge with coefficient 3

FIG. 2.3. *Edge coefficients of a complemented comb inequality.*

Now suppose that there exists $x \in Q^n_{2E}$, which satisfies all the inequalities (i), ..., (viii) with equality. Without loss of generality, we can assume that x is the incidence vector x^F of a 2-edge connected spanning subgraph of K_n with edge set F. Let

$$E' = \bigcup \left\{ [C_6 \cup \overset{o}{T}_i : C_6 \cup \overset{o}{T}_j] : 1 \leq i \neq j \leq k \right\}$$

and let $E^* = F \cap E'$. We claim that $|\delta(v) \cap E^*| = 2$ for any $v \in C_6$ and that $|\delta(\overset{o}{T}_i) \cap E^*| = 1$ for $1 \leq i \leq k$.

For any $v \in C_6$, we have

$$x^F(\delta(v)) = x^F(\delta(v) \cap E') + x^F[v : H].$$

Since $x^F(\delta(v)) = 2$ by (i) and $x^F_e = 0$ for all $e \in [v : H]$ by (vi), it follows that $x^F(\delta(v) \cap E') = 2$. Thus $|\delta(v) \cap E^*| = 2$, as required.

For T_i, $p + 1 \leq i \leq k$, we have

$$x^F(\delta(\overset{o}{T}_i)) = x^F(\delta(\overset{o}{T}_i) \cap E') + x^F(E(T_i)) + x^F([\overset{o}{T}_i : H] \backslash E(T_i)).$$

By (vii), $x^F_e = 0$ for all $e \in [\overset{o}{T_i} : H]\backslash E(T_i)$, and thus

$$x^F(\delta(\overset{o}{T_i}) \cap E') = x^F(\delta(\overset{o}{T_i})) - x^F(E(T_i)) = 1$$

by (ii). It follows that $|\delta(\overset{o}{T_i}) \cap E^*| = 1$, as required.

For $T_i,\ 1 \le i \le p$, we have

$$x^F(\delta(\overset{o}{T_i})) = x^F(\delta(\overset{o}{T_i}) \cap E') + x^F[\overset{o}{T_i} : \underset{o}{T_i}] + x^F([\overset{o}{T_i} : H]\backslash E(T_i)).$$

By (viii), $x^F_e = 0$ for all $e \in [\overset{o}{T_i} : H]\backslash E(T_i)$. Also, by (iii)–(v), we have $x^F(\delta(\overset{o}{T_i})) = x^F(\delta(\underset{o}{T_i})) = x^F(\delta(T_i)) = 2$, which implies that $x^F[\overset{o}{T_i} : \underset{o}{T_i}] = 1$. Thus $x^F(\delta(\overset{o}{T_i}) \cap E') = 1$. It follows that $|\delta(\overset{o}{T_i}) \cap E^*| = 1$, and the claim has been proved.

Since $2|E^*| = \Sigma(|\delta(v) \cap E^*| : v \in C_6) + \Sigma(|\delta(\overset{o}{T_i}) \cap E^*| : 1 \le i \le k)$, it now follows that $2|E^*| = 2|C_6| + k$. However, k is odd, and thus $|E^*| = |C_6| + k/2$, which is not an integer, leading to a contradiction. □

THEOREM 3.2. *The complemented comb constraints* (2.5) *are valid for* Q^n_{2E}.

Proof. Given comb C, let $bx \ge b_0$ be the corresponding complemented comb constraint and let $\hat{b}x \ge \hat{b}_0$ be the valid inequality for Q^n_{2E} obtained by adding all the inequalities in (i), (ii), ..., (viii) described in Lemma 3.1. Since not all of these inequalities can hold with equality simultaneously by Lemma 3.1, it follows that $\hat{b}x > \hat{b}_0$ for all $x \in Q^n_{2E}$. Furthermore, since $\hat{b}$, $\hat{b}_0$, and all the vertices of Q^n_{2E} are integer, we have

$$\hat{b}x \ge \hat{b}_0 + 1 \quad \text{for all } x \in Q^n_{2E}. \tag{3.1}$$

It is straightforward to check that $b_e = \hat{b}_e$ for all $e \in E(K_n)$. The right-hand side of (3.1) is $2|C_6| + q + 3p + 1 = b_0$, and thus (3.1) gives $bx \ge b_0$ for all $x \in Q^n_{2E}$, as required. □

The following corollary is a direct consequence of Theorem 3.2 and its proof. We will use it later to prove that some particular Hamiltonian cycles H satisfy $bx^H = b_0$.

COROLLARY 3.3. *For any* $x \in Q^n_{2E}$ *and comb* C, *we have* $bx = b_0$ *if and only if* x *satisfies all inequalities* (i), (ii), ..., (viii) *in Lemma* 3.1 *with equality except for exactly one, which is violated by* 1.

4. A characterization of facet-inducing complemented comb constraints.

In this section, we characterize the complemented comb constraints that are facet-inducing. We make use of some of the results discussed in [4], which will allow us to exploit the relationship between Q^n_T and Q^n_{2E}, thus hopefully simplifying our proof.

Given a valid inequality $ax \le a_0$ for a polyhedron F, let $F^=_a$ represent the face of F induced by $ax \le a_0$, i.e.,

$$F^=_a := \{x \in F : ax = a_0\}.$$

Given a nonempty face F of a full-dimensional polytope $P \in \mathbb{R}^E$ and a valid inequality $ax \le a_0$ for F and P, define a set $D \subseteq \mathbb{R}^E$ to be an *independent direction set* for $F^=_a$ if

(i) For every $d \in D$, there exists $\hat{x}^d \in F^=_a$ such that $x^d := \hat{x}^d + d \in P$;

(ii) For every $d \in D,\ ad = 0$;

(iii) For some minimal equation system $A^F x = b^F$ for F, $\{A^F d : d \in D\}$ are linearly independent.

THEOREM 4.1 (Corollary 2.2 in [4]). *Let $ax \leq a_0$ be an inequality that is valid for a full-dimensional polyhedron $P \in \mathbb{R}^E$ and facet-inducing for a nonempty face F of P. Let $A^F x = b^F$ be a minimal equation system for F. If there exists an independent direction set for $F_a^=$ of size $r_\ell(A^F)$, then $ax \leq a_0$ is also facet-inducing for P.*

Since Q_T^n is a nonempty face of Q_{2E}^n, $bx \geq b_0$ is valid for Q_{2E}^n, and $Ax = \mathbf{2}$ is a minimal equation system for Q_T^n by Theorem 2.1, we obtain the following corollary from Theorem 4.1.

COROLLARY 4.2. *For any comb C, the corresponding complemented comb inequality $bx \geq b_0$ is facet-inducing for Q_{2E}^n if there exists a set $D = \{d_1, d_2, \ldots, d_n\} \subseteq \mathbb{R}^E$ such that*

(P_1) For each $1 \leq i \leq n$, there exists $\hat{x}_i \in Q_T^n \cap \{x : bx = b_0\}$ such that $x_i := \hat{x}_i + d_i \in Q_{2E}^n$;

(P_2) For each $1 \leq i \leq n$, $bd_i = 0$;

(P_3) $Ad_1, Ad_2, \ldots, Ad_n$ are linearly independent, where A is the node-edge incidence matrix for the complete graph K_n.

We now find such an independent direction set D. First, we describe two specific Hamiltonian cycles and introduce several lemmas that will be used later.

Let C be any comb with teeth $T_1, T_2, \ldots, T_k$. We pair T_{2i} and T_{2i+1} for $1 \leq i \leq (k-1)/2$ and construct a Hamiltonian cycle H_1 of K_n as follows. For each pair T_{2i} and T_{2i+1}, H_1 first traverses each node in $\underset{o}{T_{2i}}$, then each node in $\overset{o}{T}_{2i}$; then it goes to $\overset{o}{T}_{2i+1}$ and traverses each node in $\overset{o}{T}_{2i+1}$, then each node in $\underset{o}{T_{2i+1}}$, before going to the next pair. In the case where $i = (k-1)/2$, H_1 traverse each node in C_6 before going to $\overset{o}{T}_k$. After traversing each node in T_k, H_1 goes to C_1 and traverses each node in C_1 before traversing each node in T_1. In $\overset{o}{T}_1$, H_1 first traverses each node in $\underset{o}{T_1}$, then each node in $\overset{o}{T}_1$, before returning to the node in $\underset{o}{T_2}$ with which H_1 started. The resulting Hamiltonian cycle H_1 is shown in Fig. 4.1.

Now suppose that comb C with teeth $T_1, T_2, \ldots, T_k$ has at least one 2-matching tooth. Without loss of generality, we assume that T_1 is a 2-matching tooth. We construct a Hamiltonian cycle H_2 as follows. First, similar to the construction of H_1, we pair T_{2i} and T_{2i+1} for $1 \leq i \leq (k-1)/2$. We then construct H_2 in the same way we constructed H_1, except for teeth T_1, T_k, and node set C_1. After visiting each node in C_6, H_2 traverses the node w in $\overset{o}{T}_1$, then each node in $\overset{o}{T}_k$, and then each node in $\underset{o}{T_k}$. Next, H_2 traverses each node in C_1 and then the node v in $\underset{o}{T_1}$ before returning to the node in $\underset{o}{T_2}$ with which H_2 started. The resulting Hamiltonian cycle is shown in Fig. 4.2.

LEMMA 4.3. *Given a comb C, let $bx \geq b_0$ be the corresponding complemented comb constraint. Then $bx^{H_1} = b_0$ and, in the case that C has T_1 as a 2-matching tooth, $bx^{H_2} = b_0$.*

Proof. It is easily verified that x^{H_1} satisfies all of the inequalities (i), (ii), ..., (viii) in Lemma 3.1 with equality except for $x_{uv} \geq 0$, where u is the first node in $\underset{o}{T_2}$ with which H_1 starts and v is the last node in $\overset{o}{T}_1$ that H_1 visits (see Fig. 4.1). We have $x_{uv}^{H_1} = 1$. Thus, by Corollary 3.3, it follows that $bx^{H_1} = b_0$.

Similarly, it is easily verified that x^{H_2} satisfies all of the inequalities (i), (ii), ..., (viii) in Lemma 3.1 with equality except for $x(\delta(\overset{o}{T}_1)) - x(E(T_1)) \geq 1$. We have $x^{H_2}(E(T_1)) = 0$ and $x^{H_2}(\delta(\overset{o}{T}_1)) = 2$, i.e., $x^{H_2}(\delta(\overset{o}{T}_1)) - x^{H_2}(E(T_1)) = 2$. Thus, by Corollary 3.3, it follows that $bx^{H_2} = b_0$. □

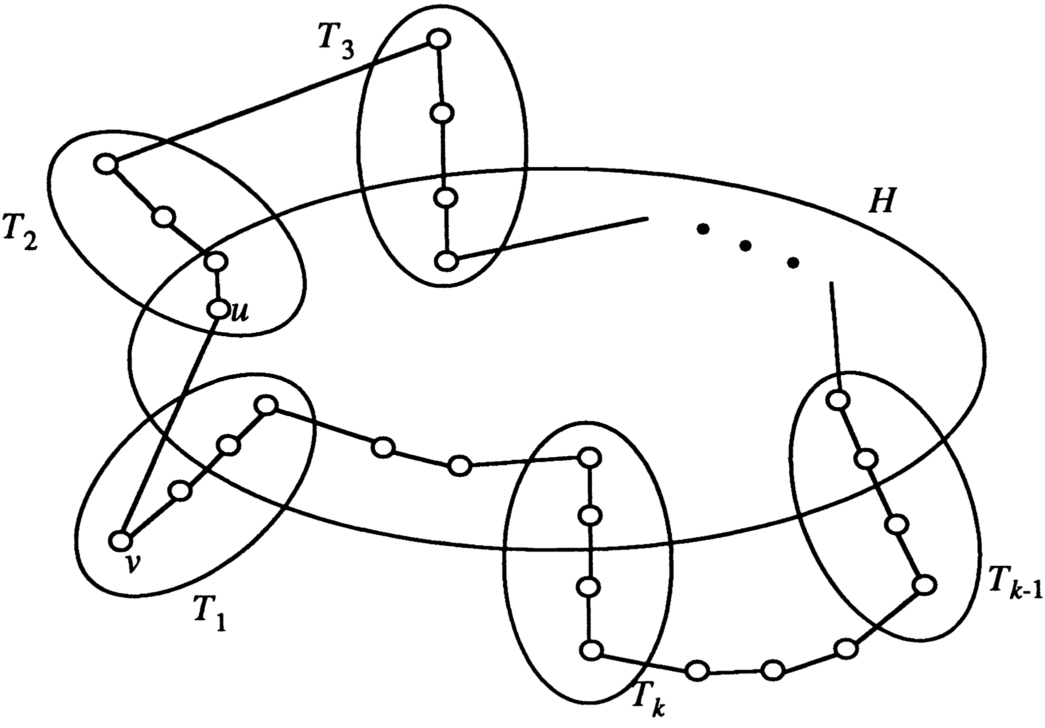


FIG. 4.1. *Hamiltonian cycle H_1 satisfying $bx^{H_1} = b_0$.*

Given a comb C and corresponding complemented comb inequality $bx \geq b_0$, we say that node $v \in V(K_n)$ *induces a tight triangle* (v, u, w) if there exists $u, w \in V(K_n)$ and Hamiltonian cycle H of K_n such that

(i) $bx^H = b_0$;

(ii) $b_{uv} + b_{vw} = b_{uw}$;

(iii) H contains uw, but not vw or uv.

Note that, if v induces a tight triangle (v, u, w), then $H' = (H \backslash \{uw\}) \bigcup \{vw, uv\}$ forms a 2-edge connected graph satisfying $bx^{H'} = b_0$.

LEMMA 4.4. *Given a comb C, node $v \in V(K_n)$ induces a tight triangle if $v \in C_1 \cup C_2 \cup C_4 \cup C_5$.*

Proof. Construct the Hamiltonian cycle H_1 as previously described. By Lemma 4.3, $bx^{H_1} = b_0$. Let w be the first node in tooth T_2 visited by H_1 and let u be the last node in T_1 visited by H_1 (see Fig. 4.3).

If $C_i \neq \emptyset$ for $i = 1, 2$, then let v_1 be any node in C_1 and let v_2 be any node in C_2. Without loss of generality, we can assume that $v_2 \in T_k$, i.e., T_k is a 2-matching tooth (see Fig. 4.3). Then H_1 contains uw, but not v_iu or v_iw for $i = 1, 2$. Moreover, for $i = 1, 2$, we have $b_{v_iw} = 0$, $b_{v_iu} = b_{uw} = 2$ if T_2 is a 2-matching tooth, and $b_{v_iw} = 1$, $b_{v_iu} = 2$, $b_{uw} = 3$, otherwise. Thus node v_i induces tight triangle (v_i, u, w) for $i = 1, 2$.

If $C_4 \neq \emptyset$, then let v_4 be any node in C_4. Without loss of generality, we can assume that $v_4 \in T_2$ (i.e., T_2 is a non-2-matching tooth), and v_4 is the last node in T_2 visited by H_1 (see Fig. 4.3). Then H_1 contains uw but not v_4u or v_4w. Moreover, we have $b_{v_4w} = 1$, $b_{v_4u} = 2$, and $b_{uw} = 3$. Thus node v_4 induces tight triangle (v_4, u, w).

If $C_5 \neq \emptyset$, then let v_5 be any node in C_5. Without loss of generality, we can assume that $v_5 \in T_1$ (i.e., T_1 is a non-2-matching tooth) and that v_5 is the first node in T_1 visited

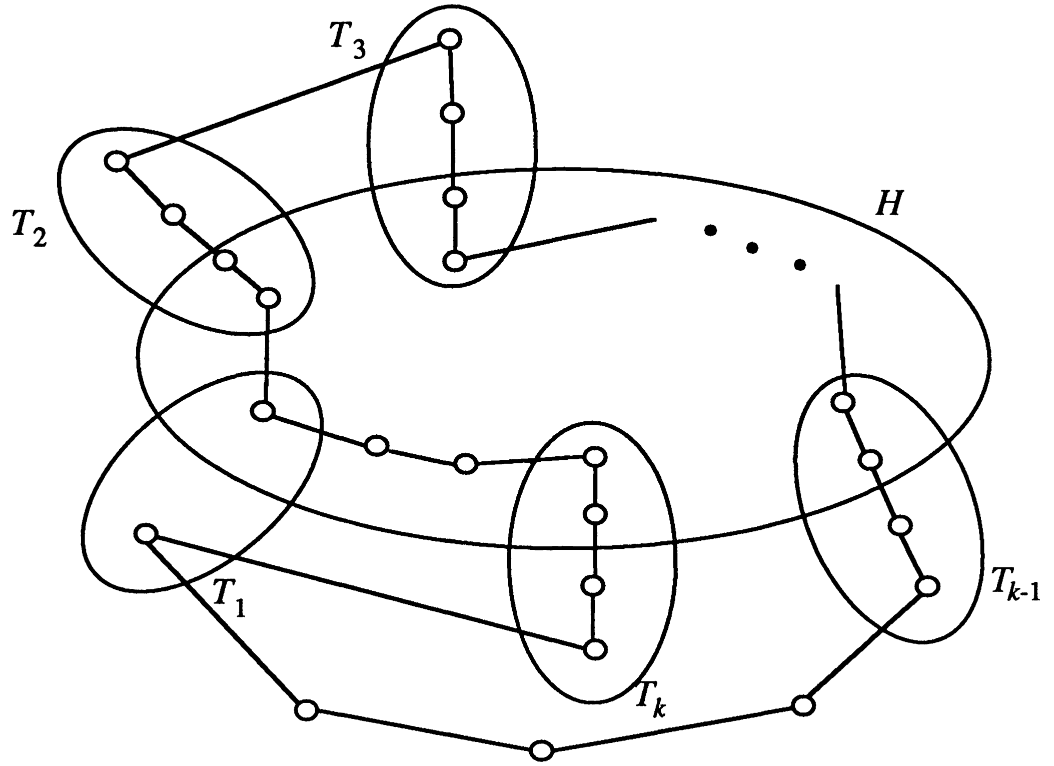


FIG. 4.2. *Hamiltonian cycle H_2 satisfying $bx^{H_2} = b_0$.*

by H_1 (see Fig. 4.3). Then H_1 contains uw but not v_5u or v_5w. Moreover, $b_{v_5w} = b_{v_5u} = 1$, $b_{uw} = 2$ if T_2 is a 2-matching tooth, and $b_{v_5w} = 2$, $b_{v_5u} = 1$, $b_{uw} = 3$ if T_2 is a non-2-matching tooth. Thus node v_5 induces tight triangle (v_5, u, w). □

LEMMA 4.5. *Let A be the node-edge incidence matrix for the complete graph $K_n = (V, E)$, let $I \in \mathbb{R}^{V \times V}$ be the identity matrix whose rows and columns are indexed by V, and let $T = (V_T, E_T)$ be a subgraph of K_n, which is a tree. Then the vectors in $\{A_e : e \in E_T\} \cup \{I_v\}$ are linearly independent for any $v \in V$.*

Proof (by induction on $|E_T|$). If $|E_T| = 0$, then the result follows. So we assume that $|E_T| \geq 1$ and that the result is true for any tree with fewer than $|E_T|$ edges. Let $\lambda \in \mathbb{R}^{E_T}$ and $\lambda_v \in \mathbb{R}$ be such that

$$(4.1) \qquad \Sigma(\lambda_e A_e : e \in E_T) + \lambda_v I_v = \mathbf{0}.$$

Since T is a tree and $|E_T| \geq 1$, T has at least two leaves, one of which is some node $u \neq v$. Let e' be the unique edge in E_T incident with u. Then, in the component indexed by u, $A_{e'}$ has 1, while A_e, $e \in E_T \backslash \{e'\}$, and I_v each has 0. Thus

$$(4.2) \qquad \lambda_{e'} = 0.$$

Furthermore, $T' = (V_T \backslash \{u\}, E_T \backslash \{e'\})$ is a tree with fewer edges than T; hence, by our induction hypothesis, the result is true for T'. This, combined with (4.1) and (4.2), gives $\lambda = \mathbf{0}$ and $\lambda_v = 0$ as desired. □

The following theorem is the main result of this chapter and characterizes the complemented comb constraints that are facet-inducing.

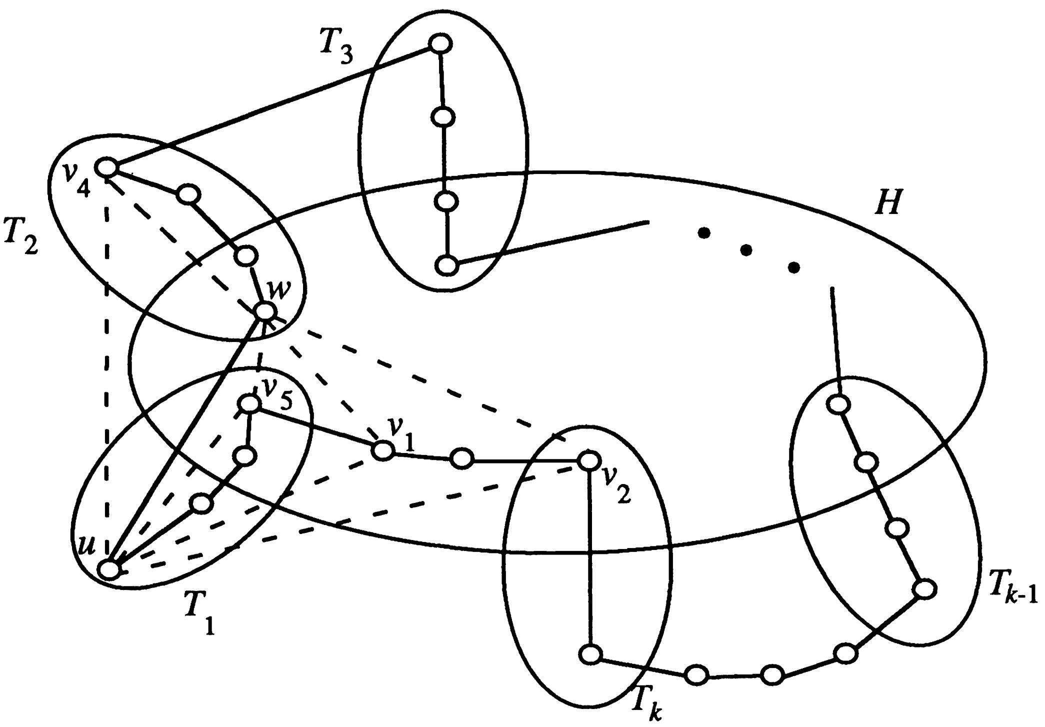


FIG. 4.3. *Induced tight triangles for* v_1, v_2, v_4, v_5.

THEOREM 4.6. *Given any comb C, the corresponding complemented comb inequality $bx \geq b_0$ is facet-inducing for Q^n_{2E} if and only if $r = 0$ or $(q + s) \geq 1$.*

Proof. Let $G_0 = (V_{G_0}, E_{G_0})$ be the spanning subgraph of $K_n = (V, E)$ whose edge set corresponds to the edges e in K_n for which $b_e = 0$, i.e., $E_{G_0} := \{e \in E : b_e = 0\}$. The graph G_0 consists of several components. To be specific, all nodes in C_1, C_2, and C_3 are in the same component, with the nodes in $C_1 \cup C_2$ inducing a complete subgraph; for each non-2-matching tooth T_i, the nodes in $\overset{o}{T_i}$ form a component, and so do those in $\underset{o}{T_i}$; each node in C_6 forms a component by itself. Altogether, G_0 has $2p + r + 1$ components if $|C_1| + |C_2| \geq 1$ and $2p + r$ components if $|C_1| = |C_2| = 0$.

Consider any component M of G_0 that does not contain a node in C_6 and let J be the edge set of a spanning tree in M. By Lemma 4.4, M has at least one node v that induces a tight triangle (v, u, w). Let $d^v := x^{vu} + x^{vw} - x^{uw}$ and let $d^e := x^e$ for all $e \in J$. Then we claim that

$$D^M := \{d^e : e \in J\} \cup \{d^v\}$$

satisfies properties (P_1)–(P_3) of Corollary 4.2 (when n is replaced by $|V(M)| = |D^M|$).

Recall that $bx \geq b_0$ is equivalent to a comb constraint for Q^n_T and thus by Theorem 2.6 does not induce the facet $x_e = 1$, $e \in E$, in Q^n_T. Hence, for each $e \in J$, there exists a Hamiltonian cycle H such that $bx^H = b_0$ and H does not contain edge e. Clearly, $H \cup \{e\}$ is a 2-edge connected subgraph of K_n, i.e., $x^H + d^e \in Q^n_{2E}$. Furthermore, by the definition of a tight triangle, there exists a Hamiltonian cycle H such that $bx^H = b_0$, and $(H \backslash \{uw\}) \cup \{uv, wv\}$ is a 2-edge connected subgraph of K_n, i.e., $x^H + d^v \in Q^n_{2E}$. Thus D^M satisfies property (P_1) of Corollary 4.2.

For each edge $e \in J$, $bd^e = b_e = 0$ by the definition of G_0. Also, $bd^v = b_{uv} + b_{vw} - b_{uw} = 0$ by the definition of a tight triangle. Thus D^M satisfies property (P_2) of Corollary 4.2.

Now consider $\{Ad : d \in D^M\}$, where A is the node-edge incidence matrix for K_n. For each $e \in J$, $Ad^e = A_e$, and $Ad^v = 2I_v$, where $I \in \mathbb{R}^{V \times V}$ is the identity matrix whose rows and columns are indexed by V. Thus, by Lemma 4.5, the vectors in $\{Ad : d \in D^M\}$ are linearly independent, and D^M satisfies property (P_3) of Corollary 4.2.

Now consider

$$D := \bigcup (D^M : M \text{ is a component of } G_0 \text{ not containing a node in } C_6).$$

We claim that D is an independent direction set for $Q_T^n \cap \{x \in \mathbb{R}^E : bx = b_0\}$ with size $n - r$. First, for each component M that does not contain a node in C_6, D^M contains $|V(M)|$ vectors, and thus $|D| = n - |C_6| = n - r$. Second, since each set D^M satisfies properties (P_1) and (P_2) of Corollary 4.2, so does D. Finally, to see that D also satisfies property (P_3), note that, for any two direction vectors d^1, $d^2 \in D$ that originate from different components in G_0, Ad^1 and Ad^2 do not have any nonzero entries in common positions. Since D^M satisfies (P_3) for each component M, it thus follows that D also satisfies property (P_3), and our claim is proved.

We now construct an independent direction set of size n. For this, four cases are considered.

Case 1. $r = 0$. In this case, D is an independent direction set of size n, and thus $bx \geq b_0$ is facet-inducing for Q_{2E}^n by Corollary 4.2.

Case 2. $r \geq 1$ and $q \geq 1$. Let v_i be any node in C_6. Without loss of generality, let tooth T_1 of comb C be a 2-matching tooth and let $\overset{o}{T_1} = \{w\}$ and $\underset{o}{T_1} = \{v_j\}$. Construct the previously described Hamiltonian cycle H_2 for C, which by Lemma 4.3 satisfies $bx^{H_2} = b_0$. Without loss of generality, let v_i be the last node in C_6 that H_2 visits and let u be the first node in T_k that H_2 visits (see Fig. 4.4). Let $d^{v_i} = x^{wv_j} + x^{uv_i} - x^{uw}$. We claim that

$$D' = D \cup \{d^{v_i} : v_i \in C_6\}$$

is an independent direction set of size n. Clearly, $(H_2 \backslash \{uw\}) \cup \{wv_j, uv_i\}$ forms a 2-edge connected subgraph of K_n; thus $x^{H_2} + d^{v_i} \in Q_{2E}^n$, and d^{v_i} satisfies property (P_1) of Corollary 4.2. Also, $bd^{v_i} = b_{wv_j} + b_{uv_i} - b_{uw} = 0 + 2 - 2 = 0$, and thus d^{v_i} also satisfies property (P_2). Now consider $\{Ad : d \in D'\}$. We know that the vectors in $\{Ad : d \in D\}$ are linearly independent. Furthermore, for each $v_i \in C_6$, $Ad^{v_i} = A_{v_i v_j}$, and this is the only vector in $\{Ad : d \in D'\}$ with a 1 in its v_ith component (all others have 0 in this component). Thus the vectors in $\{Ad : d \in D'\}$ are linearly independent, and D' satisfies property (P_3). Since $|D'| = n$, it follows from Corollary 4.2 that $bx \geq b_0$ is facet-inducing for Q_{2E}^n.

Case 3. $r \geq 1$, $q = 0$, and $s \geq 1$. In this case, comb C has $k = p$ non-2-matching. Let v_i be any node in C_6 and let v_j be a node in C_1. Construct the previously described Hamiltonian cycle H_1 for C, which by Lemma 4.3 satisfies $bx^{H_1} = b_0$. Without loss of generality, let v_i be the last node in C_6 that H_1 visits, let w be the first node in $\underset{o}{T_2}$ that H_1 visits, and let u be the last node in $\overset{o}{T_1}$ that H_1 visits (see Fig. 4.5). Let $d^{v_i} = x^{wv_j} + x^{uv_i} - x^{uw}$. Then we claim that

$$D' = D \cup \{d^{v_i} : v_i \in C_6\}$$

is an independent direction set of size n. The proof follows almost exactly as in Case 2. Thus $bx \geq b_0$ is facet-inducing for Q_{2E}^n by Corollary 4.2.

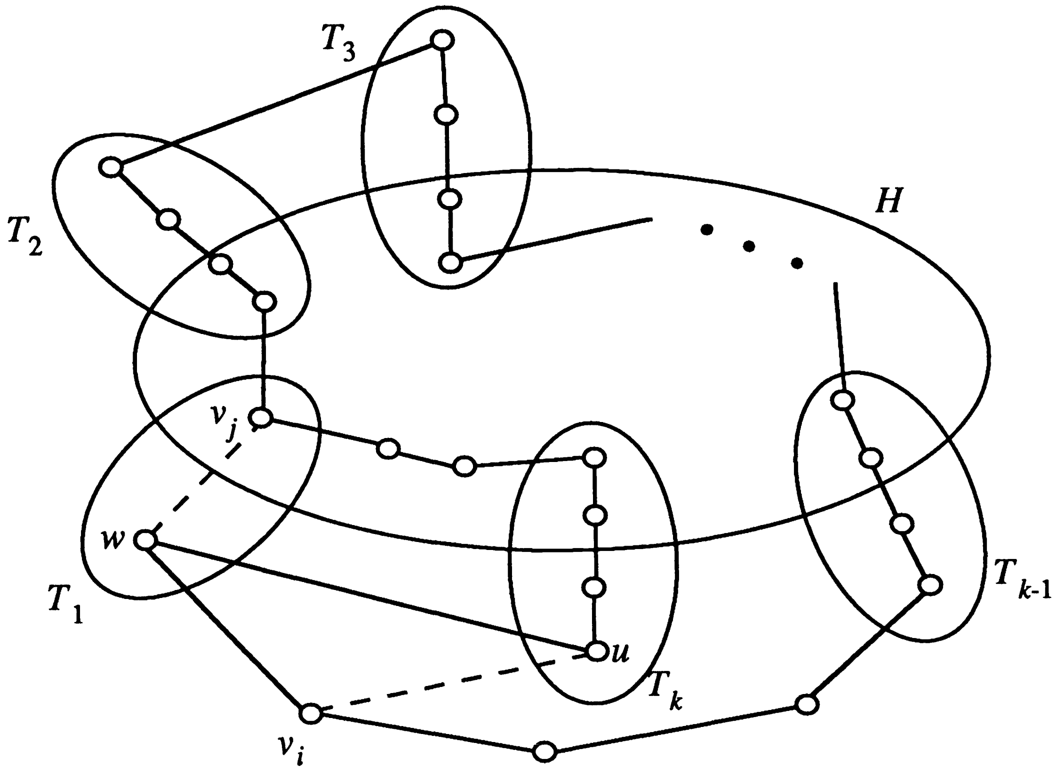


FIG. 4.4. *Hamiltonian cycle H_2 for Case* 2.

Case 4. $r \geq 1$ and $(q+s)=0$. Let v be any node in C_6. We show that $bx \geq b_0$ is not facet-inducing for Q^n_{2E} in this case by showing that $x^H(\delta(v)) = 2$ for all 2-edge connected spanning subgraphs H that satisfy $bx^H = b_0$.

Suppose that $x^H(\delta(v)) \geq 3$. Since $bx^H = b_0$, it then follows from Corollary 3.3 that $x^H(\delta(v)) = 3$, and all other inequalities (i), (ii), ..., (viii) in Lemma 3.1 must be satisfied with equality. Therefore all other nodes in C_6 have degree 2 in H. Furthermore, each tooth T_i, $1 \leq i \leq k$ satisfies $x^H(\delta(T_i)) = 2$, which implies that the number of odd-degree nodes in H in each tooth is even. Overall, we have an odd number of odd-degree nodes in H, which leads to a contradiction. Hence $bx \geq b_0$ is not facet-inducing for Q^n_{2E} when $r \geq 1$ and $q = s = 0$. □

5. Equivalence. We begin this section by investigating the equivalence of the complemented comb constraints to the other known classes of facet-inducing inequalities for Q^n_{2E} discussed in §2. We proceed to give necessary and sufficient conditions under which two complemented comb inequalities define the same facet of Q^n_{2E}.

In Theorems 5.3 and 5.4, below, we show that a complemented comb constraint defines a "new" facet of Q^n_{2E} whenever it is not a complemented 2-matching constraint.[1] First, we require the following.

THEOREM 5.1. *Let F be a nonempty face of a polyhedron P. Suppose that $ax \leq a_0$*

[1] We recently discovered that the complemented comb inequalities are the same as a class of comb inequalities described by Stoer [27] for a more general form of Q^n_{2E}. In [27] Stoer proves validity of these inequalities and remarks that several small ones define facets, but does not show that the class is facet-inducing.

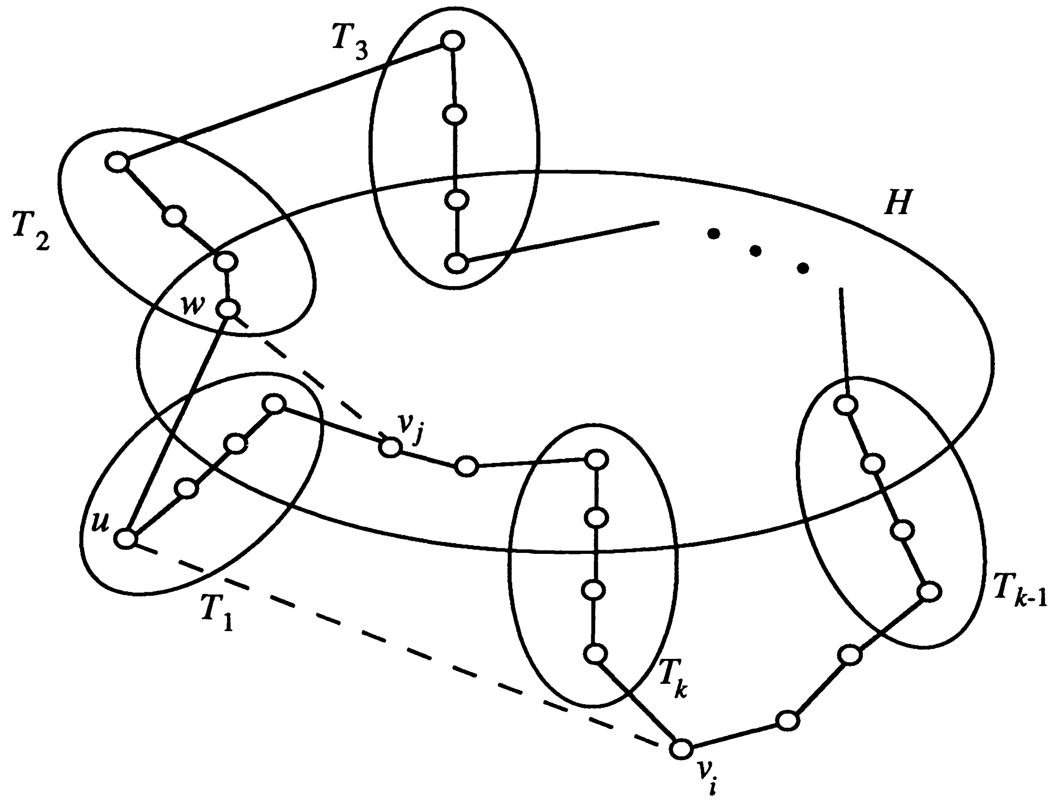


FIG. 4.5. *Hamilton cycle H_1 for Case 3.*

and $\hat{a}x \le \hat{a}_0$ are both valid for F and P. If $ax \le a_0$ and $\hat{a}x \le \hat{a}_0$ are not equivalent with respect to F, then they are not equivalent with respect to P.

Proof. We prove the converse. Suppose that they are equivalent with respect to P and $F = \{x \in P | Ax = d\}$. Then $\{x \in P | ax = a_0\} = \{x \in P | \hat{a}x = \hat{a}_0\}$. So

$$\begin{aligned}\{x \in F | ax = a_0\} &= \{x \in P | ax = a_0\} \cap \{x | Ax = d\} \\ &= \{x \in P | \hat{a}x = \hat{a}_0\} \cap \{x | Ax = d\} = \{x \in F | \hat{a}x = \hat{a}_0\}.\end{aligned}$$

So they are equivalent with respect to F. □

COROLLARY 5.2. *Two inequalities, both valid for Q_T^n and Q_{2E}^n, are not equivalent with respect to Q_{2E}^n if they are not equivalent with respect to Q_T^n.*

THEOREM 5.3. *The complemented comb inequality $bx \ge b_0$ is not equivalent to*

(i) $x_e \le 1$ *for* $e \in E$ *or*

(ii) $x_e \ge 0$ *for* $e \in E$ *or*

(iii) $x(\delta(S)) \ge 2$ *for* $1 \le |S| \le |V| - 1$

with respect to Q_{2E}^n.

Proof. This follows directly from Theorems 2.6 and 2.1 and Corollary 5.2. □

THEOREM 5.4. *A complemented comb inequality is equivalent to a lifted 2-cover inequality* (2.3) *if and only if it corresponds to a complemented 2-matching inequality.*

Proof. In the case where $|H_i| = 1$ for $i = 1, 2, \ldots, p$ and the edges in T are disjoint, a lifted 2-cover inequality (2.3) is identical to the complemented 2-matching constraint with handle $V(K_n) \backslash H$ and teeth corresponding to the end nodes for each edge in T.

Furthermore, since the lifted 2-cover inequalities have left-hand side coefficients 0, 1, they cannot be equivalent to any other complemented comb constraint, since a complemented comb constraint has left-hand side coefficients 0,1,2, and 3 whenever the corresponding comb has at least one non-2-matching tooth. □

We now address the question of equivalence amongst complemented comb constraints.

THEOREM 5.5. *Let $bx \geq b_0$ and $\hat{b}x \geq \hat{b}_0$ be two facet-inducing complemented comb inequalities for Q^n_{2E} corresponding to combs C and $\widehat{C}$, respectively. Let H and $\widehat{H}$ be the handles and let $T(C)$ and $T(\widehat{C})$ be the set of teeth for C and $\widehat{C}$, respectively. Then $bx \geq b_0$ and $\hat{b}x \geq \hat{b}_0$ induce the same facet of Q^n_{2E} if and only if $\widehat{H} = V(K_n)\backslash H$, $T(\widehat{C}) = T(C)$, and $s = q = r = 0$, where s, q, and r are the number of nodes in C_1, C_2, and C_6 in comb C, respectively.*

Proof. Let $K_n = (V, E)$. Suppose that we have $\widehat{H} = V\backslash H$, $T(\widehat{C}) = T(C)$, and $s = q = r = 0$. Then it is straightforward to verify that $b_0 = \hat{b}_0$ and $b_e = \hat{b}_e$ for all $e \in E$, and thus $bx \geq b_0$ and $\hat{b}x \geq \hat{b}_0$ induce the same facet of Q^n_{2E}.

Now suppose that $bx \geq b_0$ and $\hat{b}x \geq \hat{b}_0$ induce the same facet of Q^n_{2E}. Since Q^n_{2E} is full-dimensional, this implies that there exists a $\gamma > 0$ such that $\hat{b} = \gamma b$ and $\hat{b}_0 = \gamma b_0$. From Theorem 2.6 and Corollary 5.2, it follows that $\widehat{H} = V\backslash H$ and $T(\widehat{C}) = T(C)$, which implies that $\gamma = 1$, i.e., $b = \hat{b}$. We now show that, if $s \geq 1$ or $r \geq 1$ or $q \geq 1$, then there is some edge $e \in E$ for which $b_e \neq \hat{b}_e$, showing that we must have $s = r = q = 0$.

Suppose that $p = 0$, where p is the number of non-2-matching teeth in C. Then $q \geq 3$, and, for distinct nodes $u, v \in C_2$, we have $b_{uv} = 0 \neq 2 = \hat{b}_{uv}$. Thus we must have $p \geq 1$. Let $u \in C_5$ in comb C. If $q \geq 1$, then for $w \in C_2$ we have $b_{uw} = 1 \neq 2 = \hat{b}_{uw}$. If $r \geq 1$, then for $w \in C_6$ we have $b_{uw} = 3 \neq 2 = \hat{b}_{uw}$. If $s \geq 1$, then for $w \in C_1$ we have $b_{uw} = 1 \neq 2 = \hat{b}_{uw}$. Thus $s = r = q = 0$, as required. □

Note that Theorem 5.5 shows that two complemented comb inequalities may induce different facets for Q^n_{2E} even when they induce the same facet of Q^n_T.

6. Complemented comb constraints and 2-node connected spanning subgraphs.

We conclude with some remarks on the problem of finding a minimum cost 2-node connected spanning subgraph in a given complete weighted graph $K_n = (V, E)$. The polytope associated with this problem is

$$Q^n_{2N} := \text{conv}\{x^F \in \mathbb{R}^E : (V, F) \text{ is a 2-node connected spanning subgraph of } K_n\}.$$

It is easy to see that a 2-node connected graph is also 2-edge connected, and thus $Q^n_{2N} \subseteq Q^n_{2E}$. Therefore any valid inequality for Q^n_{2E} is also valid for Q^n_{2N}. In particular, the complemented comb inequalities are valid for Q^n_{2N}.

An interesting question is to find conditions under which the complemented comb inequalities are facet-inducing for Q^n_{2N}. In the case where the inequalities originate from a 2-matching comb, we obtain the complemented 2-matching constraints that are shown to be facet-inducing for Q^n_{2N} in [4]. However, we show that not all of the complemented comb constraints are facet-inducing for Q^n_{2N}.

A comb C with handle H and teeth $T_1, T_2, \ldots, T_k$ is called *simple* if $|T_i \cap H| = 1$ for $i = 1, 2, \ldots, k$. In [4] a class of inequalities called the *complemented simple comb constraints* is introduced and shown to be facet-inducing for Q^n_{2N}. These inequalities have the form

$$ax \geq |\overset{o}{V}| + \frac{k+1}{2},$$

where $\overset{o}{V} \subset V$ is the set of nodes not contained in H or any T_i, $i = 1, 2, \ldots, k$ and a is defined by

$$a_e = \begin{cases} 0 & \text{for } e \in E(H) \text{ or } e \in E(T),\ i = 1, 2, \ldots, k, \\ 1 & \text{otherwise.} \end{cases}$$

Given a simple comb C, if we sum two times the associated complemented simple comb constraint plus the constraints

$$x(\delta(v)) \geq 2 \quad \text{for all } v = T_i \cap H \quad \text{for some non-2-matching tooth } T_i \text{ of } C,$$

we obtain the associated complemented comb inequality. Thus the complemented comb constraints originating from simple combs with at least one non-2-matching tooth are not facet-inducing for Q^n_{2N}.

REFERENCES

[1] N. Ascheuer, L. Escudero, M. Grötschel, and M. Stoer, *On* LP *bounds for the sequential order problem*, in preparation.

[2] A. Bachem and M. Grötschel, *New aspects of polyhedral theory*, in Modern Applied Mathematics—Optimization and Operations Research, B. Korte, ed., North–Holland, Amsterdam, 1982, pp. 51–106.

[3] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, Macmillan, London, 1976.

[4] S. C. Boyd and W. R. Pulleyblank, *Facet generating techniques*, preprint, 1991.

[5] V. Chvátal, *Edmonds polytopes and weakly Hamiltonian graphs*, Math. Programming, 5 (1973), pp. 29–40.

[6] H. Crowder and M. Padberg, *Solving large-scale Symmetric Traveling Salesman Problems to Optimality*, Management Sci., 26 (1980), pp. 495–509.

[7] G. Dantzig, D. Fulkerson, and S. Johnson, *Solution of a large-scale Traveling Salesman Problem*, Oper. Res., 2 (1954), pp. 393–410.

[8] J. Edmonds, *Paths, trees, and flowers*, Canadian J. Math., 17 (1965), pp. 449–467.

[9] R. E. Erikson, C. L. Monma, and A. F. Veinott, Jr., *Send-and-split method for minimum-concave-cost network flows*, Math. Oper. Res., 12 (1987), pp. 634–664.

[10] M. Grötschel, *On the Symmetric Traveling Salesman Problem*: *Solution of a* 120-*city problem*, Math. Programming Study, 12 (1980), pp. 61–77.

[11] M. Grötschel, J. Junger, and G. Reinelt, *A cutting plane algorithm for the linear ordering problem*, Oper. Res., 34 (1984), pp. 1195–1220.

[12] M. Grötschel and C. L. Monma, *Integer polyhedra arising from certain network design problems with connectivity constraints*, SIAM J. Discrete Math., 3 (1990), pp. 502–523.

[13] M. Grötschel, C. L. Monma, and M. Stoer, *Facets for Polyhedra Arising in the Design of Communication Networks with Low-Connectivity Constraints*, Schwerpunktprogramm der Deutschen Forschungsgemeinschaft, Anwendungsbezogene Optimierung und Steuerung, Report No. 187, 1989.

[14] ———, *Computational Results with a Cutting Plane Algorithm for Designing Communication Networks with Low-Connectivity Constraints*, Schwerpunktprogramm der Deutschen Forschungsgemeinschaft, Anwendungsbezogene Optimierung und Steuerung, Report No. 188, 1989.

[15] M. Grötschel and M. Padberg, *On the Symmetric Traveling Salesman Problem* I: *Inequalities*, Math. Programming, 16 (1979), pp. 265–280.

[16] ———, *On the Symmetric Traveling Salesman Problem* II: *Lifting theorems and facets*, Math. Programming, 16 (1979), pp. 281–302.

[17] ———, *Polyhedral theory*, in The Traveling Salesman Problem, E.L. Lawler et al., eds., John Wiley, New York, 1985.

[18] R. Karp and C. Papadimitriou, *On linear characterizations of combinatorial optimization problems*, SIAM J. Comput., 11 (1982), pp. 620–632.

[19] A. R. MAHJOUB, *Two Edge Connected Spanning Subgraphs and Polyhedra*, Report No. 8850-OR, Institute für Operations Research, Universität Bonn, Germany, 1988.

[20] C. L. MONMA AND D. F. SHALLCROSS, *Methods for Designing Survivable Communication Networks*, Technical Memorandum, Bellcore, Morristown, NJ, 1986.

[21] G. L. NEMHAUSER AND L. A. WOLSEY, *Integer and Combinatorial Optimization*, Wiley-Interscience, New York, 1988.

[22] M. PADBERG AND M. GRÖTSCHEL, *Polyhedral computations*, in The Traveling Salesman Problem, E.L. Lawler et al., eds., John Wiley, New York, 1985.

[23] M. PADBERG AND S. HONG, *On the Symmetric Traveling Salesman Problem: A computational study*, Math. Programming Study, 12 (1980), pp. 78–107.

[24] M. PADBERG AND G. RINALDI, *A branch-and-cut algorithm for the resolution of large-scale Symmetric Traveling Salesman Problems*, Report R. 247, IASI-CNR, Rome, 1988.

[25] M. PADBERG AND G. RINALDI, *Optimization of a 532-city Symmetric Traveling Salesman Problem by branch and cut*, Oper. Res. Let., 6 (1987), pp. 1–8.

[26] W. R. PULLEYBLANK, *Polyhedral combinatorics*, in Handbooks in OR and MS, Vol. 1, G. L. Nemhauser et al., eds., Elsevier Science Publishers B.V., North–Holland, Amsterdam, 1989.

[27] M. STOER, *Design of Survivable Communication Networks*, Ph.D. dissertation, Augsburg University, Germany, 1991.

SIAM J. Discrete Math.
Vol. 6, No. 4, pp. 631-641, November 1993

EDGE-CHROMATIC SCHEDULING WITH SIMULTANEITY CONSTRAINTS*

D. DE WERRA†, N. V. R. MAHADEV‡, AND U. N. PELED§

Abstract. An edge-coloring model for some types of scheduling problems is described; the case is handled where some collections of (nonadjacent) edges are required to have the same color. This corresponds to simultaneity constraints. The complexity of this problem is studied. Next, some classes of graphs for which such colorings exist are characterized, and a recognition algorithm is derived.

Key words. chromatic scheduling, automated production system, edge-coloring

AMS subject classifications. 90B35, 05C15, 05C75, 68R10

1. Introduction. In some types of scheduling problems, there are only a few types of constraints that must be considered in the construction of a schedule. Such situations occur in various contexts ranging from simple class-teacher timetabling problems to special cases of machine scheduling problems. We describe an edge-coloring model where some additional requirements may be included. This allows us to define a class of colorings with constraints, and the family of graphs for which such colorings exist is characterized. Furthermore, a recognition algorithm is derived.

Let us first formulate the basic open shop problem for which some additional conditions are later introduced.

We are given m processors $P_1, P_2, \ldots, P_m$ and a collection of jobs $J_1, J_2, \ldots, J_n$ to be processed within a period of k consecutive time units. Each job J_j consists of tasks $T_{1j}, \ldots, T_{mj}$; task T_{ij} of job J_j has to be processed on processor P_i; its processing time p_{ij} is given. We assume that the p_{ij}'s are integers. We consider that, if $p_{ij} = 0$, then the task T_{ij} does not exist. No processor can handle two tasks simultaneously, and no two tasks of the same job can be processed at the same time. The tasks of the same job can be processed in any order. Furthermore, we assume that preemptions are allowed (after any integer number of time units) during the processing of a task on a processor. We may ask whether it is possible to schedule the jobs within k time units and satisfy the above requirements.

This situation is similar to the simple class-teacher timetabling problem; assume that each P_i is a teacher and that each J_j is a class, i.e., a group of students taking exactly the same program. Then T_{ij} is the set of p_{ij} lectures (of one time unit each) that teacher P_i must give to class J_j.

A feasible schedule exists if and only if k is not smaller than both the maximum number of lectures that a single teacher must give and the maximum number of lectures that a single class must take. This can be seen from the following graph-theoretical formulation. We associate with the problem a bipartite multigraph $G = (\mathcal{P}, \mathcal{J}, E)$ constructed as follows: Each P_i corresponds to a node in the left set $\mathcal{P}$ of nodes, and each J_j corresponds to a node in the right set $\mathcal{J}$ of nodes. Furthermore, node P_i is linked to node J_j by p_{ij} parallel edges.

*Received by the editors, September 1, 1991; accepted for publication (in revised form) September 24, 1992. This paper was written while the first and the third authors were visiting the Department of Mathematics at Northeastern University, Boston, Massachusetts.

†Département de Mathématiques, Ecole Polytechnique Fédérale de Lausanne, Ecublens – CP 121, CH-1015, Lausanne, Switzerland, (dewerra@elma.epfl.ch).

‡Department of Mathematics, Northeastern University, 360 Huntington Avenue, Boston, Massachusetts 02115, (nmahadev@lynx.northeastern.edu).

§Department of Mathematics, Statistics, and Computer Science (M/C 249), University of Illinois at Chicago, 851 South Morgan Street, Chicago, Illinois 60607-7045, (peled@math.uic.edu).

A *classical edge k-coloring* of G is an assignment F of one color $F(e) \in \{1, \dots, k\}$ to each edge e of G such that $F(e) \neq F(g)$ whenever edges e and g are adjacent (i.e., have at least one node in common).

There is a correspondence between classical edge k-colorings of G and feasible schedules in k time units for our timetabling problem as well as for our open shop scheduling problem.

From the theorem of König, such an edge k-coloring exists if and only if $k \geq \Delta(G)$, where $\Delta(G)$ is the maximum degree in G (i.e., the maximum number of edges incident to the same node). See [1].

In the next section, we introduce some additional requirements that occur in some timetabling and in some production scheduling problems. The complexity of the general case is studied. Section 3 is devoted to the above coloring model with some special simultaneity requirements. This leads to a characterization of the family of graphs for which such colorings exist. A recognition algorithm is given. Some variations of the coloring model are formulated in §4.

Refer to [1] for graph-theoretical terms not defined here.

2. Simultaneity requirements. In the timetabling model described above, we may only consider simple requirements related to the classes (no class can take two lectures simultaneously) and to the teachers (no teacher can give two lectures simultaneously).

It often happens that it is required to have two lectures, represented by edges $e = P_iJ_j$, $g = P_rJ_s$ (with $P_i \neq P_r, J_j \neq J_s$), scheduled at the same hour. An example is gymnastics, where two classes are grouped together; the girls are then assigned to teacher P_i, and the boys to teacher P_r for one hour.

The data for this problem consist of a bipartite multigraph $G = (\mathcal{P}, \mathcal{J}, E)$ and a set R of pairs $\{e, g\}$ of nonadjacent edges. An integer $k \geq \Delta(G)$ is given. Does there exist a classical edge k-coloring F of G such that $F(e) = F(g)$ for each pair $\{e, g\} \in R$? Let us call this problem OSSR (open shop with simultaneity requirements). Such constraints requiring that some fraction of tasks be processed simultaneously may also appear in preemptive open shop scheduling problems for technological reasons. For example, we may need a specialist, able to supervise several tasks processed at the same time.

PROPOSITION 2.1. OSSR *is* NP-*complete.*

Proof. We sketch a reduction from NODE k-COLORING [4]. Observe first that a classical edge k-coloring in a multigraph G corresponds to a node k-coloring in the line-graph $L(G)$ of G (each edge of G becomes a node in $L(G)$, and we link two nodes of $L(G)$ if the corresponding edges of G are adjacent). If in $L(G)$ two nodes e, g correspond to edges of G that must have the same color (because they form a pair $\{e, g\} \in R$), then we identify the nodes e and g in $L(G)$. Any node k-coloring of the resulting graph corresponds to a classical edge k-coloring of G that satisfies the requirements in R, and the opposite also holds.

Let us show that any instance (G, k) of NODE k-COLORING can be transformed into an instance of OSSR, i.e., of the classical edge k-coloring problem with requirements in R.

If G is already the line-graph $L(H)$ of some bipartite multigraph H, then we set $R = \emptyset$, and we are done. Otherwise, we apply repeatedly the *node-separation procedure* NSP(x) to some node x as follows: Node x with neighborhood set $N(x)$ ($|N(x)| \geq 2$) is replaced by two nonadjacent nodes x', x'' with $N(x'), N(x'') \neq \emptyset$, $N(x') \cap N(x'') = \emptyset$, $N(x') \cup N(x'') = N(x)$. Then we introduce into R the pair $\{x', x''\}$. Such a procedure NSP(x) is repeated until the resulting graph is the line-graph of a bipartite multigraph H. This happens in at most $2m - n$ applications of the node-separation procedure, where m

is the number of edges and where n is the number of nonisolated nodes of G (because, after $2m - n$ applications, G reduces to a matching and isolated node). The resulting bipartite multigraph H has an edge k-coloring with requirements in R if and only if the original G has a node k-coloring. □

Remark 2.2. The above proof shows that OSSR remains NP-complete even for bipartite graphs of the form nP_3, where P_3 is a chain on three nodes.

Remark 2.3. Consider the companion problem where, for each pair of edges $\{e, g\} \in R$, we require that e, g have different colors. Such requirements may occur in open shop scheduling problems where two tasks need the presence of an expert, who cannot supervise both at the same time. It may also occur in the timetabling context because two lectures involving different classes may require the same equipment or the same classroom. When the bipartite multigraph is a matching, this problem is simply a reformulation of the node-coloring problem for an arbitrary graph.

3. A special case of simultaneity requirements. Our purpose is to characterize a class of graphs that have edge k-colorings satisfying the requirements in R for a special type of R. Although these requirements are extremely special, they are interesting because they allow us to give a complete characterization of a nontrivial class of graphs that satisfy them. Furthermore, this provides a tool that could prove useful in extending these results to more general situations.

A chain D of positive length in a graph is *attached* to a subgraph H if both endpoints x, y of D, but no intermediate nodes or edges, are in H.

If a chain D is attached to a cycle C, then D and C form three chains between x and y that are node-disjoint except for the endpoints; such a configuration is called a *mouth*; it is *even* (respectively, *odd*) if the three chains have even (respectively, odd) length.

A *handle* in a graph G is a chain D of positive length attached to a cycle C at its endpoints x and y, such that every intermediate node of D as well as of one of the two chains of C between x and y has degree 2 in G. A handle is odd (respectively, even) if its length is odd (respectively, even).

An edge k-coloring F of a bipartite multigraph G satisfies condition $R(C, D)$ based on a cycle $C = \{e_1, e_2, \dots, e_{2p}\}$ $(p \geq 0)$ and on a chain $D = \{g_1, g_2, \dots, g_q\}$ $(q \geq 0)$ if the following hold:

$$\begin{aligned} F(e_1) &= F(e_3) = \cdots = F(e_{2p-1}), \\ F(e_2) &= F(e_4) = \cdots = F(e_{2p}), \\ F(g_1) &= F(g_3) = \cdots, \\ F(g_2) &= F(g_4) = \cdots. \end{aligned}$$

Note that, when $p = 0$ (respectively, $q = 0$), the cycle C (respectively, the chain D) reduces to a single node and the condition on C (respectively, on D) is void.

A graph G is a *bipartite odd cactus* (or BOC *graph*) if it is bipartite and if it does not contain an even mouth as a partial subgraph.

A *block* in a graph is a maximal 2-connected partial subgraph. We say that a graph G has the *extension property* (EP) if, for any choice of cycles $C_1, \dots, C_r$ in distinct blocks of G and of chains $D_1, \dots, D_r$ attached to $C_1, \dots, C_r$, respectively, there exists an edge $\Delta(G)$-coloring of G that satisfies the conditions $R(C_1, D_1), \dots, R(C_r, D_r)$. Such an edge coloring is called an EP *coloring of G*.

We can now state the main theorem.

THEOREM 3.1. *For a bipartite multigraph G, the following statements are equivalent*:

(1) *G is a* BOC *graph*;

(2) *Every nontrivial block of G can be reduced to a cycle by repeatedly removing the intermediate nodes and all the edges of an odd handle of the remaining graph*;

(3) *All partial subgraphs of G have* EP.

The following two results are used in proving Theorem 3.1. The first was essentially proved by Whitney.

THEOREM 3.2 (see [8]). *A multigraph G is 2-connected if and only if it can be obtained by the following process: Starting from any 2-connected partial subgraph of G, repeatedly attach chains to the subgraph already constructed.*

A *subdivision* of a graph G is a graph obtained by replacing some edges of G with chordless chains having the same endpoints. We denote the complete graph on four nodes by K_4.

LEMMA 3.3. *A* BOC *graph cannot contain a subdivision of K_4 as a partial subgraph.*

Proof. Assume, if possible, that the BOC graph contains a subdivision of a K_4 with nodes a, b, c, d. The subdivision of one of the edges ab, bc, ca, say ab, must be of even length since the graph is bipartite. Then the subdivision of K_4 contains an even mouth between the nodes a and b, a contradiction. □

Proof of Theorem 3.1. (2) ⇒ (1): If G contains an even mouth, then the process of repeatedly removing odd handles cannot destroy the even mouth. Thus the block containing it cannot be reduced to a cycle.

(3) ⇒ (1): Assume that G has an even mouth M formed by three node-disjoint even chains C_1, C_2, D between two nodes. Take $C = C_1 \cup C_2$; M has no edge 3-coloring satisfying $R(C, D)$, so the partial subgraph M of G does not have EP.

(1) ⇒(2): Assume without loss of generality that G is a 2-connected BOC graph. It is enough to show that G is a cycle or G has a handle, as any handle in a BOC graph must be odd, and, after its removal, the graph remains a 2-connected BOC graph. By Theorem 3.2, the edge-set of G can be partitioned into $C, P_1, \ldots, P_r$, where C is a cycle and where each P_i is a chain attached to the partial subgraph G_i consisting of $C, P_1, \ldots, P_{i-1}$. We show by induction on $r \geq 1$ that one of the P_i is a handle of $G_r = G$. By the induction hypothesis, some P_j is a handle of G_{r-1}. Then, by the definition of a handle, G_{r-1} has chains P and Q between the endpoints x, y of P_j, such that P_j, P, and Q are edge-disjoint and the internal nodes of P have degree 2 in G_{r-1}. Let z, w be the endpoints of P_r. If each of z, w is in $\{x, y\}$ or not on $P_j \cup P$, then P_j remains a handle of G_r. If both of z, w are on P_j or both are on P, then P_r is a handle of G_r. If z is an internal node of P_j and w is an internal node of P, then P, Q, P_j, and P_r form a subdivision of K_4, contradicting Lemma 3.3. The remaining case is that z is not on $P_j \cup P$, whereas w is an internal node of P_j or of P, say of P_j. By the 2-connectivity of G_{r-1}, it has a cycle C' containing z and any edge of P. It follows that C' contains all the edges of P and none of P_j. Therefore C', P_j, and P_r form a subdivision of K_4, again contradicting Lemma 3.3.

(1) ⇒ (3): We may assume that G is connected. Assume first that each block of G has EP. Then we may reconstruct G by introducing blocks $B_1, B_2, \ldots, B_s$ consecutively, so that, for each $i \geq 2$, B_i has exactly one node in common with the graph $\widehat{G}$ formed by $B_1, \ldots, B_{i-1}$. Let x be this node. By induction on s, the graph $\widehat{G}$ has an EP coloring. Now consider any EP coloring of B_i and relabel the colors of the edges in B_i so that all the colors at x in $\widehat{G} \cup B_i$ are distinct. This gives an EP coloring of $\widehat{G} \cup B_i$. Thus G has EP if each block of G has EP. We may therefore assume that G is 2-connected.

Given the constraint $R(C, D)$, by condition (1), $C \cup D$ has an EP coloring. It follows from Theorem 3.2 that G can be constructed from $C \cup D$ by repeatedly attaching chains, say $P_2, \ldots, P_r$. By the proof of (1) ⇒ (2), we may assume that each P_i is an odd handle in the partial subgraph G_i consisting of $C, D, P_2, \ldots, P_{i-1}$. Therefore the EP coloring of G_{i-1} can be extended to an EP coloring of G_i. Thus G has EP. □

Clearly, by definition, a graph G is a BOC graph if and only if every block of G is a BOC graph. The blocks of G can be obtained in linear time [3]. We now sketch an algorithm to test whether a 2-connected graph G is a BOC graph.

If G is a cycle, it is a BOC graph. Otherwise, find any handle H of G (we soon describe how to do this). If G has no handles, it is not a BOC graph by condition (2) of Theorem 3.1. If H is an even handle, then G contains an even mouth and is not a BOC graph. If H is an odd handle, then, by condition (2) of Theorem 3.1, G is a BOC graph if and only if the 2-connected graph obtained by removing the interior nodes and edges of H is a BOC graph, and we proceed recursively.

We now describe how to detect a handle in a 2-connected graph $G = (V, E)$. Let $V = V_1 \cup V_2$, where V_1 consists of the vertices of degree 2, and V_2 of degree 3 or more. Starting from any node $x \in V_2$, travel along unused edges of G, marking them as used, until a second node $y \in V_2$ is reached. This takes $O(|V|)$-time. Then use a standard traversal algorithm such as breadth-first search [3] to detect in $O(|E|)$-time a second chain of unused edges between x and y that uses only nodes of V_1 as intermediate nodes. If such a chain is found, it is a handle of G. If not, repeat the above until all the edges of G are marked as used, in which case G has no handles.

The total time to find a handle or verify that none exists is thus $O(|E|^2)$, and the complete recognition algorithm takes $O(|E|^3)$-time.

Remark 3.4. A related class of graphs has been defined for another type of scheduling problem. We consider that the k colors are arranged in a cyclic way, so that color 1 follows color k. Each edge e must receive a prescribed number w_e of cyclically consecutive colors. Furthermore, at each node x, the cyclic intervals of colors assigned to the edges adjacent to x must be disjoint, and their union must form a cyclic interval. Let k be the maximum sum of the weights of all edges adjacent to a node. There exists such a k-coloring for any choice of weights if and only if the graph is a bipartite outerplanar graph. Such a graph is called a BEC graph (denoting bipartite edge cactus) in [2]. Its 2-connected components are constructed by starting from an (even) cycle and repeatedly introducing odd handles on distinct edges. From this construction, we note that BEC graphs are a special class of BOC graphs.

4. Some classes of graphs with special simultaneity requirements. We examine here special simultaneity requirements and derive a few classes of bipartite multigraphs that satisfy these requirements.

Remark 4.1. Recall the definition of OSSR with input (G, R). If H is the line-graph of G and if H^* is the graph obtained from H by identifying the nodes e, g for each pair $\{e, g\} \in R$, then G has an edge Δ-coloring satisfying the requirements in R if and only if $\chi(H^*) = \Delta(G)$, where χ denotes the node-chromatic number.

Remark 4.2. In general, we cannot proceed directly by identifying the edges e, g of each pair $\{e, g\}$ of R in G (in either of the two ways) and edge-color the resulting graph G^*. As an example, if G has edges $e_1 = ab$, $e_2 = bc$, $e_3 = cd$, $e_4 = ef$, $e_5 = fg$ with $R = \{e_2, e_4\}$, then $\Delta(G^*) > \Delta(G)$, and thus G^* does not have an edge $\Delta(G)$-coloring, even though G does have a $\Delta(G)$-coloring satisfying the requirements in R. Thus we must invoke the line-graph H and the graph H^* obtained from H. However, if H^* is a line-graph, then $L(G^*) = H^*$, and hence we could proceed directly with G^*.

A graph G is said to have the SIMULT *property* if, for each pair $\{e, g\}$ of nonadjacent edges, there exists an edge $\Delta(G)$-coloring F with $F(e) = F(g)$. In other words, whenever R consists of a single pair, the answer to OSSR is yes.

We say that a bipartite multigraph G is a SIMULT* graph if every partial subgraph of G has the SIMULT property.

A chordless chain (respectively, cycle) on n nodes is denoted by P_n (respectively, C_n). Its complement is denoted $\overline{P_n}$ (respectively, $\overline{C_n}$).

THEOREM 4.3. *For a bipartite multigraph G, the following statements are equivalent:*

(1) *G is a* SIMULT* *graph;*

(2) *G contains no P_5 as a partial subgraph;*

(3) *The connected components of G are obtained from the square or the bistar shown in Fig.* 1 *by multiplication or removal of edges and by removal of nodes.*

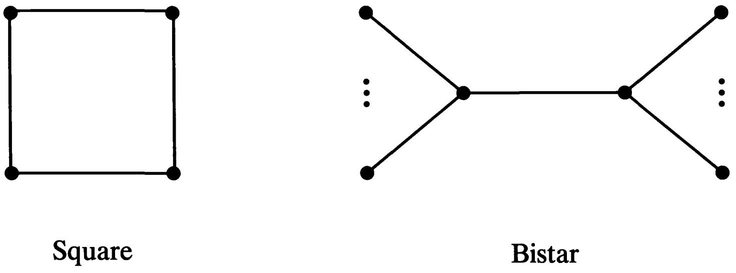


FIG. 1. SIMULT* *graphs.*

Proof. (1) ⇒ (2): P_5 does not have the SIMULT property: Let e and g be the extreme edges of P_5 and take $R = \{e, g\}$; then $\chi(H^*) = 3 > 2 = \Delta(P_5)$, where $H = L(P_5)$. Thus, by Remark 4.1, P_5 does not have the SIMULT property.

(2) ⇒ (3): This is easily verified.

(3) ⇒ (1): Let S be a connected component of G, obtained from a square or a bistar as in condition (3). For any pair $\{e, g\}$ of nonadjacent edges of S, there exists an edge $\Delta(S)$-coloring F with $F(e) = F(g) = 1$: it is obtained by coloring e and g with color 1. The remaining multigraph S' obtained by removing e and g has $\Delta(S') = \Delta(S) - 1$, so, by induction on $\Delta(S)$, it can be colored with colors $2, 3, \ldots, \Delta(S)$. □

Another simple case is the following situation: Let $\omega(G)$ denote the maximum size of a clique in G. A graph G is *perfect* if every induced subgraph G' of G satisfies $\chi(G') = \omega(G')$. Polynomial-time algorithms exist for finding the chromatic number of a perfect graph and, in fact, for finding an optimal coloring [5]. We may thus be interested in the situation, where, starting from a bipartite multigraph G, we construct its line-graph $H = L(G)$, and, by identifying *any* two nonadjacent nodes of H, we obtain a perfect graph H^*. Such a graph G is said to have the SIMULTANEOUS property. If every partial subgraph of G has the SIMULTANEOUS property, G is said to be SIMULTANEOUS-*perfect*.

THEOREM 4.4. *For a bipartite multigraph G, the following statements are equivalent:*

(1) *G is* SIMULTANEOUS-*perfect;*

(2) *G has the* SIMULTANEOUS *property;*

(3) *G contains no P_7 as a partial subgraph;*

(4) *The connected components of G are obtained from the graphs of the types $G_1, \ldots, G_5$ shown in Fig.* 2 *by multiplication or removal of edges and by removal of nodes.*

We use the following two results in proving Theorem 4.4. Two nodes x, y of G are called *twins* if they have the same neighborhood in $V - \{x, y\}$. Lovász proved [7] that

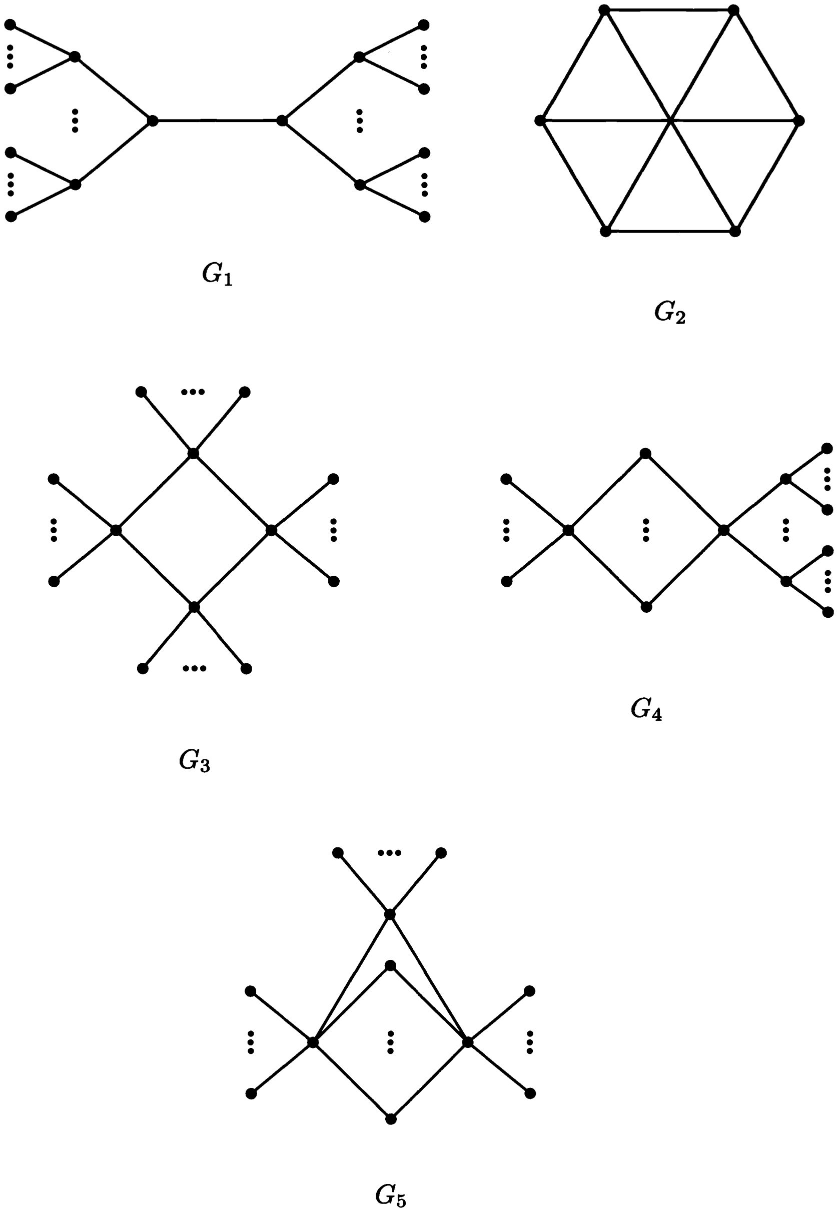


FIG. 2. SIMULTANEOUS-*perfect graphs.*

minimal imperfect graphs contain no twins. More specifically, we need the following corollary of his result.

LEMMA 4.5 (see [7]). *Let a graph G contain two adjacent twins x, y. Then G is perfect if and only if $G - \{x\}$ is perfect.*

We also need the following theorem of Hayward.

THEOREM 4.6 (see [6]). *Any graph containing neither induced cycles of length at least 5 nor their complements is perfect.*

Proof of Theorem 4.4. (1) $\Rightarrow$ (2): The proof follows directly from the definition.

(2) ⇒ (3): If G contains a P_7 as a partial subgraph, then $H = L(G)$ contains an induced P_6, and the identification of the endpoints of this P_6 gives an induced C_5 in H^*. Hence H^* is not perfect.

(3) ⇒ (4): We must verify that a connected bipartite graph without parallel edges and not containing any P_7 as a partial subgraph is contained in a graph of one of the types G_i of Fig. 2.

Case 1. G is a tree.

Since the longest chain of G has length at most 5, G must be a partial graph of a graph of type G_1.

Case 2. G has some cycles.

Then these cycles have length 4 or length 6.

(a) If G has a cycle of length 6, then there are no nodes outside the cycle, so G is a partial subgraph of a graph of type G_2.

(b) If all cycles of G have length 4, we have two possibilities. If there is exactly one C_4, then G is a partial subgraph of a graph of type G_3 or G_4. Otherwise, G has several C_4's. No two of these C_4's can share exactly zero nodes or exactly one node or exactly one edge (otherwise, G would contain a P_7 or C_6). It follows that all the C_4's have the same two nonadjacent nodes x, y in common. Hence G contains three or more chains of length 2 with common endpoints x, y. Let $z_1, \dots, z_q$ be the intermediate nodes of these chains ($q \geq 3$). No z_i can have a pendant chain of length 2 or more; otherwise, G would contain a P_7. Similarly, neither x nor y can have a pendant chain of length 3 or more. If x has a pendant chain of length 2, then no z_i has a pendant edge; y may have pendant edges, but no pendant chain of length 2. Hence G is a partial subgraph of a graph of type G_4. Finally, if neither x nor y has a pendant chain of length 2 or more, then at most one z_i can have pendant edges, so G is a partial subgraph of a graph of type G_5.

(4) ⇒ (1): If G' is any partial subgraph of G, then its associated graph $(H')^*$ is an induced subgraph of H^*. Therefore it is enough to show that H^* itself is perfect. Moreover, for the same reason, we may assume that every occurrence of "$\cdots$" in Fig. 2 represents two or more nodes and that every edge of the figure has been duplicated to two or more parallel edges of G.

We observe first that H contains no induced P_6, $\overline{P_6}$, or C_5: It contains no P_6 or C_5 since G contains no P_7 or C_5 as a partial subgraph, as verified easily by inspecting Fig. 2, and it contains no $\overline{P_6}$ since $\overline{P_6}$ is not a line-graph. It follows that H contains neither an induced C_k nor an induced $\overline{C_k}$ for any $k \geq 7$ or $k = 5$.

Next, we observe that H^* contains neither an induced C_k nor an induced $\overline{C_k}$ for any $k \geq 7$ or $k = 5$. Indeed, if H^* contains an induced C_k, then H contains an induced C_k or P_{k+1}, which is not the case for $k \geq 7$ or $k = 5$; if H^* contains an induced $\overline{C_k}$, then H contains an induced $\overline{P_{k-1}}$, which is not the case for $k \geq 7$.

If, in addition, H^* contains neither an induced C_6 nor an induced $\overline{C_6}$, then it is perfect by Theorem 4.6, and we are done. We may therefore assume the opposite.

Case 1. H^* contains an induced C_6.

In this case, H must have a C_6 (since H contains no P_7). It follows that, if G is connected (as we may assume), it is a multigraph of type G_2 of Fig. 2, which is the complete bipartite graph $K_{3,3}$.

In the line graph $H = L(G)$, each node has some adjacent twins by the assumption made at the beginning of the proof that (4) ⇒ (1). To obtain H^*, we identify a pair of nonadjacent nodes of H. By virtue of Lemma 4.5, we may remove all but one node from each maximal set of pairwise adjacent twins of H^* without affecting its perfectness. The resulting graph $H^* = (V, E)$ is shown in Fig. 3 (it does not depend on the choice of nonadjacent nodes of H to be identified; $H^* - g \simeq L(K_{3,3})$, and g results from identifying a twin of x_1 with a twin of x_2). We conclude Case 1 by showing that this particular graph is perfect.

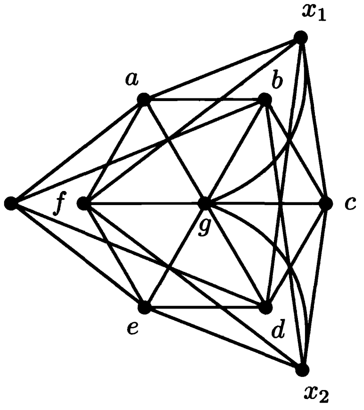


FIG. 3. *A graph used in the proof of Theorem* 4.4.

Note that H^* is 4-colorable—color $a, \ldots, f$ with two colors; x_1, x_2, x_3 with the third color; and g with the fourth. This also shows that the subgraph induced by $V - \{g\}$ is 3-colorable. It follows that, if I is any induced subgraph of H^*, then $\chi(I) = \omega(I)$. Indeed, in the case where $\omega(I) = 4$, I is 4-colorable because H^* is; in the case where $\omega(I) = 3$, if I contains g, then $I - \{g, x_3\}$ is bipartite, and hence I is 3-colorable, and, if I does not contain g, then I is 3-colorable as noted above; in the case where $\omega(I) \leq 2$, I is bipartite since H^* contains no induced odd cycles of length 5 or more. Thus H^* is perfect.

Case 2. H^* contains an induced $\overline{C_6}$.

In this case, since the line-graph H contains no induced claw $K_{1,3}$, it follows that H contains an induced $H_1 = \overline{C_6}$ or H_2, as shown in Fig. 4.

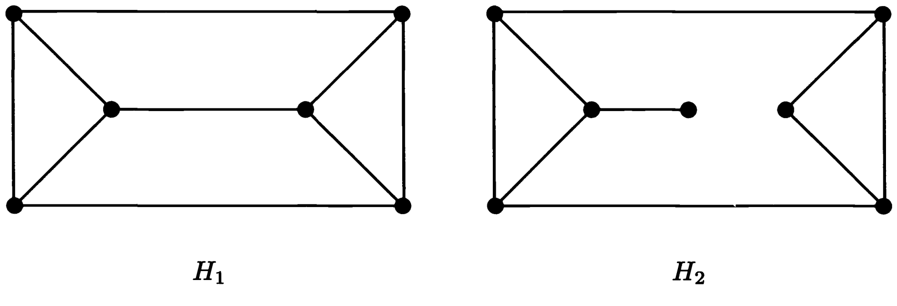
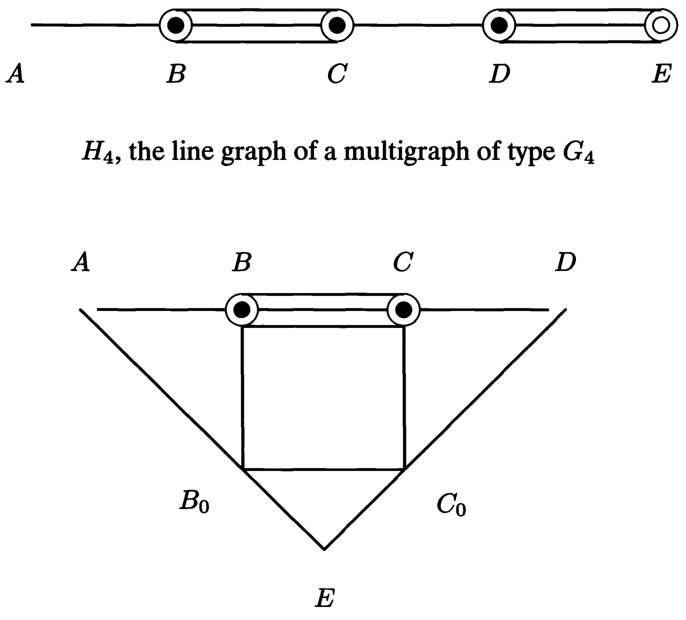


FIG. 4. *Two graphs used in the proof of Theorem* 4.4.

If H contains H_1, then G must be of type G_4 or G_5, and, if H contains H_2, then G must be of type G_4. Thus it only remains to show that when G is a multigraph of type G_4 or G_5, then H^* is perfect.



H_4, the line graph of a multigraph of type G_4

H_5, the line graph of a multigraph of type G_5

represents a clique of size 2 or more.

◉ represents the node-set of the complete join of two or more cliques referred to as component cliques.

◎ represents the node-set of two or more disjoint cliques of size 2 or more referred to as component cliques.

——— represents a complete connection between the two end-sets.

≡ represents a bijection between the component cliques of the left end-set and the component cliques of the right end-set, where matched component cliques are completely joined.

FIG. 5. *Schematic diagram of two line-graphs of bipartite multigraphs.*

Consider H_4 and H_5, the line-graphs of multigraphs of type G_4 and G_5, respectively, illustrated in Fig. 5. Let $(H_4)^*$ (respectively, $(H_5)^*$) be obtained by identifying two nonadjacent nodes of H_4 (respectively, H_5). We must show that $(H_4)^*$ (respectively, $(H_5)^*$) is perfect.

Case 2.1. $H = H_4$ and $H^* = (H_4)^*$.

We show that the nodes of $(H_4)^*$ can be partitioned into a clique K and sets R_1, R_2 with no edges between R_1 and R_2, such that $K \cup R_1$ and $K \cup R_2$ induce perfect graphs. It then follows that $(H_4)^*$ is perfect [1, Chap. 16, Thm. 3]. Let $a, \ldots, d$ be nodes of $A, \ldots, D$, respectively, and let e, e' be nonadjacent nodes of E. Assume without loss of generality that b and c are nonadjacent and d and e are nonadjacent. We obtain $(H_4)^*$ by identifying two nonadjacent nodes among $a, \ldots, d, e, e'$.

If we identify e with $x \in \{a, b, c, d\}$, let D' be the set of all neighbors of e in D and take $K = D' \cup \{x\}$, $R_2 =$ the neighbors of e in E, $R_1 =$ the remaining nodes. Then the graph induced by $K \cup R_2$ is a clique and hence perfect, and the graph induced by $K \cup R_1$ contains D as a clique cutset. Now the subgraph induced by $A \cup B \cup C \cup D$ is the complement of a bipartite graph, and the subgraph induced by $D \cup E - R_2$ is triangulated. Thus both subgraphs are perfect [1], and hence the graph induced by $K \cup R_1$ is perfect.

If both of the identified nodes are in $\{e, e'\}$ or both in $\{a, b, c, d\}$, then take $K = D$, $R_2 = E$, and $R_1 =$ the remaining nodes. Then K is a clique cutset, $K \cup R_2$ is triangulated, and $K \cup R_1$ is the complement of a bipartite graph. This shows that $(H_4)^*$ is perfect.

Case 2.2. $H = H_5$ and $H^* = (H_5)^*$.

Let $a, \ldots, e, b_0, c_0$ be nodes of $A, \ldots, E, B_0, C_0$, respectively, with b, c nonadjacent. Let $(H_5)^*$ be obtained by identifying two nonadjacent nodes among $a, \ldots, e, b_0, c_0$. Once again, we proceed as in Case 2.1. If e is not one of the identified nodes, take $K = B_0 \cup C_0$, $R_2 = E$, $R_1 =$ the remaining nodes; if e is identified with $x \in \{a, \ldots, d\}$, take $K = B_0 \cup C_0 \cup \{x\}$, $R_2 = E - \{e\}$, $R_1 =$ the remaining nodes. In either case, the graph induced by $K \cup R_2$ is a clique and the graph induced by $K \cup R_1$ is the complement of a bipartite graph. This shows that $(H_5)^*$ is perfect. □

Acknowledgments. We thank the referees for useful comments.

REFERENCES

[1] C. Berge, *Graphs and Hypergraphs*, 2nd ed., North-Holland, Amsterdam, 1976.

[2] D. de Werra, N. V. R. Mahadev, and P. Solot, *Periodic compact scheduling*, in Proc. of the Internat. Conference on Graphs and Combinatorics, Marseille, July 1990; Discrete Math., to appear.

[3] S. Even, *Graph Algorithms*, Computer Science Press, Rockville, MD, 1979.

[4] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of* NP-*Completeness*, W. H. Freeman, San Francisco, 1978.

[5] M. Grötschel, L. Lovász, and A. Schrijver, *Polynomial algorithms for perfect graphs*, Ann. Discrete Math., 21 (1984), pp. 325–356.

[6] R. Hayward, *Weakly triangulated graphs*, J. Combin. Theory Ser. B, 39 (1985), pp. 200–208.

[7] L. Lovász, *Normal hypergraphs and the weak perfect graph conjecture*, Discrete Math., 2 (1972), pp. 253–267.

[8] H. Whitney, *Non-separable and planar graphs*, Trans. Amer. Math. Soc., 34 (1932), pp. 339–362.

EMBEDDING DE BRUIJN AND SHUFFLE-EXCHANGE GRAPHS IN FIVE PAGES*

BOJANA OBRENIĆ†

Abstract. Algorithms for embedding de Bruijn and shuffle-exchange graphs in books of five pages, with cumulative pagewidth $(5/3)2^n - (2/3) - (8/3)(n \bmod 2)$ and $(5/6)2^n + (2/3) - (4/3)(n \bmod 2)$, respectively, are presented. These are the first nontrivial bounds on the pagenumber of de Bruijn and shuffle-exchange graphs.

Key words. bookembeddings, stack layouts, de Bruijn graphs, shuffle-exchange graphs, interconnection networks, algorithms, fault-tolerant computing

AMS subject classifications. 68R10, 94C15, 68Q35, 05C99

1. Introduction. A *book* of *thickness* p is a set of p half-planes, called *pages*, sharing a common boundary, called the *spine*. A *p-page bookembedding* of a graph $G = (V, A)$ is a drawing of G in a book of thickness p so that the nodes of G reside on the spine of the book, while each edge of G is drawn in exactly one page, in such a way that no edges of G cross. The *pagenumber* of a graph G is the thickness of the smallest (in number of pages) book into which G can be embedded. The *width* of a page in a bookembedding is its maximum cutwidth. The *cumulative pagewidth* of a bookembedding is the sum of the widths of all pages.

The bookembedding problem appears in several formulations and has various origins. (Chung, Leighton, and Rosenberg [5] provide a detailed summary of these variants.) Within the realm of parallel architectures, this problem is relevant for the design of fault-tolerant processor arrays of identical processing elements. Rosenberg's approach [15] to the design of such arrays, named Diogenes, assumes that processing elements are laid out in a logical line, while some number of "bundles" of wires runs in parallel with the line. The configuration of the fault-free processors into the desired topology is effected by a network of switches that connect processors to the bundles of wires. The switching mechanism behaves as a *stack*, to which wires are entered or from which they are removed as the linear array of processors is scanned during the configuration process. Chung, Leighton, and Rosenberg [4] argue that the most significant cost in a Diogenes layout of an array is the *number* of bundles of wires, organized in hardware stacks, required to configure the array. A secondary cost is the total *width* of these bundles. Therefore, a good Diogenes design requires a linearization of nodes of the target array such that the edges of its interconnection network can be laid out in few small stacks. This problem, however, is *equivalent to finding an efficient bookembedding* of the graph underlying the interconnection network. The pagenumber of a graph equals the required number of stacks, while the cumulative pagewidth equals the required stack-width; it is therefore desirable to achieve bookembeddings of important graph families, with optimal pagenumber and pagewidth.

This bookembedding problem is generally very hard. Garey et al. [7] show that, for a given linearization of the nodes of a graph G and a given integer k, the problem of deciding if the linearization admits a k-page bookembedding of G is NP-complete.

*Received by the editors, March 25, 1991; accepted for publication (in revised form) September 24, 1992. This research was supported in part by National Science Foundation grant CCR-88-12567. A preliminary version of this paper was presented at the 3rd ACM Symposium on Parallel Algorithms and Architectures, July 22-24, 1991, in Hilton Head, South Carolina.

†Department of Computer Science, University of Massachusetts, Amherst, Massachusetts 01003, (obrenic@cs.umass.edu).

At present, bookembeddings of several graph families are known, though it is less often known whether these embeddings are optimal. Exemplifying this fact are the family of complete graphs and the family of complete bipartite graphs. While Bernhart and Kainen [1] determine the pagenumber of complete graphs exactly, Muder, Weaver, and West [14] present a bookembedding of complete bipartite graphs, which is the best known, but not known to be optimal. Very few algorithms exist for achieving efficient bookembeddings of arbitrary graphs. Yannakakis [19] gives an optimal bookembedding for the class of arbitrary planar graphs. Heath and Istrail [10] provide an algorithm for constructing efficient bookembeddings of arbitrary graphs of given genus. (Malitz [12] shows by a nonconstructive proof that better bookembeddings for this graph class are possible.)

Optimal (within constant factors, or even absolutely) bookembeddings have been constructed for almost all seriously proposed interconnection networks. Chung, Leighton, and Rosenberg [5] give such algorithms for trees, grids, X-trees, and hypercubes; Games [8] provides bookembeddings of butterfly-like graphs. Yet, *no efficient bookembeddings have been found for shuffle-like graphs*, another very popular class of interconnection networks represented by de Bruijn graphs and shuffle-exchange graphs. The very weak upper bound for the much broader class of bounded-degree graphs applies, but is nonconstructive, so it follows from the results of Chung, Leighton, and Rosenberg [5] or those of Malitz [13] that there exist bookembeddings of N-node de Bruijn (shuffle-exchange) graphs with pagenumber $O(\sqrt{N})$. This paper presents an algorithm for embedding these graphs in five pages. The best-known lower bound on the pagenumber of de Bruijn and shuffle-exchange graphs remains 3, which follows from the nonplanarity of these graphs.

It may be interesting to compare our results on shuffle-like networks with the results known about butterfly-like networks. Both families are bounded-degree hypercube-derivative networks; their computational power is a frequent topic of comparative studies (cf. [2], [16]). Both families have small pagenumber: three pages are sufficient (and, in general, necessary) for butterfly-like graphs, in contrast to hypercubes themselves, whose pagenumber is unbounded (logarithmic) in the size of the graph.

We remark that Rosenberg [15] has proposed for Diogenes a switching mechanism alternative to stacks of wires; this mechanism consists of *queues* of wires, so the success of Diogenes design with queues depends on finding efficient *queue layouts* of graphs. Heath and Rosenberg [11] find the queuenumber for practically all popular interconnection networks; for both butterfly-like and shuffle-like graphs, it is 2.

Section 2 introduces the bidendral decomposition of de Bruijn graphs and adduces its relevant properties. In §3 this decomposition is exploited in the development of the five-page bookembedding of de Bruijn graphs. The development starts by embedding separately the partial subgraphs produced by the decomposition; the partial embeddings are then composed into an efficient embedding of the whole graph. In §4 the embedding is adapted to shuffle-exchange graphs.

Notation. Let $Z_2 = \{0, 1\}$. The Greek letters α, β, and γ denote variables with values in Z_2. For integer $k \geq 0$, Z_2^k denotes the set of all strings of length k over Z_2. The lowercase letters of the Roman alphabet $(a, b, \ldots, x, y, z)$ denote variables with values in Z_2^k. $Z_2^0 = \{\lambda\}$ is the singleton set consisting of the *empty string* λ. For $x \in Z_2^k$, $|x| = k$ is the length of string x. Let $\alpha^k \in Z_2^k$ be the length-k string all of whose elements are equal to α. Let $\overline{\beta} = 1 - \beta$ and $\overline{\beta y} = \overline{\beta}\overline{y}$.

Remark. The bookembedding problem is defined for undirected graphs, while de Bruijn and shuffle-exchange graphs are usually thought of as directed. We also use the directed versions of these graphs, although the only advantage in manipulating directed arcs instead of undirected edges is in presentation convenience—we change nothing in the standard statement of the bookembedding problem.

2. Bidendral decomposition of de Bruijn graphs. This section presents a decomposition of de Bruijn graphs, which is subsequently exploited to construct their bookembedding. The graphs of interest are defined below.

The *order-n de Bruijn graph* $D(n)$ (cf. [3]) has node-set Z_2^n; given $y \in Z_2^{n-1}$, two arcs are incident out of each node βy: the *shuffle* arc that goes to $y\beta$ and the *shuffle-exchange* arc that goes to $y\overline{\beta}$. Let $\mathcal{S}(\beta y) = y\beta$ and $\mathcal{E}(\beta y) = y\overline{\beta}$. (See Fig. 1.)

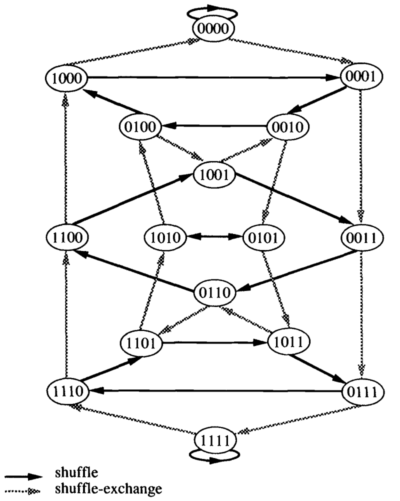


FIG. 1. *De Bruijn graph* $D(4)$.

The *complete binary tree* $T(h)$ of *height* h has node-set $\bigcup_{0\leq k\leq h} Z_2^k$ and arcs going from each $y \in Z_2^k$, $0 \leq k < h$, to its *children* $y0$ and $y1$. The *root* of the tree $T(h)$ is the empty string λ; the *leaves* of $T(h)$ are all nodes $y \in Z_2^h$.

Levels in the tree $T(h)$ are defined as follows: the 2^k nodes $x \in Z_2^k$, for $0 \leq k \leq h$, reside at level $h - k$. So, the root is the only node at level h; the leaves are at level 0.

To specify the bidendral decomposition of de Bruijn graphs, we introduce a binary tree that only slightly differs from $T(h)$. The *suspended complete binary tree* $T'(h)$ of *height* h is obtained by augmenting the complete binary tree $T(h)$ with a new node λ' and an arc incident out of node λ' to the root λ of $T(h)$. Node λ' is the *side root* of $T'(h)$; it is the second occupant of level h in tree $T'(h)$.

Let $<_Z$ be the lexicographic order on the set of all strings over Z_2. Within the tree $T'(h)$, define *tree-order* $<_{T'}$ on the node set of $T'(h)$ as follows:

$$\lambda' <_{T'} \lambda,$$

$$x <_{T'} y \iff |x| < |y| \vee (|x| = |y| \wedge x <_Z y).$$

The *reversed* lexicographic order $>_Z$ and the *reversed* tree-order $>_{T'}$ are defined naturally, so that $u >_Z v$ if and only if $v <_Z u$, and $u >_{T'} v$ if and only if $v <_{T'} u$.

The task now is to identify two suspended complete binary trees in $D(n)$ and to determine the structure of partial subgraphs induced by the arcs not contained in these trees. The node-sets of the two trees are obtained by partitioning the node-set of $D(n)$ in two sets, thereby inducing a partition of the arc-set of $D(n)$ into four sets. So, for each of the two values of $\gamma \in Z_2$, $D(n)$ contains partial subgraphs T'_γ and L_γ, which are defined in the following sections.

2.1. Trees. $T'_\gamma = (V_\gamma, A_\gamma)$ is the subgraph of $D(n)$ induced on the set of nodes

$$V_\gamma = \{\gamma y \mid y \in Z_2^{n-1}\}.$$

T'_γ is isomorphic to the suspended complete binary tree $T'(n-2)$. The isomorphism Φ_γ of the node-set of $T'(n-2)$ to V_γ is defined as follows: $\Phi_\gamma(\lambda') = \gamma^n$; for $x \in Z_2^k$, $k \le n-2$,

$$\Phi_0(x) = 0^{n-2-k}01x, \qquad \Phi_1(x) = 1^{n-2-k}10\overline{x}.$$

By definition Φ_γ is injective; it is also surjective by equal cardinality of its domain and its range ($|V_\gamma| = |Z_2^{n-1}| = 2^{n-1}$). Φ_γ preserves arcs, since

$$\Phi_\gamma(\lambda) = \mathcal{E}(\Phi_\gamma(\lambda')), \quad \Phi_\gamma(x0) = \mathcal{S}(\Phi_\gamma(x)), \quad \Phi_\gamma(x1) = \mathcal{E}(\Phi_\gamma(x)),$$

whenever $x \in Z_2^k$, $k < n-2$. To verify that every arc in A_γ that is not a self-loop is the image under Φ_γ of some arc in $T'(n-2)$, note that, when $x \in Z_2^{n-2}$, then $\Phi_\gamma(x) = \gamma\overline{\gamma}y$ (where $y = x$ or $y = \overline{x}$), whence $\mathcal{S}(\Phi_\gamma(x)) = \overline{\gamma}x\gamma \notin V_\gamma$ and $\mathcal{E}(\Phi_\gamma(x)) = \overline{\gamma}x\overline{\gamma} \notin V_\gamma$. Finally, $\mathcal{S}(\Phi_\gamma(\lambda')) = \gamma^n = (\Phi_\gamma(\lambda'))$.

Call the two arc-sets A_0 and A_1 the *tree arcs* of $D(n)$. Since $T'(n-2)$ has $2^{n-1}-1$ arcs, there are $2 \times (2^{n-1}-1) = 2^n - 2$ tree arcs in $A_0 \cup A_1$.

2.2. Leaf subgraphs. Let

$$V_\gamma^L = \{\gamma\overline{\gamma}x \mid x \in Z_2^{n-2}\}$$

be the set of *leaves* of the tree T'_γ. The graph $L_\gamma = (V_\gamma^L \cup V_{\overline{\gamma}}, A_\gamma^L)$ has node-set consisting of the leaves of tree T'_γ and all nodes of the other tree $T'_{\overline{\gamma}}$. Define the arc-set A_γ^L as the arcs incident out of leaves of tree T_γ. To verify that every arc in A_γ^L goes from a node in V_γ^L to some node of the other tree $T'_{\overline{\gamma}}$, note that, for each $x \in V_\gamma^L$, there exists $x' \in Z_2^{n-2}$ such that $x = \gamma\overline{\gamma}x'$; so

$$\mathcal{S}(x) = \overline{\gamma}x'\gamma \in V_{\overline{\gamma}}, \qquad \mathcal{E}(x) = \overline{\gamma}x'\overline{\gamma} \in V_{\overline{\gamma}}.$$

Call the two arc-sets A_0^L and A_1^L the *leaf arcs* of $D(n)$. Since two arcs are incident out of each of the 2^{n-2} leaves of T'_γ, there are $2 \times 2^{n-2} \times 2 = 2^n$ leaf arcs in $A_0^L \cup A_1^L$.

The two trees T'_0 and T'_1 are node-disjoint, so embedding one of them does not constrain the embedding of the other; however, each leaf arc connects nodes from different trees. The difficulty in embedding $D(n)$ in a small book is in finding a linearization of the nodes of the two trees that simultaneously accommodates the leaf arcs and respects the relative ordering of nodes prescribed by the embedding of the tree arcs. The following lemma clarifies the structure of the leaf subgraphs L_γ, thereby preparing for the desired linearization.

LEMMA 2.1. *Let $a, b \in V_\gamma^L$ be two leaves of T'_γ and let (a,u) and (b,v) be two leaf arcs incident into $T'_{\overline{\gamma}}$. Then $a <_{T'_\gamma} b$ if and only if $v <_{T'_{\overline{\gamma}}} u$.*

Proof. We first show that the tree-order $<_{T'_0}$, induced by T'_0 on V_0, is the same as the lexicographic order $<_Z$, while the tree-order $<_{T'_1}$, induced by T'_1 on V_1, is the same as the reversed lexicographic order $>_Z$. Indeed, for every pair of nodes $x, y \neq \lambda'$ of tree $T'(n-2)$ such that $x <_{T'} y$,

$$\Phi_0(x) = 0^{n-2-|y|}00^{|y|-|x|}1x <_Z 0^{n-2-|y|}01y = \Phi_0(y),$$

$$dst\Phi_1(x) = 1^{n-2-|y|}11^{|y|-|x|}0\overline{x} >_Z 1^{n-2-|y|}10\overline{y} = \Phi_1(y).$$

When $x = \lambda'$, then $\Phi_0(x) = 0^n <_Z \Phi_0(y)$, and $\Phi_1(x) = 1^n >_Z \Phi_1(y)$.

We complete the proof for the case where $\gamma = 0$, the case where $\gamma = 1$ being dual. Because $a, b \in V_0^L$, there exist $a', b' \in Z_2^{n-2}$ such that $a = 01a'$, $b = 01b'$. As just noted, $a <_{T_0'} b$ if and only if $a <_Z b$, or, equivalently $a' <_Z b'$, which means that

$$\mathcal{S}(a) = 1a'0 <_Z 1a'1 = \mathcal{E}(a) <_Z 1b'0 = \mathcal{S}(b) <_Z 1b'1 = \mathcal{E}(b).$$

Recall that $u \in \{\mathcal{S}(a), \mathcal{E}(a)\}$ and $v \in \{\mathcal{S}(b), \mathcal{E}(b)\}$; so this chain of relations is true if and only if $u <_Z v$, or, equivalently, $v <_{T_1'} u$. □

Given a subset $U \subseteq V_\gamma$, let node $U[<_{T_\gamma'}, i] \in U$, $0 \leq i < |U|$, have rank i in U, according to the order $<_{T_\gamma'}$. Define $U[>_{T_\gamma'}, i]$ analogously. Let $V_\gamma^{(\ell)}$ consist of the level-ℓ nodes of T_γ'.

Linearize the set V_γ^L of the leaves of T_γ', so that the leaves appear in the tree-order $<_{T_\gamma'}$. Partition V_γ^L into $n-1$ successive contiguous *segments* and let $S_\gamma^{(k)}$ denote segment k. Segment $S_\gamma^{(k)}$, $0 \leq k \leq n-3$ consists of 2^{n-3-k} nodes, starting with $V_\gamma^L[<_{T_\gamma'}, 2^{n-2} - 2^{n-3-k+1}]$ and ending with $V_\gamma^L[<_{T_\gamma'}, 2^{n-2} - 2^{n-3-k} - 1]$. Segment $S_\gamma^{(n-2)}$ contains only one node $V_\gamma^L[<_{T_\gamma'}, 2^{n-2} - 1]$.

By Lemma 2.1, our view of the partial subgraphs L_γ is summarized in the following. (See Fig. 2.)

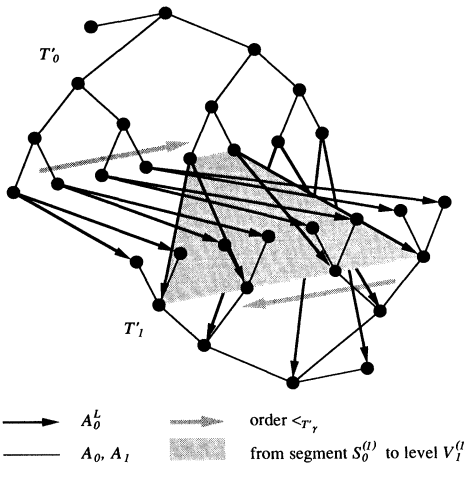


FIG. 2. *Bidendral decomposition of $D(5)$: The trees and leaf arcs A_0^L.*

PROPOSITION 2.2. *Let $0 \leq k \leq n-2$, $0 \leq j < 2^{n-3-k}$. The leaf arcs incident out of node $S_\gamma^{(k)}[<_{T_\gamma'}, j]$ are incident into the pair of level-k nodes $V_{\overline{\gamma}}^{(k)}[>_{T_\gamma'}, 2j]$ and $V_{\overline{\gamma}}^{(k)}[>_{T_\gamma'}, 2j+1]$ of $T_{\overline{\gamma}}'$.*

3. Embedding de Bruijn graphs in five pages. This section develops the bookembedding of $D(n)$ in five pages that is the main result of this paper. Theorem 3.1 states the result more precisely.

THEOREM 3.1. *The order-n de Bruijn graph $D(n)$ admits a bookembedding in five pages, with cumulative pagewidth* $(5/3)2^n - (2/3) - (8/3)(n \bmod 2)$.

The embedding that establishes Theorem 3.1 is developed in three stages. In the first stage, the subembeddings of the two trees T_0' and T_1' are specified. These subembeddings are independent, because node-sets V_0 and V_1 are disjoint. Each tree requires two pages, so four pages may be required for the first-stage subembeddings, since V_0 and V_1 appear on the spine interleaved in some way dictated by the subembeddings of subsequent stages, thereby preventing the pages used by one tree from being reused by the other. Four pages are also sufficient, as the tree-subembeddings do not constrain each other. In the second stage, each set of leaf arcs A_γ^L is embedded. The resulting leaf-subembeddings are not mutually independent, as each involves the leaf nodes V_γ^L of one tree and the node-set $V_{\bar{\gamma}}$ of the other tree. The consideration of interference between the second-stage subembeddings is deferred until the last stage, so these two subembeddings are constructed independently; each requires one page. In the last stage, a node layout consistent with the four subembeddings of the first two stages is exhibited. Finally, two pages of the first stage that can be combined into a single page are identified, thereby arriving at the total of five pages for the complete embedding.

3.1. Embedding the trees. This section presents two varieties of *spiral embedding* of trees, in particular, of $T'(h)$. In both spiral embeddings, the trees are laid out by an appropriate alternation of their levels, while each level is contiguous and ordered. We think of the levels being laid out starting from the lowest-numbered level, the leaves, and progressing sequentially toward the highest-numbered level, the roots. In the *inward* spiral embedding, the last levels to be laid out are the innermost levels, while in the *outward* spiral embedding the last levels are the outermost levels. See Figs. 3 and 4. The following definition makes the layout precise.

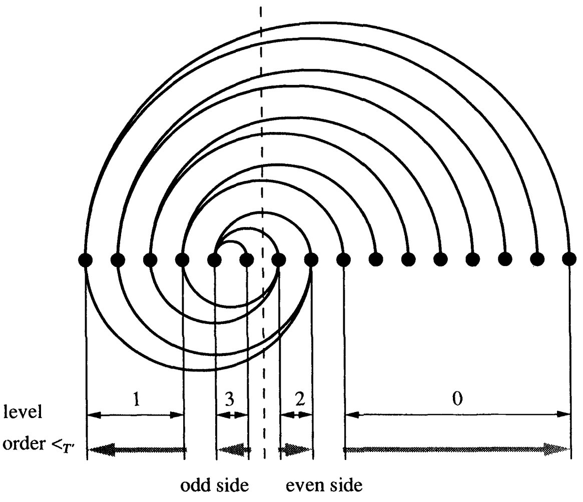


FIG. 3. *Inward spiral embedding of* $T'(3)$.

DEFINITION 3.2. Let $0 \le k \le \lfloor (h-1)/2 \rfloor$ and $0 \le \ell \le \lfloor h/2 \rfloor$.

(1) In the *inward spiral embedding* of $T'(h)$, the layout of nodes from left to right along the spine is: nodes at levels $1, 3, \ldots, 2k+1, \ldots, h-1+(h \bmod 2)$, in that order, each level in reversed tree-order $>_{T'}$, followed by nodes at levels $h-(h \bmod 2), h-2-(h \bmod 2), \ldots, h-2\ell-(h \bmod 2), \ldots, 0$, in that order, each level in tree-order $<_{T'}$.

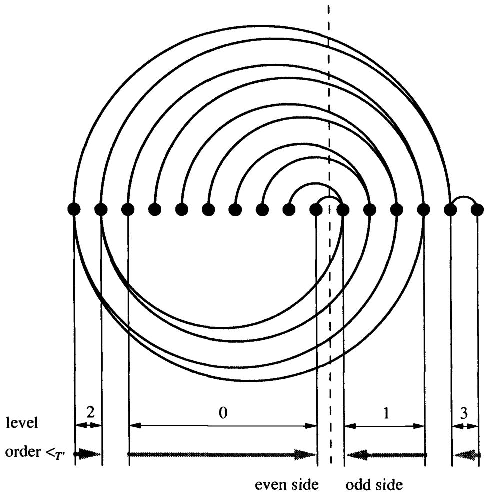


FIG. 4. *Outward spiral embedding of $T'(3)$.*

(ii) In the *outward spiral embedding* of $T'(h)$, the layout of nodes from left to right along the spine is: nodes at levels $h - (h \bmod 2), h - 2 - (h \bmod 2), \ldots, h - 2\ell - (h \bmod 2), \ldots, 0$, in that order, each level in tree-order $<_{T'}$, followed by nodes at levels $1, 3, \ldots, 2k+1, \ldots, h - 1 + (h \bmod 2)$, in that order, each level in reversed tree-order $>_{T'}$.

The spiral embeddings separate odd-numbered tree levels from even-numbered ones, so that all levels of equal parity appear at one side of some point on the spine, while the levels of opposite parity appear at the other side. Call these sides the *even* and the *odd* sides of the spine, according to the tree levels that occupy them. In the inward spiral embedding the odd side is the left side of the spine: in the outward spiral embedding the odd side is the right side of the spine. Otherwise, both embeddings place identically the levels of equal parity relative to one other—the odd-numbered ones in the order of increasing level number, the even-numbered ones in the order of decreasing level number. They also identically place the nodes inside each level—in tree-order within even-numbered levels, in reversed tree-order within odd-numbered levels. In summary, we have the following conclusion.

PROPOSITION 3.3. *The even side of a spiral embedding is laid out in tree-order. The odd side of a spiral embedding is laid out in reversed tree-order.*

The properties of the arc-assignment in the spiral embeddings are summarized in the following lemma.

LEMMA 3.4. *Both outward and inward spiral embeddings of $T'(h)$ require two pages. Cumulative pagewidth of the outward spiral embedding is $2^{h+1} - 2$, while cumulative pagewidth of the inward spiral embedding is $2^{h+1} - 1$.*

Proof. The arcs of $T'(h)$ go from nodes of one level to nodes at the level below; thus each arc goes either from the even side of the spine to the odd side, or vice versa. Assign to the *upper* page of a spiral embedding those arcs that go from the odd side to the even side and assign to the *lower* page those arcs that go from the even side to the odd side.

Consider two arcs $(x, x\alpha)$ and $(y, y\beta)$, where $|x|, |y| < h$. Say that these arcs are assigned to the same page of a spiral embedding; so x and y are at the same side of the spine, while $x\alpha$ and $y\beta$ are both at the other side. If $x = y$, then the two arcs share an endpoint; thus they cannot cross. If $x <_{T'} y$, then $x\alpha <_{T'} y\beta$. However, by

Proposition 3.3, the even and the odd sides of the spine are ordered oppositely; so $x\alpha$ and $y\beta$ appear on the spine ordered oppositely to x and y; the two arcs therefore nest inside one another.

To verify the cumulative pagewidth, note that the total cutwidth of both pages in the outward spiral embedding equals the number of arcs in the complete binary tree $T(h)$, the maximum occurring at the division point between the odd and even sides; the total cutwidth of both pages in the inward spiral embedding equals the number of arcs in the suspended complete binary tree $T'(h)$, the maximum occurring between the two roots. □

The first stage of the embedding, the layout of the two trees, is now complete.

The node layout of the two trees T'_γ is as follows.

(i) T'_0 is laid out by *outward* spiral embedding;

(ii) T'_1 is laid out by *inward* spiral embedding.

Recalling that the height of each tree is $n-2$, Lemma 3.4 yields Corollary 3.5.

COROLLARY 3.5. *The two component trees* T'_0 *and* T'_1 *of* $D(n)$ *are embedded in two pages each, with cumulative pagewidth* $2^{n-1}-2$ *and* $2^{n-1}-1$, *respectively.*

3.2. Embedding the leaf arcs. The next goal is to embed leaf arcs A^L_γ, which go from V^L_γ to $V_{\overline{\gamma}}$, without violating the *relative* ordering of nodes of $V_{\overline{\gamma}}$, stipulated by the spiral embedding of $T'_{\overline{\gamma}}$. See Fig. 5.

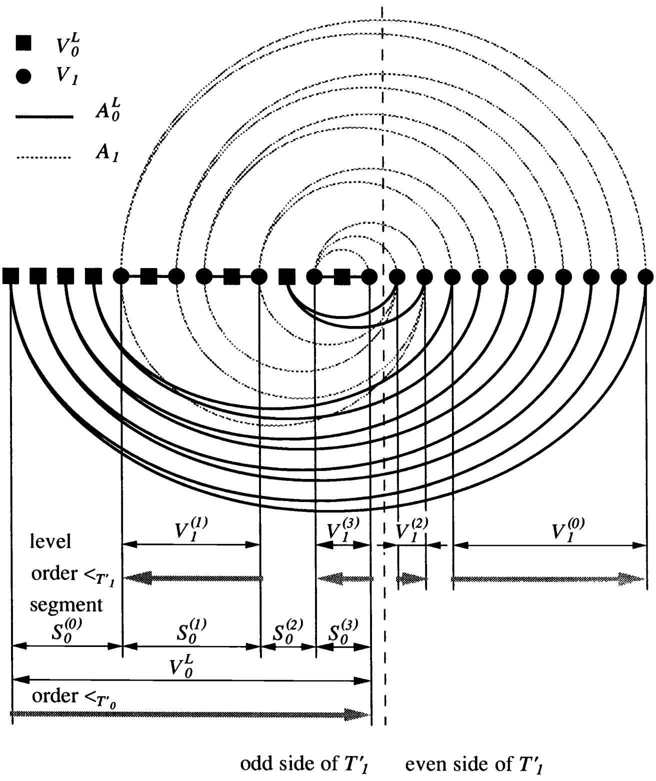


FIG. 5. *Embedding leaf arcs* A^L_0 *of* $D(5)$.

The node layout of the leaf subgraph L_γ is as follows. Lay out the nodes $V_{\overline{\gamma}}$ as mandated by the spiral embedding of tree $T'_{\overline{\gamma}}$ (inward if $\overline{\gamma}=1$, outward if $\overline{\gamma}=0$). Then *interleave* the leaves V_γ^L of tree T'_γ with the *odd* side of the spiral embedding of $T'_{\bar\gamma}$ so that the following hold:

(i) Each even-numbered segment $S_\gamma^{(2i)}$ of V_γ^L is placed *contiguously*, *in tree-order*, *between levels* $V_{\bar\gamma}^{(2i-1)}$ *and* $V_{\bar\gamma}^{(2i+1)}$. If level $2i-1$ does not exist, $S_\gamma^{(2i)}$ is placed immediately to the left of the leftmost node of level $V_{\bar\gamma}^{(2i+1)}$; analogously, if level $2i+1$ does not exist, $S_\gamma^{(2i)}$ is placed immediately to the right of the rightmost node of level $V_{\bar\gamma}^{(2i-1)}$;

(ii) Each node of an odd-numbered segment of V_γ^L is placed *between the two nodes* of $V_{\bar\gamma}$ to which it is adjacent via leaf arcs of A_γ^L.

The properties of the second-stage embedding are summarized in the following lemma.

LEMMA 3.6. *Each leaf subgraph L_γ, generated by arcs A_γ^L that go from leaves V_γ^L to nodes of $V_{\bar\gamma}$, is embedded in one page of width* $(1/3)(2^n + 5 - 4(n \bmod 2))$. *The leaf nodes V_γ^L are laid out in tree-order.*

Proof. First, all leaf nodes of odd-numbered segments of V_γ^L are placed immediately beside their corresponding adjacent nodes in the tree $T'_{\bar\gamma}$; so arcs incident out of odd-numbered segments do not cross any other arcs on the page; these arcs contribute 1 to the pagewidth.

To complete the proof for leaf arcs incident out of even-numbered segments of V_γ^L, invoke Propositions 2.2 and 3.3. The odd-numbered levels of $T'_{\bar\gamma}$ are placed in increasing order of level numbers, thus compelling the odd-numbered segments of leaves V_γ^L to appear in order of increasing segment numbers. Furthermore, each even-numbered leaf segment, say $S_\gamma^{(k)}$, is placed between levels $V_{\bar\gamma}^{(k-1)}$ and $V_{\bar\gamma}^{(k+1)}$ of $T'_{\bar\gamma}$ (assuming both levels exist), thus between segments $S_\gamma^{(k-1)}$ and $S_\gamma^{(k+1)}$. This imposes the order of increasing segment numbers on even-numbered segments. So, all nodes of even-numbered segments appear in tree-order and lie within the odd side, while the even-numbered levels of $T'_{\bar\gamma}$ are also in tree-order and lie within the even side. By Proposition 2.2, this results in opposite orders of sources and destinations of these leaf arcs; therefore, no two leaf arcs cross.

By Proposition 3.3, the order within odd-numbered levels of $T'_{\bar\gamma}$ is reversed tree-order. By Proposition 2.2, the order within odd-numbered leaf segments of V_γ^L must be tree-order. Since the leaves in even-numbered segments are also in tree-order, and since all segments are laid out in order of increasing segment number, the entire leaf set V_γ^L is in tree-order.

The contribution of leaf arcs incident out of even-numbered segments of leaves V_γ^L into even-numbered levels of tree $T'_{\bar\gamma}$ is

$$\left(\sum_{k=0}^{\lfloor\frac{n-2}{2}\rfloor} 2^{2k+(n \bmod 2)}\right) + 1 - (n \bmod 2) = \frac{1}{3}(2^n + 2 - 4(n \bmod 2)),$$

which yields the claimed pagewidth after accounting for the leaf arcs incident out of odd-numbered segments. □

3.3. The complete embedding. The partial embeddings defined in the previous sections must yet be proved consistent. The embeddings of the trees in the first stage are trivially so, since they involve disjoint sets of both nodes and arcs. By construction, the second-stage embedding of each leaf subgraph L_γ is consistent with the corresponding spiral embedding of tree $T'_{\bar\gamma}$. It is also consistent with the embedding of tree T'_γ, because the embedding of L_γ involves only the leaves V_γ^L of T'_γ, and requires only that these appear in tree-order (by Lemma 3.6). However, this is exactly the order required by the spiral embedding of T'_γ. It remains to confirm that both leaf sets V_γ^L can be laid out simultaneously in the odd sides of the spiral embeddings of the corresponding trees $T'_{\bar{\gamma}}$. To that end, recall that the two spiral embeddings have their odd (even) sides in opposite sides of the spine, so the constraint is readily satisfied by identifying the odd side of one spiral embedding with the even side of the other. See Fig. 6.

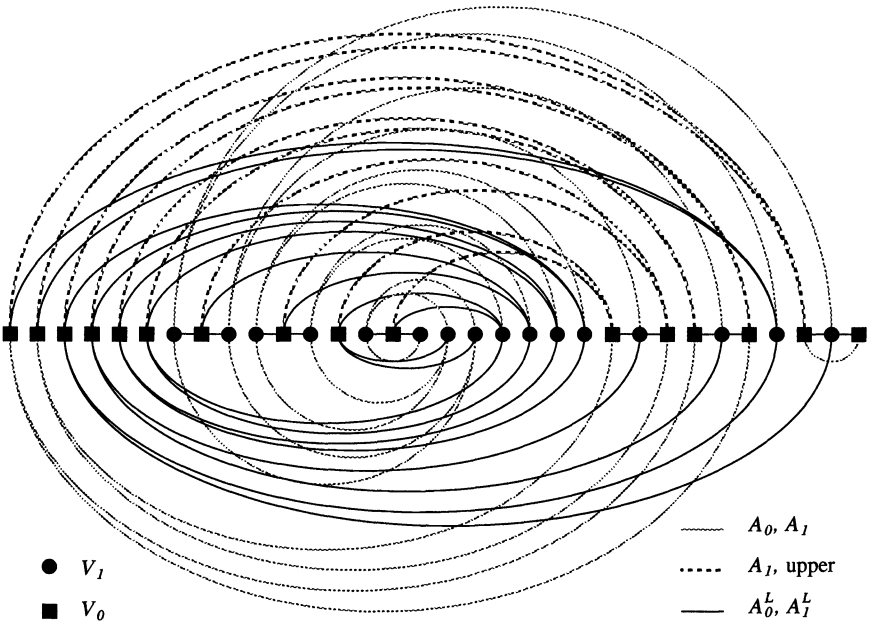


FIG. 6. *Embedding $D(5)$ in five pages.*

COROLLARY 3.7. *The four partial embeddings of the two trees T'_γ and the two leaf subgraphs L_γ define an embedding of $D(n)$ in six pages.*

The final task is to show that two of the six pages can be coalesced.

LEMMA 3.8. *Assume that the lower page of a spiral embedding accommodates those arcs that go from the even side to the odd side. Then, the two lower pages of the spiral embeddings of the trees T'_0 and T'_1 can be coalesced.*

Proof. Let (x_0, y_0) be an arc on the lower page of the outward spiral embedding of T'_0 and let (x_1, y_1) be an arc on the lower page of the inward spiral embedding of T'_1. We prove that the only possible ordering of the endpoints of these arcs on the spine is x_0, y_1, x_1, y_0, in which ordering the two arcs do not cross. Since the sources of the arcs are in the even sides and the destinations in the odd sides of the corresponding spiral embeddings, both x_0 and y_1 are to the left of x_1 and y_0, as the left side is even for T'_0 and odd for T'_1.

To prove that x_0 is to the left of y_1, we find a node that is both to the left of y_1 and to the right of x_0. Indeed, x_0 is in some nonleaf even-numbered level of T'_0, so it is to the left of all leaves of V_0^L, by properties of the outward spiral embedding. However, y_1 is in some odd-numbered level of T'_1, hence is to the right of leaf segment $S_0^{(0)}$, by properties of the embedding of leaf arcs. Thus, all nodes in segment $S_0^{(0)}$ are to the left of y_1 and to the right of x_0. Analogously, all nodes in segment $S_1^{(0)}$ are to the left of y_0 and to the right of x_1, whence the claimed ordering. □

The proof of Theorem 3.1 is completed by combining the cumulative pagewidths of the component embeddings, as established in Corollary 3.5 and Lemma 3.6, to derive the claimed cumulative pagewidth.

4. Embedding shuffle-exchange graphs in five pages. The order-n *shuffle-exchange* graph $S(n)$ (cf. [18]) has node-set Z_2^n; given $y \in Z_2^{n-1}$ and $\beta \in Z_2$, the *shuffle* arc goes from node βy to node $y\beta$, and the *exchange* arc goes from node $y\beta$ to node $y\overline{\beta}$.

Feldmann and Unger [6] show that the undirected shuffle-exchange graph $S(n)$ is a subgraph of the de Bruijn graph $D(n)$. The bookembedding of Theorem 3.1 thus contains a bookembedding of $S(n)$, given the appropriate renaming of nodes of $D(n)$. The following theorem announces that this bookembedding of de Bruijn graph $D(n)$ almost contains that of shuffle-exchange graph $S(n)$ even without renaming of nodes. (See Fig. 7.)

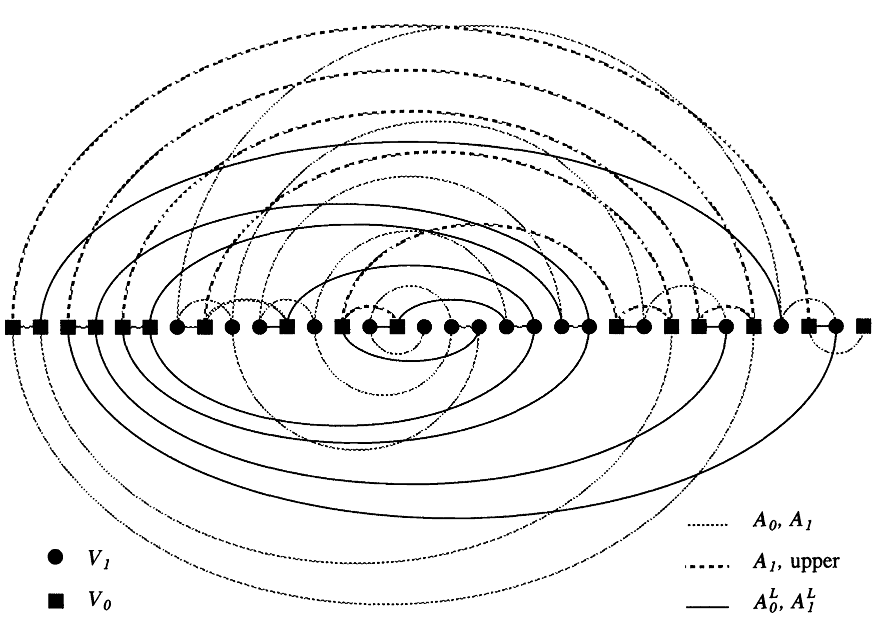


FIG. 7. *Embedding* $S(5)$ *in five pages.*

THEOREM 4.1. *The order-n shuffle-exchange graph $S(n)$ admits a bookembedding in five pages, with cumulative pagewidth* $(5/6)2^n + (2/3) - (4/3)(n \bmod 2)$.

Proof. The node layout is identical to that of $D(n)$. All shuffle arcs of $S(n)$ are identified with shuffle arcs of $D(n)$. Each exchange arc of $S(n)$ is incident to nodes $y\gamma$ and $y\overline{\gamma}$, for some $y \in Z_2^{n-1}$. However, one of $\{y\gamma, y\overline{\gamma}\}$ is the immediate successor of the other in the tree-order of one of the trees, say in $<_{T'_\gamma}$. There are no nodes of V_γ between $y\gamma$ and $y\overline{\gamma}$, so a new arc between them does not cross any other arc in either of the two pages of the spiral embedding of T'_γ.

The claimed cumulative pagewidth is arrived at after removing the shuffle-exchange arcs from the embedding of $D(n)$ and subsequently inserting exchange arcs into the spiral embeddings of the two trees. □

5. Conclusion. We have presented algorithms for embedding shuffle-like graphs in books of five pages. It remains unknown whether five pages are necessary, as the best-known lower bound is 3: Fig. 8, below, presents an embedding of the order-5 de Bruijn graph in four pages.

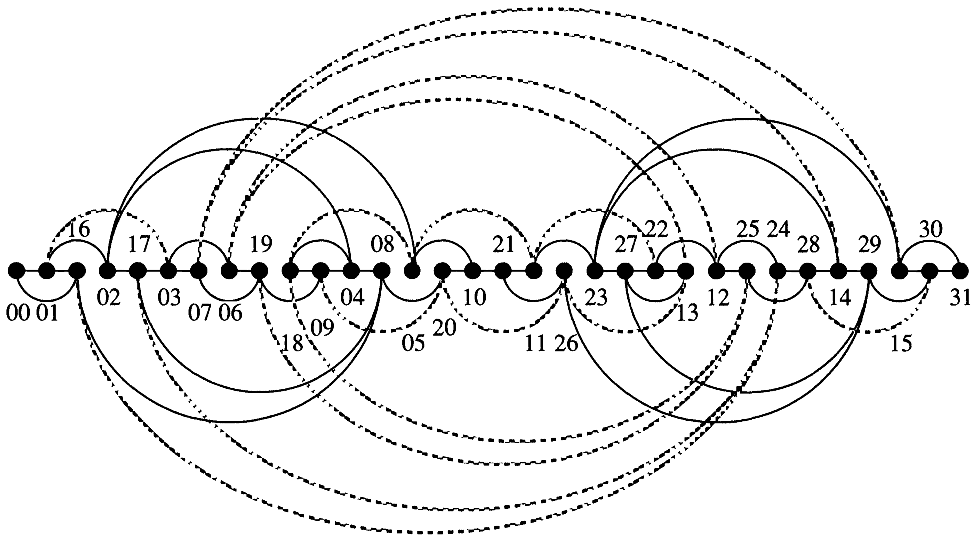


FIG. 8. *Embedding $D(5)$ in four pages.*

The *pagewidths* of our bookembeddings are greater than optimal by a factor logarithmic in the size of the graphs. This weakness is found in other bookembeddings of popular interconnection networks (cf. [8], [5]); it would be very interesting to bring the pagewidths of these embeddings closer to optimal, while retaining small pagenumbers, or to find some pagenumber-pagewidth tradeoffs. The general problem of transforming a bookembedding with optimal pagenumber and suboptimal pagewidth into one having pagewidth of optimal order and pagenumber not much greater than optimal is open for all but one-page graphs: Heath [9] presents an algorithm that converts one-page bookembeddings into two-page bookembeddings having logarithmic (asymptotically optimal) cumulative pagewidth. Although the general problem for graphs with arbitrary pagenumber is open, some special cases offer evidence that good solutions are possible: Chung, Leighton, and Rosenberg [5] describe families of one-page and two-page graphs whose cumulative pagewidth decreases dramatically (from linear in the number of nodes to a constant) when only one additional page is used; Stöhr [17] constructs families with the same property, but for an arbitrary value of the pagenumber. For the Diogenes approach to fault-tolerant design of processor arrays, simultaneous optimization of both cost measures in the bookembeddings of the prevailing interconnection networks would reduce notably the price of fault-tolerance.

Acknowledgments. I am greatly indebted to my research advisor Arnold L. Rosenberg for proposing the problem and for supplying many valuable suggestions from which the presentation has substantially improved, to Antony L. Hosking for his many helpful comments, and to the referee from whose numerous, careful, and insightful remarks the presentation has highly benefited.

REFERENCES

[1] F. Bernhart and P. C. Kainen, *The book thickness of a graph*, J. Combin. Theory Ser. B, 27 (1979), pp. 320–331.

[2] S. N. Bhatt, F. R. K. Chung, J.-W. Hong, F. T. Leighton, B. Obrenić, A. L. Rosenberg, and E. J. Schwabe, *Optimal emulations by butterfly-like networks*, J. Assoc. Comput. Mach., to appear.

[3] N. G. de Bruijn, *A combinatorial problem*, Proc. Kon. Nederl. Akad. Wetensch., 49 (1946), Part 2, pp. 758–764.

[4] F. R. K. Chung, F. T. Leighton, and A. L. Rosenberg, DIOGENES—*A methodology for designing fault-tolerant processor arrays*, in Proc. 13th Internat. Conf. on Fault-Tolerant Computing, 1983, Milan, Italy, pp. 26–32.

[5] ———, *Embedding graphs in books: A layout problem with applications to* VLSI *design*, SIAM J. Algebraic Discrete Meth., 8 (1987), pp. 33–58.

[6] R. Feldmann and W. Unger, *The cube-connected cycles network is a subgraph of the butterfly network*, Parallel Proc. Let., 2 (1992), pp. 13–19.

[7] M. R. Garey, D. S. Johnson, G. L. Miller, and C. H. Papadimitriou, *The complexity of coloring circular arcs and chords*, SIAM J. Algebraic Discrete Meth., 1 (1980), pp. 216–227.

[8] R. A. Games, *Optimal book embeddings of the* FFT, *Beneš, and barrel shifter networks*, Algorithmica, 1 (1986), pp. 233–250.

[9] L. S. Heath, *Embedding outerplanar graphs in small books*, SIAM J. Algebraic Discrete Meth., 8 (1987), pp. 198–218.

[10] L. S. Heath and S. Istrail, *The pagenumber of genus g graphs is $O(g)$*, J. Assoc. Comput. Mach., 39 (1992), pp. 479–501.

[11] L. S. Heath and A. L. Rosenberg, *Laying out graphs using queues*, SIAM J. Comput., 21 (1992), pp. 927–958.

[12] S. M. Malitz, *Genus g graphs have pagenumber $O(\sqrt{g})$*, J. Algorithms, to appear; in Proc. 29th IEEE Sympos. on Foundations of Computer Science, 1988, White Plains, NY, pp. 458–468.

[13] ———, *Graphs with E edges have pagenumber $O(\sqrt{E})$*, J. Algorithms, to appear.

[14] D. J. Muder, M. L. Weaver, and D. B. West, *Pagenumber of complete bipartite graphs*, J. Graph Theory, 12 (1988), pp. 469–489.

[15] A. L. Rosenberg, *The Diogenes approach to testable fault-tolerant arrays of processors*, IEEE Trans. Comput., C-32 (1983), pp. 902–910.

[16] ———, *Product-shuffle networks: Toward reconciling shuffles and butterflies*, Discrete Appl. Math., 37 (1992), pp. 465–488.

[17] E. Stöhr, *A trade-off between page number and page width of book embeddings of graphs*, Inform. Comput., 79 (1988), pp. 155–162.

[18] H. Stone, *Parallel processing with the perfect shuffle*, IEEE Trans. Comput., C-20 (1971), pp. 153–161.

[19] M. Yannakakis, *Embedding planar graphs in four pages*, J. Comput. System Sci., 38 (1989), pp. 36–67.

SIAM J. DISCRETE MATH
Vol. 6, No. 4, pp. 655-676, November 1993

REPRESENTATIONS OF BOREL CAYLEY GRAPHS*

K. WENDY TANG† AND BRUCE W. ARDEN‡

Abstract. There is a continuing search for dense (δ, D) interconnection graphs, that is, regular, undirected, degree δ graphs with diameter D and having a large number of nodes. Cayley graphs formed by Borel subgroups currently contribute to some of the densest known $(\delta = 4, D)$ graphs for a range of D [1]. However, the group theoretic representation of these graphs makes the development of efficient routing algorithms difficult. In an earlier report, it was shown that all Cayley graphs have generalized chordal ring (GCR) representations [2]. In this paper, it is shown that all degree-4 Borel Cayley graphs can also be represented by the more restrictive chordal rings (CR) through a constructive proof. A step-by-step algorithm to transform any degree-4 Borel Cayley graph into a CR graph is provided. Examples are used to illustrate this concept.

Key words. interconnection network, massively parallel computer, Cayley graph, Borel Cayley graph, generalized chordal ring, chordal ring

AMS subject classifications. 68R10, 68M07, 68RXX

1. Introduction. *Multiprocessors* and *multicomputers* are two major categories of parallel computers [3]. In the former, processors communicate via shared memory, whereas in the latter, each processor has its own local memory (hence a computer), and communication is via message passing. Whether it is a shared-memory multiprocessor or a message-passing multicomputer, an efficient *interconnection network* to interconnect the communicating elements is critical to the performance of the parallel computer [4]. In the design of an interconnection network, there are two major issues, the interconnection *topology* and *routing algorithms*.

An interconnection topology can be modeled as a graph. To model a multicomputer system, we consider *regular*, *undirected* graphs with no *multiple edges* between any pair of nodes. A graph is regular when it has the same number of incident edges, or *degree*, at every node [5]. Nodes of the graph correspond to processors with local memory, and the edges represent connections between these elements. Due to the limited number of connections that can be made to real chips, we are interested primarily in regular graphs of small degree. For a given small degree, we are interested in *dense graphs* [6]. A *dense graph* is one with a large number of nodes for a given *diameter*. The diameter is the maximum *distance* between all node pairs. Here *distance* between two nodes refers to the smallest number of hops between the two nodes. A dense graph allows the interconnection of a large number of processing elements with a potentially small communication delay. Furthermore, a *symmetric graph* is also desirable, because then an identical routing algorithm can be used at every node [3].

A variety of network topologies and routing algorithms have been proposed as interconnection models [7]–[12]. However, graphs originally generated from these topologies have not been the densest for their interconnection degree. The search for (δ,D) graphs that connect the maximum number of nodes with a degree δ and diameter D continues [6]. Among these (δ,D) graphs, the degree-4 graphs (i.e., $\delta = 4$) receive special attention because of the realizability of degree-4 interconnections. The TRANSPUTER™ chips are examples of such connectability [13].

Amid the many interconnection models, a special class of symmetric graphs, Cayley graphs, is an attractive candidate [1], [14], [15]. Besides their symmetric property, Cayley

*Received by the editors August 1, 1992; accepted for publication November 19, 1992.

†Department of Electrical Engineering, State University of New York at Stony Brook, Stony Brook, New York 11794-2350.

‡Department of Electrical Engineering, University of Rochester, Rochester, New York 14627.

graphs from the Borel subgroup, *Borel Cayley graphs* for short, are the densest known degree-4 graphs for a range of diameters ($D = 7, \ldots, 13$) [1]. In other words, these degree-4 graphs interconnect the largest number of nodes for this degree and range of diameter ($D = 7, \ldots, 13$), thus potentially minimizing communication delay in a parallel computer. However, practical implementation of these graphs as an interconnection model in a multicomputer system is hampered by the lack of a systematic *representation* or *structure* of Borel Cayley graphs. Originally, Borel Cayley graphs are defined over a group of matrices, which has no simple ordering and hence no regular graph structure. This representation problem of Borel Cayley graphs makes the development of routing algorithms difficult.

Generalized chordal rings (GCR) [12] and the more specialized *chordal rings* (CR) [10], on the other hand, are two existing topologies that are defined in the integer domain and have a systematic and regular structure. The definitions and properties of GCR and CR graphs are reviewed in the next section. In an earlier report, we proved that *any* Cayley graph can be represented as GCRs and provided a sufficient condition for Cayley graphs to have CR representations [2]. This paper concentrates on degree-4 *Borel Cayley graphs*. We present another interesting result concerning the representations of these graphs. Namely, *all* degree-4 Borel Cayley graphs have the more restrictive CR representations, in addition to other GCR representations. A CR is a special case of a GCR. It includes a Hamiltonian cycle formed by edges connecting adjacent integers in the modulo n labels, thus permitting a *distance-reduction* routing algorithm, called *CR routing*. Given a degree-4 Borel Cayley graph with $n = pk$ nodes, where p is a prime number and $k < p$, is a factor of $p - 1$, this distance-reduction algorithm requires a small table of $O(k)$. However, the algorithm is *suboptimal* in the sense that a shortest path is not guaranteed. Simulation shows that a more dynamic approach produces pathlength closer to optimal. The details of CR routing, its simulation, and other routing algorithms are discussed in other papers [16]–[18].

This paper is organized as follows. In §2 we review the definitions of GCRs, CRs, Cayley graphs, and Borel Cayley graphs. The proposition that all Cayley graphs have GCR representations and the sufficient condition for a Cayley graph to have a CR representation are also restated. In §3 we prove that all degree-4 Borel Cayley graphs have CR representations. Section 4 includes three examples to illustrate the transformation of degree-4 Borel Cayley graphs to CRs. Finally, in §5 we present a summary and conclusions.

2. Review. In this section, we review the definitions of GCRs, CRs, Cayley graphs in general, and Borel Cayley graphs in particular. We begin with the definition of GCR.

DEFINITION 1. A graph $\mathbf{R}$ is a GCR if nodes of $\mathbf{R}$ can be labeled with integers mod n (the number of nodes) and if there is a divisor q of n such that node i is connected to node j if and only if node $i + q \pmod{n}$ is connected to node $j + q \pmod{n}$.

According to this definition, vertices of a GCR are classified into q classes, each class with n/q elements. The classification is based on modulo q arithmetic. Two vertices having the same residue (mod q) are considered to be in the same class. That is, class i consists of the following nodes: $i,\ i + q,\ i + 2q,\ \ldots,\ i + (m - 1)q \pmod{n}$, where $m = n/q$ and node i is the *representing element* of class i. Since i connects to j implies that $i + q$ connects to $j + q \pmod{n}$, nodes in the same class have the same connection rules defined by the *connection constants* or *GCR constants*. When the GCR constants for the different classes are known, connections of the entire graph are defined.

For example, Fig. 1 shows a degree-4 GCR with ten nodes and $q = 2$ classes. The connection rules for these classes can be defined as follows: Let $\mathbf{V} = \{0, 1, \ldots, 9\}$. For any $i \in \mathbf{V}$, if

$$\begin{aligned} i \bmod 2 =: \text{“0”} &: i \text{ is connected to } i+2, i+3, i-1, i-2 \pmod{10};\\ =: \text{“1”} &: i \text{ is connected to } i+1, i+4, i-4, i-3 \pmod{10}. \end{aligned}$$

In this case, the vertices of the graph are numbered from 0 to 9 and are divided into even and odd classes. For the even vertices, the connection constants are +2, +3, -1, and -2, and, for the odd vertices, the connection constants are +1, +4, -4, and -3. The addition of these connection constants to the node label is done in modulo n arithmetic.

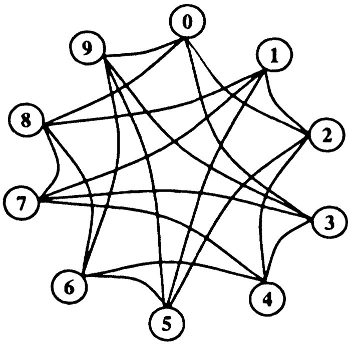


FIG. 1. *A degree*-4 GCR ($n = 10, q = 2$).

This class-structure of a GCR provides a *regular structure* and a *concise* and *simple* way of describing connectivity in the integer domain, therefore making GCR an attractive representation.

A CR is a special case of GCR, in which every node has +1 and -1 modulo n connections. In other words, a CR satisfies the connection condition in Definition 1, and, in addition, all the nodes on the peripheral of the ring are connected to form a Hamiltonian cycle.

Figure 2 shows a degree-4 CR with ten nodes and $q = 2$ classes. The connection rules for these classes can be defined as follows: Let $\mathbf{V} = \{0, 1, \dots, 9\}$. For any $i \in \mathbf{V}$, if

$$\begin{aligned} i \bmod 2 =: \text{“0”} &: i \text{ is connected to } i+1, i-1, i+2, i-2 \pmod{10};\\ =: \text{“1”} &: i \text{ is connected to } i+1, i-1, i+4, i-4 \pmod{10}. \end{aligned}$$

Note that every class has +1 and -1 as GCR constants and that nodes on the peripheral of the ring are connected.

The construction of Cayley graphs is described by finite (algebraic) group theory. Recall that a group ($\mathbf{V}$, $*$) consists of a set $\mathbf{V}$, which is closed under inversion, and a single law of composition $*$, also known as group multiplication. There also exists an identity element $I \in \mathbf{V}$. A group is finite if there is a finite number of elements in $\mathbf{V}$.

DEFINITION 2. A graph $\mathbf{C} = (\mathbf{V}, \mathbf{G})$ is a Cayley graph with vertex set $\mathbf{V}$ if two nodes $v_1, v_2 \in \mathbf{V}$ are adjacent $\Leftrightarrow v_1 = v_2 * g$ for some $g \in \mathbf{G}$, where ($\mathbf{V}$, $*$) is a finite group and $\mathbf{G} \subset \mathbf{V} \backslash \{I\}$. $\mathbf{G}$ is called the generator set of the graph and I is the identity element of the finite group ($\mathbf{V}$, $*$).

The definition of a Cayley graph requires nodes to be elements in a group but does not specify a particular group. A class of Cayley graphs that contributes to the densest

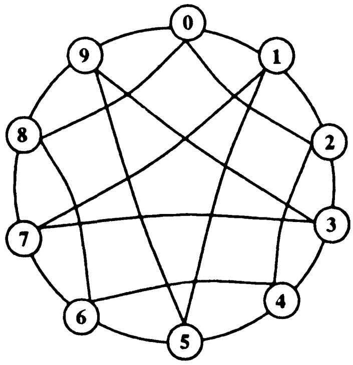


FIG. 2. *A degree*-4 CR ($n = 10, q = 2$).

degree-4 graphs arises from a subgroup, the Borel subgroup $\mathbf{BL}_2(\mathbf{Z}_p)$, of the general linear 2×2 matrices $\mathbf{GL}_2(\mathbf{Z}_p)$. The definition of the Borel subgroup is as follows.

DEFINITION 3. If $\mathbf{V}$ is a Borel subgroup, $\mathbf{BL}_2(\mathbf{Z}_p)$, of $\mathbf{GL}_2(\mathbf{Z}_p)$, then

$$\mathbf{V} = \left\{ \begin{pmatrix} x & y \\ 0 & 1 \end{pmatrix} : x = a^t \ (\mathrm{mod}\, p),\ y \in \mathbf{Z}_p,\ t \in \mathbf{Z}_k \right\},$$

where a is a fixed parameter $\in \mathbf{Z}_p \backslash \{0, 1\}$, p is prime, and k is the order of a. That is, $a^k = 1 \ (\mathrm{mod}\, p)$, and k is a factor of $p - 1$.

Thus, the nodes of Borel Cayley graphs are 2×2 matrices that satisfy the definition of a Borel subgroup, and modular matrix multiplication is chosen as the group operation $*$. Note that the variables of a Borel matrix are $t \in \mathbf{Z}_k$ and $y \in \mathbf{Z}_p$. In other words, there are $n = | \mathbf{V} | = p \times k$ nodes. By choosing specific generators, Chudnovsky, Chudnovsky, and Denneau [1] constructed the densest, nonrandom ($\delta = 4, D$) graphs known for $D = 7, \ldots, 13$ from Borel Cayley graphs (Table 1). In a separate research effort, Dinneen [20] and Campbell et al. [21] have constructed small diameter symmetric networks from Cayley graphs formed by linear groups. Interestingly, for the cases of $\delta = 4, D = 7, \ldots, 13$, these graphs have the same number of nodes as the Borel Cayley graphs in Table 1. Our investigation [22] showed that the Borel group can be formulated as a special case of the linear group described in [20].

TABLE 1
Comparisons of degree-4 graphs.

Diameter	Borel Cayley graphs	Moore bound	Known graphs (1987)
7	1,081	4,371	856
8	2,943	13,119	1,872
9	7,439	39,363	4,352
10	15,657	118,095	13,056
11	41,831	354,291	–
12	82,901	1,062,879	–
13	140,607	3,118,643	–

The *Moore bound* shown in Table 1 is an upper bound for the number of nodes in a degree-4 graph with diameter D. By arranging the nodes of a graph as a tree, the Moore bound shows that

$$n \leq 1 + \delta + \delta(\delta - 1) + \cdots + \delta(\delta - 1)^{D-1} = \frac{\delta(\delta - 1)^D - 2}{\delta - 2}.$$

Graphs attaining this Moore bound are called *Moore graphs* and are the densest possible for that degree and diameter. However, Moore graphs have been proved to be nonexistent except for some trivial cases. Specfically, these include complete graphs ($D = 1$) and rings ($\delta = 2$). Otherwise, it has been shown that Moore graphs exist only for diameter equals 2, degree equals 3, the Peterson graph, or diameter equals 2, degree equals 7, the Hoffman–Singleton graph, and possibly for diameter equals 2, degree 57 [12]. Given this general impossibility of constructing Moore graphs, there has been a long-standing search to find the densest regular graphs of a given degree and diameter. It is also worth noting that the Borel Cayley graph discovered by Chudnovsky, Chudnovsky, and Denneau [1] with $D = 11$, $\delta = 4$ has $n = 38,764$. In our research, we have discovered yet another denser Borel Cayley graph with $n = 41,831$ for $D = 11$, $\delta = 4$.

However, useful representations of Borel Cayley graphs are a challenge. These graphs are defined over a group of matrices, which lack a simple ordering that is very helpful in the development of efficient routing schemes. Furthermore, in this original matrix definition, there is no concise description of connections. Adjacent nodes can be identified only through modular matrix multiplications. The problem of finding an optimal path between nonadjacent nodes is not trivial. In an earlier report, we proved that all Cayley graphs can be represented by GCR [2]. This GCR representation is useful for routing because nodes are defined in the integer domain and there is a systematic description of connections. Different time and space efficient routing algorithms are devised for Borel Cayley graphs as a result of their GCR representations [16]–[18].

We restate this proposition as follows.

PROPOSITION 1. *For any finite Cayley graph* $\mathbf{C}$ *with vertex set* $\mathbf{V}$ *and any* $T \in \mathbf{V}$ *such that* $T^m = I$, *there exists a GCR representation of* $\mathbf{C}$ *with divisor* $q = n/m$, *where* $n = | \mathbf{V} |$.

The proof of this proposition is included in [2] and not repeated here. In the course of proving this proposition, we have constructed a step-by-step algorithm to transform any Cayley graph into a GCR. This algorithm is summarized in Table 2. The element T is referred to as the transform element, and it can be any element in the vertex set. In other words, this transformation is not unique. In the next section, we show that, by choosing a specific transform element **T** and class representing elements a_i (Table 2), all degree-4 Borel Cayley graphs have CR representations.

TABLE 2
An algorithm to generate a GCR representation.

To generate a GCR with divisor q,
choose an element T in $\mathbf{V}$ where $T^m = I$ and $m = n/q$.
For any element a in $\mathbf{V}$, define $\mathbf{N}(a)$ as
$\mathbf{N}(a) = \{x \in \mathbf{V}\colon x = T^s a\} \quad s = 0, 1, \dots, (m-1)$

1. Construct $\mathbf{N}(a_i)$, $i = 0, \dots, (q-1)$ by picking arbitrary $a_i \in V \setminus \mathbf{N}(a_0) \setminus \dots \setminus \mathbf{N}(a_{i-1})$; $a_0, a_1, \dots, a_{q-1}$ are the representative elements in partitions $\mathbf{N}(a_0), \mathbf{N}(a_1), \dots, \mathbf{N}(a_{q-1})$.
2. Associate $a_i \to i$, $\quad i = 0, 1, \dots, (q-1)$ and $T^s a_i \to i + sq$, $s = 0, \dots, (m-1)$. This forms the q classes of the GCR.
3. Obtain the connecting constant for each class:
   For each class i of the GCR, find the neighboring nodes of the representing element, a_i.
   e.g., if a_i is adjacent to a node, $b = T^s a_j$,
   then any node w in class i is connected to $w + j + sq - i$.

In [2] we also provided a sufficient condition for a Cayley graph to have a CR representation. For convenience, we restate this proposition as follows.

PROPOSITION 2. *Let A, B be two distinct generators of a finite Cayley graph* **C**. *Assume that $A \neq B^{-1}$, $A^q = I$, and $m = n/q$. If $(AB)^m = I$ or $(A^{-1}B)^m = I$, then CR representations with divisor q exist. The transform element $T = AB$ or $A^{-1}B$ and the representing element of class 0 is I and of class i is A^i, $i = 1, \ldots, q-1$.*

3. CR representations. In this section, we show that all connected degree-4 Borel Cayley graphs have CR representations. During our studies of Borel Cayley graphs, we discovered some useful properties of the subgroup. These properties and their proofs are presented here. Throughout this section, we assume a *connected* degree-4 Borel Cayley graph with n nodes and parameters a, p, and k, as defined in Definition 3, and generators $\mathbf{A}$, $\mathbf{B}$, $\mathbf{A}^{-1}$, and $\mathbf{B}^{-1}$, where

$$\mathbf{A} = \begin{pmatrix} a^{t_1} & y_1 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{B} = \begin{pmatrix} a^{t_2} & y_2 \\ 0 & 1 \end{pmatrix}.$$

Furthermore, the order of $\mathbf{A}$ and $\mathbf{B}$ are k_1 and k_2, where $k_1, k_2 \in \mathbf{Z}_k$.

PROPOSITION 3. *Let*

$$\mathbf{X} = \begin{pmatrix} a^t & y \\ 0 & 1 \end{pmatrix} \in \mathbf{BL}_2(\mathbf{Z}_p)$$

and $\mathbf{X} \neq \mathbf{I}$ be a Borel matrix, as defined in Definition 3. If q is the order of $\mathbf{X}$, i.e., q is the smallest positive integer such that $\mathbf{X}^q = \mathbf{I}$, then

$$q = \begin{cases} \dfrac{\mathrm{LCM}(t,k)}{t} & \text{if } t \neq 0, \\ p & \text{if } t = 0, \end{cases}$$

where LCM(t,k) *denotes the least common multiple of t and k.*

Proof. We have

$$\mathbf{X}^q = \begin{pmatrix} a^{qt} & (a^{(q-1)t} + a^{(q-2)t} + \cdots + a^0)y \\ 0 & 1 \end{pmatrix},$$

$$\mathbf{X}^q = \mathbf{I} \Rightarrow \begin{cases} qt = 0 \pmod{k} \text{ and } (a^{(q-1)t} + a^{(q-2)t} + \ldots + a^0) = 0 \pmod{p} \\ \text{or} \\ qt = 0 \pmod{k} \text{ and } y = 0. \end{cases}$$

Case 1. $t \neq 0$. In this case,

$$(a^{(q-1)t} + a^{(q-2)t} + \cdots + a^0) = 0 \pmod{p} \Rightarrow qt = 0 \pmod{k}$$

because

$$\begin{aligned} & (a^{(q-1)t} + a^{(q-2)t} + \cdots + a^0) = 0 && \pmod{p} \\ \Rightarrow\ & (a^t - 1)(a^{(q-1)t} + a^{(q-2)t} + \cdots + a^0) = 0 && \pmod{p} \\ \Rightarrow\ & a^{qt} - 1 = 0 && \pmod{p} \\ \Rightarrow\ & qt = 0 && \pmod{k}. \end{aligned}$$

Hence

$$\begin{aligned} \mathbf{X}^q = \mathbf{I} \ &\Rightarrow qt = 0 \pmod{k} \\ &\Rightarrow q = \frac{\mathrm{LCM}\,(t,k)}{t}. \end{aligned}$$

Case 2. $t = 0$. In this case, $y \neq 0$; otherwise $\mathbf{X} = \mathbf{I}$. Hence

$$\mathbf{X}^q = \mathbf{I} \Rightarrow \begin{cases} qt = 0 & \pmod{k} \quad \text{and} \\ (a^{(q-1)t} + a^{(q-2)t} + \cdots + a^0) = 0 & \pmod{p}, \end{cases}$$

$$\begin{aligned} & (a^{(q-1)t} + a^{(q-2)t} + \cdots + a^0) = 0 && \pmod{p} \\ \Rightarrow\ & q = 0 && \pmod{p} \\ \Rightarrow\ & q = p. \quad \square \end{aligned}$$

PROPOSITION 4. *We have*

$$\mathbf{B} \neq \mathbf{A}^m \quad \textit{for any integer } m \in \mathbf{Z}_k.$$

Proof. If $\mathbf{B} = \mathbf{A}^m$, the generators of the graph are $\mathbf{A}, \mathbf{A}^m, \mathbf{A}^{k_1-1}, \mathbf{A}^{k_1-m}$, which implies that all nodes in the graph can be written as multiples of $\mathbf{A}$. This means that some nodes in the graph are not connected because there are at most $k_1 < n$ different multiples of $\mathbf{A}$. $\square$

PROPOSITION 5. *We have*

$$(1) \qquad (1 - a^{t_2})y_1 = (1 - a^{t_1})y_2 \pmod{p} \Leftrightarrow \mathbf{AB} = \mathbf{BA}.$$

The proof of this proposition is a straightforward substitution and is omitted.

PROPOSITION 6. *If* $\mathbf{AB} = \mathbf{BA}$, *then, for any path* $\mathbf{X}$ *with* m_1 *as the net number of generator* $\mathbf{A}$ *and with* m_2 *as the net number of generator* $\mathbf{B}$,

$$\mathbf{X} = \mathbf{A}^{m_1}\, \mathbf{B}^{m_2},$$

where

$$\begin{aligned} m_1 &= \textit{number of } \mathbf{A} - \textit{number of } \mathbf{A}^{-1} && (\bmod k_1), \\ m_2 &= \textit{number of } \mathbf{B} - \textit{number of } \mathbf{B}^{-1} && (\bmod k_2). \end{aligned}$$

Proof. Since $\mathbf{A}^{-1} = \mathbf{A}^{k_1-1}$ and $\mathbf{B}^{-1} = \mathbf{B}^{k_2-1}$, it suffices to consider paths composed of generators $\mathbf{A}$ and $\mathbf{B}$ only. We use mathematical induction to prove this proposition.

If $m_1 = m_2 = 1$, $\mathbf{AB} = \mathbf{BA}$. Obviously, the proposition also holds for $m_1 = 1, m_2 = 0$ and $m_1 = 0, m_2 = 1$. Hence the proposition is true for $m_1 \leq 1$ and $m_2 \leq 1$.

Assume the proposition holds for $m_1 \leq m_1'$ and $m_2 \leq m_2'$ for some integers $m_1' \in \mathbf{Z}_{k_1}$ and $m_2' \in \mathbf{Z}_{k_2}$.

Consider $m_1 = m_1' + 1$ and $m_2 = m_2'$. There exists an integer $l = 0, \ldots, m_2'$ such that

$$\begin{aligned} \mathbf{X} &= \underbrace{\cdots\cdots\cdots}_{m_1' A,\ (m_2'-l)B} \mathbf{AB}^l \\ &= \mathbf{A}^{m_1'}\mathbf{B}^{m_2'-l}\mathbf{AB}^l \qquad \text{(by assumption).} \end{aligned}$$

Furthermore, $\mathbf{B}^{m_2'-l}\mathbf{A} = \mathbf{AB}^{m_2'-l}$ by assumption. Hence

$$\mathbf{X} = \mathbf{A}^{m_1'+1}\, \mathbf{B}^{m_2'}.$$

Similarly, the proposition is true for $m_1 = m_1' + 1$ and $m_2 = m_2' + 1$. By the principle of mathematical induction, the proposition is true for all $m_1 \in \mathbf{Z}_{k_1}$ and $m_2 \in \mathbf{Z}_{k_2}$. □

Based on Propositions 5 and 6, we have three useful corollaries.

COROLLARY 1. *If* $\mathbf{AB} = \mathbf{BA}$, *then the graph is disconnected.*

Proof. If $\mathbf{AB} = \mathbf{BA}$, from Proposition 6, an element $\mathbf{X}$ in the graph is represented as

$$\mathbf{X} = \mathbf{I} \quad \text{or} \quad \mathbf{A}^{m_1} \quad \text{or} \quad \mathbf{B}^{m_2} \quad \text{or} \quad \mathbf{A}^{m_1}\mathbf{B}^{m_2},$$

where $m_1 = 1, \dots, k_1 - 1, m_2 = 1, \dots, k_2 - 1$. In other words, there are at most

$$1 + (k_1 - 1) + (k_2 - 1) + (k_1 - 1)(k_2 - 1) \leq 1 + 2(k-1) + (k-1)^2 = k^2$$

different $\mathbf{X}$. Since k is a factor of $p - 1$ (Definition 3),

$$n = p \times k > k^2,$$

which implies that some nodes of the graph cannot be generated by $\mathbf{A}$, $\mathbf{B}$, and hence the graph is disconnected. □

COROLLARY 2. *The values of* t_1 *and* t_2 *cannot be both zero.*

Proof. We have

$$\begin{aligned} & t_1 = t_2 = 0 \\ \Rightarrow\quad & (1 - a^{t_2})y_1 = (1 - a^{t_1})y_2 \\ \Leftrightarrow\quad & \mathbf{AB} = \mathbf{BA} \qquad \text{(by (1))}, \end{aligned}$$

which implies the graph is disconnected by Corollary 1. □

COROLLARY 3. *The values of* y_1 *and* y_2 *cannot be both zero.*

Proof. We have

$$\begin{aligned} & y_1 = y_2 = 0 \\ \Rightarrow\quad & (1 - a^{t_2})y_1 = (1 - a^{t_1})y_2 \\ \Leftrightarrow\quad & \mathbf{AB} = \mathbf{BA} \qquad \text{(by (1))}, \end{aligned}$$

which implies that the graph is disconnected by Corollary 1. □

PROPOSITION 7. *For any path* $\mathbf{X}$ *composed of generators* $\mathbf{A}$, $\mathbf{B}$, $\mathbf{A}^{-1}$, *and* $\mathbf{B}^{-1}$,

$$\mathbf{X} = \begin{pmatrix} a^{\langle it_1 + jt_2 \rangle_k} & \langle gy_1 + hy_2 \rangle_p \\ 0 & 1 \end{pmatrix}$$

$$\Rightarrow \quad (1 - a^{t_1})g + (1 - a^{t_2})h = 1 - a^{it_1 + jt_2} \qquad (\operatorname{mod} p),$$

where $\langle x \rangle_k$ *denotes* x mod k.

Proof. We prove this proposition by induction on the length of the path. For the single step path $\mathbf{X} = \mathbf{A}$,

$$\begin{aligned} i &= 1, & j &= 0, \\ g &= 1, & h &= 0, \\ (1 - a^{t_1})g &= 1 - a^{t_1}. \end{aligned}$$

Therefore, the proposition holds. Similarly, the proposition holds for $\mathbf{X} = \mathbf{B}, \mathbf{A}^{-1}, \mathbf{B}^{-1}$.

Assume the proposition holds for some path $\mathbf{X}'$. That is,

$$\mathbf{X}' = \begin{pmatrix} a^{\langle i't_1+j't_2\rangle_k} & \langle g'y_1 + h'y_2\rangle_p \\ 0 & 1 \end{pmatrix}$$

and

$$(1-a^{t_1})g' + (1-a^{t_2})h' = 1 - a^{i't_1+j't_2} \qquad (\bmod p).$$

Consider the path

$$\mathbf{X}'\mathbf{A} = \begin{pmatrix} a^{\langle (i'+1)t_1+j't_2\rangle_k} & \langle (g' + a^{i't_1+j't_2})y_1 + h'y_2\rangle_p \\ 0 & 1 \end{pmatrix};$$

$$\begin{aligned} &(1-a^{t_1})(g' + a^{i't_1+j't_2}) + (1-a^{t_2})h' \qquad (\operatorname{mod} p) \\ &= (1-a^{t_1})g' + (1-a^{t_2})h' + (1-a^{t_1})a^{i't_1+j't_2} \qquad (\operatorname{mod} p) \\ &= 1 - a^{i't_1+j't_2} + (1-a^{t_1})a^{i't_1+j't_2} \qquad (\operatorname{mod} p) \quad \text{(by assumption)} \\ &= 1 - a^{(i'+1)t_1+j't_2} \qquad (\operatorname{mod} p). \end{aligned}$$

That is, the proposition holds for $\mathbf{X}'\mathbf{A}$. Similarly, the proposition is true for $\mathbf{X}'\mathbf{A}^{-1}$, $\mathbf{X}'\mathbf{B}, \mathbf{X}'\mathbf{B}^{-1}$. By the principle of mathematical induction, the proposition is true for any path $\mathbf{X}$. □

PROPOSITION 8. *For any paths* $\mathbf{X}$, $\mathbf{Y}$, *composed of generators* $\mathbf{A}$, $\mathbf{B}$, $\mathbf{A}^{-1}$, *and* $\mathbf{B}^{-1}$, *let*

$$\mathbf{X} = \begin{pmatrix} a^{\langle it_1+jt_2\rangle_k} & \langle gy_1 + hy_2\rangle_p \\ 0 & 1 \end{pmatrix} \quad \textit{and} \quad \mathbf{Y} = \begin{pmatrix} a^{\langle i't_1+j't_2\rangle_k} & \langle g'y_1 + h'y_2\rangle_p \\ 0 & 1 \end{pmatrix},$$

where $\langle x\rangle_p$ *denotes* x (mod p). *Then*

$$\begin{aligned} &\mathbf{X} = \mathbf{Y} \\ \Leftrightarrow\quad & it_1 + jt_2 = i't_1 + j't_2 \qquad (\operatorname{mod} k) \end{aligned}$$

and

$$\begin{cases} g = g' \textit{and } h = h' \qquad (\operatorname{mod} p) \quad \textit{or} \\ (1-a^{t_2})y_1 = (1-a^{t_1})y_2 \qquad (\operatorname{mod} p). \end{cases}$$

Proof. Since

$$\begin{aligned} \mathbf{X} &= \begin{pmatrix} a^{\langle it_1+jt_2\rangle_k} & \langle gy_1 + hy_2\rangle_p \\ 0 & 1 \end{pmatrix}, \\ \mathbf{Y} &= \begin{pmatrix} a^{\langle i't_1+j't_2\rangle_k} & \langle g'y_1 + h'y_2\rangle_p \\ 0 & 1 \end{pmatrix}, \end{aligned}$$

from Proposition 7,

$$\begin{aligned} (1-a^{t_1})g + (1-a^{t_2})h &= 1 - a^{it_1+jt_2} \qquad (\operatorname{mod} p), \\ (1-a^{t_1})g' + (1-a^{t_2})h' &= 1 - a^{i't_1+j't_2} \qquad (\operatorname{mod} p) \end{aligned} \tag{2}$$

(⇒)

$$
\begin{array}{llll}
 & \mathbf{X} = \mathbf{Y} & & \\
\Rightarrow & it_1 + jt_2 = i't_1 + j't_2 & \pmod{k} & \\
(3) \Rightarrow & (1 - a^{t_1})g + (1 - a^{t_2})h = (1 - a^{t_1})g' + (1 - a^{t_2})h' & \pmod{p} & \text{from (2)} \\
\Rightarrow & (1 - a^{t_1})(g - g') = (1 - a^{t_2})(h' - h) & \pmod{p} &
\end{array}
$$

Also,

$$
\begin{array}{lll}
 & \mathbf{X} = \mathbf{Y} & \\
(4) \quad \Rightarrow & gy_1 + hy_2 = g'y_1 + h'y_2 & \pmod{p} \\
\Rightarrow & (g - g')y_1 = (h' - h)y_2 & \pmod{p}.
\end{array}
$$

From (3) and (4), we have

$$
\begin{array}{ll}
(1 - a^{t_2})y_1 = (1 - a^{t_1})y_2 & \pmod{p} \quad \text{or} \\
g = g' \quad \text{and} \quad h = h' & \pmod{p}.
\end{array}
$$

(⇐) Obviously,

$$
\left.\begin{array}{ll}
it_1 + jt_2 = i't_1 + j't_2 & \pmod{k} \\
g = g' \quad \text{and} \quad h = h' & \pmod{p}
\end{array}\right. \Rightarrow \mathbf{X} = \mathbf{Y}.
$$

On the other hand, from (2),

$$
\begin{array}{lll}
 & it_1 + jt_2 = i't_1 + j't_2 & \pmod{k} \\
\Rightarrow & (1 - a^{t_1})g + (1 - a^{t_2})h = (1 - a^{t_1})g' + (1 - a^{t_2})h' & \pmod{p}.
\end{array}
$$

Since $(1 - a^{t_2})y_1 = (1 - a^{t_1})y_2 \pmod{p}$ and from Corollaries 2 and 3, t_1, t_2 and y_1, y_2 are not both zero, we have

$$
\begin{array}{lll}
 & (1 - a^{t_2})y_1(1 - a^{t_1})g + (1 - a^{t_1})y_2(1 - a^{t_2})h & \\
= & (1 - a^{t_2})y_1(1 - a^{t_1})g' + (1 - a^{t_1})y_2(1 - a^{t_2})h' & \pmod{p} \\
\Rightarrow & gy_1 + hy_2 = g'y_1 + h'y_2 \qquad \pmod{p} & \\
\Rightarrow & \mathbf{X} = \mathbf{Y}. \qquad \square &
\end{array}
$$

COROLLARY 4. *Let* $\mathbf{X}$, $\mathbf{Y}$ *be defined as in Proposition* 8. *For a connected degree*-4 *Borel Cayley graph*,

$$
\mathbf{X} = \mathbf{Y} \Leftrightarrow \left\{ \begin{array}{ll}
it_1 + jt_2 = i't_1 + j't_2 & \pmod{k} \quad \textit{and} \\
g = g' \textit{ and } h = h' & \pmod{p}.
\end{array} \right.
$$

Proof. From Proposition 8,

$$
\begin{array}{lll}
 & \mathbf{X} = \mathbf{Y} & \\
\Leftrightarrow & it_1 + jt_2 = i't_1 + j't_2 & \pmod{k}
\end{array}
$$

and

$$
\begin{array}{ll}
g = g' \text{ and } h = h' & \pmod{p} \quad \text{or} \\
(1 - a^{t_2})y_1 = (1 - a^{t_1})y_2 & \pmod{p}.
\end{array}
$$

However, from Proposition 5 and Corollary 1,

$$(1-a^{t_2})y_1 = (1-a^{t_1})y_2 \pmod{p} \Leftrightarrow \mathbf{AB}=\mathbf{BA} \;\Rightarrow\; \text{the graph is disconnected.}$$

Hence, for a connected degree-4 Borel Cayley graph,

$$\mathbf{X}=\mathbf{Y} \Leftrightarrow \begin{cases} it_1+jt_2=i't_1+j't_2 & \pmod{k} \quad \text{and} \\ g=g' \text{ and } h=h' & \pmod{p}. \end{cases} \qquad \Box$$

With the above propositions and corollaries, we are now ready to state the main result of this paper.

PROPOSITION 9. *All connected degree-4 Borel Cayley graphs have CR representations.*

Proof. We consider three cases. In the first two cases, the idea of the proof is to construct a specific GCR with $q=k$ classes. We choose the transform element

$$\mathbf{T}=\begin{pmatrix}1 & y'\\ 0 & 1\end{pmatrix}, \qquad y'\neq 0,$$

and the representing element of class j to be

$$a_j=\begin{pmatrix}a^i & \bar{y}_j\\ 0 & 1\end{pmatrix},$$

where $i,\ j=0,\ldots,k-1$, $\bar{y}_j\in\mathbf{Z}_p$ and no two classes have the same value for i. These choices ensure that any Borel matrix element

$$\begin{pmatrix}a^t & y\\ 0 & 1\end{pmatrix}$$

can be classified by the value t. Furthermore, if we can choose the class representing elements such that

$$a_0\sim a_1\sim\cdots\sim a_{k-1}\sim \mathbf{T}*a_0$$

(the symbol $\sim$ denotes adjacency), we have a CR representation.

For the third case, we prove that the sufficient condition in Proposition 2 is satisfied and hence a CR representation.

Case 1. $t_1,t_2\neq 0$ and either $(t_1,k)=1$ or $(t_2,k)=1$. Without loss of generality, we assume that $(t_1,k)=1$, (t_1 and k are relatively prime). In other words, multiples of t_1 (mod k) span the set $\{1,\ldots,(k-1)\}$. Since $t_2\in\{1,\ldots,(k-1)\}$, we have

$$m\,t_1=t_2 \pmod{k} \qquad \text{for some } m=1,\ldots,(k-1).$$

We consider a GCR with

$$\mathbf{T}=\mathbf{BA}^{k-1-m}\mathbf{B}(\mathbf{A}^{-1})^{m-1}. \tag{5}$$

Claim. It holds that

$$\mathbf{T}=\begin{pmatrix}1 & y'\\ 0 & 1\end{pmatrix}$$

for some $y'\in\mathbf{Z}_p$ and $y'\neq 0$.

Proof. Note that the superscript t of the first element of any matrix

$$\begin{pmatrix} a^t & y \\ 0 & 1 \end{pmatrix}$$

can be found by counting the net number of generators $\mathbf{A}$ and $\mathbf{B}$ that composed the matrix. As an example, for matrix $\mathbf{X} = \mathbf{AB}$, its t value is $t_1 + t_2 \pmod{k}$. Counting the net number of generators $\mathbf{A}$ and $\mathbf{B}$ in (5),

$$t_2 + (k-1-m)t_1 + t_2 + (m-1)(k-t_1) = 0 \qquad (\text{mod } k).$$

Hence the first element of $\mathbf{T}$ is 1. We proceed to prove that $\mathbf{T} \neq \mathbf{I}$. Since $mt_1 = t_2 \pmod{k}$, we let $\mathbf{B} = \mathbf{HA}^m$, where $\mathbf{H} = \left(\begin{smallmatrix} 1 & z \\ 0 & 1 \end{smallmatrix}\right)$ for some $z \in Z_p$ and $z \neq 0$ because $\mathbf{B} \neq \mathbf{A}^m$ as stated in Proposition 4,

$$\begin{array}{lrl}
\mathbf{T} = \mathbf{I} \Rightarrow & \mathbf{BA}^{k-1-m}\mathbf{B} = \mathbf{A}^{m-1} & \\
\Rightarrow & \mathbf{HA}^m\mathbf{A}^{k-1-m}\mathbf{HA}^m = \mathbf{A}^{m-1} & \\
\Rightarrow & \mathbf{HA}^{-1}\mathbf{H} = \mathbf{A}^{-1} & \\
\Rightarrow & \begin{pmatrix} 1 & z \\ 0 & 1 \end{pmatrix}\begin{pmatrix} a^{k-t_1} & z' \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & z \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} a^{k-t_1} & z' \\ 0 & 1 \end{pmatrix}, & z' = \langle -a^{-t_1}y_1 \rangle_p \\
\Rightarrow & \begin{pmatrix} a^{k-t_1} & (a^{k-t_1}+1)z + z' \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} a^{k-t_1} & z' \\ 0 & 1 \end{pmatrix} & \\
\Rightarrow & (a^{k-t_1}+1)z = 0 & (\text{mod } p) \\
\Rightarrow & a^{k-t_1} = -1 & (\text{mod } p) \\
\Rightarrow & 2(k-t_1) = 0 & (\text{mod } k) \\
\Rightarrow & (t_1, k) \neq 1 & (\text{a contradiction}).
\end{array}$$

Hence $\mathbf{T} \neq \mathbf{I}$. According to Proposition 3,

$$\mathbf{T} = \begin{pmatrix} 1 & y' \\ 0 & 1 \end{pmatrix} \Rightarrow \mathbf{T}^p = \mathbf{I}.$$

We can construct a GCR with divisor $q = k$ and choose the representing elements according to (5). That is, the representing element of class j, a_j is the composition of the first j elements in (5). Specifically,

$$\begin{aligned}
a_0 &= \mathbf{I}; \\
a_1 &= \mathbf{B}; \\
a_2 &= \mathbf{BA}; \\
&\vdots \\
a_{q-m} &= \mathbf{B}\ \mathbf{A}^{k-1-m}; \\
a_{q-m+1} &= \mathbf{B}\ \mathbf{A}^{k-1-m}\ \mathbf{B}; \\
a_{q-m+2} &= \mathbf{B}\ \mathbf{A}^{k-1-m}\ \mathbf{B}\ \mathbf{A}^{-1}; \\
&\vdots \\
a_{q-1} &= \mathbf{B}\ \mathbf{A}^{k-1-m}\ \mathbf{B}\ (\mathbf{A}^{-1})^{m-2}.
\end{aligned}$$

Note that

$$\begin{aligned}
a_0 &\sim a_1 = a_o * \mathbf{B};\\
a_1 &\sim a_2 = a_1 * \mathbf{A};\\
&\vdots\\
a_{q-2} &\sim a_{q-1} = a_{q-2} * \mathbf{A}^{-1};\\
a_{q-1} &\sim \mathbf{T} * a_0 = \mathbf{T} = a_{q-1} * \mathbf{A}^{-1},
\end{aligned}$$

where the symbol $\sim$ denotes adjacency. Furthermore, given

$$a_j = \begin{pmatrix} a^i & \bar{y}_j \\ 0 & 1 \end{pmatrix},$$

the i values for representing elements $a_0, \ldots, a_{q-1}$ are

$$0, m\, t_1,\ (m+1)\, t_1,\ \ldots,\ (k-1)\, t_1,\ (m-1)\, t_1,\ (m-2)\, t_1,\ \ldots,\ t_1,$$

where $t_2 = mt_1 \pmod{k}$. Since $(t_1, k) = 1$, these values of i span the entire set of $\{0, \ldots, k-1\}$. In other words, we have a CR representation.

An alternate way to construct a CR representation is to choose

$$\mathbf{T} = \mathbf{B}^{-1}(\mathbf{A}^{-1})^{k-1-m}\mathbf{B}^{-1}\mathbf{A}^{m-1}.$$

In this case,

$$\begin{aligned}
a_0 &= \mathbf{I};\\
a_1 &= \mathbf{B}^{-1};\\
a_2 &= \mathbf{B}^{-1}\ \mathbf{A}^{-1};\\
&\vdots\\
a_{q-m} &= \mathbf{B}^{-1}\ (\mathbf{A}^{-1})^{k-1-m};\\
a_{q-m+1} &= \mathbf{B}^{-1}\ (\mathbf{A}^{-1})^{k-1-m}\ \mathbf{B}^{-1};\\
a_{q-m+2} &= \mathbf{B}^{-1}\ (\mathbf{A}^{-1})^{k-1-m}\ \mathbf{B}^{-1}\ \mathbf{A};\\
&\vdots\\
a_{q-1} &= \mathbf{B}^{-1}\ (\mathbf{A}^{-1})^{k-1-m}\ \mathbf{B}^{-1}\ \mathbf{A}^{m-2}.
\end{aligned}$$

The proof of this construction is similar to the one shown above and is not repeated.

Case 2. $t_1, t_2 \neq 0$ and $(t_1, k) \neq 1$ and $(t_2, k) \neq 1$.

In this case, $(t_1, t_2) = 1$ (t_1 and t_2 are relatively prime) because otherwise the graph is disconnected. Furthermore, $t_1 k_1 = t_2 k_2 = k$. Since t_1 and t_2 are relatively prime, we can divide the set $\{0, \ldots, k-1\}$ into t_1 distinct subsets each with k_1 elements as follows:

$$\begin{gathered}
\{0, t_1, \ldots, (k_1 - 1)t_1\},\\
\{t_2, t_2 + t_1, \ldots, t_2 + (k_1 - 1)t_1\},\\
\vdots \qquad\qquad \vdots \qquad\qquad \vdots\\
\{(t_1 - 1)t_2, (t_1 - 1)t_2 + t_1, \ldots, (t_1 - 1)t_2 + (k_1 - 1)t_1\}.
\end{gathered}$$

If each number in the above subsets represents the superscript i of a class representing element

$$a_j = \begin{pmatrix} a^i & \bar{y}_j \\ 0 & 1 \end{pmatrix},$$

where y_j is an integer in z_p, the corresponding class representing element within one subset (on the same row) can be cyclically connected by generator $\mathbf{A}$, and those on the same column can be connected, but not cyclically, by generator $\mathbf{B}$. As discussed at the outset of this proof, the idea is to construct a specific GCR by choosing the transform element

$$\mathbf{T} = \begin{pmatrix} 1 & y' \\ 0 & 1 \end{pmatrix}, \qquad y' \neq 0,$$

and the representing element of class j,

$$a_j = \begin{pmatrix} a^i & \bar{y}_j \\ 0 & 1 \end{pmatrix}$$

such that the superscript i spans the set $\{0, 1, \ldots, k-1\}$ and $a_0 \sim a_1 \sim \cdots \sim a_{k-1} \sim \mathbf{T} * a_0$, where the symbol $\sim$ denotes adjacency.

In this case, the problem of finding such choices for $\mathbf{T}$ and class representing elements $a_0, \ldots, a_{q-1}$ is the same as finding a Hamiltonian cycle to "march through" the k numbers in the subsets, starting from 0. There are two ways of constructing this Hamiltonian cycle, depending on whether t_1 is odd or even. Figures 3 and 4 show a Hamiltonian cycle for $t_1 = 2, 3$. In these cases, $\mathbf{T} = \mathbf{B}\mathbf{A}^{k_1-1}\mathbf{B}^{-1}(\mathbf{A}^{-1})^{k_1-1}$ and $\mathbf{T} = \mathbf{B}^{t_1-1}\mathbf{A}^{k_1-1}\{\mathbf{B}^{-1}(\mathbf{A}^{-1})^{k_1-2}\mathbf{B}^{-1}\mathbf{A}^{k_1-2}\}\mathbf{A}$. The mathematical formulations of these two subcases are as follows.

Subcase 1. t_1 is odd. We define the integer $d = (t_1 - 1)/2$. In this case, we consider a GCR with

$$\mathbf{T} = \mathbf{B}^{t_1-1}\mathbf{A}^{k_1-1}\{\mathbf{B}^{-1}(\mathbf{A}^{-1})^{k_1-2}\mathbf{B}^{-1}\mathbf{A}^{k_1-2}\}^d\mathbf{A}. \tag{6}$$

Claim. It holds that

$$\mathbf{T} = \begin{pmatrix} 1 & y' \\ 0 & 1 \end{pmatrix}$$

for some $y' \in \mathbf{Z}_p$ and $y' \neq 0$.

Proof. By counting the net numbers of $\mathbf{A}$ and $\mathbf{B}$ in (6),

$$(t_1 - 1)t_2 - t_1 + \frac{t_1 - 1}{2}(-t_2 + 2t_1 - t_2 - 2t_1) + t_1 = 0 \qquad (\text{mod } k),$$

the first element of $\mathbf{T}$ is 1. We proceed to prove that $\mathbf{T} \neq \mathbf{I}$. Let

$$\mathbf{T} = \mathbf{I},$$

$$\Rightarrow \quad \begin{pmatrix} 1 & gy_1 + hy_2 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0y_1 + 0y_2 \\ 0 & 1 \end{pmatrix}, \tag{7}$$

where

$$\begin{tikzcd}
0 \arrow[d, "\mathbf{B}"'] & t_1 \arrow[l, "\mathbf{A}^{-1}"'] & \cdots \arrow[l, "\mathbf{A}^{-1}"'] & (k_1-1)t_1 \arrow[l, "\mathbf{A}^{-1}"'] \\
t_2 \arrow[r, "\mathbf{A}"'] & t_2+t_1 \arrow[r, "\mathbf{A}"'] & \cdots \arrow[r, "\mathbf{A}"'] & t_2+(k_1-1)t_1 \arrow[u, "\mathbf{B}^{-1}"']
\end{tikzcd}$$

FIG. 3. *A Hamiltonian cycle for* $t_1 = 2$.

$$\begin{tikzcd}
0 \arrow[d, "\mathbf{B}"'] & t_1 \arrow[r, "\mathbf{A}"] & \cdots \arrow[r, "\mathbf{A}"] & (k_1-1)t_1 \arrow[r, "\mathbf{A}"] & 0 \\
t_2 \arrow[d, "\mathbf{B}"'] & t_2+t_1 \arrow[u, "\mathbf{B}^{-1}"'] & \cdots \arrow[l, "\mathbf{A}^{-1}"] & t_2+(k_1-1)t_1 \arrow[l, "\mathbf{A}^{-1}"] & \\
2t_2 \arrow[r, "\mathbf{A}"'] & 2t_2+t_1 \arrow[r, "\mathbf{A}"'] & \cdots \arrow[r, "\mathbf{A}"'] & 2t_2+(k_1-1)t_1 \arrow[u, "\mathbf{B}^{-1}"'] &
\end{tikzcd}$$

FIG. 4. *A Hamiltonian cycle for* $t_1 = 3$.

$$(8) \qquad g = -a^{2dt_2-t_1} + \sum_{i=1}^{d}\{a^{(2i-1)t_2-t_1} + a^{(2i-1)t_2} - a^{2(i-1)t_2} - a^{2(i-1)t_2-t_1}\} \\ +a^{-t_1} \pmod{p},$$

$$(9) \qquad h = \sum_{i=0}^{2d-1} a^{it_2} - \sum_{i=1}^{d}\{a^{(2i-1)t_2-t_1} + a^{2(i-1)t_2+t_1}\} \qquad \pmod{p}.$$

Equations (8) and (9) are obtained by observing that, from (6),

$$\begin{aligned}\mathbf{T} &= \mathbf{B}^{t_1-1}\mathbf{A}^{k_1-1}\{\mathbf{B}^{-1}(\mathbf{A}^{-1})^{k_1-2}\mathbf{B}^{-1}\mathbf{A}^{k_1-2}\}^d\mathbf{A} \\ &= \mathbf{B}^{2d}\mathbf{A}^{-1}\{\mathbf{B}^{-1}\mathbf{A}^{2}\mathbf{B}^{-1}\mathbf{A}^{-2}\}^d\mathbf{A},\end{aligned}$$

and, for any Borel matrix,

$$\begin{pmatrix} a^t & y \\ 0 & 1 \end{pmatrix},$$

$$\begin{pmatrix} a^t & y \\ 0 & 1 \end{pmatrix}\mathbf{A} = \begin{pmatrix} a^{t+t_1} & \langle y + a^t y_1\rangle_p \\ 0 & 1 \end{pmatrix};$$

$$\begin{pmatrix} a^t & y \\ 0 & 1 \end{pmatrix}\mathbf{A}^{-1} = \begin{pmatrix} a^{t-t_1} & \langle y - a^{t-t_1} y_1\rangle_p \\ 0 & 1 \end{pmatrix};$$

$$\begin{pmatrix} a^t & y \\ 0 & 1 \end{pmatrix}\mathbf{B} = \begin{pmatrix} a^{t+t_2} & \langle y + a^t y_2\rangle_p \\ 0 & 1 \end{pmatrix};$$

$$\begin{pmatrix} a^t & y \\ 0 & 1 \end{pmatrix}\mathbf{B}^{-1} = \begin{pmatrix} a^{t-t_2} & \langle y - a^{t-t_2} y_2\rangle_p \\ 0 & 1 \end{pmatrix}.$$

Hence

$$\mathbf{B}^{2d} = \begin{pmatrix} a^{2dt_2} & \sum\limits_{i=0}^{2d-1} a^{it_2} y_2 \\ 0 & 1 \end{pmatrix},$$

$$\mathbf{B}^{2d}\mathbf{A}^{-1} = \begin{pmatrix} a^{2dt_2-t_1} & \sum\limits_{i=0}^{2d-1} a^{it_2} y_2 - a^{2dt_2-t_1} y_1 \\ 0 & 1 \end{pmatrix},$$

$$\mathbf{B}^{2d}\mathbf{A}^{-1}\mathbf{B}^{-1} = \begin{pmatrix} a^{(2d-1)t_2-t_1} & \left(\sum\limits_{i=0}^{2d-1} a^{it_2} - a^{(2d-1)t_2-t_1}\right) y_2 - a^{2dt_2-t_1} y_1 \\ 0 & 1 \end{pmatrix},$$

$$\vdots$$

$$\mathbf{B}^{2d}\mathbf{A}^{-1}\{\mathbf{B}^{-1}\mathbf{A}^{2}\mathbf{B}^{-1}\mathbf{A}^{-2}\}^{d}\mathbf{A} = \begin{pmatrix} 1 & gy_1 + hy_2 \\ 0 & 1 \end{pmatrix},$$

where g and h are described by (8) and (9).

From (7) and Corollary 4, $g = h = 0 (\operatorname{mod} p)$. That is,

$$\begin{aligned} g &= 0 \pmod p \\ \Rightarrow a^{2dt_2-t_1} - a^{-t_1} &= \sum_{i=1}^{d} \left\{ a^{(2i-1)t_2-t_1} - a^{2(i-1)t_2-t_1} \right\} \\ &\quad + \sum_{i=1}^{d} \left\{ a^{(2i-1)t_2} - a^{2(i-1)t_2} \right\} \pmod p \\ &= (a^{t_2}-1)\sum_{i=1}^{d} a^{2(i-1)t_2-t_1} + (a^{t_2}-1)\sum_{i=1}^{d} a^{2(i-1)t_2} \pmod p \\ &= (a^{-t_1}+1)(a^{t_2}-1)\sum_{i=1}^{d} a^{2(i-1)t_2} \pmod p. \end{aligned} \tag{10}$$

Similarly,

$$\begin{aligned} h = 0 \pmod p \Rightarrow \sum_{i=0}^{2d-1} a^{it_2} &= (a^{t_2-t_1} + a^{t_1}) \sum_{i=1}^{d} a^{2(i-1)t_2} \pmod p \\ \Rightarrow \sum_{i=1}^{d} a^{2(i-1)t_2} &= (a^{t_2-t_1} + a^{t_1})^{-1} \sum_{i=0}^{2d-1} a^{it_2} \pmod p. \end{aligned} \tag{11}$$

Using (10) and (11), we have

$$
\begin{aligned}
(a^{t_2-t_1}+a^{t_1})\,(a^{2dt_2-t_1}-a^{-t_1}) &= (a^{-t_1}+1)\,(a^{t_2}-1)\sum_{i=0}^{2d-1} a^{it_2} \qquad (\operatorname{mod} p)\\
\Rightarrow \quad a^{(2d+1)t_2-2t_1}-a^{t_2-2t_1}+a^{2dt_2}-1 &= (a^{t_2-t_1}-a^{-t_1}+a^{t_2}-1)\sum_{i=0}^{2d-1} a^{it_2}\\
&= \sum_{i=0}^{2d-1}\{a^{(i+1)t_2-t_1}-a^{it_2-t_1}+a^{(i+1)t_2}-a^{it_2}\}\\
&= (a^{-t_1}+1)(a^{2dt_2}-1)\\
&= 2^{2dt_2-t_1}-a^{-t_1}+a^{2dt_2}-1\\
\Rightarrow \qquad a^{(2d+1)t_2-t_1}-a^{t_2-t_1} &= a^{2dt_2}-1\\
\Rightarrow \qquad a^{t_2-t_1}(a^{2dt_2}-1) &= a^{2dt_2}-1\\
\Rightarrow \qquad (a^{t_2-t_1}-1)(a^{(t_1-1)t_2}-1) &= 0 \qquad \text{(because } 2d=t_1-1).
\end{aligned}
$$

That is, $\mathbf{T}=\mathbf{I}\Rightarrow t_1=t_2$ or $t_1=1$ or $t_2=0$, which contradict $(t_1,t_2)=1$, $(t_1,k)\neq 1$, and $t_1,t_2\neq 0$. Hence $\mathbf{T}\neq\mathbf{I}$. Similar to Case 1, we can now construct a GCR with divisor $q=k$ and choose the representing elements according to (6). That is, the representing element of class j is the composition of the first j elements in (6). Specifically,

$$
\begin{aligned}
a_0 &= \mathbf{I};\\
a_1 &= \mathbf{B};\\
a_2 &= \mathbf{B}^2;\\
&\ \vdots\\
a_{t_1-1} &= \mathbf{B}^{t_1-1};\\
a_{t_1} &= \mathbf{B}^{t_1-1}\mathbf{A};\\
a_{t_1+1} &= \mathbf{B}^{t_1-1}\mathbf{A}^2;\\
&\ \vdots\\
a_{t_1+k_1-2} &= \mathbf{B}^{t_1-1}\mathbf{A}^{k_1-1};\\
a_{t_1+k_1-1} &= \mathbf{B}^{t_1-1}\mathbf{A}^{k_1-1}\mathbf{B}^{-1};\\
a_{t_1+k_1} &= \mathbf{B}^{t_1-1}\mathbf{A}^{k_1-1}\mathbf{B}^{-1}\mathbf{A}^{-1};\\
a_{t_1+k_1+1} &= \mathbf{B}^{t_1-1}\mathbf{A}^{k_1-1}\mathbf{B}^{-1}(\mathbf{A}^{-1})^2;\\
&\ \vdots\\
a_{q-1} &= \mathbf{B}^{t_1-1}\mathbf{A}^{k_1-1}\{\mathbf{B}^{-1}(\mathbf{A}^{-1})^{k_1-2}\mathbf{B}^{-1}\mathbf{A}^{k_1-2}\}^d.
\end{aligned}
$$

Again, we assume that the representing element of class j is

$$
\begin{pmatrix} a^i & \bar{y}_j \\ 0 & 1 \end{pmatrix}.
$$

With these choices, the superscript i spans the set of $\{0,1,\ldots,k-1\}$. Furthermore, the following representing elements are connected to each other: $a_0\sim a_1\sim\cdots\sim a_{q-1}\sim \mathbf{T}*a_0$. Hence we have a CR representation.

Subcase 2. t_1 is even. We define the integer $d=(t_1/2)-1$. In this case, we consider a GCR with

$$\text{(12)} \qquad \mathbf{T}=\mathbf{B}^{t_1-1}\mathbf{A}^{k_1-1}\{\mathbf{B}^{-1}(\mathbf{A}^{-1})^{k_1-2}\mathbf{B}^{-1}\mathbf{A}^{k_1-2}\}^d\mathbf{B}^{-1}(\mathbf{A}^{-1})^{k_1-1}.$$

Again, using similar techniques as in Subcase 1, we can prove that

$$\mathbf{T}=\begin{pmatrix} 1 & y' \\ 0 & 1 \end{pmatrix}$$

for some $y' \in \mathbf{Z}_p$ and $y' \neq 0$. A GCR with divisor $q=k$ can then be constructed with class representing elements, $a_o, a_1, \ldots, a_{q-1}$, determined from the composition of the first j elements in (12). That is,

$$\begin{aligned}
a_0 &= \mathbf{I};\\
a_1 &= \mathbf{B};\\
a_2 &= \mathbf{B}^2;\\
&\vdots\\
a_{q-1} &= \mathbf{B}^{t_1-1}\mathbf{A}^{k_1-1}\{\mathbf{B}^{-1}(\mathbf{A}^{-1})^{k_1-2}\mathbf{B}^{-1}\mathbf{A}^{k_1-2}\}^d\mathbf{B}^{-1}(\mathbf{A}^{-1})^{k_1-2}.
\end{aligned}$$

As before, the superscripts of the first element of all class representing elements span the set of $\{0,1,\ldots,k-1\}$. Also, the representing elements are connected to each other: $a_0 \sim a_1 \sim \cdots \sim a_{q-1} \sim \mathbf{T} * a_0$. Hence we have a CR representation.

Case 3. $t_1 = 0$ In this case, we can assume that $(t_2, k) = 1$ (t_2 and k are relatively prime); otherwise the graph is disconnected. According to Proposition 3, $t_1 = 0 \Rightarrow \mathbf{A}^p = \mathbf{I}$. Consider

$$\mathbf{A}^{-1}\mathbf{B}=\begin{pmatrix} a^{t_2} & y_2-y_1 \\ 0 & 1 \end{pmatrix},$$

$$(\mathbf{A}^{-1}\mathbf{B})^m=\mathbf{I} \Rightarrow m=\frac{\text{LCM}\,(t_2,k)}{t_2}=k.$$

Hence $m=n/p=k=\text{LCM}\,(t_2,k)/t_2$. According to the sufficient condition in Proposition 2, we choose $\mathbf{T}=\mathbf{A}^{-1}\mathbf{B}$ and the representing element of class i, $a_i=\mathbf{A}^i$ $(i=0,\ldots,p-1)$ to construct a CR representation with divisor $q=p$. □

In the above proposition, we proved that all degree-4 Borel Cayley graphs have CR representations. In the course of proving the proposition, we provided an algorithm for the construction of a CR representation. This algorithm is summarized in Table 3. For simplicity, Table 3 only shows one possible way of constructing a CR representation in Case 1, even though an alternate way exists.

4. Examples. In this section, we use three examples to illustrate the three cases discussed in the constructive proof of CR representations (§3). Again, we assume a degree-4 Borel Cayley graph with parameters n, p, a, k as defined in Definition 3. Furthermore, $n=p\times k$ and $\mathbf{A}, \mathbf{B}, \mathbf{A}^{-1}, \mathbf{B}^{-1}$ are the generators, where

$$\mathbf{A}=\begin{pmatrix} a^{t_1} & y_1 \\ 0 & 1 \end{pmatrix}, \qquad \mathbf{B}=\begin{pmatrix} a^{t_2} & y_2 \\ 0 & 1 \end{pmatrix},$$

$t_1, t_2 \in \{0,\ldots,k-1\}$, and $y_1, y_2 \in \{0,\ldots,p-1\}$.

TABLE 3
An algorithm to generate a CR *representation.*

For any degree-4 Borel Cayley graph with $n = |\mathbf{V}| = p \times k$, assume $\mathbf{A}$, $\mathbf{B}$, and their inverses are generators

$$\mathbf{A} = \begin{pmatrix} a^{t_1} & y_1 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} a^{t_2} & y_2 \\ 0 & 1 \end{pmatrix}.$$

In each of the following cases, we construct a CR representation with divisor q, by following the procedure summarized in Table 2. Instead of using arbitrary transform element and class representing elements, we have specific choices.

Case 1. $t_1, t_2 \neq 0$ and $(t_1, k) = 1$.
Assume $t_2 = mt_1$ for some integer m;

$$\mathbf{T} = \mathbf{B}\,\mathbf{A}^{k-1-m}\,\mathbf{B}\,(\mathbf{A}^{-1})^{m-1}.$$

The representing element of class 0 is $\mathbf{I}$ and of class j is the composition of the first j elements in the above equation. With these choices, there are $q = k$ classes.

Case 2. $t_1, t_2 \neq 0$ and $(t_1, k) \neq 1$ and $(t_2, k) \neq 1$. Assume $\mathbf{A}^{k_1} = \mathbf{I}$
Subcase 1. t_1 is odd, let $d = (t_1 - 1)/2$;

$$\mathbf{T} = \mathbf{B}^{t_1-1}\mathbf{A}^{k_1-1}\{\mathbf{B}^{-1}(\mathbf{A}^{-1})^{k_1-2}\mathbf{B}^{-1}\mathbf{A}^{k_1-2}\}^{d}\mathbf{A}.$$

The representing element of class 0 is $\mathbf{I}$ and of class j is the composition of the first j elements in the above equation. With these choices, there are $q = k$ classes.
Subcase 2. t_1 is even, let $d = t_1/2 - 1$;

$$\mathbf{T} = \mathbf{B}^{t_1-1}\mathbf{A}^{k_1-1}\{\mathbf{B}^{-1}(\mathbf{A}^{-1})^{k_1-2}\mathbf{B}^{-1}\mathbf{A}^{k_1-2}\}^{d}\mathbf{B}^{-1}(\mathbf{A}^{-1})^{k_1-1}.$$

The representing element of class 0 is $\mathbf{I}$ and of class j is the composition of the first j elements in the above equation. With these choices, there are $q = k$ classes.

Case 3. $t_1 = 0$.
In this case, we can have a CR with $q = p$ classes and the transform element and class representing elements are

$$\mathbf{T} = \mathbf{A}^{-1}\mathbf{B} \quad \text{and} \quad a_j = \mathbf{A}^j, \quad j = 0, 1, \ldots, q-1.$$

4.1. Case 1. We consider a Borel subgroup with $p = 13$, $k = 12$, $a = 2$, $n = 156$. We choose parameters for the generators as $t_1 = 5$, $t_2 = 2$, $y_1 = 1$, $y_2 = 1$. That is, $\mathbf{A} = \left(\begin{smallmatrix} 6 & 1 \\ 0 & 1 \end{smallmatrix}\right)$, $\mathbf{B} = \left(\begin{smallmatrix} 4 & 1 \\ 0 & 1 \end{smallmatrix}\right)$. For this set of generators, diameter $D = 5$. Since $t_1, t_2 \neq 0$ and $(t_1, k) = 1$, the conditions for Case 1 in Table 3 are satisfied. Furthermore, $t_2 = 10\,t_1 \pmod{k}$. Accordingly, we choose

$$\mathbf{T} = \mathbf{B}\,\mathbf{A}\,\mathbf{B}\,(\mathbf{A}^{-1})^9 = \begin{pmatrix} 1 & 10 \\ 0 & 1 \end{pmatrix}.$$

We thus have a CR representation with divisor $q = k = 12$. For any $i \in \mathbf{V}$, if $i \bmod 12 =$:
"0" : i is connected to $i+1, i-1, i+14, i-38 \pmod{n}$;
"1" : i is connected to $i+1, i-1, i-22, i-69 \pmod{n}$;
"2" : i is connected to $i+1, i-1, i-14, i-57 \pmod{n}$;
"3" : i is connected to $i+1, i-1, i+22, i-58 \pmod{n}$;

"4" : i is connected to $i+1, i-1, i-34, i-69 \pmod{n}$;
"5" : i is connected to $i+1, i-1, i+74, i+58 \pmod{n}$;
"6" : i is connected to $i+1, i-1, i+14, i+34 \pmod{n}$;
"7" : i is connected to $i+1, i-1, i-22, i-74 \pmod{n}$;
"8" : i is connected to $i+1, i-1, i+50, i-14 \pmod{n}$;
"9" : i is connected to $i+1, i-1, i+62, i+22 \pmod{n}$;
"10" : i is connected to $i+1, i-1, i+38, i-50 \pmod{n}$;
"11" : i is connected to $i+1, i-1, i-57, i-62 \pmod{n}$.

4.2. Case 2. We consider the same Borel group as in Case 1, but with a different set of generators. The parameters for the generators are $t_1 = 2$, $t_2 = 3$, $y_1 = 1$, $y_2 = 1$. That is, $\mathbf{A} = \left(\begin{smallmatrix} 4 & 1 \\ 0 & 1 \end{smallmatrix}\right)$, $\mathbf{B} = \left(\begin{smallmatrix} 8 & 1 \\ 0 & 1 \end{smallmatrix}\right)$. For this set of generators, diameter $D = 6$. Since $t_1, t_2 \neq 0$, $(t_1, k) \neq 1$, and $(t_2, k) \neq 1$, the conditions for Case 2 in Table 3 are satisfied. Furthermore, $k_1 = 6$, and $t_1 = 2$ is even. Accordingly, we choose

$$\mathbf{T} = \mathbf{B}\,\mathbf{A}^4\,\mathbf{B}^{-1}\,(\mathbf{A}^{-1})^4 = \begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}.$$

We thus have a CR representation with divisor $q = k = 12$. For any $i \in \mathbf{V}$, if $i \bmod 12 =$:
"0" : i is connected to $i+1, i-1, i-5, i+64 \pmod{n}$;
"1" : i is connected to $i+1, i-1, i+5, i-16 \pmod{n}$;
"2" : i is connected to $i+1, i-1, i+54, i-51 \pmod{n}$;
"3" : i is connected to $i+1, i-1, i+28, i+67 \pmod{n}$;
"4" : i is connected to $i+1, i-1, i-64, i+77 \pmod{n}$;
"5" : i is connected to $i+1, i-1, i+18, i-33 \pmod{n}$;
"6" : i is connected to $i+1, i-1, i-5, i+40 \pmod{n}$;
"7" : i is connected to $i+1, i-1, i+5, i-28 \pmod{n}$;
"8" : i is connected to $i+1, i-1, i+33, i-54 \pmod{n}$;
"9" : i is connected to $i+1, i-1, i-77, i+16 \pmod{n}$;
"10" : i is connected to $i+1, i-1, i-67, i-40 \pmod{n}$;
"11" : i is connected to $i+1, i-1, i+51, i-18 \pmod{n}$.

4.3. Case 3. We consider a smaller Borel Cayley graph with $a = 2$, $p = 7$, $k = 3$, $n = 21$, diameter $D = 3$, and the generators $\mathbf{A} = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$, $\mathbf{B} = \left(\begin{smallmatrix} 2 & 1 \\ 0 & 1 \end{smallmatrix}\right)$. Note that, in this case, we have $t_1 = 0$, $t_2 = 1$, $q = p = 7$, and $n/q = \text{LCM}\,(t_2 - t_1, k)/(t_2 - t_1) = 3$. According to Table 3, we choose $\mathbf{T} = (\mathbf{A}^{-1}\mathbf{B})$, $a_j = \mathbf{A}^j$ to produce a CR representation with divisor, $q = p = 7$. Let $\mathbf{V} = \{0, 1, \ldots, 20\}$. For any $i \in \mathbf{V}$, if $i \bmod 7 =$:
"0" : i is connected to $i+1, i-1, i-10, i+6 \pmod{n}$;
"1" : i is connected to $i+1, i-1, i+7, i-7 \pmod{n}$;
"2" : i is connected to $i+1, i-1, i+10, i-6 \pmod{n}$;
"3" : i is connected to $i+1, i-1, i+6, i-5 \pmod{n}$;
"4" : i is connected to $i+1, i-1, i+9, i+10 \pmod{n}$;
"5" : i is connected to $i+1, i-1, i+5, i-10 \pmod{n}$;
"6" : i is connected to $i+1, i-1, i-6, i-9 \pmod{n}$.
We show this CR representation of the graph in Fig. 5.

5. Conclusions. Dense, symmetric graphs are good candidates for the interconnection topology of a multicomputer system. Being a class of symmetric graphs, Cayley graphs are attractive. In our earlier research effort, we discussed the representations and routing of Cayley graphs [2]. In this paper, we analyzed a special class of Cayley graphs, the *Borel Cayley graphs*, which generates the densest known, constructive, degree-4 graphs with diameter $D = 7, \ldots, 13$.

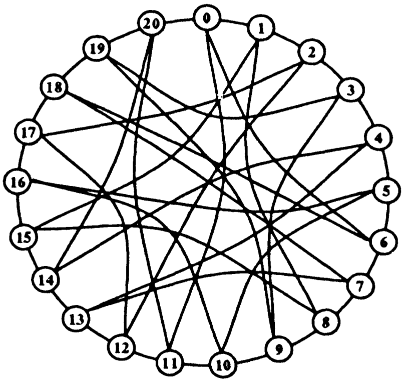


FIG. 5. *CR representation of* $\mathbf{BL}_2(\mathbf{Z}_7)$.

Borel Cayley graphs are defined over a group of matrices, the *Borel matrices*. That is, nodes are labeled as matrices. There is no inherent, simple ordering of node labels and no known computational routing algorithm with a constant or O(1) space commitment. GCRs and CRs, on the other hand, are two existing topologies defined in the integer domain and have systematic structure.

By transforming into GCRs [2], Cayley graphs have a systematic representation. Furthermore, an optimal, time-efficient routing algorithm, called *vertex-transitive routing*, is developed for Borel Cayley graphs [18]. However, the goal of developing an optimal, space-efficient, distance-reduction routing algorithm is still elusive.

Through the discovery of inherent properties of degree-4 Borel Cayley graphs, we proved that CR representations always exist for these graphs. A step-by-step algorithm and examples are used to illustrate the transformation to CR representations. This special case of a GCR includes a Hamiltonian cycle formed by edges connecting adjacent integers in the modulo n labels, thus permitting a distance-reduction routing algorithm, called *CR routing*. Given a Borel Cayley graph with $n = pk$ nodes (p is a prime and k is a factor of $p - 1$), this distance-reduction algorithm requires a small table of O(k). However, the algorithm is *suboptimal* in the sense that a shortest path is not guaranteed. Readers interested in CR routing are referred to [17].

Aside from facilitating the development of a space-efficient routing algorithm, the existence of a CR representation for any degree-4 Borel Cayley graphs also partially proved the long-standing conjecture that all Cayley graphs have Hamiltonian cycles [19]. Obviously, a CR graph, by definition, contains a Hamiltonian cycle. In fact, its class structure and connection rules impose a stronger condition. By providing a CR representaion, we have thus shown that all connected, degree-4 Borel Cayley graphs have Hamiltonian cycles.

REFERENCES

[1] D. V. CHUDNOVSKY, G. V. CHUDNOVSKY, AND M. M. DENNEAU, *Regular Graphs with Small Diameter as Models for Interconnection Networks*, Tech. Report RC 13484(60281), IBM Research Division, T. J. Watson Research Center, Yorktown Heights, NY, February 1988.

[2] B. W. ARDEN AND K. W. TANG, *Representations and routing of Cayley graphs*, IEEE Trans. Comm., 39 (1991), pp. 1533–1537.

[3] D. A. REED AND R. M. FUJIMOTO, *Multicomputer Networks*, MIT Press, Cambridge, MA, 1987.

[4] L. D. WITTIE, *Communication structures for large networks of microcomputers*, IEEE Trans. Comput., 30 (1981), pp. 264–273.

[5] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, North–Holland, New York, 1979.

[6] J. C. Bermond, C. Delorme, and J. J. Quisquater, *Tables of large graphs with given degree and diameter*, Inform. Process. Lett., 15 (1982), pp. 10–13.

[7] T. Y. Feng, *A survey of interconnection networks*, Computer, 14 (1981), pp. 12–27.

[8] G. H. Barnes, *The* ILLIAC IV *computer*, IEEE Trans. Comput., 17 (1968), pp. 746–757.

[9] J. P. Hayes et al., *Architecture of a hypercube supercomputer*, in Proc. of the 1986 Internat. Conf. on Parallel Processing, St. Charles, IL, August 1986, pp. 653–660.

[10] B. W. Arden and H. Lee, *Analysis of chordal ring network*, IEEE Trans. Comput., 30 (1981), pp. 291–295.

[11] F. P. Preparata and J. Vuillemin, *The cube-connected cycles: A versatile network for parallel computation*, Comm. Assoc. Comput. Mach., May 1981, pp. 300–309.

[12] J. C. Bermond and C. Delorme, *Strategies for interconnection networks: Some methods from graph theory*, J. Parallel Distributed Comput., 3 (1986), pp. 433–449.

[13] M. Homewood, D. May, D. Shepherd, and R. Shepherd, *The* IMS T800 *transputer*, IEEE MICRO, October 1987, pp. 10–26.

[14] S. B. Akers and B. Krishnamurthy, *A group-theoretic model for symmetric interconnection networks*, IEEE Trans. Comput., 38 (1989), pp. 555–565.

[15] G. E. Carlsson, J. E. Cruthirds, and H. B. Sexton, *Interconnection networks based on a generalization of cube-connected cycles*, IEEE Trans. Comput., 34 (1985), pp. 769–772.

[16] K. W. Tang and B. W. Arden, *Representations and routing for Borel Cayley graphs*, in Proc. of Internat. Conf. on Information Technology, Tokyo, Japan, October 1990, pp. 27–31.

[17] ———, *Class-congruence property and two-phase routing for Borel Cayley graphs*, IEEE Trans. Comput., February 1993, submitted.

[18] ———, *Vertex-transitivity and routing for Cayley graphs in* GCR *representations*, in Proc. of 1992 Sympos. on Applied Computing, Kansas City, MO, March 1992, pp. 1180–1187.

[19] D. Witte and J. A. Gallian, *A survey: Hamiltonian cycles in Cayley graphs*, Discrete Math., 51 (1984), pp. 293–304.

[20] M. J. Dinneen, *Algebraic Methods for Efficient Network Constructions*, Master thesis, Department of Computer Science, University of Victoria, Victoria, BC, Canada, 1991.

[21] L. Campbell et al., *Small diameter symmetric networks from linear groups*, IEEE Trans. Comput., 41 (1992), pp. 218–220.

[22] K. W. Tang and B. W. Arden, *Pseudo-Random Formulation of Borel Cayley Graphs*, Technical Report #661, College of Engineering and Applied Sciences, State University of New York at Stony Brook, Stony Brook, NY, March 1993.

SIAM J. DISCRETE MATH
Vol. 6, No. 4, p. 677, November 1993

011

ERRATA: LYAPUNOV FUNCTIONALS FOR AUTOMATA NETWORKS DEFINED BY CYCLICALLY MONOTONE FUNCTIONS*

E. GOLES† AND S. MARTÍNEZ†

Theorems 2 and 3 of our paper have already been proved by Poljak and Turzik [1]. (Our main contribution was determination of a Lyapunov operator for such automata.)

To prove Theorem 2 of our paper, the definition of strict cyclically monotone given on p. 201 must be changed to the following: f is strict cyclically monotone if and only if f is cyclically monotone and $g(u) = g(u) + < f(v), u - v > \Longrightarrow f(u) = f(v)$, where g is a potential associated to f. $f(u) = f(v)$, where g is a potential associated to f.

From previous definition, in the proof of Theorem 2 in our paper, the equation

$$\Delta_{t+1}\hat{H} = g(u(t+2)) - g(u(t)) - < f(u(t)), u(t+2) - u(t) > = 0$$

implies $f(u(t)) = f(u(t+2))$; that is, $x(t+3) = x(t+1)$.

REFERENCE

[1] S. POLJAK AND D. TURZIK, *On an application of convexity to discrete systems*, Discrete Appl. Math., 13(1986), pp. 27–32.

*Received by the editors August 4, 1993; accepted for publication August 6, 1993. SIAM J. Discrete Math., 4(1991), pp. 200–206.

†Departamento de Ingeniería, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, Casilla 170/3, correo 3, Santiago, Chile.