THE WEAK BRUHAT ORDER OF S_Σ, CONSISTENT SETS, AND CATALAN NUMBERS*

JAMES ABELLO†

Abstract. *Chains* in the *weak Bruhat order* β of S_Σ (the symmetric group on Σ) belong to the class of subsets of S_Σ over which unrestricted choice necessarily produces transitive relations under pairwise simple majority vote (*consistent sets*). If for $A \subset S_\Sigma$ we let $T(A) \equiv \cup_{p \in A} T(p)$ where $T(p) = \{(p_i, p_j, p_k) | i < j < k\}$ and $\Psi(A) \equiv \{w \in S_\Sigma | T(w) \subset T(A)\}$ the following theorem (among others) is obtained.

THEOREM. *For all* $q \in S_\Sigma$, *if* A *is a saturated chain under* β *then* $\Psi(qA)$ *is an upper semimodular sublattice of cardinality* $|\Psi(qA)| \leqq \frac{1}{|\Sigma|+1}\binom{2|\Sigma|}{|\Sigma|} \equiv$ *The* $|\Sigma|$*th Catalan number.*

From the Arrow's Impossibility Theorem point of view, the results obtained here indicate that majority rule produces transitive results if the collection of voters as a whole can be partitioned into no more than $(|\Sigma|^2 + |\Sigma|)/2$ groups which can be ordered according to the level of disagreement they have with respect to a fixed permutation p. On the other hand, by viewing S_Σ as a *Coxeter group* a "novel" combinatorial interpretation of the collection of maximal chains that can be obtained from one another by using only one type of Coxeter transformation is obtained.

Key words. weak Bruhat order, upper semimodular lattice, Catalan numbers, Arrow's Impossibility Theorem, Coxeter groups

AMS(MOS) subject classifications. 05, 06

Introduction. The Marquis de Condorcet recognized nearly 200 years ago [12] that majority rule can produce intransitive group preferences if the domain of possible (transitive) individual preference orderings is unrestricted. This phenomenon is commonly known as the voting paradox (see Black [9] and Riker [20] for an excellent historical account).

Domains for which the simple majority rule produces transitive results are called here "Transitive Simple Majority" domains (TSM). The study of the structure and cardinality of TSM domains has proven to be a combinatorial problem of an unusual sort (Abello [1], [2], [4], Abello and Johnson [3], Arrow [5], Black [9], Fishburn [15], Good [17], Ward [25]).

By restricting our attention to TSM domains that are subsets of the symmetric group (called here "consistent sets") we have given general constructions that produce "consistent" sets of greater cardinality than all those offered in the past (Abello [2], Abello and Johnson [3]). All the constructed sets are maximally transitive and they achieve the best known (uniform) general lower bound.

A unified view of several seemingly different constructions of "consistent" sets has been obtained by Abello [1] via the weak Bruhat order, β, of S_n (Bourbaki [10], Lehmann [19], Savage [21], Yanagimoto and Okamoto [26]).

In this paper we will present the only known global structural properties of "consistent" sets. Namely, we prove that each maximal "consistent" set that contains a maximal chain in β is an upper semimodular sublattice of $\langle S_n, \beta \rangle$. This offers a "novel"

* Received by the editors March 2, 1987; accepted for publication (in revised form) March 2, 1990. Portions of this work were presented at the Third SIAM Conference on Discrete Mathematics, May 14-16, 1986, Clemson, South Carolina and first appeared in the author's Ph.D. dissertation, University of California, San Diego, 1985. This work was partially supported by a University of California President's Postdoctoral Fellowship and National Science Foundation grant DCR86-03722.

† Department of Computer Science, Texas A&M University, College Station, Texas 77843.

combinatorial interpretation of each collection of maximal chains in β whose elements can be obtained from one another by using one type of *Coxeter transformation* (Benson and Grove [6], Coxeter and Moser [13]). Moreover, we prove that each of these maximal transitive sets has cardinality bounded by the nth *Catalan number*. This provides the unique nontrivial upper bound known to date.

We must remark that even though we restrict our attention to subsets of the symmetric group, many of the ideas contained here are extendable to the more general domains discussed in Chapter 1 of Abello [4], as they stand or with modification.

1. Preliminaries. Let $\langle \Sigma, \leqq \rangle$ be a totally ordered set of symbols of cardinality $|\Sigma| = n \in Z^+$ and S_Σ the group of permutations on Σ (we will be using one line notation for permutations).

DEFINITION 1.1. A set $\{u, v, w\} \subset S_\Sigma$ is called a *cyclic* three-set if there are three symbols $x, y, z \in \Sigma$ such that $u^{-1}(x) < u^{-1}(y) < u^{-1}(z)$, $v^{-1}(y) < v^{-1}(z) < v^{-1}(x)$, $w^{-1}(z) < w^{-1}(x) < w^{-1}(y)$.

DEFINITION 1.2. A subset C of S_Σ is called *consistent* if it contains no cyclic three-set; otherwise C is called a *cyclic* set.

DEFINITION 1.3.

i. For $p \in S_\Sigma$, let:

$$T(p) \equiv \{(x,y,z) \mid p^{-1}(x) < p^{-1}(y) < p^{-1}(z)\};$$

$$\Gamma(p) \equiv \{(x,y) \mid p^{-1}(x) < p^{-1}(y)\};$$

$$\tau(p) \equiv \{(x,y) \in \Gamma(p) \mid p^{-1}(x) + 1 = p^{-1}(y)\}.$$

We will refer to $T(p)$, $\Gamma(p)$, and $\tau(p)$ as the sets of triples, pairs, and admissible adjacent transpositions determined by p, respectively. If $t \in \tau(p)$ then $t(p)$ will denote the permutation obtained from p by interchanging the symbols x and y where $(x, y) = t$.

ii. For $C \subseteq S_\Sigma$, let $T(C) \equiv \cup_{p \in C} T(p)$, $\Gamma(C) \equiv \cup_{p \in C} \Gamma(p)$, $\tau(C) \equiv \cup_{p \in C} \tau(p)$. Note that $|T(p)| = \binom{|\Sigma|}{3}$ for $|\Sigma| \geqq 3$. We will say that T(C) is a *cyclic* or *consistent* set of triples depending on whether C is a *cyclic* or *consistent* subset of S_Σ, respectively.

The following are some elementary properties of consistent sets.

FACT 1.1.

i. *Any subset of a consistent set is consistent and any superset of a cyclic set is cyclic.*

ii. *The intersection of consistent sets is consistent but their union is not always consistent.*

iii. $|T(S_\Sigma)|$ = *the number of different* 3-*permutations out of a set of* $|\Sigma|$-*elements.*

iv. *If* C *is a consistent subset of* S_Σ *then* $|T(C)| \leqq 4\binom{|\Sigma|}{3}$.

2. A closure operator on S_Σ. The results in this section are independent of consistency.

DEFINITION 2.1.

i. Let $\Psi: 2^{S_\Sigma} \to 2^{S_\Sigma}$ be given by $\Psi(A) = M_A = \{w \in S_\Sigma \mid T(w) \subseteq T(A)\}$.

ii. If $A \subseteq S_\Sigma$ is such that $\Psi(A) = A$ then A is called a *closed* subset and if $K \subseteq A$ satisfies that $\Psi(K) = \Psi(A)$ where $|K| = \min |B|$ (taken over all subsets B of A such that $T(B) = T(A)$), then K is called a *kernel* for A.

Let $C_K = \{A \subseteq S_\Sigma \mid K \text{ is a kernel for } A\}$. The following facts are immediate from the preceding definitions.

FACT 2.1.

i. Ψ *is a closure operator on* S_Σ, *namely,* $A \subseteq \Psi(A)$; *if* $A \subseteq B$ *then* $\Psi(A) \subseteq \Psi(B)$ *and* $\Psi^2(A) = \Psi(A)$.

ii. *There is a unique closed set in* C_K, *namely,* $X_K = \Psi(K)$.

iii. *If* K *and* K′ *are kernels for* C_K *and* $C_{K'}$, *respectively, and* $T(K) \subset T(K')$ *then* $\Psi(K) \subset \Psi(K')$.

The preceding result means that a *closed* set is completely determined by its kernels; moreover, any kernel K of a *closed* set X_K will do in the sense that if $K = \{K_1, K_2, \cdots, K_j\}$ then a chain of subsets, $\{\Psi_i\}_{i=1}^{j}$, can be constructed such that $\Psi_i \subset \Psi_{i+1}$ for $i = 1, \cdots, j-1$ and $\Psi_j = X_K$, namely, $\Psi_i \equiv \Psi(\{K_1, \cdots, K_i\})$. Note also that by letting $A_1 = \Psi_1$ and $A_{i+1} = \Psi_{i+1} - \Psi_i$ for $i = 1, \cdots, j-1$, we obtain a partition $(A_1, A_2, \cdots, A_j)$ of X_K. So, if we can characterize the dependencies between A_{i+1} and A_i we will have (perhaps) some information about the cardinality of A_i, $|A_i|$, which will give us at least bounds for $|X_K| = \sum_{i=1}^{j} |A_i|$. Therefore the study of the class of *closed* sets in an *independence system* coming from a closure operator may be reduced to the study of their corresponding kernels. Unfortunately determination of even a single kernel K, for a *closed* set X_K seems to be a hard computational problem because if K and K′ are kernels for X_K and $x \in K$ it is not true, in general, that there exists $y \in K'$ such that $K - \{x\} \cup \{y\}$ is a kernel, so there is not a suitable interchange property based on Ψ (see Williamson [24] for related topics). However, by relaxing the minimality assumption of a kernel and by imposing a mild restriction on each A_i we are able to characterize the elements of A_{i+1}. This is our intention in what follows.

DEFINITION 2.2.

i. A set of triples $O \subseteq T(S_\Sigma)$ is called *realizable* if there exists $A \subseteq S_\Sigma$ such that $T(A) = O$. In this case we will denote $M_A = \Psi(A)$ by M_O.

ii. A set $M = \Psi(A)$ is called *extensible* if there is a transposition $t = (x, y)$ and an element $p \in M$ such that $t \in \tau(p)$, (x and y are adjacent in p), and for all $w \in M$, $w^{-1}(x) < w^{-1}(y)$. In this case we will say that M is *extensible* by the pair (t, p). Note that a set may be *extensible* by many different pairs (t, p).

THEOREM 2:1. *Let* $M \subset S_\Sigma$ *be* extensible *by the pair* (t, p) *where* $t = (x, y)$, $p = uxyv$ *and let* $O = T(M)$. *If* $w \in M_{O \cup T(t(p))}/M_O$ *then* $w = u'yxv'$ *where* $u' \in S_u$, $v' \in S_v$, (S_u *and* S_v *denote the symmetric groups on the synmbols of* u *and* v, *respectively*).

Proof. i. First note that because $O \cup T(t(p))$ is a realizable set of triples the notation $M_{O \cup T(t(p))}$ makes sense. $w \in M_{O \cup T(t(p))}/M_O \rightarrow T(w) \cap [T(t(p))/O] \neq \emptyset$ by the definition of Ψ and because $O = T(M)$.

ii. $\emptyset \neq T(w) \cap [T(t(p))/O] \subset T(t(p))/O = \{(-, y, x), (y, x, -)\} \rightarrow$ w cannot be of the form $w = u'xyv'$.

iii. So, w *is of the form* $w = u'yAxv'$ for some $A \subset \Sigma$. The triples in w of the form (y, A, x) (if any) must be in $T(t(p))/O$ because x precedes y in every permutation in M_O by hypothesis. On the other hand $T(t(p))$ does not contain triples of the form (y, A, x) because $t \in \tau(p)$; therefore, $A = \emptyset$ and $w = u'yxv'$.

iv. Suppose now that $u' \notin S_u$ where $p = uxyv$. This means that there exists a symbol $c \in$ symbols of u'/symbols of u and $w = \cdots c \cdots yxv'$, $p = uxy \cdots c \cdots$, $(p) = uyx \cdots c \cdots$.

v. The triple $(c, y, x) \notin O$ because x precedes y in every permutation in M_O, also $(c, y, x) \notin T(t(p))$ by (iv), so $(c, y, x) \notin O \cup T(t(p))$ which means that $w \notin M_{O \cup T(t(p))}$, (contradiction); therefore, symbols of $u' \subset$ symbols of u.

vi. Finally, assume that there exists a symbol c which appears in u but not in u'. We can assume that $w = u'yxv'$ and $t(p) = u' \cdots c \cdots yxv''$ (by v). In this case we have that c appears in v' but not in v'', then $w = u'yx \cdots c \cdots$ and again the triple $(y, x, c) \notin O \cup T(t(p))$, which means that $w \notin M_{O \cup T(t(p))}$, (contradiction); therefore, symbols of $u \subset$ symbols of u'.

(v) and (vi) together give us that if $p = uxyv$ then $w = u'yxv'$ where $u' \in S_u$ and $v' \in S_v$. □

The preceding theorem allows us to express in a very explicit way the relationship between $M_{\mathcal{O}UT(p)}$ and $M_{\mathcal{O}}$ as stated in the following corollary.

COROLLARY 2.1. *Let* $M \subset S_\Sigma$ *be extensible by* (t, p) *where* $t = (x, y)$, $p = u x y v$ *and let* $\mathcal{O} = T(M)$. *If* $w \in M_{\mathcal{O}UT(t(p))}/M_{\mathcal{O}}$ *then* $t^{-1}(w) \in M_{\mathcal{O}}$.

Proof. Let $p = uxyv$ and $t = (x, y)$. $w \in M_{\mathcal{O}UT(t(p))}/M_{\mathcal{O}} \twoheadrightarrow w = u'yxv'$, $u' \in S_u$, $v' \in S_v$ by the preceding theorem. This in turn implies that $T(w)/\mathcal{O} = T(t(p))/\mathcal{O}$, and $T(t^{-1}(w))/T(w) \subset T(p)$ because $t^{-1}(w) = u'xyv'$, $u' \in S_u$, $v' \in S_v$; therefore, $T(t^{-1}(w)) \subset T(p) \cup \mathcal{O} = \mathcal{O}$, which means that $t^{-1}(w) \in M_{\mathcal{O}}$. □

Corollary 2.1 tells us that the "extension" of a set M by a pair (t, p) is completely determined by a subset of it, namely, $\{q \in M \mid q = u'xyv'$ with $u' \in S_u$, $v' \in S_v$, $p = uxyv$ and $t = (x, y)\}$. Note that the reciprocal of Corollary 2.1 is not true in the sense that it can happen that $t^{-1}(w) \in M_{\mathcal{O}}$ and however $w \notin M_{\mathcal{O}UT(t(p))}$. This motivates the following definition.

DEFINITION 2.3. If $M \subset S_\Sigma$ is *extensible* by a pair (t, p), then *the projection set* of M with respect to (t, p) will be denoted by $\prod^{M}_{t,p}$ and is defined as follows.

$\prod^{M}_{t,p} \equiv \{q \in M \mid q = u'xyv'$ where $u' \in S_u$, $v' \in S_v$, $p = uxyv$, $t = (x, y)\}$. With this definition we have the following corollary.

COROLLARY 2.2. *If* M *is extensible by* (t, p) *and* $O = T(M)$ *then* $M_{\mathcal{O}UT(t(p))} = M \cup t(\prod^{M}_{t,p})$.

Proof. The proof follows from Theorem 2.1. and the definition of $\prod^{M}_{t,p}$.

We close this section by mentioning that if X_K is a *closed* set under Ψ and if there exists a sequence of pairs $\{(t_i, P_i)\}^{j}_{i=1}$, such that $T(K) = \bigcup^{j}_{i=1} T(P_i)$ and each of the sets $\Psi_i = \Psi(\{P_1, \cdots, P_i\})$ is *extensible* by (t_i, P_i) for $i = 1, \cdots, j-1$, then by letting $A_1 = \Psi_1$, $A_{i+1} = \Psi_{i+1} - \Psi_i$ for $i = 1, \cdots, j-1$ we obtain a partition $(A_1, \cdots, A_j)$ of X_K, even though $\{P_i\}^{j}_{i=1}$ is not, in general, a kernel for X_K. All of this is true independent of the consistency of X_K. In the case that X_K is *consistent* then we can characterize algorithmically $\prod^{\Psi_i}_{t_i,p_i}$ for $i = 1, \cdots, j-1$ by looking at the *weak Bruhat* order of S_Σ. This is the purpose of the next section.

3. The weak Bruhat order of S_Σ versus consistent sets.

DEFINITION 3.1.

i. For $u = u_1 \cdots u_n$, let $E(u) = \{(u_i, u_j) \mid i < j, u_i < u_j\}$. E(u) is commonly known as the set of noninversions of u.

ii. For $\{u, v\} \subset S_\Sigma$ we write,

a) $u \twoheadrightarrow v$ if there exists $t \in E(u) \cap \tau(u)$ such that $t(u) = v$.

We say in this case that u *weakly covers* v;

b) $u \overset{\cdot}{\twoheadrightarrow} v$ if there exists $t \in E(u)$ such that $t(u) = v$. In this case we say that u *strongly covers* v.

iii. The *weak Bruhat* order of S_Σ, β, is defined as follows.

$u \, \beta \, v$ if there exists a sequence $(P_0, \cdots, P_m)$, $P_i \in S_\Sigma$ such that $u = P_0$, $P_m = v$ and $P_{i-1} \twoheadrightarrow P_i$ for $i = 1, \cdots, m$ (Lehmann [19], Savage [21]).

iv. The *strong Bruhat* order of S_Σ, $\dot{\beta}$, is given by $u \, \dot{\beta} \, v$ if $u = P_0$, $P_m = v$ and $P_{i-1} \overset{\cdot}{\twoheadrightarrow} P_i$ for $i = 1, \cdots, m$ (Savage [21], [22]. Clearly $u \, \beta \, v \twoheadrightarrow u \, \dot{\beta} \, v$.

FACT 3.1 (see Fig. 3.1).

i. $u \, \beta \, v$ *if and only if* $E(u) \supseteq E(v)$.

ii. *The maps* $f(u) = u \cdot I^R$ *and* $f'(u) = I^R \cdot u$ *are* order reversing involutions *of* $\langle S_\Sigma, \beta \rangle$, *i.e.,* $f^2(u) = u$ *and* $u \, \beta \, v \twoheadrightarrow f(v) \, \beta \, f(u)$; *similarly for* $f'(u)$, (I *is the identity in* S_Σ, I^R *is its reverse and* $\cdot$ *denotes the usual permutation multiplication*).

iii. $\langle S_\Sigma, \beta \rangle$ *and* $\langle S_\Sigma, \dot{\beta} \rangle$ *are posets with maximum element* I *and minimum element* I^R. *Moreover* $\langle S_\Sigma, \beta \rangle$ *is a* lattice *by defining the join* $u \vee v$ *of two elements* u *and* v *as the* minimum element p (*in the weak Bruhat order* β) *such that* $p \, \beta \, u$ *and* $p \, \beta \, v$ *while*

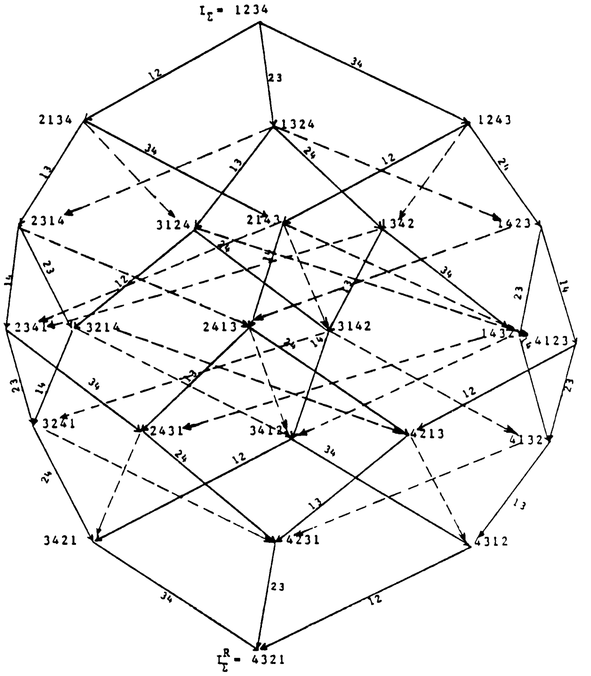


FIG. 3.1. *Bruhat orders on* S_Σ *for* $\Sigma = \{1, 2, 3, 4\}$. *Solid lines denote the covering relations in the weak order and dotted lines correspond to the additional covering relations in the strong ordering. The relevant transpositions are indicated on each edge.*

defining the meet $u \wedge v$ *dually, namely, as the* maximum element p' *such that* $u\ \beta\ p'$, $v\ \beta\ p'$. *In other words* $u \vee v$ = *least upper bound of* u *and* v *in* β *and* $u \wedge v$ = *greatest lower bound of* u *and* v *in* β.

Proof of i. That $u\ \beta\ v$ implies $E(u) \supseteq E(v)$ follows from the definition of β. In the other direction, let j be the minimum i such that $u_i \neq v_i$, (if such j does not exist then $u = v$ and we are done). For this choice of j we have that $u_j < v_j$ ($<$ is the order of Σ) and if $v_j = u_k$ then $u_{k-1} < u_k$ because we are assuming that $E(u) \supseteq E(v)$; therefore, $E(u) \supset E(t(u)) \supseteq E(v)$ where $t = (u_{k-1}, u_k)$. By repeating the argument we construct a chain $u = P_0 \rightarrow \cdots \rightarrow P_m$ with $E(P_m) = E(v)$, so $P_m = v$, which completes the proof. □

Proof of ii. Without loss of generality, take $\Sigma = \{1, 2, \cdots, n\}$. Then we have $f(u) = u \cdot I^R = u^R$, $f'(u) = I^R \cdot u = u'$ with $u_j' = (n+1) - u_j$ and the result immediately follows. □

Proof of iii. For the proof see Yanagimoto and Okamoto [26].

The following two lemmas give the first relation between the poset $\langle S_\Sigma, \beta \rangle$ and the class of consistent subsets of S_Σ. These results appear in Abello and Johnson [3] and Abello [1], [4] but we reproduce their proofs here for completeness.

LEMMA 3.1. *If* L *is a chain in* $\langle S_\Sigma, \beta \rangle$ *then* L *is a consistent subset of* S_Σ.

Proof (by contradiction). Assume that L is cyclic. Then there are three permutations u, v, w in L and three symbols x, y, z in Σ such that

$$u = \cdots x \cdots y \cdots z \cdots,$$

$$v = \cdots y \cdots z \cdots x \cdots,$$

$$w = \cdots z \cdots x \cdots y \cdots.$$

We can assume without loss of generality that $x < y < z$ (the only other essentially different case is $x > y > z$, which can be treated similarly).

i. $E(u)$ contains the ordered pairs (x, y), (x, z), (y, z) and at least two of these pairs do not belong to $E(v)$; thus $E(v) \not\supseteq E(u)$ which means that $v \not\beta u$. Similarly $E(w) \not\supseteq E(u)$ and then $w \not\beta u$. On the other hand $E(v)$ contains (y, z), which does not belong to $E(w)$, then $E(w) \not\supseteq E(v)$, which means $w \not\beta v$.

ii. $E(w)$ contains (x, y), which does not belong to $E(v)$, then $E(v) \not\supseteq E(w)$ and $v \not\beta w$.

(i) and (ii) together give us that v and w are not comparable and therefore u, v, and w cannot be in the same chain (a contradiction). □

Example 3.1. The set $\{1234, 1243, 1423, 4123, 4132, 4312, 4321\}$, which is a subset of $S_{\{1,2,3,4\}}$, is consistent because it is a chain in $\langle S_{\{1,2,3,4\}}, \beta\rangle$ (see Fig. 3.1).

It is interesting to notice that Lemma 3.1 is not true for the *strong Bruhat ordering* $\dot\beta$. For example, $\{2143, 3142, 4321\}$ is a chain in $\langle S_\Sigma, \dot\beta\rangle$; however, it is not *consistent*. This is due to the fact that $\dot\beta$ allows the interchange of nonadjacent elements.

The following is a simple but important property of *maximal* chains in β.

LEMMA 3.2. *If* L *is a maximal chain in* $\langle S_\Sigma, \beta\rangle$ *then* L *is a consistent subset of* S_Σ *such that* $|T(L)| = 4\binom{n}{3}$ and $|L| = \binom{n}{2} + 1$.

Proof. That L is consistent follows from the preceding lemma. Now, $|T(L)| = \binom{n}{3} + \binom{n}{2}(n - 2) = 4\binom{n}{3}$ because maximal chains in β have *length* equal to $\binom{n}{2}$. □

The interest of the preceding lemmas is that *for any consistent* set C it must be true that $|T(C)| \leqq 4\binom{n}{3}$ (see Fact 1.1 (iv)) so a maximal chain has the *maximum* number possible of consistent triples; therefore, any *maximal* (with respect to the noncyclicity property) consistent set M which contains a *maximal chain* L must satisfy that $T(M) = T(L)$. Now, if $L = (I = P_0, P_1, \cdots, P_{\binom{n}{2}} = I^R)$ with $t_{i+1}(P_i) = P_{i+1}$ for $i = 0, \cdots, \binom{n}{2} - 1$ and if L_i denotes the unrefinable subchain of L running from I to P_i, i.e., $L_i = \{q \in L, I \,\beta\, q \,\beta\, P_i\}$, then we have that for each i (as above) $\Psi(L_i)$ is a *consistent* set which is *extensible* by the pair (P_i, t_{i+1}) in the sense of § 2; therefore, Theorem 3.2.1 gives important information about the class of *maximal consistent sets* which contain a *maximal* chain in the weak Bruhat order. In fact it provides the basis of an algorithm to construct these sets (Abello [1], [2]).

The preceding ideas carry over to a more general class of consistent sets which contain subsets that are structurally equivalent to chains in the weak Bruhat order. To this end the following definitions are in order.

DEFINITION 3.2.

i. $L \subset S_\Sigma$ is called a *pseudochain* under β if there exists $p \in L$ and a map $m{:}u \rightarrow p^{-1} \cdot u$ such that $m(L)$ is a chain under β. If we want to indicate the dependency between L and p we write $L(p)$ for L. For our purposes any adjectives that apply to chains can be used with pseudochains. Stanley [23] has counted the number of maximal chains, $|\mathcal{C}|$, in β; then it follows that the number of maximal pseudochains is $(n!/2)|\mathcal{C}|$.

ii. If $L(p)$ is a maximal pseudochain and $m(L) = (I = P_0, \cdots, P_{\binom{n}{2}} = I^R)$ we write $L_i \equiv \{q \in L, I \,\beta\, m(q) \,\beta\, P_i\}$.

iii. For $A \subseteq S_\Sigma$, let $\operatorname{Cov}(A) \equiv \{(p, q) \in A \times A$, p covers q under $\beta\}$ and let $\lambda{:}\ \operatorname{Cov}(S_\Sigma) \rightarrow \{(x, y) \in \Sigma \times \Sigma, x < y\}$ be given by $\lambda(p, q) = (x, y)$ if $t(p) = q$ and $t = (x, y)$. λ is called a labelling of the edges in the Hasse diagram of $\langle S_\Sigma, \beta\rangle$. With these conventions let $\operatorname{TRAN}(A) \equiv \lambda(\operatorname{Cov}(A))$.

iv. G_n will denote the undirected (edge labelled) version of the Hasse diagram of $\langle S_\Sigma, \beta\rangle$, namely $G_n = (V, E) = (S_\Sigma, \operatorname{Cov}(S_\Sigma))$ where the edge $(p, t(p))$ is labelled by the two subset $\{x, y\}$ if $t = (x, y)$.

The following lemma states the equivalence between chains and pseudochains from the consistency point of view and it identifies pseudochains in $\langle S_\Sigma, \beta\rangle$ with shortest paths in G_n.

LEMMA 3.3. *Let* L(p) *be a pseudochain in* $\langle S_\Sigma, \beta\rangle$.

i. $\Psi(L(p))$ *is a consistent subset of* S_Σ (*see definition of* Ψ *in* § 2).

ii. *If* t, $l \in$ TRAN (L(p)) *then* $t \neq l$ *and* $t^{-1} \neq l$ (*if* t = (x, y), t^{-1} = (y, x)).

iii. L(p) *is a* saturated (*unrefinable*) *pseudochain from* p *to* q *if and only if* L(p) *is a* shortest path *from* p *to* q *in* G_n.

iv. *If* SPATH (p, q) *denotes a shortest path from* p *to* q *in* G_n *then* SPATH (p, q) *is consistent.*

Proof. For (i) note that L(p) is consistent because it is the image of a chain in β under a uniform relabelling, m, of the symbols of Σ, and chains in β are consistent by Lemma 3.1; therefore, $\Psi(L(p))$ is consistent.

For (ii) and (iii) note that if $p = p_1p_2 \cdots p_n \in S_\Sigma$ and $t = (p_i, p_{i+1})$ then $t(p) = p \cdot l(I)$ where $l = (i, i + 1)$. Now, left multiplication by a fixed permutation is an automorphism of S_Σ that preserves adjacency in the weak Bruhat order (for example, $p \to p^{-1} \cdot p = I$ and $t(p) \to p^{-1} \cdot t(p) = l(I)$); therefore, it does preserve distances. In particular a shortest path SPATH (p, q) is mapped by left multiplication to SPATH (I, $p^{-1} \cdot q$). But shortest paths, in G_n, from the identity I to any permutation w are saturated chains in β. This can be seen by induction on the path length which is nothing else than the number of inversions of w.

(iv) is just the result of putting (i) and (iii) together. □

The preceding lemma will allow us to state consistency results in terms of shortest paths in G_n even if we give proofs of them only in terms of chains in $\langle S_\Sigma, \beta\rangle$.

The following result gives information about certain subconfigurations of any consistent subset M of S_Σ. Note that no assumptions are made about the *connectivity* (in the graph sense) or *maximality* of M.

LEMMA 3.4. *Let* M *be a* consistent *subset of* S_Σ, $q \in M$, $p \in S_\Sigma$ *and let* SPATH (p, q) *and* SPATH′ (p, q) *be two different shortest paths from* p *to* q *such that* $t(p) \in$ SPATH (p, q), $t'(p) \in$ SPATH′ (p, q) *where* t *and* t′ *are two different adjacent transpositions* (*see* Fig. 3.2 *below*). *Under these conditions*, $\{t(p), t'(p)\} \subset M \to t \cap t' = \varnothing$.

Proof (by contradiction). (i) Assume that $t \cap t' \neq \varnothing$ and without loss of generality let t = (x, y), t′ = (y, z) and suppose that SPATH (p, q) and SPATH′ (p, q) are chains in $\langle S_\Sigma, \beta\rangle$. With these assumptions q becomes a *lower bound* for t(p) and t′(p) which means that the set of inversions of q, INV (q), contains INV (t(p)) ∪ INV (t′(p)); therefore, INV (q) ⊃ {(y, x), (z, y)}, which implies that (z, y, x) ∈ T(q) because SPATH and SPATH′ are shortest paths.

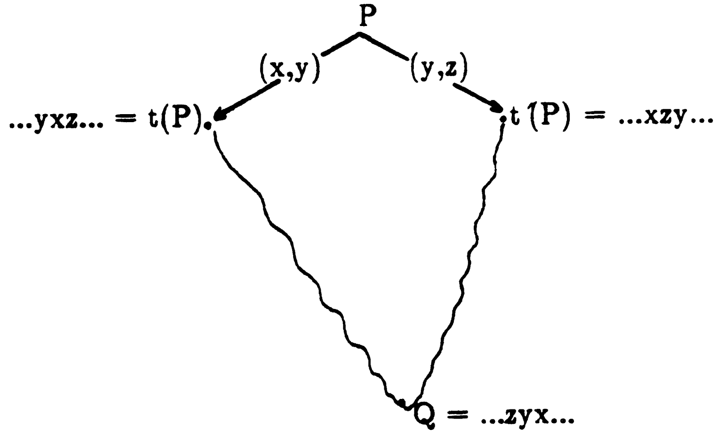


FIG. 3.2. *Illustration of Lemma* 3.4. *Note that* P *is not required to be in* M.

(ii) On the other hand, the fact that $t \cap t' \neq \emptyset$ forces $(y, x, z) \in T(t(p))$ and $(x, z, y) \in T(t'(p))$. (i) and (ii) together contradict the *consistency* of M. □

The fact that $\langle S_\Sigma, \beta \rangle$ is a lattice (Fact 1.iii) gives us the following corollary as a special case.

COROLLARY 3.1. *Let* $\{q, w, v\} \subset M \subset S_\Sigma$ *and let* t, t' *be two different adjacent transpositions.*

i. *If* $t(w \vee v) = w$, $t'(w \vee v) = v$, $w \beta q$, $v \beta q$ *and if* M *is consistent then* $t \cap t' = \emptyset$.

Dually we have,

ii. *If* $t(w) = w \wedge v$, $t'(v) = w \wedge v$, $q \beta w$, $q \beta v$ *and if* M *is consistent then* $t \cap t' = \emptyset$.

Proof. (i) and (ii) follow from the preceding lemma by taking $p = w \vee v$ and $p = w \wedge v$, respectively. □

Maximal consistent subsets in the *weak Bruhat order* exhibit a "local semimodularity" property which does not hold for the *strong Bruhat order*. This is stated precisely in the following corollary whose content will be referred to as *the Quadrilateral rule* or *the* Q *rule*.

COROLLARY 3.2 (the Quadrilateral rule). *Let* M *be a consistent subset of* S_Σ *and* $\{w, v\} \subset M$. *If there exist* $\{p, q\} \subset S_\Sigma$ *and two different adjacent transpositions* t *and* l *such that* $l(w) = q = t(v)$ *and* $t^{-1}(w) = p = l^{-1}(v)$ *then* $\{w, v, p, q\} \subset \Psi(M)$ (*see* Fig. 3.3).

Proof. The conditions imposed to l and t in the hypothesis hold if and only if $l \cap t = \emptyset$ and this in turn implies that $T(\{p, q\}) = T(\{w, v\}) \subset T(M)$; therefore, $\{p, q, w, v\} \subset \Psi(M)$ (this is not true if t and l are not adjacent transpositions and then it is not true in the *strong Bruhat* order). □

In terms of the weak Bruhat order, the Q rule says that for any two elements w, v of a maximal consistent set $\Psi(M)$, if their join, $w \vee v$, covers both w and v *and* if their meet, $w \wedge v$, is covered also by *both* w and v then $\{w, v, w \vee v, w \wedge v\} \subset \Psi(M)$. This resembles the definition of an Upper Semimodular lattice (Birkhoff [8]). However, the problem here is that *both* conditions $w \vee v \rightarrow \{w, v\}$ and $\{w, v\} \rightarrow w \wedge v$ are necessary, neither one implies the other, and moreover it is not true in general that $\Psi(M)$ is even a sublattice of $\langle S_\Sigma, \beta \rangle$. On the other hand, if M is a chain in β then $\Psi(M)$ is not only a sublattice but an upper semimodular one as will be established in Theorem 3.3.

The following result is basically an iterated application of the Quadrilateral rule.

THEOREM 3.1. *Let* M *be a* consistent *subset of* S_Σ *and let* $p, q \in \Psi(M)$ *such that* $p = uxyv$, $q = u'xyv'$ *where* $u' \in S_u$, $v' \in S_v$. *If there exists a shortest path* SPATH $(q, p) \subset \Psi(M)$ *such that for all* $w \in$ SPATH (q, p), $w^{-1}(x) < w^{-1}(y)$ *then for all* $w \in$ SPATH (q, p), $w = u''xyv''$ *where* $u'' \in S_u$, $v'' \in S_v$.

Proof (by induction on |SPATH (q, p)|).

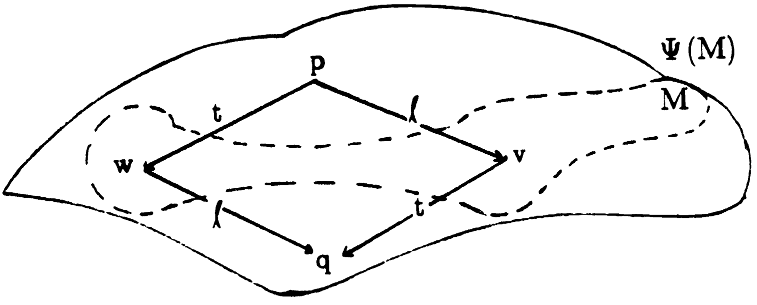


FIG. 3.3. *The Quadrilateral rule.*

Notation. If $p \in S_\Sigma$ and $a \in \Sigma$, denote by $p/_a$ the permutation in $S_{\Sigma-\{a\}}$ obtained by erasing a from p.

Basis. If $|\text{SPATH}(q, p)| = 1$ then there is nothing to prove.

(i) *Induction Hypothesis*. Assume it is true for $|\text{SPATH}(q, p)| = j \leqq k < \binom{|\Sigma|}{2}$ and let $|\text{SPATH}(q, p)| = k + 1$. Let $w' \in \text{SPATH}(q, p)$ and $l'(q) = w'$ where $l' \in \text{TRAN}(\text{SPATH}(q, p))$ and assume that $l' \cap \{x, y\} \neq \varnothing$. Without loss of generality let $l' = (a, x)$. By assumption $u' \in S_u$ and therefore a must precede x in p; therefore, there exists $l \in \text{TRAN}([w', p])$ such that $l = (x, a)$, ($[w, p]$ denotes the subpath of SPATH (q, p) running from w down to p). Take the first such l in TRAN ([w′, p]) and let w be the permutation in SPATH (q, p) to which l is applied, so $w = u''xav''$ and $w' = (u'/_a)xa(yv')$. Assume now that there exists $c \in u''$ such that $c \notin u'/_a$, so $c \neq a$ because $a \notin u''$ and $c \neq y$ because $w^{-1}(x) < w^{-1}(y)$ by hypothesis; therefore, $(c, x, a) \in T(w)$, $(x, a, c) \in T(w')$, which imply that $(c, a, x) \in T(l(w))$ and $(a, x, c) \in T(p) \cap T(q)$. This forces $[w', w]$ to contain a permutation w'' which contains the triple (x, c, a) because to go from w′ to w, a and c must be interchanged without interchanging (x, a) by the choice of l, and for c to precede x in w, at some point in $[w', w]$, c must be between x and a (the preceding argument depends exclusively on the connectivity of SPATH (q, p) and on the choice of $l = (x, a)$). Therefore, $\{w'', l(w), p\}$ contains a cyclic triple, namely, $\{(x, c, a), (c, a, x), (a, x, c)\}$ *contradicting* the consistency of M. Up to this point we have proved that symbols of $u'' \subset$ symbols of $u'/_a$ and by a symmetric argument we obtain that symbols of $u'/_a \subset$ symbols of u'', which means that $u'' \in S_{u'/_a}$, $w = u''xav''$, $w' = (u'/_a)xa(yv')$; therefore, the subpath $[w', w]$ has length $|[w', w]| \leqq k$ and satisfies the hypothesis of the theorem, so by Induction Hypothesis every permutation on it is of the form $u'''xav'''$ with $u''' \in S_{u'/_a}$, $v''' \in S_{yv'}$, and if $t \in \text{TRAN}([w', w])$ then $t \cap l = \varnothing$.

(ii) Now, the maximality of $\Psi(M)$, the fact that $[w', w] \subset \Psi(M)$, and (i) allow us to apply iteratively *the Quadrilateral rule* to get that $l([w', w]) \subset \Psi(M)$, giving us that the path $(q, l(w'), l([w', w]), [l(w), p])$ is a path from q to p that is shorter than SPATH (q, p), which is a contradiction; therefore, the original assumption that $l \cap \{x, y\} \neq \varnothing$ was false.

By (ii), $l \cap \{x, y\} = \varnothing$ and then $l(q)$ and p satisfy the hypothesis of the theorem, and by induction we will be done. □

Theorem 3.1, coupled with the results of § 2, gives the following characterization of *extensible consistent* subsets of S_Σ.

THEOREM 3.2 (see § 2 for related definitions). *Let* M *be a* consistent *subset of* S_Σ *which is also* extensible *by a pair* (t, p) *and let* $w \in t(\prod^M_{t,p})$. *If there exists a shortest path* $\text{SPATH}(t^{-1}(w), p) \subset \Psi(M)$ *and if* $l \in \text{TRAN}(\text{SPATH}(t^{-1}(w), p))$ *then* $l \cap t = \varnothing$. (*We will refer to this theorem as* the projection theorem).

(i) *Proof.* If $w \in t(\prod^M_{t,p})$ then $t^{-1}(w) \in \Psi(M)$ by Corollary 2.1 and by the definition of $\prod^M_{t,p}$.

Now, $p \in \prod^M_{t,p}$ and $\text{SPATH}(t^{-1}(w), p) \subset \Psi(M)$ satisfy the hypothesis of Theorem 3.1 because M is an *extensible consistent* subset of S_Σ; therefore, $\text{SPATH}(t^{-1}(w), p) \subset \prod^M_{t,p}$ which means that $l \cap t = \varnothing$ for every $l \in \text{TRAN}(\text{SPATH}(t^{-1}(w), p))$. □

The preceding theorem tells us that within each connected component of an *extensible* set, which is also *consistent*, the elements of $\prod^M_{t,p}$ *are precisely* those that are connected by paths all of whose transpositions are disjoint from t.

A lattice semimodular property of consistent sets. Recall that a lattice **L** is upper semimodular if it satisfies the following condition:

The **U.S.** Condition: For all elements w and v of **L** if w covers $w \wedge v$ then $w \vee v$ covers v. The following seemingly weaker condition is sufficient to prove upper semimodularity (Birkhoff [8]):

The **W.U.S.** Condition: For all elements w and v of **L**, if $w \wedge v$ is covered by both w and v then $w \vee v$ must cover both w and v.

As another application of the Q-rule we have the following result.

LEMMA 3.5. *Let* M *be a consistent subset of* $\langle S_\Sigma, \beta \rangle$. *If* $\Psi(M)$ *is a meet subsemilattice* (*join subsemilattice*) of $\langle S_\Sigma, \beta \rangle$ *with a maximum element* (*minimum element*) *then* $\Psi(M)$ *is an* upper semimodular sublattice *of* $\langle S_\Sigma, \beta \rangle$.

Proof. That $\Psi(M)$ is a meet sublattice with a maximum element automatically implies that $\Psi(M)$ is a lattice.

To prove that $\Psi(M)$ is upper semimodular is enough to prove that $\Psi(M)$ satisfies the **W.U.S.** condition. To this end let w and $v \in \Psi(M)$, $w \wedge v \in \Psi(M)$. Now let q be some upper bound for both v and w and assume that there are adjacent transpositions t and t′ such that $t(w) = w \wedge v$, $t'(v) = w \wedge v$ (i.e., $w \wedge v$ is covered by both v and w). The consistency of $\Psi(M)$ allows us then to apply Corollary 3.1 (ii) to conclude that $t \cap t' = \varnothing$, which in turn implies by the quadrilateral rule that the element $w \vee v = t^{-1}(v) \in \Psi(M)$ satisfies that $t'(w \vee v) = w$. This proves that $w \vee v$ covers both w and v which is the conclusion of the **W.U.S.** condition. □

Notation. For the remainder of this section we will follow the following notational conventions.

i. *Ch* will always denote a saturated chain (or pseudochain) $Ch = (P_0, P_1, \cdots, P_k)$ where $t_{i+1}(P_i) = P_{i+1}$ for $i = 0, \cdots, k-1$.

ii. $[P_0, P_i] \equiv \{p \in Ch \mid P_0 \,\beta\, p \,\beta\, P_i\}$; $Ch_i \equiv \Psi([P_0, P_i])$.

The following basic properties of the weak Bruhat order will be instrumental in the proof of the main result of this section.

LEMMA 3.6. *For* $p \in S_\Sigma$ *consider the set* E(p) *of noninversions of* p *as a binary relation on* Σ *and denote by* $(E(p))^*$ *its transitive closure. With these conventions, we have*:

i. $p \vee q$ *is the unique permutation satisfying that* $E(p \vee q) = (E(p) \cup E(q))^*$;

ii. *If* $p = uxyv$ *and* $q = u'xyv'$ *where* $x < y$, u *and* u′ *in* S_{Σ_1}, v *and* v′ *in* S_{Σ_2}, *then* $p \vee q = (u \vee u')xy(v \vee v')$;

iii. *If* $t = (x, y) \in E(p) \cap E(q)$ *and if* t *is an admissible transposition of* p *then* $p \vee q = t(p) \vee q$.

Proof.

i. For the proof, see Berge [7].

ii. Note that E(p) and E(q) differ only in E(u), E(u′), E(v), and E(v′), respectively. This forces $(E(p) \cup E(q))^*$ to be equal to $E((u \vee u')xy(v \vee v'))$, which together with (i) implies that $p \vee q = ((u \vee u')xy(v \vee v'))$.

iii. The fact that $(x, y) \in E(p) - E(t(p))$, $E(t(p)) \subset E(p)$, and $(x, y) \in E(q)$ implies that $E(t(p)) \cup E(q) = E(p) \cup E(q)$ and again by (i), $t(p) \vee q = p \vee q$. □

Theorem 3.2 (the projection theorem) and the Q-rule, together with the fact that $[P_0, P_i]$ is a saturated chain (or pseudochain) imply that $Ch_i = \Psi([P_0, P_i])$ is a connected subset of S_Σ.

Now, if $i = 1$, $\Psi([P_0, P_i]) = (P_0, P_1)$, which is clearly a join sublattice with top element P_0. For the general case note that $Ch_{k+1} - Ch_k = t_{k+1}(\prod^{Ch_k}_{t_{k+1},P_k})$ by Corollary 2.2. But this is saying that $Ch_{k+1} - Ch_k$ is obtained from $\prod^{Ch_k}_{t_{k+1},P_k}$ by right multiplication by a fixed permutation, namely the one corresponding to the transposition t_{k+1}. Moreover, if two elements are adjacent in $\prod^{Ch_k}_{t_{k+1},P_k}$, their images under t_{k+1} must be adjacent. So we have here a one-to-one mapping that preserves adjacencies and therefore distances under β. Therefore, if $v, w \in Ch_{k+1} - Ch_k$ then $t^{-1}_{k+1}(w)$ and $t^{-1}_{k+1}(v) \in \prod^{Ch_k}_{t_{k+1},P_k}$, and by Lemma 3.6 (ii) we can assume that $z = t^{-1}_{k+1}(w) \vee t^{-1}_{k+1}(v) \in \prod^{Ch_k}_{t_{k+1},P_k}$, which allows us to conclude that $t_{k+1}(z) = w \vee v \in Ch_{k+1} - Ch_k$. If $v \in Ch_k$ and $w \in Ch_{k+1} - Ch_k$, then the fact that Ch_k is extensible by (t_{k+1}, P_k) allows us to apply Lemma 3.6 (iii) by letting $q = v$ and $p = t_{k+1}^{-1}(w)$ to obtain that $v \vee w \in Ch_{k+1}$.

The preceding arguments show that Ch_i is a sublattice of $\langle S_\Sigma, \beta\rangle$ with top element, and therefore by Lemma 3.5 we have the following promised result.

THEOREM 3.3. *If* M *is a saturated chain in the weak Bruhat order then* $\Psi(M)$ *is an upper semimodular sublattice of* $\langle S_\Sigma, \beta\rangle$.

Remarks. The preceding results play a central role in the algorithmic construction of maximal consistent sets which contain a saturated chain (or pseudochain) *Ch* in $\langle S_\Sigma, \beta\rangle$. It says that if $Ch_i \equiv \Psi([P_0, P_i])$ has been constructed then to find $\prod_{t_{i+1},P_i}^{Ch_i}$ one backtracks (in Ch_i) from P_i by following any path whose transpositions are disjoint from t_{i+1}. At every step all that is required is to find one incoming transposition l disjoint from t_{i+1}. Theorem 3.3 guarantees that the process will stop if and only if at some point we reach one permutation all of whose incoming transpositions intercept t_{i+1} (the formal algorithm can be found in Abello [1], [4], where it is called the MCCS algorithm).

4. Weak Bruhat order, consistent sets and Catalan numbers. We will prove here that the *n*th Catalan number is an upper bound for those consistent sets containing a Maximal pseudochain in the weak Bruhat order.

DEFINITION 4.1.

i. If M is a connected subset of S_Σ, its diameter, diam (M), is defined as $\operatorname{diam}(M) \equiv \max_{\{P,Q\}\subset M} |\mathrm{SPATH}(P, Q)|$.

ii. For a saturated chain (pseudochain) $Ch = [P, Q]$ in $\langle S_\Sigma, \beta\rangle$ denote by OTRAN (*Ch*) the *ordered* set of transpositions used in *Ch*, namely $\mathrm{OTRAN}(Ch) \equiv \{t_i\}_{i=|Ch|-1}$ where $t_{i+1}(P_i) = P_{i+1}$, $P_i \in Ch$; and let Ch^x be the *subsequence* of OTRAN (*Ch*) consisting of transpositions involving $x \in \Sigma$. Elements of Ch^x will be distinguished by having a superscript x, namely, $Ch^x = (t_1^x, t_2^x, \cdots, t_j^x)$.

iii. $[l_j, l_k] \equiv \{l_i \in \mathrm{OTRAN}(Ch) \text{ such that } j \leqq i \leqq k\}$.

iv. For a *subsequence* $(l_1, l_2, \cdots, l_j)$ of Ch^x and a permutation Q, we will write $(l_j, \cdots, l_1)(Q)$ to denote the *sequence* of permutations $(Q = Q_0, Q_1, \cdots, Q_j)$ where $Q_{i+1} = l_i(Q_i)$ for $i = 1, \cdots, j-1$.

The following is a technical lemma that will allow us to single out a very special canonical subchain in M_{Ch}.

FACT 4.1. *Assume that* $[p, v]$ *is a saturated chain in* $\langle S_\Sigma, \beta\rangle$ *such that* $p_1 = v_i = x \in \Sigma$ *and* $p_n = v_{i+1} = y \in \Sigma$ *and let us recall that if* $p \in S_\Sigma$, $\tau(p)$ *denotes its admissible set of transpositions. If* $t_q, t_r \in \mathrm{OTRAN}([p, v])$ *are such that* $t_q = t_1^x = (x, a)$, $t_r = t_2^x = (x, b)$, $a \neq y$, $b \neq y$ *with* $t_q \in \tau(Q)$, $t_r \in \tau(R)$ *and* $[Q, R] \subset [p, v]$ *then* $M_{[p,R]} = M_{[p,Q]\cup(t_q,t_{r-1},\cdots,t_{q+1})(Q)}$. *We will say in this case that the sequence* $(t_{q+1}, \cdots, t_{r-1})$ *has been lifted by the transposition* t_q (*see Fig*. 4.1).

Proof. $M_{[p,R]} = M_{[p,Q]\cup(t_{r-1},\cdots,t_{q+1},t_q)(Q)}$ by the definition of t_q and t_r.

(i) $t_r = t_2^x \rightarrow$ each transposition in $(t_{q+1}, \cdots, t_{r-1})$ does not involve x.

(ii) $t_q = t_1^x$ and the assumption that $[Q, R]$ is a chain $\rightarrow$ each transposition in $(t_{q+1}, \cdots, t_{r-1})$ does not involve the symbol a.

Therefore, the Quadrilateral rule (Corollary 3.2), can be applied (iteratively) to $(t_{q+1}, \cdots, t_{r-1})$ by (i) and (ii) and the result follows by the maximality of $M_{[p,R]}$. □

Remark. The idea of lifting one sequence, by one transposition (Fact 4.1), can be used iteratively, in certain cases, to lift one sequence by another as follows. Consider two permutations p and q such that $p \,\beta\, q$, $p_j = q_j = x$ and assume that there is a saturated chain *Ch* from p to q such that if $t = (a, b) \in \mathrm{OTRAN}(Ch)$ then $a \neq x \neq b$. Now, let $\mathrm{LEFT}(Ch) = (t_{i_1}, \cdots, t_{i_k})$ denote the subsequence of OTRAN (*Ch*) obtained by deleting from it those transpositions using symbols in $\{p_1, \cdots, p_{j-1}\}$. Simi-

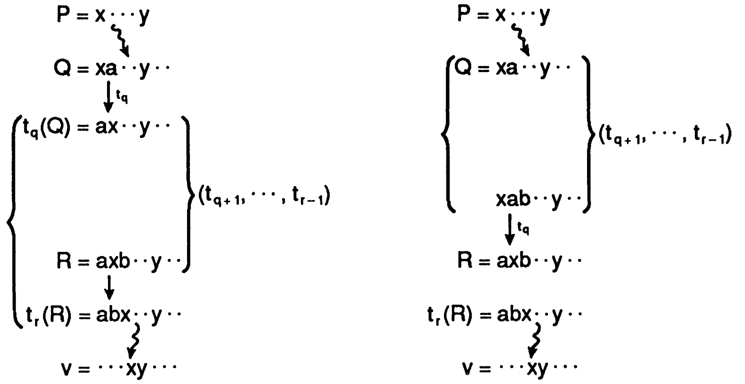


FIG. 4.1. *The lifting of a sequence* $(t_{q+1}, \cdots, t_{r-1})$ *by a transposition* t_q.

larly, let RIGHT (*Ch*) = $(t_{j_1}, \cdots, t_{j_{k'}})$ denote the subsequence of OTRAN (*Ch*) obtained by deleting from it those transpositions using symbols in $\{p_{j+1}, \cdots, p_k\}$. (For our purpose assume that both LEFT (*Ch*) and RIGHT (*Ch*) are nonempty and that the last transposition of OTRAN (*Ch*) is an element of RIGHT (*Ch*)). Note that if $t \in$ LEFT (*Ch*) and $t' \in$ RIGHT (*Ch*) then $t \cap t' = \emptyset$. This together with the assumption that *Ch* is a saturated chain in β all of whose elements have the symbol x exactly in the same position implies that the sets of permutations $(t_{i_k}, \cdots, t_{i_1})(p)$ and $(t_{j_{k'}}, \cdots, t_{j_1})(p)$ are saturated chains in $\langle S_\Sigma, \beta\rangle$. This can be seen by an iterated lifting of certain subsegments of the sequence LEFT (*Ch*) by each of the elements of RIGHT (*Ch*) (in reverse order) in an iterated fashion. The figure below illustrates this process for the case where RIGHT (*Ch*) consists of two transpositions only. Note that because here we use only the Quadrilateral rule, then the set of ordered triples of $(t_{i_k}, \cdots, t_{i_1})(p)$, $T((t_{i_k}, \cdots, t_{i_1})(p))$, together with the set of ordered triples of $(t_{j_{k'}}, \cdots, t_{j_1})(p)$, $T((t_{j_{k'}}, \cdots, t_{j_1})(p))$, is precisely equal to the set of ordered triples of *Ch*, T(*Ch*).

Note that because the process depicted in Fig. 4.2 consists of repeated applications of the Quadrilateral rule, we can be sure that all the saturated chains *Ch'* from p to q that are obtained in this manner satisfy that T(*Ch'*) = T(*Ch*) which means that *Ch'* $\subset$ Ψ(*Ch*). In particular this is true for the chain determined by using first (in order) the transpositions of LEFT (*Ch*) and then the transpositions of RIGHT (*Ch*), which in our unwanted (very clumsy) notation is denoted by $((t_{j_{k'}}, \cdots, t_{j_1})(t_{i_k}, \cdots, t_{i_1}))(p)$.

We collect the preceding remarks and the process depicted in Fig. 4.2 in the following result.

FACT 4.2. Let p, q be permutations in S_Σ that satisfy p β q, $p_j = q_j = x$ and let *Ch* denote a saturated chain from p to q such that if t = (a, b) $\in$ OTRAN (*Ch*) then $a \neq x \neq b$. Under these conditions it is possible to find a saturated chain *Ch'* from p to q such that:

i. OTRAN (*Ch'*) consists first of all tranpositions in OTRAN (*Ch*) which use only symbols in $\{p_{j+1}, \cdots, p_n\}$ (call this set LEFT (*Ch*)) followed by all transpositions in OTRAN (*Ch*) using only symbols in $\{p_1, \cdots, p_{j-1}\}$ (call this set RIGHT (*Ch*)) (or vice versa). In symbols: OTRAN (*Ch'*) = (LEFT (*Ch*), RIGHT (*Ch*)) or OTRAN (*Ch'*) = (RIGHT (*Ch*), LEFT (*Ch*)).

ii. T(*Ch'*) = T(*Ch*) or equivalently *Ch'* $\subset$ Ψ(*Ch*).

iii. (a) If RIGHT (*Ch*) = $(t_{j_1}, \cdots, t_{j_{k'}})$ then all the permutations in the set $(t_{j_{k'}}, \cdots, t_{j_1})(p)$ have as a common suffix the subpermutation $p_{j+1} \cdots p_n$. By deleting this common suffix from all of them we obtain a saturated pseudochain in

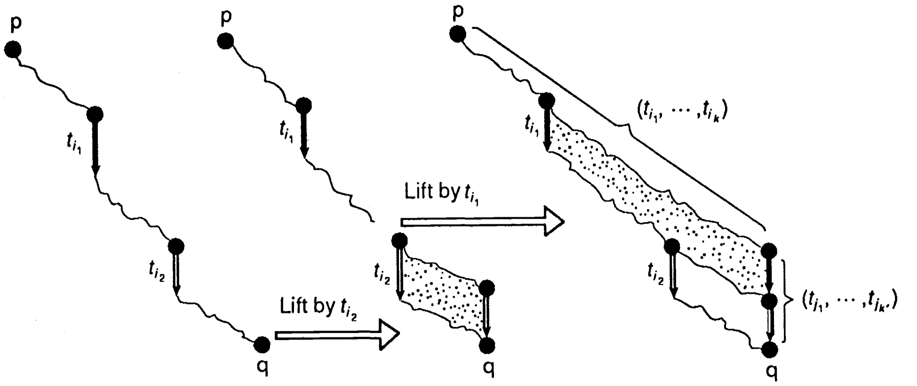


FIG. 4.2. *Lifting of a sequence* LEFT(*Ch*) *by another sequence* RIGHT(*Ch*) = (t_{i_1}, t_{i_2}). *This assumes that all the elements in the chain Ch from* p *to* q *contain a fixed symbol x in exactly the same position.*

$\langle S_{\{p_1, \cdots, p_{j-1}\}}, \beta\rangle$ from $p_1 \cdots p_{j-1}$ to $q_1 \cdots q_{j-1}$. Call this pseudochain RESTRICTED_RIGHT (*Ch*) and its closure FIRST_HALF $\Psi(Ch)$.

(b) If LEFT $(Ch) = (t_{i_1}, \cdots, t_{i_k})$ then all the permutations in the set $(t_{i_k}, \cdots, t_{i_1})(p)$ have as a common prefix $p_1 \cdots p_{j-1}$. By deleting this common prefix from all of them we obtain a saturated pseudochain in $\langle S_{\{p_{j+1}, \cdots, p_n\}}, \beta\rangle$ from $p_{j+1} \cdots p_n$ to $q_{j+1} \cdots q_n$. Call this pseudochain RESTRICTED_LEFT (*Ch*) and its closure SECOND_HALF $\Psi(Ch)$.

As a justification (if any) for the definitions given in (a) and (b) above we have the following:

(c) For a chain *Ch* satisfying the restrictions given above we have that $\Psi(Ch)$ = FIRST_HALF $(\Psi(Ch)) \times [x] \times$ SECOND_HALF $(\Psi(Ch))$, (here $\times$ denotes cross product).

Note. Everything we have discussed after Fact 4.1 is put very concisely in the following definition and theorem. However, if the reader feels comfortable he/she may jump directly to the remarks preceding Theorem 4.2 without losing continuity.

DEFINITION 4.2.

i. For $(t_{i_1}, t_{i_2}, \cdots, t_{i_j})$ a subsequence of OTRAN$([P, P^R])$ such that $(t_{i_1}, t_{i_2}, \cdots, t_{i_j}) = (t_1^x, t_2^x, \cdots, t_j^x)$ denote by $\{Q_l\}_{l=1}^{j}$ the subchain of $[P, P^R]$ such that $t_l^x \in \tau(Q_l)$.

ii. Let LEFT (t_l^x, t_{l+1}^x) denote the subsequence of $[t_{i_l}, t_{i_{l+1}}]$ obtained by deleting from it those transpositions using symbols that precede x in Q_l. Similarly, let RIGHT (t_l^x, t_{l+1}^x) denote the subsequence of $[t_{i_l}, t_{i_{l+1}}]$ obtained by deleting from it those transpositions using symbols that follow x in Q_l.

iii. Let TRANSFORM $(t_l^x, t_{l+1}^x) \equiv$ (RIGHT (t_l^x, t_{l+1}^x), LEFT (t_l^x, t_{l+1}^x), t_l^x) and TRANSFORM $(t_l^x, t_j^x) \equiv$ (TRANSFORM (t_l^x, t_{l+1}^x), TRANSFORM (t_{l+1}^x, t_{l+2}^x), $\cdots$, TRANSFORM (t_{j-1}^x, t_j^x)).

The following result is just an iterated application of Fact 4.1 in which a sequence was lifted by one transposition. In the following theorem a sequence is lifted by another sequence.

THEOREM 4.1. *If* $(t_{i_1}, t_{i_2}, \cdots, t_{i_j}) = (t_1^x, t_2^x, \cdots, t_j^x)$ *with* $t_1^x \in \tau(Q)$, $t_j^x = t_s \in \tau(S)$*and* $[Q, S] \subset [p, v]$ *then* $M_{[p,S]} = M_{[p,Q] \cup (t_j^x, \text{TRANSFORM}\,(t_1^x, t_j^x))(Q)}$.

Proof. (By induction on j).

Basis. If j = 2, the result follows from Fact 4.1.

Induction Hypothesis. Assuming the result is true for j, we will prove it for j + 1.

Suppose $(t_{i_1}, \cdots, t_{i_j}, t_{i_{j+1}}) = (t_1^x, t_2^x, \cdots, t_j^x, t_{j+1}^x)$ and let $t_{i_j} \in \tau(R)$, $t_{i_{j+1}} \in \tau(u)$ with $[p, R] \cup [R, u] \subset [p, v]$. By Induction Hypothesis $M_{[p,R]} =$

$M_{[p,Q] \cup (t_j^x, \mathrm{TRANSFORM}\,(t_1^x, t_j^x))(Q)}$. By definition of t_{j+1}^x every transposition in OTRAN $[t_{i_j}(R), u]$ does not use the symbol x. This implies that the quadrilateral rule may be applied to OTRAN $[t_{i_j}(R), u]$ to lift the transpositions in LEFT (t_j^x, t_{j+1}^x) by those transpositions in RIGHT $(t_j^x, t_{j+1}^x)/t_{j+1}^x$. But this means that instead of OTRAN $[t_{i_j}(R), u]$ we may use (RIGHT (t_j^x, t_{j+1}^x), LEFT (t_j^x, t_{j+1}^x)). Therefore, $M_{[p,u]} = M_{[p,Q] \cup (t_{j+1}^x, \mathrm{RIGHT}\,(t_j^x, t_{j+1}^x), \mathrm{LEFT}\,(t_j^x, t_{j+1}^x), (t_j^x, \mathrm{TRANSFORM}\,(t_1^x, t_j^x)))(Q)}$ by Induction Hypothesis and by the maximality of $M_{[p,u]}$. Now, by noticing that the right-hand side of the last equation is equal to $M_{[p,Q] \cup (t_{j+1}^x, \mathrm{TRANSFORM}\,(t_1^x, t_{j+1}^x))}$ the result follows. □

Remarks. We have seen that a shortest path SPATH (p, q) is mapped bijectively to a saturated chain *Ch* in $\langle S_\Sigma, \beta \rangle$ by left multiplication by p^{-1}. This induces a map from the ordered triples of SPATH (p, q), T(SPATH (p, q)), to the ordered triples of *Ch*, T(*Ch*); namely, if R ∈ SPATH (p, q), (x, y, z) ∈ T(R) if and only if $(p^{-1}(x), p^{-1}(y), p^{-1}(z)) \in T(p^{-1} \cdot R)$. But this means that $w \in \Psi(\mathrm{SPATH}\,(p, q))$ if and only if $p^{-1} \cdot w \in \Psi(Ch)$; therefore, $|\Psi(\mathrm{SPATH}\,(p, q))| = |\Psi(Ch)|$. Therefore, for every *maximal connected consistent set* (m.c.c.s.) $M \subset S_\Sigma$ of diameter $\binom{n}{2}$ where $n = |\Sigma|$ there exists a m.c.c.s. $M' \subset S_\Sigma$ that contains a maximal chain such that $|M| = |M'|$. This is not saying that all such sets (with the same diameters) have the same cardinality (in fact their cardinalities are in general quite different as proved in Abello [2–4]). With this in mind we will denote by M_j any maximal connected consistent subset of S_Σ where $|\Sigma| = j$. Now if M_j has diameter $\binom{j}{2}$ we may assume that it contains a maximal chain under β.

Finally, we will prove the next result which relates *Catalan numbers* and *maximal connected consistent sets*.

THEOREM 4.2. *For* $|\Sigma| = n$. *If* M_n *denotes a maximum connected consistent subset of* $\langle S_\Sigma, \beta \rangle$ *of diameter*, diam $(M_n) = \binom{n}{2}$ *then* $\langle M_n, \beta \rangle$ *is an upper semimodular lattice with cardinality* $|M_n| < (1/n + 1)\binom{2n}{n} =$ *the* nth *Catalan number* C_n *for* $n > 2$.

Proof. The upper semimodularity of $\langle M_n, \beta \rangle$ was established in the preceding section (Theorem 3.3), so we will prove here that $|M_n| \leqq C_n$.

For simplicity in notation we will write $\prod_t^B$ to denote the projection set $\prod_{t,P}^B$ of *B* with respect to (*t*, *P*), if there is no danger of confusion.

(i) By the remarks preceding this theorem we may assume that M_n contains a maximal chain $Ch = [I, I^R]$ in β. Let $I_1 = x \in \Sigma$ and $I_n = y \in \Sigma$. By noting that x never moves to the left in *Ch* we have that OTRAN $(Ch) = (t_1, \cdots, t_{\binom{n}{2}})$ imposes a total order < on $\Sigma - x$ given by $b_i < b_j$ if and only if $t_i = (x, b_i)$, $t_j = (x, b_j)$ and $i < j$.

(ii) Now, by letting $M^i = \{w \in M_n \colon w_i = x\}$ we have an ordered partition of M, namely, $(M^1, \cdots, M^n)$ and $\exists\, u \in M^i$ such that $t_i(u) \in M^{i+1}$ where $t_i = (x, b_i)$ and b_i is as defined in (i).

(iii) By the *projection theorem* (Theorem 3.2), the definition of M^i and (ii), we have that $\prod_{t_i}^{M^i} \subset M^i$ and $t_i(\prod_{t_i}^{M^i}) \subset M^{i+1}$.

(iv) On the other hand, if $v \in M^{i+1}/t_i(\prod_{t_i}^{M^i})$ then the set of symbols $\{v_l, l < i + 1\} = \{b_l, l < i + 1\}$ by (i) and by the order imposed on *Ch*.

(v) (iii), (iv), and the fact that $v_{i+1} = x$ allow us to conclude that
$M^{i+1} \subseteq \Psi(Ch^i)$

(vi) where Ch^i is the saturated chain of *Ch* between $t_{i-1}(p)$ and $t_i^{-1}(q)$, with the understanding that $t_0(p)$ should be taken as I. By Fact 4.2 (iii) (c) we know that $\Psi(Ch^i) =$ FIRST_HALF $(\Psi(Ch^i))$ x $\{x\}$ x SECOND_HALF $(\Psi(Ch^i))$ where FIRST_HALF $(\Psi(Ch^i)) \subset S_{\{b_l, l < i+1\}}$ and SECOND_HALF $(\Psi(Ch^i)) \subset S_{\Sigma - \{b_l, l < i+1\}}$ are consistent and connected sets, each of which contains a pseudochain. Therefore, $|$FIRST_HALF $(\Psi(Ch^i))| \leqq |M_i|$ and $|$SECOND_HALF $(\Psi(Ch^i))| \leqq |M_{n-i-1}|$, which in turn imply by (v) that $|M^{i+1}| \leqq |M_i| * |M_{n-i-1}|$.

(vii) This, together with (ii) above, give us $|M_n| = \sum_{i=0}^{n-1} |M^{i+1}| \leqq \sum_{i=0}^{n-1} |M_i| * |M_{n-i-1}|$ with $|M_0| = 1$, $|M_1| = 1$, $|M_2| = 2$, $|M_3| = 4$.

Inequality (vii) and the fact that the Catalan numbers $\{C_n\}$ satisfy that $C_n = \sum_{i=0}^{n-1} C_i * C_{n-i-1}$ with the same boundary conditions allow us to apply induction on n to get that $|M_n| < C_n$ for every $n > 2$. □

COROLLARY 4.1. *If* M_n *is a maximal consistent subset of* S_Σ *of diameter* diam $(M_n) = \binom{n}{2}$ *then* $|M_n| < 4^{n-1}$.

Proof. The proof follows from the preceding theorem and from the fact that $C_n \leqq 4^{n-1}$. □

Remarks. The preceding results suggest the possibility of studying the structure of *maximal consistent* sets by looking at them as representing a certain restricted collection of binary trees or as a certain subcollection of stack permutations (de Bruijn [11]). The multiple interpretations offered in the literature to the Catalan numbers, C_n, (de Bruijn [11], Feller [14], Gardner [16], Klamer [18]), could be a good source of ideas to shed new light on the problem in question. This approach has not yet been pursued.

The unexpected relationship between C_n and $|M_n|$ established in Theorem 4.3 offers the (unique) best known upper bound at present. In a forthcoming paper we will prove that $|M_n|$ is not bounded by 2^n for all n, as was conjectured in [2]. We conjecture that in general *any consistent set* $M \subset S_\Sigma$ *satisfies that* $|M| < 4^{|\Sigma|-1}$ *for* $|\Sigma| > 2$ *and that if M contains a maximal pseudochain in the weak Bruhat order then* $|M| < 3^{|\Sigma|-1}$.

We suspect that a general bound for *connected consistent* sets between $3^{|\Sigma|-1}$ and $4^{|\Sigma|-1}$ is a very hard result to obtain because the structure of general connected sets is as random as that of unconnected ones. Moreover, relating *connected consistent* sets to *unconnected* ones appears to be a very hard problem. In Abello [1] we present a very surprising bijection of this type that gives a unified view of several constructions (connected and unconnected) offered in the past.

Conclusions. We have seen that maximal pseudochains in $\langle S_\Sigma, \beta \rangle$ are a very important substructure of those *maximal consistent sets* which contain them. From the *Arrow's Impossibility Theorem* point of view (Abello [4], Arrow [5]), the results obtained here indicate that the majority rule produces transitive results if the collection of voters as a whole (at least in the *extensible* cases covered by Theorem 3.2), can be partitioned into no more than $(n^2 + n)/2$ groups that can be ordered according to the level of disagreement they have with respect to a fixed permutation p. On the other hand, by viewing S_Σ as a *Coxeter* group (Benson and Grove [6], Bourbaki [10], Coxeter and Moser [13], Stanley [23]), these results provide a "*novel*" *interpretation* of the following partition of the collection Ω of maximal chains in the *weak Bruhat order*. Namely, if for *Ch* and $Ch' \in \Omega$ we let M_{Ch} and $M_{Ch'}$ be the maximal consistent sets containing them, respectively, then the relation $\sim$ given by $Ch \sim Ch'$ if and only if $M_{Ch} = M_{Ch'}$ partition Ω and our results say that $\langle \cup_{Ch' \sim Ch} Ch', \beta \rangle$ is an *upper semimodular sublattice* of $\langle S_\Sigma, \beta \rangle$ such that $\lceil \cup_{Ch' \sim Ch} Ch' \rceil \leqq$ the $|\Sigma|$th Catalan number. Now, if $\gamma = (t_1, \cdots, t_i)$ is a *reduced decomposition of* w_0 = minimum element in $\langle S_\Sigma, \beta \rangle$, any other reduced decomposition of w_0 may be obtained from γ by using two types of transformations known as Coxeter relations of type I and of type II (see Benson and Grove [6]). Our Projection Theorem (Theorem 3.2) shows that $Ch \sim Ch'$ if and only if Ch' may be obtained from *Ch* by using transformations of type I only; therefore, we have obtained a "new" combinatorial interpretation of the collection of chains which can be obtained from one another by using Coxeter transformations of type I or type II exclusively. Namely, for $Ch' \in \Omega$, if $\Omega_{Ch'} = \{Ch \in \Omega$: *Ch* can be obtained from Ch' by using Coxeter transformations of type I only$\}$ then the set $\cup_{Ch \in \Omega_{Ch'}} Ch$ does not contain a cyclic triple (or Latin square) in the sense of Definition 1.1.

If one is puzzled by the fact that we never said what these transformations were, it should suffice to say that what we call transformations of type I correspond to interchanging t_i and t_{i+1}, in the reduced decomposition γ of w_0, if and only if they are "disjoint."

We close with the following question: What is the corresponding combinatorial interpretation of the projection theorem for general coxeter groups?

Acknowledgments. We thank professors Gill Williamson, Anders Bjorner, Adriano Garsia, and Jeff Remmel for valuable suggestions, an anonymous referee whose detailed comments improved the presentation, and Ms. Ruth Cintron for encouragement throughout this project.

REFERENCES

[1] J. M. ABELLO, *Algorithms for consistent sets*, Congressus Numerantium, 53 (1987), pp. 23–38.

[2] ———, *Intrinsic limitations of the majority rule, an algorithmic approach*, SIAM J. Algebraic Discrete Meth., 6 (1985), pp. 133–144.

[3] J. M. ABELLO AND C. R. JOHNSON, *How large are transitive simple majority domains?*, SIAM J. Algebraic Discrete Meth., 5 (1984), pp. 603–618.

[4] J. M. ABELLO, *A study of an independence system arising in group choice via the weak Bruhat order*, Ph.D. Thesis, University of California, San Diego, CA, 1985.

[5] K. J. ARROW, *Social Choice and Individual Values*, John Wiley, New York, 1951.

[6] C. T. BENSON AND L. C. GROVE, *Finite Reflection Groups*, Bogden and Quigley, New York, 1971.

[7] C. BERGE, *Principles of Combinatorics*, Academic Press, New York, 1971.

[8] G. BIRKOFF, *Lattice Theory*, Amer. Math. Soc. Colloq. Publ. No. 25, American Mathematical Society, Providence, R.I., 1967.

[9] D. J. BLACK, *The Theory of Committees and Elections*, Cambridge Press, London, 1958.

[10] N. BOURBAKI, *Groupes et algébres de Lie*, chapters 4–6, Fascicule XXXIV, Eléments de mathématique, Hermann, Paris, 1968.

[11] N. G. DE BRUIJN AND B. J. M. MORSELT, *A note on plane trees*, J. Combin. Theory, 2 (1967), pp. 27–34.

[12] MARQUIS DE CONDORCET, *Essai sur l'Application de l'Analyse à la Probabilité des Decisions Rendues à la Pluralité des Voix*, Paris, 1785.

[13] H. S. M. COXETER AND W. O. J. MOSER, *Generators and Relations for Discrete Groups*, 2nd edition, Springer-Verlag, New York, 1965.

[14] W. FELLER, *An Introduction to Probability Theory and Its Applications*, John Wiley, New York, 1950.

[15] P. C. FISHBURN, *Conditions for simple majority decision with intransitive individual indifference*, J. Econom. Theory, 2 (1970), pp. 354–367.

[16] M. GARDNER, *Mathematical games, Catalan numbers: an integer sequence that materializes in unexpected places*, Scientific American, 234 (1976), pp. 120–125.

[17] I. J. GOOD, *The number of ordering of n candidates when ties are permitted*, Fibonacci Quart., 13 (1975), pp. 11–18.

[18] D. A. KLAMER, *Correspondence between plane trees and binary sequences*, J. Combin. Theory, 9 (1970), pp. 401–411.

[19] E. L. LEHMANN, *Some concepts of dependence*, Ann. Math. Statist., 37 (1966), pp. 1137–1153.

[20] W. H. RIKER, *Arrow's theorem and some examples of the paradox of voting*, Foundation monograph, Southern Methodist University Press, Dallas, 1961.

[21] I. R. SAVAGE, *Contributions to the theory of rank order statistics, the "trend" case*, Ann. Math. Statist., 28 (1957), pp. 968–977.

[22] ———, *Contributions to the theory of rank order statistics: Application of lattice theory*, Rev. Internat. Statist. Inst., 32 (1964), pp. 52–64.

[23] R. P. STANLEY, *On the number of reduced decompositions of elements of Coxeter groups*, Europ. J. Combinatorics, 5 (1984), pp. 359–372.

[24] S. WILLIAMSON, *Combinatorics for Computer Science*, Computer Science Press, MD, 1985.

[25] B. WARD, *Majority voting and alternative forms of public enterprises*, in *The Public Economy of Urban Communities*, J. Margolis, ed., Johns Hopkins Press, Baltimore, 1965, Chapter 6, pp. 112–126.

[26] T. YANAGIMOTO AND M. OKAMOTO, *Partial ordering of permutations and monotonicity of a rank correlation statistic*, Ann. Inst. Statist. Math., 21 (1969), pp. 489–506.

SIAM J. DISC. MATH.
Vol. 4, No. 1, pp. 17–27, February 1991

002

AN EXTREMAL PROBLEM ON SPARSE 0-1 MATRICES*

DAN BIENSTOCK† AND ERVIN GYÖRI‡

Abstract. The problem of estimating the number of 1's in a square 0-1 matrix with certain forbidden configurations is considered, and nearly tight bounds are provided. This is motivated by a problem in computational geometry.

Key words. extremal problems

1. Introduction. In this paper we study a problem of a nature typical to extremal combinatorics: that of finding the maximum number of 1's that can occur in a 0-1 $n \times n$ matrix with a certain forbidden configuration of 1's.

We remark that various problems in extremal graph theory may be described in a similar way. For example, Zarankiewicz's problem, which asks for the largest number of edges in an n-vertex graph with no 4-circuit, can be stated as follows. What is the maximum number of 1's in a 0-1 $n \times n$ matrix with 0's on the main diagonal, that contains no "rectangle" with a 1 at each corner? It is well known that the tight answer is $n^{3/2}$ (see [B]).

We call the forbidden configurations in our problem *trapezoids*. For integers $1 \leqq i_1 \leqq i_2 < i_3 \leqq n$ and $1 \leqq j_1 < j_2 \leqq j_3 \leqq n$, a trapezoid is a pattern of four 1's, occurring at entries (given by (row, column) in standard numbering) (i_1, j_1), (i_2, j_2), (i_3, j_1), and (i_3, j_3) (see Fig. 1). We denote by t_n the maximum number of 1's in a trapezoid-free $n \times n$ matrix. The problem of computing t_n has appeal of its own, as the proofs are not immediate. Moreover, this problem arises in computational geometry, as outlined next.

Recently, Mitchell produced an algorithm for computing a shortest rectilinear path to join two points in the plane while avoiding certain rectilinear obstacles. The complexity of this algorithm is difficult to estimate, but may be shown to be bounded above (up to other, unrelated factors) by t_n [M1]. Thus, it is important to investigate t_n. Our results, described below, imply that Mitchell's algorithm is one of two best algorithms for the geometry problem. We prove the following theorem.

THEOREM 1.

(i) *There exists a constant* $c_1 > 0$, *so that for all* $n \geqq 1$, $t_n \leqq c_1\, n \log n$.

(ii) *There exists a constant* $c_2 > 0$, *so that for infinitely many* $n \geqq 1$, $t_n \geqq c_2\, n \log n / \log \log n$.

We conjecture that the lower bound in (ii) is the correct answer.

We will use the following definitions and conventions.

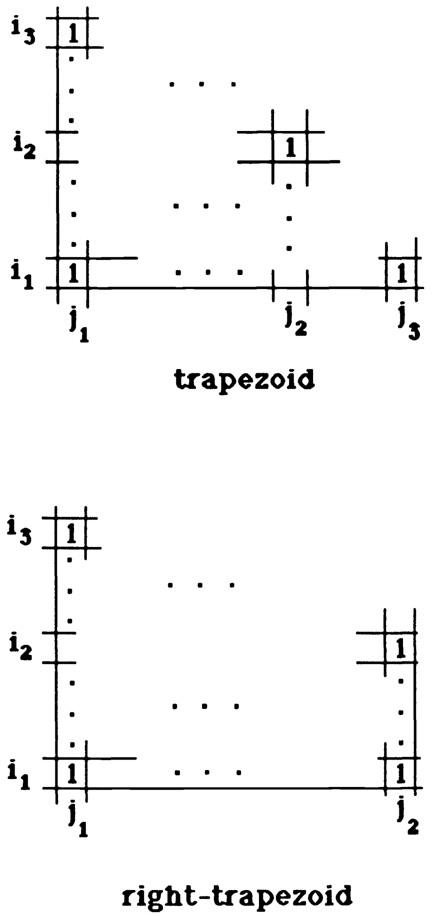
In what follows, the rows of matrices will be numbered from bottom to top, and the columns from left to right. Thus, the $(1, 1)$ entry of a matrix is its bottom left corner entry. We say that two columns of a matrix *overlap* at a given row if both columns contain a 1 in that row. Given a matrix A, the submatrix corresponding to the column indices $c_1, c_2, \cdots, c_m$ is denoted by $A[c_1, c_2, \cdots, c_m]$.

Let A be a 0-1 matrix. The total number of 1's in A is denoted by $\#(A)$. Let c be a column of A. Then:

* Received by the editors January 21, 1988; accepted for publication (in revised form) May 8, 1990.

† Columbia University, New York, New York 10027. The work of this author was carried out at Bellcore, Morristown, New Jersey 07960.

‡ Mathematical Institute of the Hungarian Academy of Sciences, Budapest, Hungary.



trapezoid

right-trapezoid

FIG. 1

(1) If X is a subset of the entries in c, the *interval spanned by* X is the set of all entries of c (including those not in X), in the rows between the lowest and the highest containing an element of X (inclusive). The *row-set* of the interval consists precisely of those rows of A that contain the elements of the interval.

(2) We say that a set Z of 1's of c are *consecutive* if the interval they span contains no other 1's but those in Z. We define the *spread* of c to be the minimum number of rows spanned by any three consecutive 1's in c. The spread of A, sp (A), is the minimum spread of any column of A.

2. The upper bound. Given integers $1 \leqq i_1 \leqq i_2 < i_3 \leqq n$ and $1 \leqq j_1 < j_2 \leqq n$, a *right trapezoid* is a pattern of four 1's, at positions (i_1, j_1), (i_2, j_2), (i_3, j_1), and (i_3, j_2). Let T'_n denote the class of $n \times n$ 0-1 right trapezoid-free matrices, and t'_n the maximum number of 1's in a matrix of T'_n. Clearly, $t_n \leqq t'_n$. We will show that $t'_n = O(n \log n)$. Below we prove the following theorem.

THEOREM 2. *There are constants* $\varepsilon \geqq 0$, $\lambda > 1$, *so that if* $A \in T'_n$, *there exists* $B \in T'_n$ *satisfying* sp $(B) \geqq \lambda$ sp (A) *and such that* A *has at most* ε n *more* 1's *than* B.

Pending the proof of Theorem 2, the upper bound on t'_n is clear. For if $A \in T'_n$, after applying Theorem 2 $O(\log n)$ times we will obtain a matrix in T'_n with $O(n)$ 1's, but also at most $O(n \log n)$ fewer 1's than A. This would conclude the proof of the upper bound.

The proof of Theorem 2 will be broken up into several steps. First we will give an informal description. We stress that this description will be slightly incorrect but will contain the main ideas.

2.1. Informal description of the construction. The heart of the proof is an algorithm that processes the columns of A from left to right, generating the corresponding columns of B in the process. Thus, after $i-1$ such steps, we will have generated the first $i-1$ columns of B. We will also have generated a right trapezoid-free matrix C_{i-1}, which corresponds to the semiprocessed matrix A. In other words, the last (rightmost) $n-i+1$ columns of C_{i-1} are identical to the last $n-i+1$ columns of A. The remaining (leftmost) columns of C_{i-1}, on the other hand, have a very special structure, and intuitively can be regarded as containing some carefully rearranged 1's obtained from the leftmost $i-1$ columns of A. The objective of the rearrangement is to facilitate the processing of 59additional columns of A, in particular to attain the desired increase of the spread in B. More precisely, let C_{i-1} have $m_{i-1}+n-i$ columns (possibly $m_{i-1} \neq i-1$, but for intuition, first regard $m_{i-1}=i-1$).

The 1's in the first m_{i-1} columns of C_{i-1} satisfy the following crucial property, to be maintained inductively. For each such column, consider the interval spanned by the highest and lowest 1's occurring in the column. *Then the row-sets of all such intervals are pairwise disjoint, and each interval is "long."* Here, long means having at least a prescribed number of rows (this will be made precise later). We stress here that a given interval may contain 0's; what is important is that it is spanned by the extreme 1's in the column. The fact that the intervals are long is used in the following way.

Suppose the construction has been carried out in $i-1$ steps and now we want to process the ith column of A (i.e., the ith column of C_{i-1}). Call this column v_i. The 1's of v_i are classified into two types.

First, consider the 1's where v_i overlaps any of the first m_{i-1} columns of C_{i-1}. Note that because C_{i-1} is right trapezoid-free (and thus rectangle-free), and because of the interval structure that we have constructed, the gaps between successive overlaps, on the average, have to be "long" (no three overlaps can correspond to the same interval). The column containing a 1 precisely on those rows where an overlap occurs will be the ith column of B. This new column satisfies the desired sparsity condition of the gaps between consecutive 1's, on the average, being long. This is the heart of the procedure. What we must show now is how to deal with the remaining 1's in v_i without having to remove an excessive number of these (ideally, a bounded number per column).

We partition these remaining 1's into consecutive blocks that correspond to the intervals of C_{i-1} and the gaps between these intervals. We will use most of the 1's in these blocks to define a new column C_i. Now, the 1's in any block *except the top one* may be jointly shifted to the left (this is possible since there is no overlap anymore). For example, if a block B_j corresponds to an interval I_h of C_{i-1}, we can shift all 1's in B_j from column $m_{i-1}+1$ of C_{i-1} to the column containing I_h (here, each shifted 1 remains in its original row). It is seen that this will not create a right trapezoid because of the existence of the top block. Similarly, the blocks corresponding to gaps can also be shifted to an appropriate column. Thus, we are left with a single block B_1. If B_1 contains few 1's (say, a bounded number), then just remove them. If B_1 corresponds to a gap, then make a new interval out of B_1.

The difficult case occurs when B_1 corresponds to some interval I, and B_1 has many 1's. But let us assume for now that the difficult case does not arise. Then we define C_i to be the matrix obtained from C_{i-1} by the above shifting and removal operations. Note that the number of 1's that were removed, if any, is bounded.

Proceeding inductively then, we will eventually process all the columns of A. At this point, the matrix B will have the desired sparsity condition, and the total number of 1's that were permanently removed in the process is at most linear. Furthermore, the matrix C_n does not have more than one 1 in any row (because of the disjoint interval structure). Hence if we also permanently remove all those 1's, altogether a linear number of 1's is removed in the worst case. In other words, the difference between the number of 1's in A and that in B is at most $O(n)$, as desired.

However, we have to consider the possibility of the difficult case arising when processing the top block B_1 of some column, say in the ith step, to keep notation as above. Thus, B_1 corresponds to some interval I, and B_1 has many 1's. In this case, no matter what we do, it appears that we may have to remove many 1's. This potentially dangerous situation is remedied by making a stronger inductive assumption about the interval structure. Given one of the first m_{i-1} columns of C_{i-1}, consider, within the interval corresponding to that column, the largest subinterval, starting from the top, such that the gap between any two consecutive 1's is "small" (made precise later). The rows in this subinterval (there is always at least one such row) are the "special rows" of the column. The inductive assumption is that the total number of special rows (added over the first m_{i-1} columns) does not decrease, i.e., for any $a > b$ the total number of special rows of C_a is at least that of C_b.

How can we use this assumption to handle B_1 and I as above? The general idea is to use 1's from B_1 to *increase* the number of special rows in I. This involves moving 1's within I, and from B_1 to I, to enlarge the interval of special rows in I (and thus, a move may involve a change of the row occupied by the 1). Naturally, such a set of moves may create right trapezoids, with one of the moved 1's acting as the left-bottom element, and one of the 1's in the rightmost $n - i$ columns of C_{i-1} acting as the top right element. *But a counting argument shows that the total number of such right trapezoids is at most proportional to the increase in the number of special rows.* We eliminate such trapezoids by removing their right top elements, and we will be done processing the new column of A. This removal operation, on the average, will not be expensive: since the total number of special rows is nondecreasing, altogether we will not remove more than $O(n)$ 1's in this manner, in the course of processing all columns of A.

This concludes the informal description. To summarize the above, there are two main facts concerning the construction. (1) First, the interval structure in the leftmost columns of the matrices C_i is used to achieve the sparsity condition of the matrix B. (2) To achieve the interval structure, some of the 1's are shifted left (with a few also changing rows). That the shifting usually works is a consequence of the matrix being right trapezoid-free (this is the main instance in the proof that this fact is actually used). But we also have to remove some 1's. To avoid too many removals, we introduce the special row structure of the intervals (this is the only reason why the special rows are used), which in turn must be inductively maintained.

2.2. Formal statement of the inductive assumption and its proof. Let $k > 4$ be a fixed (independent of n) integer. The proof of Theorem 2 will be based on the following lemma.

LEMMA 1. *There exists $B \in T'_n$, and for $1 \leqq i \leqq n$, there exist integers m_i and* 0-1 *matrices C_i, so that C_i is $n \times (m_i + n - i)$, and*

(1.1) *C_i is right trapezoid-free.*

(1.2) *The last $n - i$ columns of C_i are a copy of the last $n - i$ columns of A, possibly with some* 1's *removed (changed into* 0's).

(1.3) *The first column of B is made up of* 0's, *and for $i > 1$ $B[i]$ contains* 1's *precisely on those rows where $C_{i-1}[m_{i-1} + 1]$ overlaps any of the first (leftmost) m_{i-1} columns of C_{i-1}.*

(1.4) *Let* $1 \leqq j \leqq m_i$. *The row-set of the interval* $I_{j,i}$ *spanned by the highest and lowest* 1 *in* $C_i[j]$ *has cardinality at least* $\lfloor (k-2)/2 \rfloor \operatorname{sp}(A)$. *The row-sets of the intervals* $I_{j,i}$, $1 \leqq j \leqq m_i$, *are pairwise disjoint.*

(1.5) *Let* $1 \leqq j \leqq m_i$. *For some* $h_{j,i} \geqq 1$ (*possibly* ($h_{j,i} = 1$), *the first* (*topmost*) $h_{j,i}$ 1's *of* $I_{j,i}$ *have the property that the number of rows spanned by each* 1 *and the next is at most* $\lceil \operatorname{sp}(A)/2 \rceil$, *and* $h_{j,i}$ *is defined largest with this property. The rows spanned by these* $h_{j,i}$ 1's *are called the special rows of* $I_{j,i}$.

(1.6) *If* d_j *denotes the total number of special rows of* C_j, *then* $d_{i+1} \geqq d_i$, *for* $i < n$.

(1.7) $\#(A) \leqq \#(C_i) + \#(B[1, 2, \cdots, i]) + O(k(i + kd_i))$.

Figure 2 shows a typical matrix C_i. The function of these matrices is auxiliary. We postpone the proof of Lemma 1 until later. Let us next see how to use it to prove Theorem 2. We have the following results.

LEMMA 2. $\operatorname{sp}(B) \geqq \lfloor (k-2)/2 \rfloor \operatorname{sp}(A)$.

Proof. Consider three consecutive 1's of some column of B, say column $B[i]$. By (1.3) these 1's indicate overlaps of column $C_{i-1}[m_{i-1} + 1]$ with previous columns of C_{i-1}. Since by (1.1) C_{i-1} is right trapezoid-free (and thus rectangle-free), these overlaps must correspond to different intervals in C_{i-1}. Consequently, the number of rows from the first to the third 1 is at least $\lfloor (k-2)/2 \rfloor \operatorname{sp}(A)$, by (1.4), as desired. □

LEMMA 3. $\#(A) \leqq \#(C_n) + \#(B) + O(n)$.

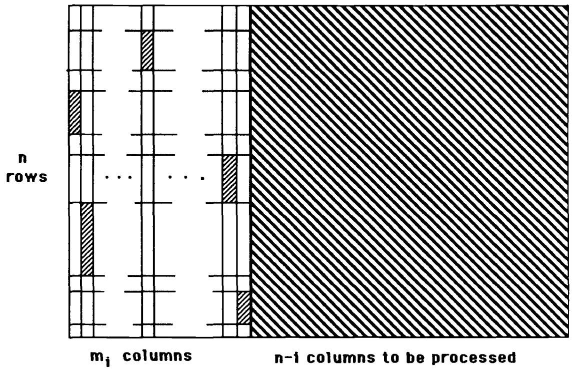
Proof. The result follows from (1.7), since $d_n \leqq n$. □

Now by Lemmas 2 and 3, the proof of Theorem 2 is complete with $\lambda = \lfloor (k-2)/2 \rfloor$ and $\varepsilon = O(k)$, since no row of C_n contains more than one 1, by the row-disjoint property of the intervals, as in (1.4).

In the remainder of this section, we will prove Lemma 1.

2.3. Proof of Lemma 1. We will prove (1.1)–(1.7) by induction on i. For $i = 1$, we set $B[1]$ to consist of 0's. If $A[1]$ has at least k 1's, we let $m_1 = 1$, $C_1 = A$, and we set $I_{1,1}$ to be the set of rows spanned by the 1's in $A[1]$, and, if necessary, set $h_{1,1} = 1$. If $A[1]$ contains fewer than k 1's, we remove them, and set $m_1 = 0$ and $C_1 = A[2, \cdots, n]$. Clearly, this satisfies (1.1)–(1.7).

To prove the general inductive step, we will use an algorithmic construction given next. The algorithm will be described in full, with proofs afterwards. We will assume $i > 1$, and assume that the constructions and proofs in (1.1)–(1.7) have been carried out for $1, 2, \cdots, i-1$.



A typical C$_i$

FIG. 2

Algorithm

Step (I). Let $B[i]$ be the column that contains 1's precisely where $C_{i-1}[m_{i-1}+1]$ overlaps any of $C_{i-1}[j]$, $1 \leqq j \leqq m_{i-1}$. This choice of $B[i]$ clearly satisfies the inductive hypothesis (3).

Step (II). We proceed as follows to define C_i. Replace $C_{i-1}[m_{i-1}+1]$ by $C_{i-1}[m_{i-1}+1] - B[i]$ to obtain a matrix C'_i (that is $C'_i[m_{i-1}+1]$ contains a 1 on those rows where $C_{i-1}[m_{i-1}+1]$ contains a 1 and does *not* overlap any of $C_{i-1}[j]$, $1 \leqq j \leqq m_{i-1}$).

Step (III). Consider the partition of $C'_i[m_{i-1}+1]$ into intervals induced by the row-sets of the intervals $I_{j,i-1}$, $1 \leqq j \leqq m_{i-1}$, as well as the "gaps" between them. To simplify notation, we call such intervals *blocks* (note that a block does not necessarily begin or end with a 1). Suppose that $X_0, \cdots, X_s$, $s \geqq 0$, are the blocks that contain 1's, numbered as they appear from bottom to top. Then, for $0 \leqq r < s$, do the following: if X_r corresponds to some $I_{j,i-1}$, then *shift* all 1's in X_r from column $m_{i-1}+1$ to column j (keeping them on the same row). If X_r corresponds to the gap between some $I_{x,i-1}$ and $I_{y,i-1}$; with $I_{x,i-1}$ "higher" than $I_{y,i-1}$, then we shift all the 1's in X_r from column $m_{i-1}+1$ to column x (changing $I_{x,i-1}$).

Denote by v_i the column obtained from $C'_i[m_{i-1}+1]$ after the shifts (i.e., after removing the 1's in the blocks $X_0, \cdots, X_{s-1}$), and let C''_i denote the resulting matrix.

Step (IV). If v_i contains fewer than $2k+2$ 1's, then remove column v_i from C''_i. The resulting matrix is C_i, where $m_i = m_{i-1}$. **Algorithm** terminates.

Otherwise, proceed with:

Step (V). If X_s corresponds to a gap, we set $m_i = m_{i-1}+1$ and create an interval in column m_i spanned by the 1's in X_i.

Algorithm terminates.

If X_s does not correspond to a gap, proceed with:

Step (VI). Let X_s correspond to some $I_{j,i-1} = I^*$. Refer to Fig. 3.

We partition I^* into two consecutive intervals, I_1^*, I_2^*, where the row-set of I_2^* contains exactly $k+2$ 1's of X_s, and the row-set of I_1^* contains at least k 1's of X_2 (the numbering of the intervals is from bottom to top). Let Y_1, Y_2 be the corresponding blocks of X_s. Now we *split* the column containing I^* into 2 columns, each containing one interval I_j^* in the obvious way. (The idea here will be to shift the 1's in Y_j to I_j^*, while removing few 1's. This will require special care in the case of Y_2 and I_2^*.) There are two cases.

(1) The set of special rows of I^* is contained within I_2^*.

(2) The set of special rows of I^* extends beyond I_2^*.

In either case, we first shift all the 1's from Y_1 to I_1^*.

Case 1. Let h^* be the number of 1's that span the special rows of I^* (see Fig. 3(a)). Now, if I^* has at least $\lfloor (k-2)/2 \rfloor \operatorname{sp}(A)$ special rows, we are done: we simply remove all 1's from Y_2, and the resulting matrix will be called C_i (with $m_i = m_{i-1}+1$). Otherwise, we consider the (h^*+1)st 1 in I_2^*, and shift it up so that the gap to the previous 1 is exactly $\lceil \operatorname{sp}(A)/2 \rceil$ (thus, we add $\lceil \operatorname{sp}(A)/2 \rceil$ special rows). If no such 1 exists, then use the first 1 from Y_2.

This action, of course, may create some right trapezoids. The 1's in the top right positions of the right trapezoids are removed (note that this may involve removing 1's from the rightmost $n-i$ columns of C''_i).

Next, we consider the (h^*+2)nd 1 in I_2^* and we shift it up to reduce the gap to the $(h^*.+1)$st to $\lceil \operatorname{sp}(A)/2 \rceil$, if necessary. We continue inductively, until I_2^* contains at least $\lfloor (k-2)/2 \rfloor \operatorname{sp}(A)$ special rows, or else we run out of 1's of I_2^*. In the latter case, we start using the 1's from Y_2, and clearly, these will suffice, since there are k of them. Once we are done, we remove any remaining 1's from Y_2.

The resulting matrix is C_i, and the transformed interval I_2^* will be a new interval in C_i. This concludes the description of Case 1.

Case 2. Let j be the first (lowest) row where I_2^* contains a 1. We remove all 1's from Y_2, and place a new 1 in the same column as I_1^*, on row $j-1$ (unless a 1 is already present there). The effect of such changes will be that a new interval of special rows starts at the top of I_1^*.

The resulting matrix is C_i. This concludes Case 2.

We set $m_i = m_{i-1} + 1$. In either Case 1 or 2, the interval Z_1 (Z_2), spanned by the top and bottom 1's in the column corresponding to I_1^* (I_2^*), will be an interval in matrix C_i, and we retain all other ones except, of course, I^*.

End of Algorithm

Now we will prove (1.1)–(1.7) of Lemma 1. By construction, (1.2), (1.3), and (1.5) hold.

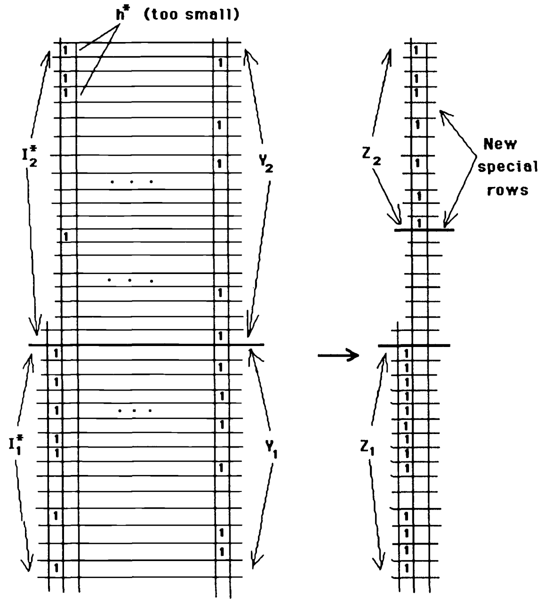
CLAIM 1. *Statement* (1.4) *holds.*

Proof. Inductively, (1.4) held before executing the algorithm, and it is clear that if additional 1's are placed in a given interval, then (1.4) still holds for that interval. We have to verify that (1.4) holds after Step (VI). This is clear for Z_1, since we moved into its column at least k 1's. Furthermore, the bottom 1 of Z_2 must be no more than sp $(A)/2$ rows higher than the bottom of I_2^*, by definition of Case 2. But in the column containing X_s those rows cannot contain more than two 1's of X_s. In other words, the row-set of Z_2 must contain at least k 1's of X_s, and the result follows. □

CLAIM 2. *Statement* (1.6) *holds.*

Proof. Again, we only need to check that the condition is maintained after Step (VI). This is clear in Case 1. In Case 2, the 1 added at the top of I_1^* ensures that Z_1, together with Z_2, contains at least as many special rows as I^*. □

CLAIM 3. *Statement* (1.1) *holds.*



(a) Case 1

FIG. 3 (a)

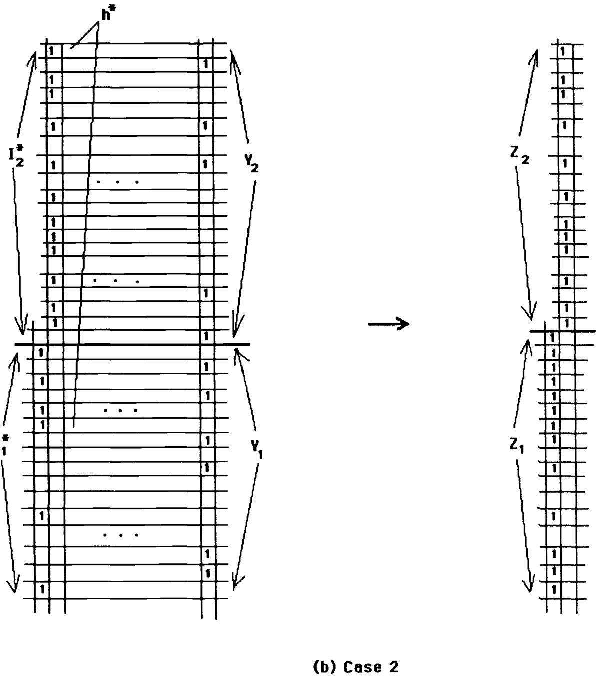


FIG. 3 (b)

Proof. We only have to check that the shifts in Step (III), or the execution of Case 2 in Step (VI), do not create any right trapezoids. Consider first Step (III).

If one of the shifted 1's is the bottom left corner of a right trapezoid, then C_i', before the shifts, had a right trapezoid, using a 1 in X_s as the top left corner—a contradiction. Similarly, if a shifted 1 is a top left corner, then we must have shifted this 1 into some interval $I_{j,i-1}$, and thus C_i' has a right trapezoid with both left corners either in $I_{j,i-1}$, or in column $C_i'[m_{i-1}+1]$—again a contradiction. The analysis is similar for Case 2 of Step (VI). □

CLAIM 4. *Statement* (1.7) *holds.*

Proof. Suppose Case 1 of Step (VI) is not executed. Then (1.7) holds since it does for $i-1$ and we only removed at most $O(k)$ 1's in processing the new column. So assume Case 1 of Step (VI) was executed.

Here, we may create right trapezoids every time a 1 is moved to increase the number of special rows in I_2^*, which was currently less than $\lfloor (k-2)/2 \rfloor \operatorname{sp}(A)$ (and by construction, the move increased the number of special rows by exactly $\lfloor \operatorname{sp}(A)/2 \rfloor$). It is not difficult to see that the moved 1 must be the bottom left element of all such right trapezoids (and hence the bottom right elements of the right trapezoids are all on the same row). But since C_i'' is rectangle-free, the total number of these right trapezoids is at most $\lfloor (k-2)/2 \rfloor \operatorname{sp}(A)$. Consequently, the number of 1's removed to eliminate all the right trapezoids, *per added special row*, is $O(k)$. This concludes the proof. □

The proof of Lemma 1 is now complete.

Remarks. (1) With a little care, the proof above will in fact show that A has at most $O(k \cdot n \log_k n)$ 1's. Thus, it is best to choose k bounded as a function of n.

(2) The upper bound $O(n \log n)$ on t_n' is in fact tight [M2]. In fact, the lower bound example for t_n that we give in the next section can be modified to yield a $cn \log n$ lower bound on t_n' as well.

3. The lower bound. In this section we prove the $n \log n/\log\log n$ lower bound. For an arbitrary integer $k \geqq 2$, and $n = k^k$, we will construct a 0-1 matrix A with n rows and cn columns (c a constant), such that A is trapezoid-free and A contains k 1's in every column.

We will use the following notation. If B is a matrix, then $B\{m\}$ will denote the matrix obtained by removing the top m rows of B and adding m new rows, consisting of 0's only, at the bottom. If $B_1, B_2, \cdots, B_m$ are matrices with equal number of rows, then $[B_1, B_2, \cdots, B_m]$ is the matrix obtained by putting side by side, from left to right, $B_1, B_2, \cdots, B_m$.

Choose $k \geqq 3$. We will construct, inductively, matrices $N_1, \cdots, N_{k-1}$, with n rows each, and we will set $A = [N_2, N_3, \cdots, N_{k-1}]$. Figure 4 shows the matrix for $k = 3$. The matrices N_i are defined inductively, as follows.

For $0 \leqq j \leqq k-1$, let M_j be the $n \times (n/k^{j+1})$ matrix such that the column $M_j[r]$, $1 \leqq r \leqq n/k^{j+1}$, contains 1's at rows $1 + (r-1)k^{j+1} + sk^j$, $0 \leqq s \leqq k-1$. Then $N_1 = [M_{k-1}\{k^{k-2}\}, M_{k-1}\{2k^{k-2}\}, \cdots, M_{k-1}\{(k-1)k^{k-2}\}]$.

Assuming we have defined $N_1, \cdots, N_{i-1}$, let $P_i = [M_{k-i}, M_{k-i+1}, \cdots, M_{k-1}, N_{i-1}]$. We then set $N_i = [P_i\{k^{k-i-1}\}, P_i\{2k^{k-i-1}\}, \cdots, P_i\{(k-1)k^{k-i-1}\}]$.

Some remarks will be useful before proving the desired facts about A. These remarks are not difficult to prove.

Remark 1. Let $0 \leqq j \leqq k-1$. In M_j, each column contains exactly k 1's, spaced k^j rows between each other. For $r > 1$, the lowest 1 in $M_j[r]$ is exactly k^j rows higher than the highest 1 in $M_j[r-1]$. Thus, M_j contains 1's in precisely all rows of the form

$$1 + b_{k-j}k^j, \quad \text{where } 0 \leqq b_{k-j} \leqq k^{k-j} - 1.$$

The top $k^j - 1$ rows of M_j contain only 0's.

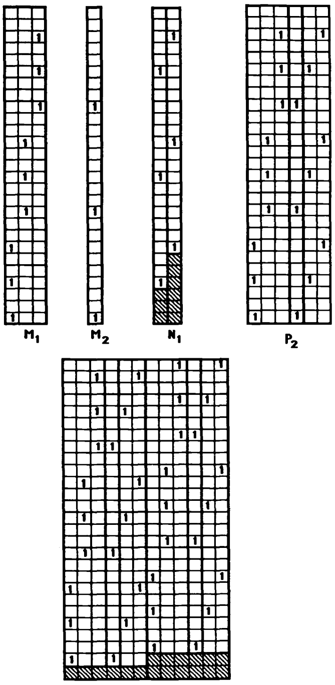


FIG. 4. *Example with* $k = 3$.

Remark 2. Inductively, we may easily show that N_i, $i \geqq 1$, contains 1's in precisely all rows of the form, for $1 \leqq r \leqq i$ and $1 \leqq s \leqq r$,

$$1 + \sum_{j=r}^{i} c_j k^{k-j-1} + b_s k^{k-s}, \quad \text{where } 1 \leqq c_j \leqq k-1, \quad 0 \leqq b_s \leqq k^s - 1.$$

From (1) we deduce that each column of N_i contains exactly k 1's.

Note that for $i \geqq 1$, every column $N_i[a]$ is essentially a copy of a column of some M_j, "shifted up" a certain positive number of rows. We call this amount the *shift* of $N_i[a]$. The highest power of k in the shift of any copy of a column of M_j is at most k^{j-1}. By induction, the following is not hard to prove.

Remark 3. Every shift of a column of N_{k-i-1} is of the form

$$\sum_{j=i}^{t} c_j k^j, \quad \text{where } 1 \leqq c_j \leqq k-1, \quad i \leqq j \leqq t$$

and, in such a case, it corresponds to a column of some M_h with $t < h$.

Now, M_j has k^{k-i-1} columns. By induction, we may prove that N_i, $i > 1$, has

$$k\{k^i - (k-1)^i\} - \frac{(k-1)^i - 1}{k-2}$$

columns. A calculation shows then that the number of columns in A is

$$k^k - (k-1)^k + o(k^k) \sim (1 - e^{-1})n.$$

Therefore, the number of 1's in A is

$$\sim (1 - e^{-1}) \frac{n \log n}{\log \log n},$$

as desired.

THEOREM 3. *A is trapezoid-free.*

Proof. Assume that there is a trapezoid. Let us first investigate under which conditions the "L" of the trapezoid can occur. Let $A[r]$ denote the column containing the right corner of the trapezoid, and $A[l]$ the column containing the two left corners of the trapezoid. Let $k^{k-i(r)}$ denote the spacing between consecutive 1's in $A[r]$, and $k - k - i(l)$ the spacing between consecutive 1's in $A[l]$. Thus, for $x = r, l$, $A[x]$ is a shifted-up copy of a column of $M_{k-i(z)}$.

Let x be one of l, r, and y the other. Note that if any term in the shift of $A[x]$ is a multiple of a power of k smaller than $k^{k-i(y)}$, then this term must also be a term in the shift of $A[y]$. Since $A[r]$ is to the right of $A[l]$, this fact is easily seen to imply that $k - i(l) < k - i(r)$. Furthermore, there is a smallest number t so that $A[l]$ and $A[r]$ are both in a shifted-up copy Z of P_t, and in that case $A[r]$ is in the copy of N_{t-1} but $A[l]$ is not.

Clearly $k - t \leqq k - i(l)$. Note that the shift of $A[r]$ has a term in $k^{k-(t-1)-1} = k^{k-t}$ (which the shift of $A[l]$ does not have), and therefore we must have that $t = i(l)$, and consequently $A[l]$ is in the copy of $M_{k-i(l)}$ contained in Z. It is easily seen that the fourth 1 in the trapezoid clearly cannot occur in any of the columns of the copies of $M_{k-i(l)}, \cdots, M_{k-1}$. It also cannot occur in the copy of $N_{i(l)-1}$ because of the convention in defining $N_{i(l)-1}$ (the first shift of $P_{i(l)-1}$ is by $k^{i(l)}$, the second by $2k^{i(l)}$, and so on).

Hence there is no trapezoid, a contradiction as desired. □

We conjecture that the lower bound $(n \log n/\log\log n)$ for t_n is in fact tight. A special case that might prove more tractable for improving the upper bound is that in which the matrix is regular; that is, it contains the same number of 1's in every row or column.

REFERENCES

[B] B. BOLLOBÁS, *Extremal Graph Theory*, Academic Press, New York, 1978.

[M1] J. MITCHELL, *Shortest rectilinear paths among obstacles*, Department of Operations Research and Industrial Engineering Technical Report No. 739, Cornell University, Ithaca, New York, 1987.

[M2] ———, *private communication*, 1987.

SIAM J. DISC. MATH.
Vol. 4, No. 1, pp. 28-35, February 1991

003

A POLYNOMIAL TIME OPTIMAL ALGORITHM FOR SATELLITE-SWITCHED TIME-DIVISION MULTIPLE ACCESS SATELLITE COMMUNICATIONS WITH GENERAL SWITCHING MODES*

M. A. BONUCCELLI†

Abstract. The Satellite-Switched Time-Division Multiple Access (SS/TDMA) is a technique effectively used in wideband communication satellites. A very important problem for SS/TDMA systems is the proper communications scheduling over the satellite equipment. This problem is equivalent to decomposing a given traffic matrix T into a positive linear combination of (0, 1)-matrices satisfying additional technology-dependant constraints. The sum of the multiplying constants represents the time taken by the satellite to handle the communications and must be minimum in order to achieve an efficient use of the equipment. A polynomial time optimal algorithm for the SS/TDMA scheduling problem for systems with variable bandwidth beams and restricted multiplexing and demultiplexing is presented. As a corollary of the presented results, another generalization of the classical Birkhoff–von Neumann Theorem is established.

Key words. network flow, combinatorial optimization, polynomial time algorithm, Birkhoff–von Neumann Theorem

AMS(MOS) subject classifications. 08-04, 90C27

1. Introduction. The rapidly growing demand of satellite communications services is exhausting the Radio Frequencies spectrum. An efficient use of such a spectrum can be achieved by the Satellite-Switched Time-Division Multiple Access (SS/TDMA) technique, which is conveniently used in wideband satellite communication systems [11]. In an SS/TDMA system, the satellite is equipped with a number of spot-beam antennas covering several geographical zones by disjoint communication channels, and a solid-state RF switch allowing a simultaneous interconnectivity between many uplink and downlink beams, and so between earth stations. Each earth station issues its connection needs at specific times. All the connection needs are gathered in a matrix T, the *traffic matrix*. Entry t_{ij} of T represents the time (in multiples of a minimal transmission time) that uplink i needs to be connected with downlink j. The transmission of the traffic described by T is called a *frame*, and is divided into subframes, called *time slots*. Each time slot represents the traffic transmitted during a specific switch configuration (also called a *switching mode*). A switching mode can be depicted as a (0, 1)-matrix, where the 1's denote the connected uplink-downlink pairs. These switching mode matrices must have some specific properties imposed by the technological features of the system under consideration.

A given traffic matrix can be decomposed in a variety of sequences (i.e., positive linear combinations) of switching modes, each of which represents a distinct frame. Different frames take different times to be completed. A very important problem in this setting is to find a frame of minimum transmission time for a given traffic matrix T. Such a frame increases the system efficiency, and therefore the operational profits. The above problem is often referred to as time slot assignment (TSA) and has been studied for several system configurations (e.g., see [2]–[4], [7], [8]). In [3], [7], systems with variable bandwidth beams and restricted multiplexing and demultiplexing have been considered. In these systems, each uplink and downlink beam can simultaneously transmit

* Received by the editors October 31, 1988; accepted for publication (in revised form) May 8, 1990.

† Dipartimento di Informatica, Università di Pisa, Pisa, Italy.

a mix of several signals. Hardware limitations impose an upper bound on the number of signals that can be mixed in each beam. In terms of TSA, this means that there is an upper bound on the number of 1's in each row and column of every switching mode. Lewandowski and Liu [7] proposed polynomial time algorithms based on a generalization of the celebrated Birkhoff–von Neumann theorem for this TSA problem. Such algorithms are optimal when the switching modes are restricted to have *exactly* as many 1's as the rows and columns upper bounds and have been proposed as suboptimal heuristics for the more general case of switching modes with *at most* as many 1's as the upper bounds. In this paper we present an optimal TSA algorithm for the last, general problem, with a further limit on the maximum number of 1's in each switching mode. Our algorithm is based on network flow in bipartite graphs and has the same time complexity of those given in [7]. The algorithm can be easily used to establish yet another generalization of the Birkhoff–von Neumann theorem.

In § 2 we formally define the problem under investigation, and give the definitions needed in the paper. In § 3 the polynomial time optimal algorithm is presented, as well as its relations to a generalization of the Birkhoff–von Neumann theorem.

2. Definitions and problem formulation. Let us assume that the system under investigation has m uplink and n downlink beams. Then, the traffic matrix T is an $m \times n$ matrix with nonnegative integer entries. Entry t_{ij} of T represents the amount to traffic to be transmitted from uplink i to downlink j, and is expressed in multiples of time slot length. Each uplink and downlink beam is a multiplex of several different signals. The hardware limitations of the system impose an upper bound on the number of different signals that can be multiplexed in each beam. Specifically, there are given two integer vectors, $\rho = (\rho_1, \cdots, \rho_m)$ and $\lambda = (\lambda_1, \cdots, \lambda_n)$, such that $\sum_{i=1}^{m} \rho_i = \sum_{j=1}^{n} \lambda_j$. Uplink beam i can be demultiplexed in at most ρ_i different signals, and downlink j is the multiplex of up to λ_j signals. Besides, there is an integer upper bound γ on the total number of messages that can be simultaneously handled by the satellite. γ is called the satellite *capacity*. Obviously, $0 \leqq \gamma \leqq \sum_{i=1}^{m} \rho_i$. Note that the satellites considered in [7] had an unlimited capacity, i.e., $\gamma \geqq \sum_{i=1}^{m} \rho_i$.

The sum of all entries in the ith row of T is called ith *row sum* and is denoted by R_i. The jth *column sum* C_j is similarly defined as the sum of all entries in column j. The symbol S is used to represent the sum of all entries in T.

Let $V(\rho, \lambda, \gamma)$ denote the class of $m \times n$ $(0, 1)$-matrices having at most ρ_i 1's in the ith row, $1 \leqq i \leqq m$, at most λ_j 1's in the jth column, $1 \leqq j \leqq n$, and at most γ 1's in the whole matrix. The TSA problem considered in this paper can be formulated as follows.

Given an $m \times n$ integer nonnegative matrix T, find integer positive constants $\nu_1, \cdots, \nu_h$ and matrices $Z_1, \cdots, Z_h$ in $V(\rho, \lambda, \gamma)$ (the switching modes) such that

$$T = \sum_{i=1}^{h} \nu_i Z_i \tag{1}$$

and $\sum_{i=1}^{h} \nu_i$ (the *cost* or *length* of the TSA) is minimum. Constant ν_i denotes the number of (not necessarily consecutive) time slots during which switching mode Z_i is used.

Note that the problem formulation given in [7] does not require integer values for the entries t_{ij} and the constants ν_i. Our integrality constraint is more realistic. Besides, it makes the problem computationally more difficult. In fact, our algorithm also solves the problem with no integrality constraint since it is based on network flow.

3. The optimal algorithm. The length of a TSA for a given traffic matrix T cannot be smaller than L, where

$$L = \max\{t_{ij}, R_i/\rho_i, C_j/\lambda_j, S/\gamma\}. \tag{2}$$

This lower bound follows directly from the problem formulation. In fact, the traffic represented by an entry t_{ij} must be transmitted in a strictly sequential way. Furthermore, at most ρ_i (λ_j) entries in row i (column j) can be allocated in a given time slot. Finally, $L \geqq S/\gamma$ since the system has a limited capacity.

The following inequalities hold true as a consequence of the lower bound given in (2).

$$\sum_{i=1}^{m}\sum_{j=1}^{n} t_{ij} \leqq \gamma L, \tag{3}$$

$$\sum_{i=1}^{m} t_{ij} \leqq \lambda_j L \quad \text{for} \quad \text{each } j, 1 \leqq j \leqq n, \tag{4}$$

$$\sum_{j=1}^{n} t_{ij} \leqq \rho_i L \quad \text{for} \quad \text{each } i, 1 \leqq i \leqq m, \tag{5}$$

$$t_{ij} \leqq L \qquad \text{for each } i \text{ and} j, 1 \leqq i \leqq m, 1 \leqq j \leqq n. \tag{6}$$

A row (column) such that $R_i = \rho_i L$ ($C_j = \lambda_j L$) is called *critical*. Similarly, the matrix T is critical if $S = \gamma L$, and the entry t_{ij} is also critical when $t_{ij} = L$. Let us define the following parameters:

$$\delta_i = \max\{0; R_i - \rho_i(L-1)\}, \quad \text{for each } i, 1 \leqq i \leqq m;$$

$$\mu_j = \max\{0; C_j - \lambda_j(L-1)\}, \quad \text{for each } j, 1 \leqq j \leqq n;$$

$$\varphi = \max\{0; S - \gamma(L-1)\}$$

$$\beta_{ij} = 1 \text{ if } t_{ij} = L, \quad \text{and} \quad \beta_{ij} = 0 \text{ if } t_{ij} < L, \text{for each } i \text{ and } j, 1 \leqq i \leqq m, 1 \leqq j \leqq n.$$

Let Z be a switching mode for T. If there are less than δ_i 1's in row i of Z, then any TSA for T containing Z will be longer than L, since the matrix $T' = T - Z$ will have a lower bound of L, and so the length of this TSA will be $L + 1$ at least. Thus, a necessary condition for a length L TSA for T is to have at least δ_i 1's in row i of any switching mode. A similar meaning pertains to the other parameters, namely μ_j for column j, φ for the whole matrix, and β_{ij} for the entry in row i and column j.

Given a traffic matrix T, we want to find a switching mode Z with at least δ_i 1's in row i, μ_j 1's in column j, and a total number of 1's not smaller than φ. Besides, entry z_{ij} in Z must be equal to 1 if $t_{ij} = L$. Such a switching mode is called *provident* (as opposed to greedy) since it does not contain the maximum number of 1's, but the minimum number of 1's necessary for a length L TSA; it can be obtained by means of network flow in a bipartite network with lower bounds and capacities.

Let (a, b) denote the arc oriented from node a to node b. We derive a network with $m + n + 2$ nodes labeled $s, r_1, \cdots, r_m, c_1, \cdots, c_n, t$, from the matrix T. The node s is called a *source* node, and is linked by one arc with each node $r_1, \cdots, r_m$. Each such arc is oriented from s to r_i, $(1 \leqq i \leqq m)$, and has lower bound δ_i and capacity ρ_i. Furthermore, there are arcs from nodes labeled r_i to nodes labeled c_j. In particular, there is the arc (r_i, c_j) if and only if entry t_{ij} is greater than zero. Such arc has capacity 1 and lower bound β_{ij}. Each node c_j is connected by an arc (c_j, t) with the *sink* node t. The

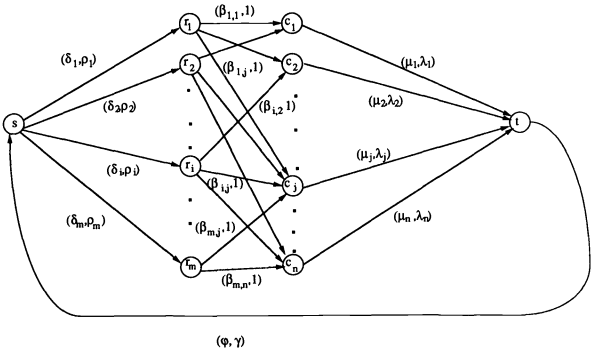


FIG. 1. *Network associated to a traffic matrix. Some* (*lower bound, capacity*) *pairs are not shown.*

capacity of this arc is λ_j and the lower bound is μ_j. Finally, there is the arc (t, s) with capacity γ and lower bound φ. It follows from (3)–(6) and from the definition of the above parameters that all the capacities and lower bounds are nonnegative. Note that in deriving the network from a traffic matrix, we put the constraints of our TSA problem as arc capacities, and the previously defined parameters as lower bounds. Figure 1 shows a network derived from a traffic matrix.

A *circulation* in a network is an assignment of numbers to the arcs such that (i) the number assigned to an arc (the arc flow) is not smaller than the lower bound and not greater than the capacity; (ii) for each node, the sum of the flows of the incoming arcs is equal to the flow sum of the outgoing arcs (the conservation law).

A circulation in a network derived from the traffic matrix T can be used to get a provident switching mode. Let Z be the $m \times n$ matrix with entry z_{ij} equal to the flow in the arc (r_i, c_j). We claim that Z is a provident switching mode. In fact, Z is a (0, 1)-matrix since the network flow problem (and so the circulation problem) is totally unimodular, and the networks derived from traffic matrices have integer capacities and lower bounds. Furthermore, Z has at most ρ_i 1's in row i, and λ_j 1's in column j since the capacity of the arc (s, r_i) is ρ_i, and that of the arc (c_j, t) is λ_j. The lower bounds on these arcs guarantee that at least δ_i (μ_j) 1's are present in row i (column j). Besides, Z has at least φ and at most γ 1's, owing to the lower bound and the capacity of the arc (t, s). Finally, if $t_{ij} = L$, then $z_{ij} = 1$ since β_{ij} is 1.

The existence of circulations in networks depends on the arcs capacities and lower bounds. A *cutset* (σ, τ) in a network is a partition of the nodes in two subsets σ and τ such that $s \in \sigma$ and $t \in \tau$. The following classical theorem states a necessary and sufficient condition for the existence of a circulation in networks. Let ξ_{ij} and χ_{ij} denote the lower bound and the capacity of the arc (i, j), respectively.

HOFFMAN'S THEOREM. *In a network with lower bounds and capacities, a circulation exists if and only if*

$$\sum_{i \in \tau} \sum_{j \in \sigma} \xi_{ij} \leqq \sum_{i \in \sigma} \sum_{j \in \tau} \chi_{ij}$$

for all cutsets (σ, τ).

The proof of this theorem can be found, e.g., in [6, p. 139].

THEOREM 1. *If a network is derived from a traffic matrix, then it has a circulation.*

Proof. Let (σ, τ) be a cutset for a network derived from a traffic matrix T. Then,

$$\sum_{i\in\tau}\sum_{j\in\sigma}\xi_{ij}=\sum_{r_i\in\tau}\sum_{c_j\in\sigma}\beta_{ij}+\varphi,$$

and

$$\sum_{i\in\sigma}\sum_{j\in\tau}\chi_{ij}=\sum_{r_i\in\tau}\rho_i+\sum_{c_j\in\sigma}\lambda_j+\sum_{r_j\in\sigma}\sum_{c_j\in\tau}\alpha_{ij},$$

where α_{ij} is 1 if $t_{ij}>0$, and is 0 otherwise.

By Hoffman's theorem we must have

$$\sum_{i\in\sigma}\sum_{j\in\tau}\chi_{ij}-\sum_{i\in\tau}\sum_{j\in\sigma}\xi_{ij}=\sum_{r_i\in\tau}\rho_i+\sum_{c_j\in\sigma}\lambda_j+\sum_{r_i\in\sigma}\sum_{c_j\in\tau}\alpha_{ij}-\sum_{r_i\in\tau}\sum_{c_j\in\sigma}\beta_{ij}-\varphi\geqq 0. \tag{7}$$

Since $t_{ij}\leqq L$, for each i and j, $1\leqq i\leqq m$, $1\leqq j\leqq n$, then

$$\sum_{r_i\in\sigma}\sum_{c_j\in\tau}\alpha_{ij}\geqq\left\lceil 1/L\left(\sum_{r_i\in\sigma}\sum_{c_j\in\tau}t_{ij}\right)\right\rceil.$$

Hence,

$$\sum_{r_i\in\sigma}\sum_{c_j\in\tau}\alpha_{ij}\geqq 1/L\left(\gamma(L-1)+\varphi-\sum_{r_i\in\tau}\sum_{j=1}^{n}t_{ij}-\sum_{i=1}^{m}\sum_{c_j\in\sigma}t_{ij}+\sum_{r_i\in\tau}\sum_{c_j\in\sigma}t_{ij}\right)$$

$$\geqq 1/L\left(\gamma(L-1)+\varphi-L\sum_{r_i\in\tau}\rho_i-L\sum_{c_j\in\sigma}\lambda_j+\sum_{r_i\in\tau}\sum_{c_j\in\sigma}t_{ij}\right).$$

We have that

$$1/L\left(\sum_{r_i\in\tau}\sum_{c_j\in\sigma}t_{ij}\right)\geqq\sum_{r_i\in\tau}\sum_{c_j\in\sigma}\beta_{ij},$$

by definition. Then,

$$\sum_{r_i\in\sigma}\sum_{c_j\in\tau}\alpha_{ij}\geqq 1/L(\gamma(L-1)+\varphi)-\sum_{r_i\in\tau}\rho_i-\sum_{c_j\in\sigma}\lambda_j+\sum_{r_i\in\tau}\sum_{c_j\in\sigma}\beta_{ij}. \tag{8}$$

Substituting (8) in (7), we get $1/L(\gamma(L-1)+\varphi)-\varphi\geqq 0$, and so $\gamma(L-1)+\varphi-L\varphi\geqq 0$.

Therefore, $\gamma(L-1)-\varphi(L-1)\geqq 0$, which is true since $0\leqq\varphi\leqq\gamma$, by definition. The theorem then follows from the generality of the cutset (σ, τ). □

The networks derived from traffic matrices have circulations, and therefore it is always possible to obtain provident switching modes from them. A circulation in a network derived from a traffic matrix, and so a provident switching mode, can efficiently be found by means of max flow algorithms (e.g., see [5, p. 65]). For instance, the MPM algorithm [10] can be used.

The optimal algorithm produces a TSA by repeatedly generating provident switching modes via circulations in the networks derived from traffic matrices. Let us assume that the TSA has been found and that it is formed by h switching modes. In order to complete the TSA, we must also find a proper set of integer positive constants $\nu_1, \nu_2, \cdots, \nu_h$, the switching modes multipliers. Let us consider the switching mode Z obtained from a circulation in the network derived from T. We want to find the largest multiplier constant ν such that the lower bound of the traffic matrix $T_1=T-\nu Z$ is $L-\nu$. This constant must satisfy the following inequalities:

$$(9)\quad \nu \leqq \left\lfloor (L\rho_i - R_i) \Big/ \left(\rho_i - \sum_{j=1}^{n} z_{ij}\right)\right\rfloor, \quad \text{for each } i \text{ such that } \sum_{j=1}^{n} z_{ij} < \rho_i;$$

$$(10)\quad \nu \leqq \left\lfloor (L\lambda_j - C_j) \Big/ \left(\lambda_j - \sum_{i=1}^{m} z_{ij}\right)\right\rfloor, \quad \text{for each } j \text{ such that } \sum_{i=1}^{m} z_{ij} < \lambda_j;$$

$$(11)\quad \nu \leqq \left\lfloor (L\gamma - S) \Big/ \left(\gamma - \sum_{i=1}^{m}\sum_{j=1}^{n} z_{ij}\right)\right\rfloor, \quad \text{provided that } \sum_{i=1}^{m}\sum_{j=1}^{n} z_{ij} < \gamma;$$

$$(12)\quad \nu \leqq L - t_{ij} \text{ if } z_{ij} = 0, \text{ and } \nu \leqq t_{ij} \text{ if } z_{ij} = 1, \text{for each } i \text{ and } j, 1 \leqq i \leqq m, 1 \leqq j \leqq n.$$

We are now in a position to state the algorithm.

ALGORITHM OPTIMAL.

Step 1:

Set up the network derived from T.

Step 2:

Find a circulation in the network, e.g., by using MPM algorithm.

Step 3:

Get a provident switching mode Z_e from the circulation found in Step 2 by using the flow of the arcs (r_i, c_j).

Step 4:

Compute the largest solution of the inequalities system (9)–(12), and let it be the multiplier ν_e.

Step 5:

Subtract $\nu_e Z_e$ from T. If T has some nonzero entry, then go to Step 1, else Halt.

THEOREM 2. *Algorithm Optimal generates a minimum length* TSA *in* $O(\theta^5)$ *time, where* θ *is the larger of* m *and* n.

Proof. Let T be the original traffic matrix and L its lower bound. Moreover, let Z_1, $Z_2, \cdots, Z_h$ be the switching modes generated in Steps 2 and 3, and let $\nu_1, \nu_2, \cdots, \nu_h$ be the multiplier constants computed in Step 4. Finally, let $T_k = T - \nu_1 Z_1 - \nu_2 Z_2 - \cdots - \nu_k Z_k = T_{k-1} - \nu_k Z_k$, $1 \leqq k \leqq h$ be the matrix containing the traffic not yet assigned after the kth iteration of the algorithm, and let L_k be its lower bound. We now show that $L = \sum_{k=1}^{h} \nu_k$ by induction. Let $t_{ij}^{(k)}$ $(z_{ij}^{(k)})$ be the entry in row i and column j of T_k (Z_k).

Since T is a traffic matrix, then (by the above discussion on circulations) Z_1 is a provident switching mode. Consider now the inequalities (9):

$$\nu_1 \leqq \left\lfloor (L\rho_i - R_i) \Big/ \left(\rho_i - \sum_{j=1}^{n} z_{ij}^{(1)}\right)\right\rfloor, \quad \text{for each } i \text{ such that } \sum_{j=1}^{n} z_{ij}^{(1)} < \rho_i.$$

Thus, if $\sum_{j=1}^{n} z_{ij}^{(1)} < \rho_i$ we have:

$$\nu_1 \leqq \left((L\rho_i - R_i) \Big/ \left(\rho_i - \sum_{j=1}^{n} z_{ij}^{(1)}\right)\right),$$

or, equivalently,

$$\nu_1 \rho_i - \nu_1 \sum_{j=1}^{n} z_{ij}^{(1)} \leqq L\rho_i - R_i.$$

So,

$$R_i - \nu_1 \sum_{j=1}^{n} z_{ij}^{(1)} \leqq L\rho_i - \nu_1\rho_i.$$

Since $R_i - \nu_1 \sum_{j=1}^{n} z_{ij}^{(1)} = \sum_{j=1}^{n} t_{ij}^{(1)}$, we have:

$$\sum_{j=1}^{n} t_{ij}^{(1)} \leqq \rho_i(L - \nu_1), \quad \text{if} \sum_{j=1}^{n} z_{ij}^{(1)} < \rho_i.$$

Assume now that $\sum_{j=1}^{n} z_{ij}^{(1)} = \rho_i$. Then,

$$\sum_{j=1}^{n} t_{ij} - \nu_1 \sum_{j=1}^{n} z_{ij}^{(1)} = \sum_{j=1}^{n} t_{ij}^{(1)} \leqq L\rho_i - \nu_1\rho_i = \rho_i(L - \nu_1).$$

A similar argument (applied to (10) and (11)) can be used to show that $\sum_{i=1}^{m} t_{ij}^{(1)} \leqq \lambda_j(L - \nu_1)$, for each j, $1 \leqq j \leqq n$, and that $\sum_{i=1}^{m} \sum_{j=1}^{n} t_{ij}^{(1)} \leqq \gamma(L - \nu_1)$. Furthermore, if $z_{ij}^{(1)} = 0$, then $t_{ij}^{(1)} = t_{ij} \leqq L - \nu_1$, and $t_{ij}^{(1)} = t_{ij} - \nu_1 \leqq L - \nu_1$ whenever $z_{ij}^{(1)} = 1$, by (12).

Thus, T_1 is a traffic matrix with lower bound $L - \nu_1$. Let us assume now that T_{k-1} is a traffic matrix with lower bound $L_{k-1} = L - \nu_1 - \nu_2 - \cdots - \nu_{k-1}$. Then Z_k is a provident switching mode, and T_k is a traffic matrix with lower bound $L_k = L - \nu_1 - \nu_2 - \cdots - \nu_{k-1} - \nu_k$. In fact,

$$\nu_k \leqq \left[\left(L_{k-1}\rho_i - \sum_{j=1}^{n} t_{ij}^{(k-1)} \right) \Big/ \left(\rho_i - \sum_{j=1}^{n} z_{ij}^{(k)} \right) \right],$$

provided that $\sum_{j=1}^{n} z_{ij}^{(k)} < \rho_i$. Hence,

$$\sum_{j=1}^{n} t_{ij}^{(k)} = \sum_{j=1}^{n} t_{ij}^{(k-1)} - \nu_k \sum_{j=1}^{n} z_{ij}^{(k)} \leqq \rho_i(L_{k-1} - \nu_k) = \rho_i\left(L - \sum_{p=1}^{k} \nu_p \right).$$

Similarly,

$$\sum_{j=1}^{n} t_{ij}^{(k)} \leqq \lambda_j\left(L - \sum_{p=1}^{k} \nu_p \right); \qquad \sum_{i=1}^{m} \sum_{j=1}^{n} t_{ij}^{(k)} \leqq \gamma\left(L - \sum_{p=1}^{k} \nu_p \right)$$

and

$$t_{ij}^{(k)} \leqq L - \sum_{p=1}^{k} \nu_p.$$

Therefore, Algorithm Optimal generates a minimum length TSA.

For the time complexity of the algorithm, note that if ν_k is equal to an entry $t_{ij}^{(k-1)}$, T_k has more zero entries than T_{k-1}. Otherwise, either T_k becomes critical, or it has more critical items (rows, columns, entries). Since there are at most mn nonzero entries in the original traffic matrix, and we can have at most $mn + m + n$ critical items, at most $2mn + m + n + 1$ switching modes are sufficient for the optimal TSA. Steps 1–5 are performed each time a switching mode is generated, namely $2mn + m + n + 1$ times at most. Step 2 is the most time-consuming one, and needs $O(\theta^3)$ time when MPM algorithm is used. Therefore, the total time complexity of Algorithm Optimal is $O(\theta^5)$. □

The network must be set up only the first time that Step 1 is performed, since subtracting a switching mode from T can eventually lead to the deletion of some (r_i, c_j) arc and the change of the lower bound of some arc.

The problem considered in [7] is a special case of that investigated in this paper. In particular, if we drop the integrality constraint and the one that the switching modes must have γ 1's at most, we get Lewandowski and Liu's problem. It is easy to see that our algorithm optimally also solves in $O(\theta^5)$ time this last problem. It is also easy to see that we have algorithmically established the following generalization of the Birkhoff–von Neumann Theorem [1], [9], as a corollary.

THEOREM 3. *Let T be an $m \times n$ matrix with nonnegative entries t_{ij}. Then $T = \sum_{i=1}^{h} \nu_i Z_i$ (where Z_i is a $(0, 1)$ matrix in $V(\rho, \lambda, \gamma)$), and $L = \sum_{k=1}^{h} \nu_k$, if and only if*

$$\sum_{i=1}^{m} \sum_{j=1}^{n} t_{ij} \leqq \gamma L$$

$$\sum_{i=1}^{m} t_{ij} \leqq \lambda_j L \quad \textit{for each } j,\ 1 \leqq j \leqq n$$

$$\sum_{j=1}^{n} t_{ij} \leqq \rho_i L \quad \textit{for each } i,\ 1 \leqq i \leqq m$$

$$t_{ij} \leqq L \qquad \textit{for each } i \textit{ and } j,\ 1 \leqq i \leqq m,\ 1 \leqq j \leqq n.$$

4. Conclusions. In this paper we investigated the time slot assignment problem for SS/TDMA systems with variable bandwidth beams and restricted multiplexing/demultiplexing. We presented a polynomial time algorithm for the above problem, and showed that it generates minimum cost TSA's. As a corollary, we proved another generalization of the celebrated Birkhoff–von Neumann Theorem.

REFERENCES

[1] G. BIRKHOFF, *Tres observaciones sobre el algebra lineal*, Univ. Nac. Tucuman, Rev, Ser. A, 5 (1946), pp. 147–151.

[2] G. BONGIOVANNI, D. COPPERSMITH, AND C. K. WONG, *An optimal time slot assignment for an* SS/TDMA *system with variable number of transponders*, IEEE Trans. Commun., 29 (1981), pp. 721–726.

[3] I. S. GOPAL, G. BONGIOVANNI, M. A. BONUCCELLI, D. T. TANG, AND C. K. WONG, *An optimal switching algorithm for multibeam satellite systems with variable bandwidth*, IEEE Trans. Commun., 30 (1982), pp. 2475–2481.

[4] I. S. GOPAL, M. A. BONUCCELLI, AND C. K. WONG, *Scheduling in multibeam satellites with interfering zones*, IEEE Trans. Commun., 31 (1983), pp. 941–951.

[5] P. A. JENSEN AND J. W. BARNES, *Network flow programming*, John Wiley, New York, 1980.

[6] E. L. LAWLER, *Combinatorial optimization: networks and matroids*, Holt, New York, 1976.

[7] J. L. LEWANDOWSKI AND C. L. LIU, SS/TDMA *satellite communications with k-switching modes*, SIAM J. Algebraic Discrete Methods, 8 (1987), pp. 519–534.

[8] J. L. LEWANDOWSKI, J. W. S. LIU, AND C. L. LIU, SS/TDMA *time slot assignment with restricted switching modes*, IEEE Trans. Commun., 31 (1983), pp. 149–154.

[9] J. L. LEWANDOWSKI, C. L. LIU, AND J. W. S. LIU, *An algorithmic proof of a generalization of the Birkhoff–von Neumann theorem*, J. Algorithms, 7 (1986), pp. 323–330.

[10] V. M. MALHOTRA, M. PRAMODH KUMAR, AND S. N. MAHESHWARI, *An $O(|V|^3)$ algorithm for finding maximum flows in networks*, Inform. Proc. Lett., 7 (1978), pp. 277–278.

[11] W. W. WU, *Elements of Digital Satellite Communication*, Computer Science Press, Rockville, MD, 1985.

SIAM J. DISC. MATH.
Vol. 4, No. 1, pp. 36–47, February 1991

004

A ZERO-ONE LAW FOR BOOLEAN PRIVACY*

BENNY CHOR†‡ AND EYAL KUSHILEVITZ†

Abstract. A Boolean function $f: A_1 \times A_2 \times \cdots \times A_n \rightarrow \{0, 1\}$ is t-private if there exists a protocol for computing f so that no coalition of size $\leqq t$ can infer any additional information from the execution, other than the value of the function. It is shown that f is $\lceil n/2 \rceil$-private if and only if it can be represented as

$$f(x_1, x_2, \cdots, x_n) = f_1(x_1) \oplus f_2(x_2) \oplus \cdots \oplus f_n(x_n),$$

where the f_i are arbitrary Boolean functions. It follows that if f is $\lceil n/2 \rceil$-private, then it is also n-private. Combining this with a result of Ben-Or, Goldwasser, and Wigderson, and of Chaum, Crepeau, and Damgard, [*Proc. 20th Symposium on Theory of Computing*, 1988, pp. 1–10 and pp. 11–19] an interesting "zero-one" law for private distributed computation of Boolean functions is derived: every Boolean function defined over a finite domain is either n-private, or it is $\lfloor (n-1)/2 \rfloor$-private but not $\lceil n/2 \rceil$-private.

A weaker notion of privacy is also investigated, where (a) coalitions are allowed to infer a limited amount of additional information, and (b) there is a probability of error in the final output of the protocol. It is shown that the same characterization of $\lceil n/2 \rceil$-private Boolean functions holds, even under these weaker requirements. In particular, this implies that for Boolean functions, the strong and the weak notions of privacy are equivalent.

Key words. private distributed computations, Boolean functions

AMS(MOS) subject classifications. 94A15, 94A60, 68R05

1. Introduction. A set of n parties, each holding an input value x_i, wishes to distributively compute the value of a Boolean function $f(x_1, x_2, \cdots, x_n) \in \{0, 1\}$. The participants communicate via a complete network of secure channels (no eavesdropping). The participants are honest—they send messages according to the prescribed protocol for f. However, honesty is no deterrent against curiosity. A subset of the participants (a coalition) might get together after the execution of the protocol and compare notes in an attempt to infer additional information on the inputs of noncoalition parties. Additional information is any information that does not follow from the value of the function and the inputs of the coalition parties. As an example of additional information consider a case where $f(0, 0, \cdots, 0, 0, 0) = f(0, 0, \cdots, 0, 0, 1)$ and, based on the execution of the protocol, the first $n/2$ participants can infer that the nth input is more likely to be 1 than 0. In general, any information-theoretic advantage can be used by the coalition even if this requires, for example, exponential computational resources.[1]

A function f is called *t-private* if there is a protocol for computing f so that no coalition of size $\leqq t$ can get any additional information. The fundamental result in this area is due to Ben-Or, Goldwasser, and Wigderson [BGW] who have shown that every n-variable function defined over a finite domain is $\lfloor (n-1)/2 \rfloor$-private. (A similar result was independently obtained by Chaum, Crepeau, and Damgard [CCD].) Ben-Or, Gold-

* Received by the editors July 24, 1989; accepted for publication (in revised form) December 18, 1989. An early version of this paper appeared in *Proceedings of 21st Symposium on Theory of Computing*, Association for Computing Machinery, 1989, pp. 62–72.

† Department of Computer Science, Technion, Haifa 32000, Israel. This research was supported by United States–Israel Binational Science Foundation grant 88-00282.

‡ Part of this research was done while the author visited the International Computer Science Institute, Berkeley, California.

[1] The information-theoretic approach alleviates the need for restricting the computational power of the participants as well as the use of unproven intractability assumptions. The case of computationally bounded participants is entirely different and is handled in [Ya], [GMW].

wasser, and Widgerson also showed that *certain* functions (e.g., the OR function) are not $\lceil n/2 \rceil$-private, while *certain* others (e.g., the XOR function) are n-private. Other than these two examples, very little was known about t-private functions for $t \geqq \lceil n/2 \rceil$.

In this paper we address the general problem of t-privacy in the range $\lceil n/2 \rceil \leqq t \leqq n$. We raise two major questions:

1. What is the structure of the "privacy hierarchy"? Is it the case that for every t in the range $\lceil n/2 \rceil \leqq t < n$ there are functions that are t-private but not $t + 1$-private, or does the hierarchy consist of two isolated levels $\lfloor (n-1)/2 \rfloor$-privacy and n-privacy)?

2. It is possible to relate the form of a function f to its attainable privacy?

We resolve both questions for Boolean functions $f: A_1 \times A_2 \times \cdots \times A_n \rightarrow \{0, 1\}$ defined over arbitrary (possibly infinite) domains. We give a complete characterization of Boolean functions that are $\lceil n/2 \rceil$-private. It is shown that every such function can be expressed as the "exclusive-or" of n Boolean functions, each depending on a single variable. There is a simple n-private randomized protocol [Bh] for computing functions of the form $f(x_1, x_2, \cdots, x_n) = f_1(x_1) \oplus f_2(x_2) \oplus \cdots \oplus f_n(x_n)$. Thus our characterization implies that if a Boolean function f is $\lceil n/2 \rceil$-private, then it is also n-private. Interestingly, the same characterization remains valid under a weaker definition of privacy. Specifically, this weaker definition allows coalitions to infer a limited (quite substantial) amount of additional information, and there can be a positive probability of error in computing the final output of the protocol. Finally, combining our result with [BGW] we conclude that there is a surprising gap in the Boolean privacy hierarchy: Every Boolean function, defined over a finite domain, is either exactly $\lfloor (n-1)/2 \rfloor$-private or exactly n-private, and there is nothing in between.

The rest of this paper is organized as follows: In § 2 we present the model and the definitions of privacy. In § 3 we consider the two-party case. Section 4 contains our main results: the characterization for the multiparty case as well as some implications and conclusions.

2. Model and definitions.

In this section we define the model of distributed computation that is used in the following. We then give formal definitions of strong and weak privacy in this model.

The system consists of a complete synchronous network of n honest parties P_1, $P_2, \cdots, P_n$ with secure reliable point-to-point communication (no eavesdropping). (By saying that the parties are honest it is meant that they send messages according to the protocol.) At the beginning of an execution, each party P_i has an input x_i taken from a nonempty set of possible inputs A_i (no probability space is associated with A_i). In addition, each party has a random input r_i taken from a source of randomness R_i. The parties wish to compute a Boolean function $f: A_1 \times A_2 \times \cdots \times A_n \rightarrow \{0, 1\}$. To this end they exchange messages as prescribed by a protocol F. Messages are sent in rounds, where in each round every processor sends a message to every other processor. Each message a party sends in the kth round is determined using its input, its random input, the messages it received so far, and the identity of the receiver. As commonly assumed, the messages sent at each round are prefix-free. We say that a protocol F computes the function f if the last message in the protocol, $F(x_1, \cdots, x_n)$, is an identical message sent by party P_1 to all parties, which contains the value $f(x_1, \cdots, x_n)$.

The *communication* passed in the network when the parties have inputs $\vec{x}$ and random inputs $\vec{r}$ is denoted $S(\vec{x}, \vec{r})$. Formally the communication S is an n-by-n matrix whose (i, j) entry is the concatenation of all messages sent from P_i to P_j. For any $T \subset \{1, 2, \cdots, n\}$, S_T denotes the matrix S where entries (i, j) with either $i, j \in T$ or $i, j \notin T$ are omitted. That is, S_T is the communication between processors in T and in $\bar{T}$.

We say that a coalition (i.e., a set of parties) T *does not learn any additional information* (other than what follows from its input and the function value) from the execution of a randomized protocol F, which computes f, if the following holds: For every two inputs $\vec{x}$, $\vec{y} \in A_1 \times A_2 \times \cdots \times A_n$ that agree in their T entries (i.e., for all $i \in T : x_i = y_i$) and satisfy $f(\vec{x}) = f(\vec{y})$, and for every choice of random inputs $\{r_i\}_{i\in T}$, the messages passed between T and $\bar{T}$ are identically distributed. That is:

$$\langle S_T(\{x_i\}_{i\in T}, \{r_i\}_{i\in T}, \{x_i\}_{i\in \bar{T}})\rangle = \langle S_T(\{y_i\}_{i\in T}, \{r_i\}_{i\in T}, \{y_i\}_{i\in \bar{T}})\rangle$$

where the probability space is over all random inputs in $\bar{T}$, namely $\{r_i\}_{i\in \bar{T}}$ (each r_i is distributed according to R_i and they are all independent).

We say that a protocol F for computing f is (*strongly*) *t-private* if any coalition T of size $\leqq t$ does not learn any additional information from the execution of the protocol. We say that a function f is (*strongly*) *t-private* if there exists a (strongly) t-private protocol that computes it.

The weak notion of privacy is different from the strong one, described above, in two ways: (a) coalitions may get some (limited) additional information (other than what follows from the inputs of the coalition members and the function value), and (b) the protocol may not always compute the correct value of the function. This is formalized as follows:

(a) Given $0 \leqq \delta \leqq 1$, we say that a protocol F for computing f is *(δ, t)-private* if the following holds: Let T be any coalition of size $\leqq t$ and let $\vec{x}$, $\vec{y} \in A_1 \times A_2 \times \cdots \times A_n$ be any two inputs that agree in their T entries and satisfy $f(\vec{x}) = f(\vec{y})$. Then the *variation distance* on the space $\mathcal{S}_T$ (messages passed between T and $\bar{T}$), given $\vec{x}$ and given $\vec{y}$, is bounded above by δ. That is,

$$\frac{1}{2}\sum_{s\in \mathcal{S}_T} |\Pr(s\,|\,\vec{x}) - \Pr(s\,|\,\vec{y})| \leqq \delta.$$

(b) Given $0 \leqq \varepsilon < 1$, we say that a protocol F has *ε-error* in computing f if

$$\forall \vec{x} : \Pr(F(\vec{x}) = f(\vec{x})) \geqq 1 - \varepsilon,$$

(in both (a) and (b) the probabilities are taken over the random inputs of all the participants).

We remark that even for $\delta = 0$, error free (δ, t)-privacy is a weaker requirement than (strong) t-privacy. In the special case of $n = 2$ we say that a function is (strongly) *private* if it is (strongly) 1-private, and it is δ-private if it is (δ, 1)-private.

3. The two-party case. In this section we consider the case where f is a Boolean function of two variables. We show that if f is weakly private then it can be expressed as $f(x_1, x_2) = f_1(x_1) \oplus f_2(x_2)$, where f_1 and f_2 are also Boolean functions. On the other hand, we show that functions of the form $f(x_1, x_2) = f_1(x_1) \oplus f_2(x_2)$ can be computed in a strongly private way.

We now present a lemma that will play a central rule in the proof of the characterization theorem.

LEMMA 1. *Let ε, $\delta \geqq 0$ satisfy $\varepsilon + \delta < \frac{1}{2}$. Let A_1, A_2, B be nonempty sets and let $f: A_1 \times A_2 \twoheadrightarrow B$ be a function that can be computed δ-privately with ε-error. Under these assumptions, for every $b \in B$, x_1, $y_1 \in A_1$, and x_2, $y_2 \in A_2$ the following condition holds: If $f(x_1, x_2) = f(x_1, y_2) = f(y_1, x_2) = b$, then $f(y_1, y_2) = b$.*

Before proving Lemma 1 we introduce three technical lemmas. The first one holds for any two-party communication protocol. It relates the probabilities of any communication string s to be sent on the four corners of any "input rectangle" (x_1, x_2), (x_1, y_2), (y_1, x_2), (y_1, y_2). (An equivalent lemma is implicitly used by Paturi and Simon [PS]).

LEMMA 2. *Let A_1, A_2 be nonempty sets and F a two-party communication protocol. For every communication string s and all inputs $x_1, y_1 \in A_1, x_2, y_2 \in A_2$*

$$\Pr(s|(x_1,x_2)) \cdot \Pr(s|(y_1,y_2)) = \Pr(s|(x_1,y_2)) \cdot \Pr(s|(y_1,x_2)).$$

Proof. Let $s = m_1 \circ m_2 \circ \cdots \circ m_k$, where m_i is the message sent in the ith round and $\circ$ denotes concatenation. Assume, without loss of generality, that the first message of the protocol is sent by P_1 and that k (the number of rounds) is even. Let $\Pr_1(s|x_1)$ denote the probability that P_1 will send the messages $m_1, m_3, m_5 \cdots$ given that its input is x_1, and that the messages received from P_2 are $m_2, m_4, m_6 \cdots$. That is,

$$\Pr_1(s|x_1) = \Pr(m_1|x_1) \cdot \Pr(m_3|x_1, m_1, m_2) \cdot \cdots \cdot \Pr(m_{k-1}|x_1, m_1, m_2, \cdots, m_{k-2}),$$

and similarly $\Pr_2(s|x_2)$ is defined as

$$\Pr_2(s|x_2) = \Pr(m_2|x_2, m_1) \cdot \Pr(m_4|x_2, m_1, m_2, m_3) \cdot \cdots \cdot \Pr(m_k|x_2, m_1, m_2, \cdots, m_{k-1}).$$

(Recall that every message is a function of the local input, the messages received from the other party, and the random input.) By these definitions, for every input (x_1, x_2) and every communication string s we have:

$$\Pr(s|(x_1,x_2)) = \Pr_1(s|x_1) \cdot \Pr_2(s|x_2)$$

and therefore

$$\begin{aligned}\Pr(s|(x_1,x_2)) \cdot \Pr(s|(y_1,y_2)) &= \Pr_1(s|x_1) \cdot \Pr_2(s|x_2) \cdot \Pr_1(s|y_1) \cdot \Pr_2(s|y_2) \\ &= \Pr(s|(x_1,y_2)) \cdot \Pr(s|(y_1,x_2)).\end{aligned}$$

This completes the proof of the lemma. □

The second technical lemma gives a lower bound on the probability of any communication string s to be sent on the input (y_1, y_2), given the probability that s will be sent on each of the inputs (x_1, x_2), (x_1, y_2), and (y_1, x_2).

LEMMA 3. *Let $0 \leqq p_1, p_2, p_3, p_4 \leqq 1$ such that $p_1 \cdot p_4 = p_2 \cdot p_3$:*

(1) *If $p_1 \leqq p_2, p_3$ then $p_4 \geqq p_1$.*

(2) *If $p_1 \geqq p_2, p_3$ then $p_4 \geqq p_1 - (p_1 - p_2) - (p_1 - p_3)$.*

(3) *If $p_2 \leqq p_1 \leqq p_3$ then $p_4 \geqq p_1 - (p_1 - p_2)$.*

(4) *If $p_3 \leqq p_1 \leqq p_2$ then $p_4 \geqq p_1 - (p_1 - p_3)$.*

Proof. We prove each of the four cases using simple arithmetic manipulations.

(1) In the case that $p_1 \leqq p_2, p_3$ if $p_1 = 0$ then clearly $p_4 \geqq p_1$. Otherwise, the following holds:

$$p_4 = \frac{p_2 \cdot p_3}{p_1} \geqq \frac{p_1 \cdot p_1}{p_1} = p_1.$$

(2) In the case that $p_1 \geqq p_2, p_3$, if $p_1 = 0$ then $p_2 = p_3 = 0$, and thus the inequality holds. Otherwise we have

$$\begin{aligned} p_4 &= \frac{p_2 \cdot p_3}{p_1} \\ &= \frac{(p_1-(p_1-p_2))\cdot(p_1-(p_1-p_3))}{p_1} \\ &= \frac{p_1^2-(p_1-p_2)\cdot p_1-(p_1-p_3)\cdot p_1+((p_1-p_2)\cdot(p_1-p_3))}{p_1} \\ &\geqq p_1-(p_1-p_2)-(p_1-p_3). \end{aligned}$$

(3) In the case that $p_2 \leqq p_1 \leqq p_3$ we have to show that $p_4 \geqq p_2$. If $p_3 = 0$ then so is p_2 and the inequality trivially holds. Otherwise, assume by way of contradiction that $p_4 < p_2$. Since $p_1 \leqq p_3$ then $p_1 \cdot p_4 < p_3 \cdot p_2$, contradicting $p_1 \cdot p_4 = p_2 \cdot p_3$.

(4) This case is similar to the proof of (3).

This completes the proof of the lemma. □

The next lemma is a trivial property of the variation distance.

LEMMA 4. *Let S be the set of all communication strings and let p_1 and p_2 be two probability distributions defined over S. Denote by $S_1 \subseteq S$ the set $\{s \mid p_1(s) \geqq p_2(s)\}$, then*

$$\sum_{s\in S_1} (p_1(s)-p_2(s)) = \frac{1}{2} \sum_{s \in S} |p_1(s)-p_2(s)|.$$

Proof. The proof is obtained by simple arithmetic manipulations:

$$\begin{aligned} 2\cdot \sum_{s\in S_1}(p_1(s)-p_2(s)) &= 2\cdot \sum_{s\in S_1} p_1(s) - 2\cdot \sum_{s\in S_1} p_2(s) \\ &= \sum_{s\in S_1} p_1(s) + \left(1-\sum_{s\in \bar{S}_1} p_1(s)\right) - \sum_{s\in S_1} p_2(s) - \left(1-\sum_{s\in\bar{S}_1} p_2(s)\right) \\ &= \sum_{s\in S_1}(p_1(s)-p_2(s)) - \sum_{s\in\bar{S}_1}(p_1(s)-p_2(s)) \\ &= \sum_{s\in S} |p_1(s)-p_2(s)|. \end{aligned}$$

(The last equality follows from the definition of S_1.) □

Using these three lemmas we can now prove Lemma 1.

Proof of Lemma 1. Let F be a protocol that computes f δ-privately with ε-error. Let S_b be the set of all communication strings whose last message is b. Recall that on the first three points, these strings correspond to executions computing the correct value of the function (which equals b). Define

$$\begin{aligned} S_b^1 &\overset{\text{def}}{=} \{s \mid s \in S_b \text{ and } \Pr(s \mid (x_1,x_2)) \leqq \Pr(s \mid (x_1,y_2)), \Pr(s \mid (y_1,x_2))\}, \\ S_b^2 &\overset{\text{def}}{=} \{s \mid s \in S_b \text{ and } \Pr(s \mid (x_1,x_2)) \geqq \Pr(s \mid (x_1,y_2)), \Pr(s \mid (y_1,x_2))\}, \\ S_b^3 &\overset{\text{def}}{=} \{s \mid s \in S_b \text{ and } \Pr(s \mid (x_1,y_2)) < \Pr(s \mid (x_1,x_2)) < \Pr(s \mid (y_1,x_2))\}, \\ S_b^4 &\overset{\text{def}}{=} \{s \mid s \in S_b \text{ and } \Pr(s \mid (y_1,x_2)) < \Pr(s \mid (x_1,x_2)) < \Pr(s \mid (x_1,y_2))\}. \end{aligned}$$

The protocol F has at most ε-error for every input. Thus to prove that $f(y_1, y_2) = b$, it suffices to show that on input (y_1, y_2) the probability of having a communication string whose last message is b, (that is $s \in S_b$), is greater than ε:

$$\sum_{s\in S_b} \Pr(s|(y_1,y_2)) = \sum_{s\in S_b^1} \Pr(s|(y_1,y_2)) + \sum_{s\in S_b^2} \Pr(s|(y_1,y_2)) + \sum_{s\in S_b^3} \Pr(s|(y_1,y_2)) + \sum_{s\in S_b^4} \Pr(s|(y_1,y_2)).$$

Now for each set S_b^i we use the appropriate part of Lemma 3 together with Lemma 2. These imply that this sum is bounded below by

$$\begin{aligned}&\sum_{s\in S_b^1} \Pr(s|(x_1,x_2))\\ &+ \sum_{s\in S_b^2} \Pr(s|(x_1,x_2)) - \sum_{s\in S_b^2} (\Pr(s|(x_1,x_2)) - \Pr(s|(x_1,y_2)))\\ &- \sum_{s\in S_b^2} (\Pr(s|(x_1,x_2)) - \Pr(s|(y_1,x_2)))\\ &+ \sum_{s\in S_b^3} \Pr(s|(x_1,x_2)) - \sum_{s\in S_b^3} (\Pr(s|(x_1,x_2)) - \Pr(s|(x_1,y_2)))\\ &+ \sum_{s\in S_b^4} \Pr(s|(x_1,x_2)) - \sum_{s\in S_b^4} (\Pr(s|(x_1,x_2)) - \Pr(s|(y_1,x_2))).\end{aligned}$$

This last expression equals

$$\sum_{s\in S_b} \Pr(s|(x_1,x_2)) - \sum_{s\in S_b^2 \cup S_b^3} (\Pr(s|(x_1,x_2)) - \Pr(s|(x_1,y_2))) - \sum_{s\in S_b^2 \cup S_b^4} (\Pr(s|(x_1,x_2)) - \Pr(s|(y_1,x_2))).$$

The first summand is at least $1 - \varepsilon$ since ε is the maximum error permitted on the input (x_1, x_2). According to Lemma 4 and the fact that the protocol F is δ-private, each of the other two summands is at most δ. Thus we have

$$\sum_{s\in S_b} \Pr(s|(y_1,y_2)) \geqq 1 - \varepsilon - \delta - \delta.$$

Since $\varepsilon + \delta < \frac{1}{2}$, we have $1 - \varepsilon - \delta - \delta > \varepsilon$. This completes the proof of Lemma 1. □

Lemma 1 suffices for showing that certain Boolean functions of two variables are not weakly private. The first example, which was given in [BGW] (with respect to strong privacy), is the OR function ($A_1 = A_2 = \{0, 1\}$ and $f(x_1, x_2) = x_1 \vee x_2$). Clearly this function does not satisfy Lemma 1. For an additional example we take A_1 and A_2 to be the set of all integers and f the IDENTITY function ($f(x_1, x_2) = 1 \Leftrightarrow x_1 = x_2$). This function is not private since for any c we have $f(c-1, c) = f(c-1, c+1) = f(c, c+1) = 0$ but $f(c, c) = 1$.

THEOREM 1. *Let $\varepsilon, \delta \geqq 0$ satisfy $\varepsilon + \delta < \frac{1}{2}$. Let A_1, A_2 be nonempty sets and $f : A_1 \times A_2 \twoheadrightarrow \{0, 1\}$ an arbitrary Boolean function. Then f can be computed δ-privately with ε-error if and only if there exist Boolean functions $f_1 : A_1 \twoheadrightarrow \{0, 1\}$, $f_2 : A_2 \twoheadrightarrow \{0, 1\}$ such that $f(x_1, x_2) = f_1(x_1) \oplus f_2(x_2)$.*

Proof. First we present a private protocol for computing any function f of the form $f(x_1, x_2) = f_1(x_1) \oplus f_2(x_2)$:

(1) P_1 computes $f_1(x_1)$ and sends its value (one bit) to P_2.

(2) P_2 computes $f_2(x_2)$ and sends $f(x_1, x_2) = f_1(x_1) \oplus f_2(x_2)$ (one bit) to P_1.

It is clear that P_1 does not learn any additional information since the only message it received during the protocol contains the function value, and P_2 does not learn any additional information since it can compute by itself (from the function value and its input) $f_1(x_1) = f(x_1, x_2) \oplus f_2(x_2)$. The above protocol computes f with no errors and with strong privacy. In addition, this protocol is deterministic and the number of bits exchanged is exactly 2.

Now we assume that f can be computed δ-privately with ε-error and we show how to find f_1 and f_2 as stated. Let α be an arbitrary element of A_2 and define the following sets:

$$B \overset{\text{def}}{=} \{x_1 \in A_1 \mid f(x_1, \alpha) = 0\},$$

$$C \overset{\text{def}}{=} \{x_2 \in A_2 \mid \forall x_1 \in B : f(x_1, x_2) = 0\}.$$

We assume, without loss of generality, that there exists some x_1 such that $f(x_1, \alpha) = 0$ (that is, B is not empty). We will show now that the function is constant over each of the domains $B \times C$, $B \times \bar{C}$, $\bar{B}$, $\times C$, $\bar{B} \times \bar{C}$:

Claim 1. *For all $x_1 \in B$ for all $x_2 \in C$: $f(x_1, x_2) = 0$.*

Proof. This proof follows directly from the definition of C.

Claim 2. *For all $x_1 \in B$ for all $x_2 \in \bar{C}$: $f(x_1, x_2) = 1$.*

Proof. Assume to the contrary that there exist $x_1 \in B$ and $x_2 \in \bar{C}$ such that $f(x_1, x_2) = 0$. Now, by the definition of $C, x_2 \in \bar{C} \Rightarrow$ there exists $y_1 \in B$ such that $f(y_1, x_2) = 1$. By the definition of B, $x_1, y_1 \in B \Rightarrow f(x_1, \alpha) = f(y_1, \alpha) = 0$. Since we have $f(x_1, \alpha) = f(y_1, \alpha) = f(x_1, x_2) = 0$, then according to Lemma 1 we must also have $f(y_1, x_2) = 0$ — contradiction.

Claim 3. *For all $x_1 \in \bar{B}$ for all $x_2 \in C$: $f(x_1, x_2) = 1$.*

Proof. Assume to the contrary that there exist $x_1 \in \bar{B}$ and $x_2 \in C$ such that $f(x_1, x_2) = 0$. It follows from the definition of $\bar{B}$ that $x_2 \neq \alpha$. Let y_1 be an arbitrary element of B (recall that B is not empty). Now, $y_1 \in B \Rightarrow f(y_1, \alpha) = 0$ and $x_2 \in C \Rightarrow f(y_1, x_2) = 0$. Since we have $f(y_1, \alpha) = f(y_1, x_2) = f(x_1, x_2) = 0$ then according to Lemma 1 we must also have $f(x_1, \alpha) = 0$ — contradicting the fact that $x_1 \in \bar{B}$.

Claim 4. *For all $x_1 \in \bar{B}$ for all $x_2 \in \bar{C}$: $f(x_1, x_2) = 0$.*

Proof. Assume to the contrary that there exists $x_1 \in \bar{B}$ and $x_2 \in \bar{C}$ such that $f(x_1, x_2) = 1$. Recall that $x_1 \in \bar{B}$ implies $f(x_1, \alpha) = 1$ and let y_1 be an arbitrary element of B, i.e., $f(y_1, \alpha) = 0$. According to Claim 2 $f(y_1, x_2) = 1$. Since we have $f(x_1, x_2) = f(y_1, x_2) = f(x_1, \alpha) = 1$ then according to Lemma 1 we must also have $f(y_1, \alpha) = 1$ — contradicting the fact that $y_1 \in B$.

We now define:

$$f_1(x_1) = \begin{cases} 0 & \text{if } x_1 \in B \\ 1 & \text{if } x_1 \notin B \end{cases}$$

$$f_2(x_2) = \begin{cases} 0 & \text{if } x_2 \in C \\ 1 & \text{if } x_2 \notin C \end{cases}$$

then by Claims 1–4 we have $f(x_1, x_2) = f_1(x_1) \oplus f_2(x_2)$ for each of the four possible combinations $(x_1, x_2) \in B \times C$, $B \times \bar{C}$, $\bar{B} \times C$, $\bar{B} \times \bar{C}$. This completes the proof of Theorem 1. □

One conclusion of Theorem 1 is that, in the two-party case, if f can be privately computed then it can be privately computed by a deterministic protocol. We emphasize that this does not hold for the multiparty case. A second conclusion is that in the two-party case whatever can be privately computed under the weak notion can also be privately computed under the strong notion. Thus (for any ε, $\delta \geqq 0$ such that $\varepsilon + \delta < \frac{1}{2}$) these two notions are equivalent. As we will see in the next section, this conclusion holds in the multiparty case as well.

4. The multiparty case. In this section we prove the main result of our paper: A complete characterization of n-variable Boolean functions that are $\lceil n/2 \rceil$-private. We start with a lemma that helps reduce the multiparty case to the two-party scenario. Using this lemma, we proceed to a detailed proof of the characterization theorem. Finally, we give some implications and corollaries. Throughout this section, we will say that a function f is *weakly t-private* if there are ε, $\delta \geqq 0$ satisfying $\varepsilon + \delta < \frac{1}{2}$, such that f can be computed (δ, t)-privately with ε-error.

LEMMA 5. *Let $A_1, A_2, \cdots, A_n$ be nonempty sets, ε, $\delta \geqq 0$ satisfying $\varepsilon + \delta < \frac{1}{2}$, and $f : A_1 \times A_2 \times \cdots \times A_n \to \{0, 1\}$* be *$(\delta, \lceil n/2 \rceil)$-privately computable with ε-error. Let $S \subseteq \{1, 2, \cdots, n\}$ be any subset of size $\lceil n/2 \rceil$. Denote by D (respectively, E) the Cartesian product of the A_i with $i \in S$ (respectively, $i \in \bar{S}$). Then, viewing f as a two argument function $f : D \times E \to \{0, 1\}$, f is δ-private with ε-error.*

Proof. Given an n-party protocol for computing $f : A_1 \times A_2 \times \cdots \times A_n \to \{0, 1\}$ $(\delta, \lceil n/2 \rceil)$-privately with ε-error, we convert it into a two party protocol for computing $f : D \times E \to \{0, 1\}$. Denote the two parties by Q_1 and Q_2. Q_1 simulates the role of the $\lceil n/2 \rceil$ processors P_i with $i \in S$ using its source of random bits as $\lceil n/2 \rceil$ independent sources of random bits. (Q_2 acts similarly with respect to $\bar{S}$ whose size is $\lfloor n/2 \rfloor$.) The messages exchanged between Q_1 and Q_2 in this two-party protocol correspond to messages exchanged between S and $\bar{S}$ processors in the original multiparty protocol. Using the definitions, it is easy to see that this two-party protocol computes $f : D \times E \to \{0, 1\}$ δ-privately with ε-error. □

We remark that to make use of the $\lceil n/2 \rceil$-privacy, both S and $\bar{S}$ must be of size not exceeding $\lceil n/2 \rceil$. Our main theorem states that if $f : A_1 \times A_2 \times \cdots \times A_n \to \{0, 1\}$ is weakly $\lceil n/2 \rceil$-private, then f can be expressed as the exclusive-or of n Boolean functions $f_1, f_2, \cdots, f_n$. The proof makes use of Theorem 1 and Lemma 5.

THEOREM 2. *Let $A_1, A_2, \cdots, A_n$ be nonempty sets, and $f : A_1 \times A_2 \times \cdots \times A_n \to \{0, 1\}$. Suppose f is weakly $\lceil n/2 \rceil$-private. Then there are n Boolean functions $f_1 : A_1 \to \{0, 1\}, f_2 : A_2 \to \{0, 1\}, \cdots, f_n : A_n \to \{0, 1\}$ such that*

$$f(x_1, x_2, \cdots, x_n) = f_1(x_1) \oplus f_2(x_2) \oplus \cdots \oplus f_n(x_n).$$

Proof. The proof consists of two parts. In the first part we show that for every i the set A_i can be partitioned into two disjoint sets

$$A_i = B_i \cup C_i \qquad (B_i \cap C_i = \emptyset, B_i \neq \emptyset)$$

such that for all $b_i \in B_i$, $\check{b}_i \in B_i$, $c_i \in C_i$, $x_j \in A_j$ $(j \neq i)$

$$(1) \qquad f(x_1, \cdots, x_{i-1}, b_i, x_{i+1}, \cdots, x_n) \neq f(x_1, \cdots, x_{i-1}, c_i, x_{i+1}, \cdots, x_n)$$

and

$$(2) \qquad f(x_1, \cdots, x_{i-1}, b_i, x_{i+1}, \cdots, x_n) = f(x_1, \cdots, x_{i-1}, \check{b}_i, x_{i+1}, \cdots, x_n).$$

In the second part of the proof we show how to derive the desired characterization of f from this property.

To simplify the exposition, the first part is proven for $i = 1$ and the subscript is omitted from the two sets in the partition. We use the following notations:

$$R \overset{\text{def}}{=} \left\{2, 3, \cdots, \left\lceil \frac{n}{2} \right\rceil\right\}$$

$$T \overset{\text{def}}{=} \left\{\left\lceil \frac{n}{2} \right\rceil + 1, \cdots, n-1\right\}$$

$$S_1 \overset{\text{def}}{=} \{1\} \cup R$$

$$S_2 \overset{\text{def}}{=} \{n\} \cup R.$$

Each of the sets S_1 and S_2 are of size $\lceil n/2 \rceil$, and thus Lemma 5 applies to both. We establish the existence of the desired partition by examining the effect of "switching" the variable x_1 — from S_1 to $\bar{S}_2$. By Lemma 5 together with Theorem 1, there are functions

$$g: \times_{i \in S_1} A_i \to \{0, 1\}, \qquad h: \times_{i \in \bar{S}_1} A_i \to \{0, 1\}$$

such that for every $x_i \in A_i$

$$f(x_1, \cdots, x_n) = g(x_1, \cdots, x_{\lceil n/2 \rceil}) \oplus h(x_{\lceil n/2 \rceil + 1}, \cdots, x_n). \tag{3}$$

Similarly, there are functions

$$\tilde{g}: \times_{i \in S_2} A_i \to \{0, 1\}, \qquad \tilde{h}: \times_{i \in \bar{S}_2} A_i \to \{0, 1\}$$

such that

$$f(x_1, \cdots, x_n) = \tilde{g}(x_n, x_2, \cdots, x_{\lceil n/2 \rceil}) \oplus \tilde{h}(x_{\lceil n/2 \rceil + 1}, \cdots, x_{n-1}, x_1). \tag{4}$$

We distinguish between two cases. If f does not depend on x_1, its first argument, then we simply take $B = A_1$ and $C = \emptyset$. The interesting case is where f does depend on x_1. That is, there are $b, c \in A_1$ $(b \neq c)$, $\vec{\alpha} \in \times_{i \in R} A_i$, $\vec{\beta} \in \times_{i \in T} A_i$, $d \in A_n$ such that

$$f(b, \vec{\alpha}, \vec{\beta}, d) \neq f(c, \vec{\alpha}, \vec{\beta}, d).$$

Define the sets

$$B \overset{\text{def}}{=} \{a_1 \in A_1 \mid f(a_1, \vec{\alpha}, \vec{\beta}, d) = f(b, \vec{\alpha}, \vec{\beta}, d)\},$$

$$C \overset{\text{def}}{=} \{a_1 \in A_1 \mid f(a_1, \vec{\alpha}, \vec{\beta}, d) = f(c, \vec{\alpha}, \vec{\beta}, d)\}.$$

Since f is a Boolean function, $A_1 = B \cup C$. By the definition, for all $b \in B$, $c \in C$

$$f(b, \vec{\alpha}, \vec{\beta}, d) \neq f(c, \vec{\alpha}, \vec{\beta}, d). \tag{5}$$

Assume, by way of contradiction, the existence of $b \in B$, $c \in C$, $\vec{x}_R \in \times_{i \in R} A_i$, $\vec{x}_T \in \times_{i \in T} A_i$, $x_n \in A_n$ such that

$$f(b, \vec{x}_R, \vec{x}_T, x_n) = f(c, \vec{x}_R, \vec{x}_T, x_n).$$

By (3), we have

$$g(b, \vec{x}_R) \oplus h(\vec{x}_T, x_n) = g(c, \vec{x}_R) \oplus h(\vec{x}_T, x_n)$$

and thus

$$g(b, \vec{x}_R) = g(c, \vec{x}_R).$$

Using (3) again, this implies

$$f(b, \vec{x}_R, \vec{\beta}, d) = f(c, \vec{x}_R, \vec{\beta}, d).$$

Now, using (4), this implies

$$\tilde{h}(\vec{\beta},b)=\tilde{h}(\vec{\beta},c)$$

and thus

$$\begin{aligned}f(b,\vec{\alpha},\vec{\beta},d)&=\tilde{g}(d,\vec{\alpha})\oplus\tilde{h}(\vec{\beta},b)\\&=\tilde{g}(d,\vec{\alpha})\oplus\tilde{h}(\vec{\beta},c)\\&=f(c,\vec{\alpha},\vec{\beta},d)\end{aligned}$$

contradicting (5).

Thus for every $b\in B$, $\check{b}\in B$, $c\in C$, $\vec{x}_R\in\times_{i\in R}A_i$, $\vec{x}_T\in\times_{i\in T}A_i$, $x_n\in A_n$

$$f(b,\vec{x}_R,\vec{x}_T,x_n)\neq f(c,\vec{x}_R,\vec{x}_T,x_n)$$

$$f(\check{b},\vec{x}_R,\vec{x}_T,x_n)\neq f(c,\vec{x}_R,\vec{x}_T,x_n).$$

Again, since f is Boolean, these two inequalities imply

$$f(b,\vec{x}_R,\vec{x}_T,x_n)=f(\check{b},\vec{x}_R,\vec{x}_T,x_n),$$

which completes the first part of the proof.

In the second part of the proof, we show that if for every i the set A_i can be partitioned such that (1) and (2) hold, then the function f has the desired form. We begin the second part of the proof by fixing an element $\tilde{b}_i\in B_i$ for each $i=1,2,\cdots n$. Without loss of generality, assume

$$f(\tilde{b}_1,\tilde{b}_2,\cdots,\tilde{b}_n)=0.$$

Define the functions $f_i:A_i\rightarrow\{0,1\}$ by

$$f_i(x_i)=\begin{cases}0 & \text{if } x_i\in B_i\\ 1 & \text{if } x_i\in C_i.\end{cases}$$

Given any $x_1\in A_1,\cdots,x_n\in A_n$, let $J\subseteq\{1,\cdots,n\}$ be the set of indices of the x_i's in C_i, and $M\subseteq\{1,\cdots,n\}$ its complement. Denote by l the size of J, and let $k=n-l$. We will index the elements in J and M separately, that is

$$(x_1,\cdots,x_n)=(x_{j_1},x_{j_2},\cdots,x_{m_1},\cdots,x_{m_k},\cdots,x_{j_l}).$$

By (1) and (2)

$$\begin{aligned}f(x_1,\cdots,x_n)&=f(x_{j_1},x_{j_2},\cdots,x_{m_1},\cdots,x_{m_k},\cdots,x_{j_l})\\&=f(x_{j_1},x_{j_2},\cdots,\tilde{b}_{m_1},\cdots,\tilde{b}_{m_k},\cdots,x_{j_l})\\&=f(\tilde{b}_{j_1},x_{j_2},\cdots,\tilde{b}_{m_1},\cdots,\tilde{b}_{m_k},\cdots,x_{j_l})\oplus 1\\&=f(\tilde{b}_{j_1},\tilde{b}_{j_2},\cdots,\tilde{b}_{m_1},\cdots,\tilde{b}_{m_k},\cdots,x_{j_l})\oplus 1\oplus 1\\&\;\vdots\\&=f(\tilde{b}_{j_1},\tilde{b}_{j_2},\cdots,\tilde{b}_{m_1},\cdots,\tilde{b}_{m_k},\cdots,\tilde{b}_{j_l})\oplus(l \bmod 2)\\&=l(\bmod 2).\end{aligned}$$

By the definition of the f_i's

$$f_1(x_1)\oplus f_2(x_2)\oplus\cdots\oplus f_n(x_n)=l(\bmod 2)$$

and thus

$$f(x_1,x_2,\cdots,x_n)=f_1(x_1)\oplus f_2(x_2)\oplus\cdots\oplus f_n(x_n).$$

for each $\vec{x}\in\times_{i=1}^{n}A_i$. □

We now turn to some implications of Theorem 2. First we note that if $f: A_1\times A_2\times\cdots\times A_n\rightarrow\{0,1\}$ has the form

$$f(x_1,x_2,\cdots,x_n)=f_1(x_1)\oplus f_2(x_2)\oplus\cdots\oplus f_n(x_n),$$

then there is a very simple protocol [Bh] for computing f n-privately. The ith participant locally computes the bit $y_i\overset{\text{def}}{=}f_i(x_i)$. Then, it picks $n-1$ random independent bits $y_{i,1}$, $y_{i,2},\cdots,y_{i,n-1}$, and $y_{i,n}$ such that $y_i=y_{i,1}\oplus y_{i,2}\oplus\cdots\oplus y_{i,n}$ holds. It sends $y_{i,j}$ to the jth participant over their joint channel. After getting the n splits $y_{1,i},y_{2,i},\cdots,y_{n,i}$, the ith participant adds them modulo 2 and sends the result $z_i\overset{\text{def}}{=}y_{1,i}\oplus y_{2,i}\oplus\cdots\oplus y_{n,i}$ to every other participant. The sum modulo 2 of these n z_i's equals $f(x_1,x_2,\cdots,x_n)$. This protocol is (strongly) t-private for any $1\leqq t\leqq n$. Thus we have

THEOREM 3. *Let $f: A_1\times A_2\times\cdots\times A_n\rightarrow\{0,1\}$. If f is weakly $\lceil n/2\rceil$-private, then it is (strongly) n-private.*

In particular, there is no Boolean function that is t-private but not $t+1$-private for any t in the range $\lceil n/2\rceil\leqq t<n$.

We now consider the case where f is a Boolean function of Boolean variables. There are only four Boolean functions of a Boolean variable (the two constants, the variable, and its complement). The form of private Boolean functions of Boolean variables is thus particularly simple.

THEOREM 4. *A function $f:\{0,1\}^n\rightarrow\{0,1\}$ is n-private if and only if there is a subset $J\subseteq\{1,\cdots,n\}$ such that*

$$\text{either}\quad f(x_1,x_2,\cdots,x_n)=\bigoplus_{j\in J}x_j$$

$$\text{or}\quad \bar{f}(x_1,x_2,\cdots,x_n)=\bigoplus_{j\in J}x_j.$$

Finally, we remark that our characterization of Boolean functions that are $\lceil n/2\rceil$-private is valid even for functions defined over infinite domains. The $\lfloor(n-1)/2\rfloor$-private protocol of [BGW], on the other hand, relied heavily on the finiteness of the domains. We conjecture that for infinite domains there exist functions that are not $\lfloor(n-1)/2\rfloor$-private. Indeed, the secret-sharing techniques used in that $\lfloor(n-1)/2\rfloor$-private protocol cannot be utilized in countable domains, as shown in [BS], [CK].

Acknowledgments. We would like to thank Shai Ben-David and Oded Goldreich for helpful discussions and comments on the topics of this paper.

Note added in proof. Chor, Gereb-Gravs, and Kushilevitz [Proc. 31st IEEE Conference on Foundations of Computer Science, 1990] have recently proved this conjecture for various Boolean and non-Boolean functions defined over countable domains.

REFERENCES

[Bh] J. D. BENALOH-COHEN, *Secret sharing homomorphisms: keeping shares of a secret secret*, in Advances in Cryptography—Crypto86 (proceedings), A. M. Odlyzko, ed., Lecture Notes in Computer Science, 263, Springer-Verlag, Berlin, New York, 1987, pp. 251–260.

[BGW] M. Ben-or, S. Goldwasser, and A. Wigderson, *Completeness theorems for noncryptographic fault-tolerant distributed computation*, in Proc. of 20th Symposium on Theory of Computing, ACM, 1988, pp. 1–10.

[BS] G. R. Blakley and L. Swanson, *Security proof for information protection systems*, in Proc. IEEE Symposium on Security and Privacy, 1981, pp. 75–88.

[CCD] D. Chaum, C. Crepeau, and I. Damgard, *Multiparty unconditionally secure protocols*, in Proc. of 20th Symposium on Theory of Computing, ACM, 1988, pp. 11–19.

[CK] B. Chor and E. Kushilevitz, *Secret sharing over infinite domains*, in Advances in Cryptography—Crypto89 (proceedings), Springer-Verlag, Berlin, New York, to appear.

[GMW] O. Goldreich, S. Micali, and A. Wigderson, *How to play any mental game*, in Proc. of 19th Symposium on Theory of Computing, ACM, 1987, pp. 218–229.

[PS] R. Paturi and J. Simon, *Probabilistic communication complexity*, J. Comput. System Sci., 33 (1986), pp. 106–123.

[YA] A. C. Yao, *How to generate and exchange secrets*, in Proc. of 27th Foundations of Computer Science, IEEE, 1986, pp. 162–167.

SIAM J. DISC. MATH.
Vol. 4, No. 1, pp. 48-57, February 1991

005

A SIMPLE PROOF OF THE $O(\sqrt{n}\log^{3/4} n)$ UPRIGHT MATCHING BOUND*

E. G. COFFMAN, JR.† AND P. W. SHOR†

Abstract. The stochastic upright matching problem has had many important applications, most notably in statistics and the average-case analysis of algorithms. A problem instance is a set of n points chosen uniformly at random in the unit square. The points are labeled with signs; the signs are chosen independently and each is equally likely to be a plus or minus. An up-right matching of S is a matching of minus points to plus points such that if (x, y) is a minus point matched to the plus point (x', y'), then $x \leqq x'$ and $y \leqq y'$. The problem is to estimate the expected number of points left unmatched in a maximum upright matching of S. It is well known that if U_n denotes the number of unmatched points, then $E[U_n] = \Theta(\sqrt{n}\log^{3/4} n)$. Existing proofs of the upper bound $O(\sqrt{n}\log^{3/4} n)$ are quite long and difficult to follow. This paper presents a much simpler and more compact proof. A distinctive feature of the new proof is the use of Fourier expansions.

Key words. stochastic planar matching, Fourier series, analysis of algorithms, empirical discrepancy

AMS(MOS) subject classifications. 60D05, 68Q25

1. Introduction. Consider a set S of n points chosen uniformly at random in the unit square. Each point carries a sign; the signs of the points are chosen independently and each is equally likely to be a plus or minus. An *upright matching* of S is a matching of minus points to plus points such that if (x, y) is a minus point matched to the plus point (x', y'), then $x \leqq x'$ and $y \leqq y'$. Figure 1 shows an example.

An efficient algorithm for finding maximum upright matchings can be found in [3]. Our interest focuses on estimates of the number U_n of points left unmatched in such matchings. Shor [6] gave a relatively simple proof of the lower bound

$$E[U_n] = \Omega(\sqrt{n}\log^{3/4} n).$$

Leighton and Shor [4] then proved the corresponding upper bound

$$E[U_n] = O(\sqrt{n}\log^{3/4} n). \tag{1}$$

Subsequently and independently, Rhee and Talagrand [5] also proved (1) using different methods. The proofs of (1) in [4], [5] are ingenious but quite complicated; a significant effort is required to follow the many details of the arguments. Our purpose here is to give a much simpler, more compact proof of this important result.

We refer the reader to [4] for a digest of the background and applications of these and closely related results; the statistical application is amplified in [5].

The proof of (1) in [4] falls out as a special case of a corresponding result for the more general and more difficult problem of minimax grid matching. The combinatorial properties needed in [4] require an especially long analysis compared to the one given here in § 3. In [5] upright matching is related to an essentially equivalent problem dealing with empirical measures in statistics. The probabilistic arguments in the proof of (1) are drawn from the techniques of majorizing measures developed by Fernique and others (see [5] for references). But again, the combinatorial results needed in support of the approach are difficult.

The proof of (1) given in the next two sections adheres to well known and elementary methods. A key to the greater simplicity of this proof is the use of Fourier expansions.

* Received by the editors April 7, 1989; accepted for publication (in revised form) November 21, 1989.

† AT&T Bell Laboratories, Murray Hill, New Jersey 07974.

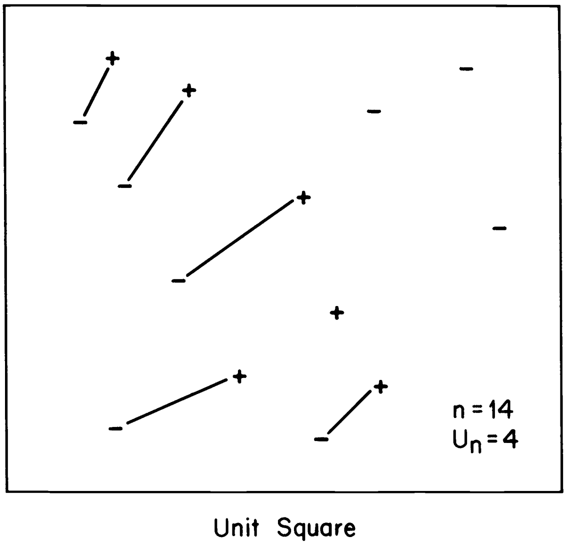


FIG. 1. *A maximum upright matching.*

In rough outline, there are points in common among the methods used here and in [4], [5]. A brief discussion of these is best deferred to § 4, after the new proof is given.

2. Preliminaries. As in [4], [5] it is convenient to reformulate our problem in terms of *discrepancies*. Consider any subset L of the unit square and define the plus discrepancy of L, $\Delta^+(L)$, as the number of plus points in L less the number of minus points in L. The term discrepancy by itself refers to $\Delta(L) = |\Delta^+(L)|$. L is called a *lower layer* if it is closed and if $(x, y) \in L$ implies $(x', y') \in L$ whenever $x' \leqq x$ and $y' \leqq y$. It follows from Hall's matching theorem that U_n is equal in distribution to $\sup_{L \in \mathscr{L}} \Delta^+(L)$, where $\mathscr{L}$ is the set of all lower layers. We will prove

$$E[\sup_{L \in \mathscr{L}} \Delta(L)] = O(\sqrt{n} \log^{3/4} n); \tag{2}$$

the desired result follows trivially.

For each lower layer L there exist lower layers L' such that $\Delta(L) = \Delta(L')$ with probability 1 and such that the boundaries of L' are the unit intervals on the x and y axis and a third, nonincreasing boundary extending from (0, 1) to (1, 0). This third boundary is called a *lower layer function*. The following lemma furnishes a basis for the probability estimates needed to prove (1). The result can be found, without proof, in [4]. A simple proof is given below. The notation Δf refers to the discrepancy of the lower layer defined by f.

LEMMA 1. *Let f_1 and f_2 be two lower layer functions with $\int_0^1 |f_1(x) - f_2(x)|\, dx = \alpha$. Then there exists a $c > 0$ such that*

$$\Pr\{|\Delta f_1 - \Delta f_2| > x\} = O(e^{-cx^2/(\alpha n)}), \qquad x \leqq \alpha n$$
$$= O(e^{-cx}), \qquad x > \alpha n.$$

Proof. Enumerate the points in S and let R denote the region bounded entirely by f_1 and f_2 and having area α. Figure 2 illustrates the definition. Define $p_k = 0$ if the kth point of S is not in R; otherwise, $p_k = +1$ or -1 according to whether the kth point is a plus or minus. Then $\Delta(R) = \sum_{k=1}^n p_k$ is a sum of independently and identically distributed

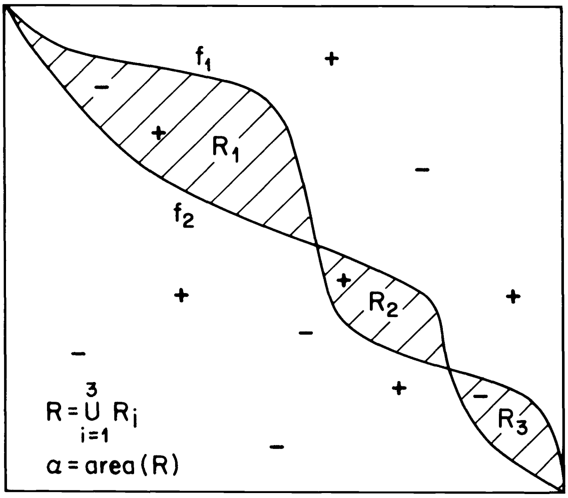


FIG. 2. *Illustration for Lemma* 1.

(i.i.d.) random variables on $[-1, +1]$ with mean 0 and variance $n\alpha$. It is easily seen that, by symmetry, $\Delta(R)$ is equal in distribution to $|\Delta^+f_1 - \Delta^+f_2|$. Since $|\Delta f_1 - \Delta f_2| \leqq |\Delta^+f_1 - \Delta^+f_2|$, the result follows easily from Bernstein's bound [1] applied to $\Delta(R)$:

$$\Pr\{|\Delta f_1 - \Delta f_2| > x\} \leqq \Pr\{\Delta(R) > x\} \leqq \exp\left\{-\frac{x^2/2}{n\alpha + x/3}\right\}. \qquad \square$$

It is readily verified that the set of partitions of S created by lower layer functions is also created by the subset consisting only of decreasing *step* functions. Now rotate the unit square 45° counterclockwise, center it at the point $(\frac{1}{2}, 0)$, and scale it down by a factor of $\sqrt{2}$. The problem instance changes as illustrated in Fig. 3, where a lower layer step function becomes a piecewise linear function $f(x)$, $0 \leqq x \leqq 1$, with the slopes of the pieces alternating between $+1$ and -1.

Hereafter, our terminology refers to this transformed version of the problem. Lower layer functions are defined on $[0, 1]$, they are completely contained in the rotated square, and they vanish at $x = 0$ and 1. Note that Lemma 1 continues to apply in this new setup. Let $\mathscr{F}$ denote the subset of piecewise linear lower layer functions with slopes alternating between -1 and $+1$.

Another useful lemma is given next. It uses a convention that applies throughout the remainder of the paper: When we write "$g_1(n) = O(g_2(n))$ with high probability" for given functions $g_1(n)$, $g_2(n)$, $n = 1, 2, \cdots$, we mean that there exist constants $\beta > 0$ and $c \geqq 1$ such that for all n sufficiently large, $\Pr\{g_1(n) > \beta g_2(n)\} \leqq 1/n^c$. Occasionally, we write whp as an abbreviation for "with high probability." Also, the symbol c will be used generically to denote constants; unless noted otherwise, constraints on c are determined by the immediate context only.

LEMMA 2. *Let $f_1 \in \mathscr{F}$ and let f_2 be any other function over $[0, 1]$ such that for some $c > 0$, $|f_1(x) - f_2(x)| \leqq c\sqrt{\log n/n}$*[‡] *uniformly in x, $0 \leqq x \leqq 1$. Then $|\Delta f_1 - \Delta f_2| = O(\sqrt{n}\log n)$ with high probability.*

Proof. Place a grid of squares of sizes $\sqrt{\log n/n} \times \sqrt{\log n/n}$ over the unit square, as shown in Fig. 4. The number of points within a grid square entirely inside the rotated

[‡] Unless noted otherwise, logarithms are base 2.

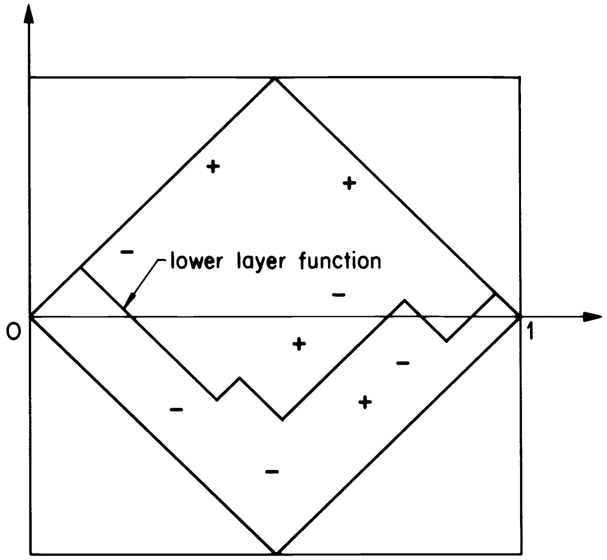


FIG. 3. *The transformed problem.*

square is binomially distributed with mean 2 log n. We find from standard estimates for this distribution that *all* squares in the grid have $O(\log n)$ points with high probability.

Now in any column of the grid, f_1 intersects at most two squares (since $|f'_1(x)| = 1$ on the pieces of f_1), so $|f_1(x) - f_2(x)| \leqq c\sqrt{\log n/n}$ shows that the difference in the discrepancies of f_1 and f_2 within any column is concentrated in at most a constant number of squares. Then over [0, 1] the difference in the discrepancies is concentrated in at most $O(\sqrt{n/\log n})$ squares. The lemma follows at once from the fact that all squares have $O(\log n)$ points with high probability. □

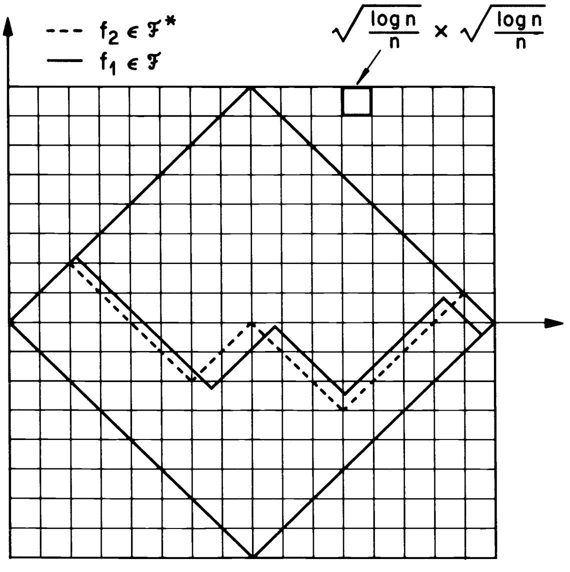

--- $f_2 \in \mathcal{F}^*$
— $f_1 \in \mathcal{F}$
$\sqrt{\frac{\log n}{n}} \times \sqrt{\frac{\log n}{n}}$

FIG. 4. *Approximating* $f_1 \in \mathcal{F}$ *by* $f_2 \in \mathcal{F}^*$.

Consider the grid introduced in the proof of Lemma 2. It is clear that for any function $f_1 \in \mathscr{F}$ we can construct another function $f_2 \in \mathscr{F}$ such that the vertices of f_2 coincide with vertices in the grid and such that $|f_1(x) - f_2(x)| \leqq \sqrt{\log n/n}$ uniformly in x, $0 \leqq x \leqq 1$. Let $\mathscr{F}^*$ be the subset of such functions. Figure 4 illustrates the construction. Clearly, by Lemma 2, (1) will be proved if we can show that

$$E[\sup_{f \in \mathscr{F}^*} \Delta f] = O(\sqrt{n} \log^{3/4} n). \tag{3}$$

Elementary Fourier analysis shows that a lower layer function $f \in \mathscr{F}$ may be represented by the sine series

$$f(x) = \sum_{i \geqq 1} a_i \sin \pi i x, \qquad 0 \leqq x \leqq 1. \tag{4}$$

These expansions play a key role in simplifying the proof of (1). Note that, since $|f'(x)| = 1$ on the pieces of f, we have

$$1 = \int_0^1 [f'(x)]^2 \, dx = \frac{\pi^2}{2} \sum_{i \geqq 1} i^2 a_i^2$$

and

$$\sum_{i \geqq 1} i^2 a_i^2 = 2/\pi^2. \tag{5}$$

Our final preliminary result describes the convergence in (4) for $f \in \mathscr{F}^*$, and is an immediate consequence of a result in [2, p. 21]. Let $f_n(x)$, $n \geqq 1$, denote the nth partial sum in (4).

LEMMA 3. *There exists a universal constant $c > 0$ such that for all $f \in \mathscr{F}^*$ and for all x, $0 \leqq x \leqq 1$,*

$$|f(x) - f_n(x)| \leqq \frac{c}{\sqrt{n \log n}}.$$

Remark. Lemma 3 in fact holds for any $f \in \mathscr{F}$ having at most $\sqrt{n/\log n}$ vertices.

3. The main result.

THEOREM. *For the expected number of unmatched points in a maximum upright matching, we have*

$$E[U_n] = O(\sqrt{n} \log^{3/4} n).$$

Proof. Trivially, $U_n \leqq n$, so for any $c > 0$

$$E[U_n] \leqq c\sqrt{n} \log^{3/4} n + n \Pr\{U_n > c\sqrt{n} \log^{3/4} n\}.$$

Then, since U_n is stochastically smaller than $\sup_{L \in \mathscr{L}} \Delta(L)$, it is enough to prove that $\sup_{L \in \mathscr{L}} \Delta(L) = O(\sqrt{n} \log^{3/4} n)$ with high probability. This in turn will be proved if we can show that $\sup_{f \in \mathscr{F}^*} \Delta f = O(\sqrt{n} \log^{3/4} n)$ with high probability (see (3)). This last result is proved below.

Let $f^{(1)}, f^{(2)}, \cdots, f^{(\log n)}$ be successively better approximations of $f \in \mathscr{F}^*$ defined by

$$f^{(k)}(x) = \sum_{i=1}^{2^{k+1}} a_i(k) \sin \pi i x, \qquad 1 \leqq k \leqq \lfloor \log n \rfloor, \tag{6}$$

where $a_i(k)$ is a_i truncated to the $\lfloor \log(\sqrt{\log n}\ 2^{3k/2})\rfloor$ most significant bits of its binary representation. (Hereafter, we shall omit the floor notation and treat the affected quantities as integers; extension of the analysis to noninteger values is trivial and influences only hidden constants.) By (6), differences in accuracy are bounded by

$$(7) \qquad |a_i(k)-a_i(k-1)| \leqq \frac{1}{\sqrt{\log n}\, 2^{3(k-1)/2}}.$$

Clearly, for fixed k, the possible functions $f^{(k)}$ make up a finite set. We will be counting certain subsets of these functions in terms of properties defined by

$$(8) \qquad t(f^{(k)}) = \sum_{j=0}^{k} r_j(f^{(k)})2^{j-k} \quad \text{with} \quad r_j(f^{(k)}) = 4^j \sum_{i=2^j+1}^{2^{j+1}} a_i^2(k), \tau \geqq 1$$

$$\tau_0(f^{(k)}) = a_1^2(k)+a_2^2(k)$$

From (5), we have

$$(9) \qquad \sum_{j=0}^{\log n} r_j(f^{(k)}) = a_1^2(k) + \sum_{j=0}^{\log n} \sum_{i=2^j+1}^{2^{j+1}} [2^j a_i(k)]^2 \leqq \sum_{i=1}^{\infty} i^2 a_i^2 = 2/\pi^2$$

and hence

$$(10) \qquad \sum_{k=1}^{\log n} t(f^{(k)}) = \sum_{k=1}^{\log n} \sum_{j=0}^{k} r_j(f^{(k)})2^{j-k} \leqq \sum_{j=0}^{\log n} r_j(f^{(\log n)}) \sum_{k=j}^{\log n} 2^{j-k} \leqq 4/\pi^2.$$

The following lemma comprises the combinatorial part of the proof.

LEMMA 4. *There exists a universal constant c and a mapping from $\mathscr{F}^*$ into $\mathscr{R}^{\log n}$, with values denoted by $(s_1(f), s_2(f), \cdots, s_{\log n}(f))$, such that for each $f \in \mathscr{F}^*$, we have $\sum_{k=1}^{\log n} s_k(f) \leqq c$,*

$$(11) \qquad \int_0^1 |f^{(1)}(x)|\ dx \leqq \sqrt{s_1(f)}/2,$$

$$\int_0^1 |f^{(k)}(x)-f^{(k-1)}(x)|\ dx \leqq \sqrt{s_k(f)}/2^k, \qquad 2 \leqq k \leqq \log n,$$

and if $\eta_k(\sigma)$ denotes the number of functions $g^{(k)}$, $g \in \mathscr{F}^*$, such that $s_k(g) \leqq \sigma$, then

$$(12) \qquad \eta_k(\sigma) \leqq (\sigma \log n)^{2^k} \wedge 2^{\sqrt{n \log n}}, \qquad 1 \leqq k \leqq \log n.$$

Proof. We will show that the mapping $s_k(f) = \gamma t(f^{(k)}) + \gamma/\log n$, for an appropriately chosen constant γ has properties (11) and (12). To prove the first property, consider a function $f \in \mathscr{F}^*$ and write from (6)

$$(13) \quad f^{(k)}(x)-f^{(k-1)}(x) = \sum_{i=2^k+1}^{2^{k+1}} a_i(k)\sin \pi i x + \sum_{i=1}^{2^k} [a_i(k)-a_i(k-1)] \sin \pi i x,$$

$$2 \leqq k \leqq \log n.$$

Let $g_k(x)$ and $h_k(x)$ denote the first and second sums in (13), respectively. We have from (8),

$$\int_0^1 g_k^2(x)\ dx = \frac{1}{2} \sum_{i=2^k+1}^{2^{k+1}} a_i^2(k) = \frac{1}{2} 4^{-k} r_k(f^{(k)}),$$

so by Schwarz's inequality, with $r_k \equiv r_k(f^{(k)})$,

$$\int_0^1 |g_k(x)|\, dx \leqq 2^{-k}\sqrt{r_k}. \tag{14}$$

Next, by (7),

$$\int_0^1 h_k^2(x)\, dx \leqq \frac{1}{2}\sum_{i=1}^{2^k}[a_i(k)-a_i(k-1)]^2 \leqq 2^{k+2}/(\log n 2^{3k}),$$

so,

$$\int_0^1 |h_k(x)|\, dx \leqq 2^{-k}\sqrt{4/\log n}. \tag{15}$$

Now add (14) and (15) and note that $\sqrt{r_k}+\sqrt{4/\log n} \leqq \sqrt{2(r_k+4/\log n)}$ by Cauchy's inequality. Then by (8)–(10) and (13)–(15), any $\gamma \geqq 8$ gives $s_k(f) = \gamma(t(f^{(k)})+1/\log n) \geqq 2(r_k+4/\log n)$ and $\sum_{k=1}^{\log n} s_k(f) \leqq c$, with c independent of f and n.

For the second part of the lemma, we note first that, for all k, $\eta_k(\sigma) \leqq 2^{\sqrt{n/\log n}}$ follows from the definition of $\mathscr{F}^*$ (there are at most $2^{\sqrt{n/\log n}}$ functions $f \in \mathscr{F}^*$, since the vertices of these functions are restricted to the vertices of a $\sqrt{n/\log n} \times \sqrt{n/\log n}$ grid on the unit square). To prove $\eta_k(\sigma) \leqq (\sigma \log n)^{2^k}$, we first establish a bound on $\mu_k(\tau)$, defined as the number of functions $g^{(k)}$, $g \in \mathscr{F}^*$, such that $t(g^{(k)}) \leqq \tau$. This bound will hold for $\tau \geqq 1/\log n$.

Consider a function $f \in \mathscr{F}^*$ and rewrite $f^{(k)}$ in (6) as

$$f^{(k)}(x) = a_1(k)\sin \pi x + a_2(k)\sin 2\pi x + \sum_{j=1}^{k}\sum_{i=2^j+1}^{2^{j+1}} a_i(k)\sin \pi i x. \tag{16}$$

Now consider the number of possibilities in the jth inner sum, i.e., the number of vectors of coefficients $a_i(k)$, $2^j+1 \leqq i \leqq 2^{j+1}$, $1 \leqq j \leqq k$. It is more convenient to work with the numbers $b_i(k) = a_i(k)2^{3k/2}\sqrt{\log n}$, since these are integers by definition of the $a_i(k)$'s. By (8), the $b_i(k)$'s satisfy

$$\begin{gathered} \sum_{i=2^j+1}^{2^{j+1}} b_i^2(k) = 2^{3k}\log n \sum_{i=2^j+1}^{2^{j+1}} a_i^2(k) \leqq 2^{3k-2j} r_j(f^{(k)})\log n, \\ \tau \geqq 1, \qquad 1 \leqq k \leqq \log n, \\ b_1^2(k)+b_2^2(k) = 2^{3k}\log n(a_1^2(k)+a_2^2(k)) \leqq 2^{3k}\tau_0(f^{(k)})\log n. \end{gathered} \tag{17}$$

Now divide both sides by 2^{4k-3j} and sum over $j \geqq 0$. Assuming that $t(f^{(k)}) \leqq \tau$, this leads to

$$\begin{aligned} 2^{-4k}[b_1^2(k)+b_2^2(k)]^2 + \sum_{j=1}^{k} 2^{3j-4k}\sum_{i=2^j+1}^{2^{j+1}} b_i^2(k) &\leqq \log n \sum_{j=0}^{k} r_j(f^{(k)})2^{j-k} \\ &\leqq \tau \log n. \end{aligned} \tag{18}$$

Then the number of functions $f^{(k)}$ with $t(f^{(k)}) \leqq \tau$ is clearly bounded by the number of vectors $(b_1(k), \cdots, b_{2^{k+1}}(k))$ satisfying (18). This is the number of lattice points in a 2^{k+1} dimensional ellipsoid with 2^j axes of lengths $2\sqrt{\tau \log n 2^{4k-3j}}$, for each $j = 1, 2, \cdots, k$, plus two additional axes of length $2\sqrt{2^{4k}\tau \log n}$. This in turn is approximately the volume of the ellipsoid. Now a d dimensional ellipsoid with axis-lengths $l_1, \cdots, l_d$

and d even has volume

$$V=\left(\prod_{i=1}^{d}\frac{l_i}{2}\right)\pi^{d/2}\Big/(d/2)!\leqq\left(\prod_{i=1}^{d}\frac{l_i}{2}\right)(2\pi e/d)^{d/2}, \tag{19}$$

where the inequality is obtained from Stirling's formula. Substituting for the l_i's and $d = 2^{k+1}$, we find that

$$\mu_k(\tau)\approx\frac{(\tau\log n)^{2^k}(2\pi e)^{2^k}}{2^{(k+1)2^k}}\prod_{j=0}^{k}2^{1/2(4k-3j)2^j}. \tag{20}$$

It is easy to verify that an increase in the constant $2\pi e$ will make the volume approximation in (20) an upper bound. Routine algebra then shows that there is a constant γ such that $\mu_k(\tau)\leqq(\gamma\tau\log n)^{2^k}$. Finally, we can choose $s_k(f)=\gamma t(f^{(k)})+\gamma/\log n\leqq\sigma$ for all $f\in\mathscr{F}^*$ and obtain $\eta_k(\sigma)\leqq\mu_k(\sigma/\gamma)\leqq(\sigma\log n)^{2^k}$. By (10), $\sum_{k=1}^{\log n}s_k(f)\leqq 4\gamma/\pi^2+\gamma$, and we are done. □

As a trivial extension of Lemma 4, it is convenient to assume for each f that $s_k(f)$ is a positive multiple of $1/\log n$, $1\leqq k\leqq\log n$.

We turn now to the probabilistic part of the proof. Consider any function $f^{(\log n)}$ with $f\in\mathscr{F}^*$. Comparing $f^{(\log n)}$ and the partial sums f_n we have by the definition of the $a_i(k)$'s that $|a_i-a_i(\log n)|=O(1/(\sqrt{\log n}\,n^{3/2}))$ and hence

$$|f_n(x)-f^{(\log n)}(x)|\leqq\sum_{i=1}^{n}|a_i-a_i(\log n)|=O\left(\frac{1}{\sqrt{n\log n}}\right),\qquad 0\leqq x\leqq 1. \tag{21}$$

By Lemma 3 we have $|f(x)-f_n(x)|=O(1/\sqrt{n\log n})$, $0\leqq x\leqq 1$. This together with (21) yields $|f(x)-f^{(\log n)}(x)|=O(1/\sqrt{n}\log n)$, $0\leqq x\leqq 1$. We conclude from Lemma 2 applied to $f^{(\log n)}$ and f that if

$$\sup_{f\in\mathscr{F}^*}\Delta f^{(\log n)}=O(\sqrt{n}\log^{3/4}n)\text{ whp}, \tag{22}$$

then $\sup_{f\in\mathscr{F}^*}\Delta f=O(\sqrt{n}\log^{3/4}n)$ whp as well. We prove (22) below.

Consider any $f\in\mathscr{F}^*$ and write

$$\Delta f^{(\log n)}=\sum_{k=1}^{\log n}(\Delta f^{(k)}-\Delta f^{(k-1)}), \tag{23}$$

with $\Delta f^{(0)}\equiv 0$. Below, we introduce numbers $q_k=q_k(s_k(f))$, $1\leqq k\leqq\log n$, such that $\sum_{k=1}^{\log n}s_k(f)\leqq c$ implies $\sum_{k=1}^{\log n}q_k=O(\sqrt{n}\log^{3/4}n)$ for all $f\in\mathscr{F}^*$. If f is such that $\Delta f^{(\log n)}>\sum_{k=1}^{\log n}q_k$, then there exist k and σ, $1\leqq k\leqq\log n$, $1/\log n\leqq\sigma\leqq c$ (with c as given in Lemma 4), and a pair of functions $(f^{(k)}, f^{(k-1)})$ such that $s_k(f)=\sigma$ and $\Delta f^{(k)}-\Delta f^{(k-1)}>q_k(\sigma)$. Over all $f\in\mathscr{F}^*$, the number of pairs of functions $(f^{(k)}, f^{(k-1)})$ for given k, σ, and $s_k(f)=\sigma$ is at most $\eta_k(\sigma)$, so by Boole's inequality

$$\begin{aligned}\Pr\left\{\max_{f\in\mathscr{F}^*}\Delta f^{(\log n)}>\sum_{k=1}^{\log n}q_k(s_k(f))\right\}\\ \leqq c\log^2 n\max_{\substack{1\leqq k\leqq\log n\\ 1/\log n\leqq\sigma\leqq c}}\ \max_{\{f\in\mathscr{F}^*|s_k(f)=\sigma\}}\eta_k(\sigma)\Pr\{\Delta f^{(k)}-\Delta f^{(k-1)}>q_k(\sigma)\},\end{aligned} \tag{24}$$

where the $c\log^2 n$ factor comes from the $\log n$ values of k and the at most $c\log n$ values of σ (recall that the $s_k(f)$ are chosen as multiples of $1/\log n$). With $\eta_k(\sigma)$ bounded by

(12), and with $q_k(\sigma)$ as defined below, we will show that for some $c > 1$

$$(25)\qquad (2^{\sqrt{n/\log n}} \wedge (\sigma \log n)^{2^k}) \Pr\{\Delta f^{(k)} - \Delta f^{(k-1)} > q_k(\sigma)\} \leqq 1/n^c$$

for any choice of k, σ and $f \in \mathscr{F}^*$ with $s_k(f) = \sigma$. Since $c > 1$ implies that $\log^2 n/n^c = O(1/n^{c'})$ for some $c' \geqq 1$, the proof of (22) will be complete once we have verified that, in the left-hand side of (24), $\sum_{k=1}^{\log n} q_k(s_k(f)) = O(\sqrt{n} \log^{3/4} n)$ for all $f \in \mathscr{F}^*$.

Now consider the pair of functions $(f^{(k)}, f^{(k-1)})$ for an f such that $s_k(f) = \sigma$. To apply Lemma 1 to (25), let $f^{(k)}$ and $f^{(k-1)}$ be f_1 and f_2, let $\alpha = \sqrt{\sigma}/2^k$ from (11), and assume that k is such that $u_k = u_k(\sigma) \leqq \alpha n$, where for some $c > 1$

$$(26)\qquad u_k^2 \equiv \beta n \sqrt{\sigma}[[\log(\sigma \log n)] + c2^{-k} \log n],$$

with $\beta = \ln 2 = 1/\log e$. Substitute (26) into the first of the bounds in Lemma 1 and then substitute the result into (25). A little algebra shows that

$$(27)\quad (2^{\sqrt{n/\log n}} \wedge (\sigma \log n)^{2^k}) \Pr\{\Delta f(k) - \Delta f^{(k-1)} > u_k\} \leqq (\sigma \log n)^{2^k} 2^{-u_k^2 2^k/(\beta n \sqrt{\sigma})} = \frac{1}{n^c},$$

as desired. To take care of the case $u_k > \alpha n$ we put $q_k \equiv q_k(\sigma) = u_k + v$, where for some $c > 1$,

$$(28)\qquad v = \beta[\sqrt{n/\log n} + c \log n].$$

For, if $q_k > \alpha n$, then Lemma 1 and substitution into (25) give

$$(2^{\sqrt{n/\log n}} \wedge (\sigma \log n)^{2^k}) \Pr\{\Delta f^{(k)} - \Delta f^{(k-1)} > q_k\} \leqq 2^{\sqrt{n/\log n}} 2^{-v/\beta} = \frac{1}{n^c},$$

again as desired.

It remains to show that $\sum_{k=1}^{\log n} (u_k(s_k(f)) + v) = O(\sqrt{n} \log^{3/4} n)$ for all $f \in \mathscr{F}^*$. For the contribution of the $u_k \equiv u_k(s_k(f))$, use Cauchy's inequality, let $s_k \equiv s_k(f)$, and write

$$(29)\qquad u_k = O(\sqrt{n} s_k^{1/4} \sqrt{\log(s_k \log n)} + \sqrt{n} s_k^{1/4} 2^{-k/2} \sqrt{\log n}).$$

Since s_k is bounded by a constant, the contribution to $\sum u_k$ of the second term in (29) is easily seen to be $O(\sqrt{n} \log n)$. By Lemma 4 the sum of the s_k, $1 \leqq k \leqq \log n$, is at most a constant c, so the contribution of the first term is $O(w_n)$, where

$$(30)\qquad w_n = \max_{\{z_k\}} \left\{ \sqrt{n} \sum_{k=1}^{\log n} z_k^{1/4} \log(z_k \log n) \,\middle|\, z_k \geqq \frac{1}{\log n}, \sum_{k=1}^{\log n} z_k \leqq c \right\}.$$

A calculation shows that the function $w(z) = z^{1/4} \log(z \log n)$ is increasing and concave ($w''(z) \leqq 0$) for all $z \geqq 1/\log n$. Then by Jensen's inequality the maximum in (29) is achieved by putting all z_k's equal to $c/\log n$. Then

$$w_n = \sqrt{n} \sum_{k=1}^{\log n} \frac{c^{1/4} \log c}{\log^{1/4} n} = O(\sqrt{n} \log^{3/4} n),$$

and hence $\sum_{k=1}^{\log n} u_k = O(\sqrt{n} \log^{3/4} n)$. It can be seen by inspection that $v \log n = O(\sqrt{n} \log n)$, so (22) and hence the theorem is proved. □

4. Final remarks. The probabilistic parts of the proofs in § 3 and in [4], and the use of Fernique's theorem in [5] all seem to have elements in common. In particular, it appears that the desired result could be obtained by applying Fernique's theorem to inequalities similar to those in Lemma 4. The difficult part in all cases lies in proving the combinatorial properties needed for these techniques. All three proofs involve the construction of successive approximations to a lower layer function. The use of Fourier approximations enables us to take advantage of standard properties of Fourier series to simplify the proof. For example, our proof of (5) is much simpler than the proof of its analogue in [4].

It is clear from §§ 2 and 3 that for any fixed constant $c > 1$ we can write $U_n = O(\sqrt{n} \log^{3/4} n)$ with probability $1 - 1/n^c$. In fact, as shown in [4], [5], an even stronger statement is possible, namely that there exists a $c > 0$ such that $U_n = O(\sqrt{n} \log^{3/4} n)$ with probability $1 - O(n^{-c\sqrt{\log n}})$. Our methods do not preclude such a result. For this tighter bound we can easily modify the conclusion of Lemma 2 to $|\Delta f_1 - \Delta f_2| = O(\sqrt{n} \log^{3/4} n)$ with probability $1 - O(n^{-c\sqrt{\log n}})$ for some $c > 0$. With somewhat more effort, a tighter analysis of suitably larger functions q_k will then yield the desired result. The details are left to the interested reader.

REFERENCES

[1] W. HOEFFDING, *Probability inequalities for sums of bounded random variables*, J. Amer. Statist. Assoc., 58 (1963), pp. 13–30.

[2] D. JACKSON, *Fourier series and orthogonal polynomials*, in The Carus Math. Mono., Mathematical Association of America, Washington, DC, 1941.

[3] R. M. KARP, M. LUBY, AND A. MARCHETTI-SPACCAMELA, *A probabilistic analysis of multidimensional bin-packing problems*, Proc. 16th ACM Symp. Theoret. Comput., 1984, pp. 289–298.

[4] T. LEIGHTON AND P. SHOR, *Tight bounds for minimax grid matching, with applications to the average-case analysis of algorithms*, Combinatorica, 9 (1989), pp. 161–187.

[5] W. T. RHEE AND M. TALAGRAND, *Exact bounds for the stochastic upward matching problem*, Trans. Amer. Math. Soc., 307 (1988), pp. 109–125.

[6] P. W. SHOR, *The average-case analysis of some on-line algorithms for bin packing*, Combinatorica, 6 (1986), pp. 179–200.

SIAM J. DISC. MATH.
Vol. 4, No. 1, pp. 58-60, February 1991

006

THE CYCLE CONSTRUCTION*

P. FLAJOLET† AND M. SORIA†‡

Abstract. A direct generating function construction is given for cycles of combinatorial structures.

Key words. combinatorial enumerations, generating functions, combinatorial theory of words

AMS(MOS) subject classification. C05

Let $\mathscr{A}$ be a class of combinatorial structures, with $A(z)$ its corresponding *ordinary generating function*: $A(z) = \sum_{\alpha \in \mathscr{A}} z^{|\alpha|}$. We use corresponding letters for classes and generating functions. Consider the class $\mathscr{C}$ whose elements are cycles of elements of $\mathscr{A}$. The following result is classical [6], [1]:

$$C(z) = \sum_{k \geqq 1} \frac{\phi(k)}{k} \log \frac{1}{1 - A(z^k)}, \tag{0}$$

where $\phi(k)$ is the Euler totient function. This result is proved by Read [6] using Pólya's theory [5] and a classical computation of the *Zyklenzeichner* of the cyclic group. De Bruijn and Klarner [1] have another derivation, which amounts to the Lyndon factorization of free monoids [4, p. 64]. Our purpose in this note is to show that equality (0) follows directly from basic principles of combinatorial analysis [3], using elementary concepts of combinatorics on words from Lothaire [4].

PRINCIPLE 1. *Every nonempty word over $\mathscr{A}$ has a unique root that is a primitive word.*

For instance with α, $\beta \in \mathscr{A}$, word $\alpha\beta\alpha\beta\beta\alpha\beta\alpha\beta\beta\alpha\beta\alpha\beta\beta$ decomposes into $\alpha\beta\alpha\beta\beta \mid \alpha\beta\alpha\beta\beta \mid \alpha\beta\alpha\beta\beta$ and its root is the primitive (also called aperiodic) word $\alpha\beta\alpha\beta\beta$. Let $\mathscr{S} = \mathscr{A}^+$ be the set of nonempty words formed with elements of $\mathscr{A}$, and $\mathscr{P}\mathscr{S}$ the set of primitive words. From Principle 1, we have[1]

$$S(z,u) \equiv \frac{uA(z)}{1 - uA(z)} = \sum_{k \geqq 1} PS(z^k, u^k). \tag{1a}$$

From Moebius inversion applied to (1a), we get an explicit form for $PS(z, u)$:

$$PS(z,u) = \sum_{k \geqq 1} \mu(k) S(z^k, u^k) = \sum_{k \geqq 1} \mu(k) \frac{u^k A(z^k)}{1 - u^k A(z^k)}. \tag{1b}$$

PRINCIPLE 2. *Every primitive k-cycle has k distinct primitive word representations.*

A cycle is said to be primitive if and only if any associated word is primitive. We use the notation $[\cdots]$ to denote a cycle. Then, for instance, the 5-cycle $[ababb] = [babba] = \cdots = [babab]$ is primitive, while the 6-cycle $[abbabb]$ is not. We let $\mathscr{P}\mathscr{C}$ denote the class of primitive cycles. Principle 2 permits us to express the bivariate gen-

* Received by the editors July 11, 1988; accepted for publication (in revised form) October 17, 1989.

† Institut National de Recherche en Informatique et en Automatique, Domaine de Voluceau, Rocquencourt, 78153-Le Chesnay, France.

‡ LRI Université Paris-Sud 91405-Orsay, France.

[1] We introduce bivariate generating functions, and make a consistent use of variable u to mark the number of letters (called length) in a sequence (word) or a cycle: The coefficient of $[u^l z^n]$ in a generating function $F(z, u)$ of $\mathscr{F}$ represents the number of structures in $\mathscr{F}$ of total size n having length l.

erating function $PC(z, u)$ via the transformation $u^k \mapsto u^k/k$ applied to $PS(z, u)$:

$$PC(z,u)=\int_0^u PS(z,t)\frac{dt}{t}. \tag{2a}$$

Integrating with respect to t, we derive

$$PC(z,u)=\sum_{k\geqq 1}\frac{\mu(k)}{k}\log\frac{1}{1-u^kA(z^k)}. \tag{2b}$$

Principle 3. *Every cycle has a root that is a primitive cycle.*

A cycle like $[\alpha\beta\alpha\beta\beta\alpha\beta\alpha\beta\beta\alpha\beta\alpha\beta\beta]$ has a unique root defined up to cyclic order that is here $[\alpha\beta\alpha\beta\beta] \equiv [\beta\alpha\beta\beta\alpha] \equiv \cdots$. For generating functions, this entails the relation

$$C(z,u)=\sum_{k\geqq 1} PC(z^k,u^k) \quad \text{and} \quad C(z)=\sum_{k\geqq 1} PC(z^k,1). \tag{3a}$$

Using the relation $\sum_{p|k}\mu(p)/p = \phi(k)/k$ in summation (3a), we obtain

$$C(z,u)=\sum_{k\geqq 1}\frac{\phi(k)}{k}\log\frac{1}{1-u^kA(z^k)}. \tag{3b}$$

Specializing (3b) with $u = 1$ establishes Equation (0).

Thus the generating function for l-cycles, which is obtained by extracting the coefficient of $[u^l]$ in (3b), is found to be

$$\frac{1}{l}\sum_{k|l}\phi(k)A(z^k)^{l/k}.$$

Other results from [1] can also be derived from (3a). The multiset construction $\mathscr{F} = \mathscr{M}(\mathscr{G})$ ($\mathscr{F}$ is the class of all finite multisets of elements of $\mathscr{G}$) is known [5] to translate into

$$F(z)=\exp\sum_k\frac{1}{k}G(z^k).$$

Using identities $\sum_{d|n}\mu(d) = \delta_{n,1}$ and $\sum_{d|n}\phi(d) = n$, the generating functions for multisets of primitive cycles and multisets of cycles (with u again marking length) are found to be

$$\frac{1}{1-uA(z)} \quad \text{and} \quad \prod_{k\geqq 1}\frac{1}{1-u^kA(z^k)}.$$

By considering singularities of corresponding generating functions [5], it is easy to derive asymptotic results. Assume for instance that the radius of convergence ρ of $A(z)$ satisfies $\rho < 1$ and that $A(\rho) = +\infty$. Then, we have the following:

- The number of $\mathscr{A}$-cycles of size n and length l is asymptotically $1/l$ times the number of $\mathscr{A}$-sequences having size n and length l.
- The number of $\mathscr{A}$-cycles of size n is asymptotically $1/n$ times the number of $\mathscr{A}$-sequences of size n.
- The length of a random $\mathscr{A}$-cycle of size n is asymptotically Gaussian with mean and variance that are $O(n)$. (See [2] for similar results).

These results can be extended to the case when $\rho = 1$ and $A(z)$ has only a pole at $z = 1$ on its circle of convergence.

Note added in proof. Related results appear in [7].

REFERENCES

[1] N. G. De Bruijn and D. A. Klarner, *Multisets of aperiodic cycles*, SIAM J. Algebraic Discrete Methods, 3 (1982), pp. 359–368.

[2] P. Flajolet and M. Soria, *Gaussian limiting distributions for the number of components in combinatorial structures*, J. Combin. Theory Ser. A, 53 (1990), pp. 165–182.

[3] I. Goulden and D. Jackson, *Combinatorial Enumerations*, John Wiley, New York, 1983.

[4] M. Lothaire, *Combinatorics on words*, in Encyclopedia of Mathematics and Its Applications, vol. 17, Academic Press, New York, 1983.

[5] G. Pólya, *Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemische Verbindungen*, Acta Math. 68 (1937), pp. 145–254. Translated in: G. Pólya and R. C. Read, *Combinatorial Enumeration of Groups, Graphs and Chemical Compounds*, Springer-Verlag, New York, 1987.

[6] R. C. Read, *A note on the number of functional digraphs*, Math. Ann., 143 (1961), pp. 109–110.

[7] M. Soria, *Méthodes d'analyse pour les constructions combinatoires et les algorithmes*, Doctorate in Sciences, Université de Paris–Sud, July, 1990.

ANALYSIS OF A COMPOUND BIN PACKING ALGORITHM*

DONALD K. FRIESEN† AND MICHAEL A. LANGSTON‡

Abstract. Consider the classic bin packing problem, in which we seek to pack a list of items into the minimum number of unit-capacity bins. The worst-case performance of a *compound* bin packing algorithm that selects the better packing produced by two previously analyzed heuristics, namely, FFD (first fit decreasing) and B2F (best two fit) is investigated. FFD and B2F can asymptotically require as many as $\frac{11}{9}$ and $\frac{5}{4}$ times the optimal number of bins, respectively. A new technique, *weighting function averaging*, is introduced to prove that our compound algorithm is superior to the individual heuristics on which it is based, never using more than $\frac{6}{5}$ times the optimal number of bins.

Key words. bin packing, compound algorithms, heuristics, weighting functions, worst-case analysis

AMS(MOS) subject classifications. 68Q20, 68Q25

1. Introduction. In the usual definition of the bin packing problem, we seek to pack the items of a list $L = \{l_1, l_2, \cdots, l_N\}$, each item with size in the range $(0,1]$, into the minimum number of unit-capacity bins. It is easily verified that this problem is NP-hard. Therefore, we focus our efforts on practical, efficient approximation algorithms in hopes of guaranteeing near-optimal results. (Note that there are algorithms guaranteed to produce results as close to the optimum as desired [1], [7]. Unfortunately, these algorithms are not practical to implement because the time required to ensure results at most $(1 + \varepsilon)$ times the optimum grows extremely rapidly as ε approaches zero.)

We use worst-case analysis as a measure of the worth of a bin packing heuristic. The heuristic may not discover the best packing, but we endeavor to show that it always provides results close to the optimum. For some algorithm, ALG, let ALG (L) represent the number of nonempty bins required by ALG to pack L. For instance, OPT (L) denotes the number of bins required in an optimal packing of L. We restrict our attention to two off-line[1] algorithms: FFD (first fit decreasing) and B2F (best two fit). Given any list L, it is known from [6] that FFD (L) does not exceed $(\frac{11}{9})$ OPT $(L) + 4$, and from the Appendix to this paper that B2F (L) does not exceed $(\frac{5}{4})$ OPT $(L) + 4$. Moreover, examples exist that demonstrate that these bounds are asymptotically tight.

It seems reasonable to suggest that these two heuristics produce particularly inferior packings for rather small, distinct regions of the input space. Based on this conjecture, we analyze a *compound* algorithm, CFB, in which both FFD and B2F are applied and the better packing selected. This notion of combining two or more heuristics is an attractive one, but the analysis of such an algorithm can be especially difficult; only a few compound algorithms have been successfully analyzed in the literature (see, for example, [2], [8], [9]). We note that a tight worst-case bound of 71/60 has recently been reported for a modification of the FFD algorithm [5], thereby yielding the lowest bound yet published for an efficient bin packing heuristic. This bound is superior to the upper bound of $\frac{6}{5}$ that we prove here, but is inferior to the lower bound of 227/195 provided by the worst

* Received by the editors June 19, 1989; accepted for publication (in revised form) May 8, 1990. A preliminary version of a portion of this paper was presented at the twentieth Allerton Conference on Communication, Control, and Computing held in Monticello, Illinois in October, 1982.

† Department of Computer Science, Texas A&M University, College Station, Texas 77843.

‡ Department of Computer Science, University of Tennessee, Knoxville, Tennessee 37996-1301 and Department of Computer Science, Washington State University, Pullman, Washington 99164-1210. This research was supported in part by National Science Foundation grants MIP-8603879 and MIP-8919312, and by Office of Naval Research contract N00014-88-K-0343.

[1] An off-line algorithm is free to preview and rearrange items before it begins to pack them.

examples we know of for CFB. Moreover, the novel analysis we devise for our compound algorithm merits attention and may, we hope, be applicable in other settings.

We shall employ the technique of "weighting" L so that the FFD and B2F packings can be compared to an optimal packing. Although we would like to determine the minimum of $\{\text{FFD}(L), \text{B2F}(L)\}$, the analysis involved is extremely complicated. Instead, we investigate the average of $\{\text{FFD}(L), \text{B2F}(L)\}$, in an effort to obtain a weak upper bound on the minimum. In particular we show that, after eliminating certain cases where we can guarantee that one or the other algorithm performs within our bound of $\frac{6}{5}$, our weighting of L ensures that the average and hence the minimum number of bins used by the two algorithms is within the bound.

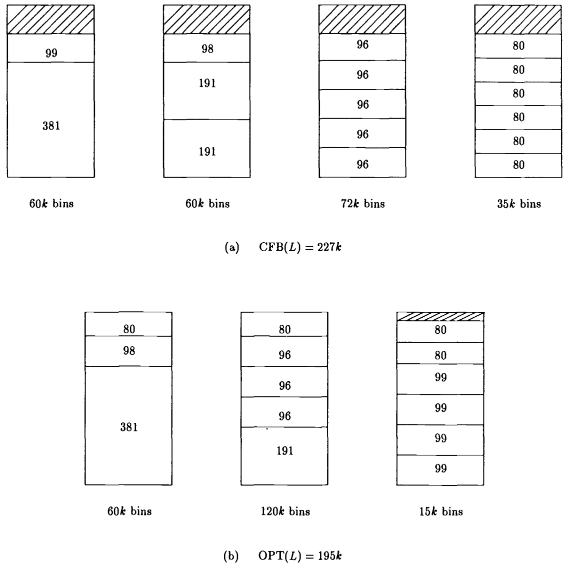
In the next section, we present some preliminary analysis and demonstrate that CFB (L) can be as great as $(227/195)$ OPT (L). We also introduce a typing scheme for the items of L based on size. In § 3, we establish the specific conditions required for the FFD packing to use more than $\frac{6}{5}$ the optimal number of bins. Section 4 contains an analogous determination for B2F. We present our main result in § 5, proving that CFB (L) does not exceed $(\frac{6}{5})$ OPT $(L) + 8$. The final section contains remarks about proving a tighter performance bound for CFB. In the Appendix, we discuss in further detail the B2F algorithm and derive its asymptotic worst-case bound.

2. Preliminary discussion. We begin by describing the FFD and B2F heuristics more precisely. The FFD algorithm can be implemented by first sorting all items so that their sizes are arranged in nonincreasing order. Each bin is packed by repeatedly placing in it the largest unpacked item that fits. When no more items are available that fit, the next bin is packed. The B2F algorithm modifies this in the following way. First a bin is packed as by the FFD rule. If the bin contains more than a single item, then the list is checked to see if the smallest item in the bin could be replaced by two items that would pack the bin more nearly full. If so, those two whose sum is largest are used in place of the smallest item in the bin. A number of other schemes could be used to decide which two replace the smallest item, but almost any choice will satisfy our analysis, subject to the following modification made to simplify the proof: items of sizes less than or equal to $\frac{1}{6}$ will be held back until all larger items are packed. An FFD-like procedure is used to complete the packing when only items of size no greater than $\frac{1}{6}$ are left. The purpose of this modification is to reduce the number of combinations to consider in proving an asymptotic $\frac{6}{5}$ bound, although it seems likely that this modification actually detracts somewhat from the performance of the compound algorithm.

Figure 1 depicts the worst example (independent, of course, of an additive constant) that we were able to contrive for the CFB algorithm. For simplicity, the bin size has been expanded to 559. All of the examples we devised that were even close to being this poor were dependent on the small items being held back, so that the FFD and B2F packings are the same.

We denote the size of an item $l_i \in L$ by $s(l_i)$. Thus, after sorting, $s(l_1) \geqq s(l_2) \geqq \cdots \geqq s(l_N)$. We use *last* to denote the index of the last item packed by FFD. Note that l_{last} may not be the smallest item in L, since smaller items may have been packed earlier where l_{last} did not fit.

To prove that $\frac{6}{5}$ is an asymptotic upper bound on the worst-case behavior of CFB, we now proceed by contradiction and henceforth assume that L denotes a counterexample. That is, we assume that both FFD (L) and B2F (L) exceed $(\frac{6}{5})$ OPT $(L) + 8$. Without loss of generality, we also assume that L is minimal. By this we mean that no counterexample exists with which OPT can use fewer bins, and that no counterexample is possible with fewer items for this minimal number of bins. (Of course, minimality for CFB does not imply minimality for either FFD or B2F alone.)



(a) CFB(L) = 227k

(b) OPT(L) = 195k

FIG. 1. *Example for which* CFB $(L) = (227/195)$ OPT (L), *using bin size* 559.

An immediate consequence of this is that L contains no item whose size is less than or equal to $\frac{1}{6}$. If it did, then minimality requires that one or more such items must be packed in the last bin by either the FFD or the B2F algorithm, in which case all preceding bins would be packed to a level of at least $\frac{5}{6}$. A simple "conservation of size" argument ensures that, for such a list, no packing could use fewer than $(\frac{5}{6})(\text{CFB}(L) - 1)$ bins.

With this in mind, we let $s(l_{\text{last}}) = \frac{1}{6} + \Delta$, for some $\Delta > 0$. Since no item has size less than or equal to $\frac{1}{6}$, we know that no bin in any packing of L has more than five items.

We use the notation B^* for an arbitrary bin of the optimal packing, and $|B^*|$ to denote the number of items B^* contains. For the bins of the FFD or B2F packing, we use $B_1, B_2, \cdots$ as the sequence of bins in the order in which they are packed.

LEMMA 2.1. *L contains no item l_i with $s(l_i) \geqq \frac{2}{3}$.*

Proof. To obtain the proof, assume otherwise. In both the FFD and B2F packings, the largest item l_1 is packed in B_1 with at most one other item, the largest that would fit. The optimal bin containing l_1 can contain at most one additional item and in fact can be packed no better than B_1. If the item or items of B_1 are removed from L, then all three of FFD (L), B2F (L), and OPT (L) can be reduced by one, contradicting the presumed minimality of L with respect to CFB. □

There can be no bin containing only one item in the FFD packing (except, possibly, for the last bin). If there were, $s(l_{\text{last}})$ must exceed $\frac{1}{3}$, since otherwise l_{last} would have fit, and it is known that FFD (L) is bounded by $(\frac{7}{6})$ OPT $(L) + 2$ whenever $s(l_{\text{last}})$ exceeds $\frac{1}{4}$. (See [6, Thm. 4.10].) From this it also follows that Δ must be less than or equal to $\frac{1}{12}$.

Each item of L is assigned a type as shown in Table 1. Although this typing scheme is motivated by the structure of a typical packing produced by the FFD rule (more will be said on this in the next section), we classify items exclusively by their size so that we can compare both FFD (L) and B2F (L) to OPT (L). Note that Δ cannot exceed $\frac{1}{30}$ if Y_4 or X_5 items exist.

3. A close look at FFD. We say that an item is "regular" if there is no larger item available when it is packed. A "fallback" item is one that is packed when one or more

TABLE 1
Item types based on size.

Type	Min size	Max size
Y_1	$>\frac{1}{2}$	$<\frac{2}{3}$
X_2	$>\frac{5}{12}-\Delta/2$	$\leqq\frac{1}{2}$
Y_2	$>\frac{1}{3}$	$\leqq\frac{5}{12}-\Delta/2$
X_3	$>\frac{5}{18}-\Delta/3$	$\leqq\frac{1}{3}$
Y_3	$>\frac{1}{4}$	$\leqq\frac{5}{18}-\Delta/3$
X_4	$>\frac{5}{24}-\Delta/4$	$\leqq\frac{1}{4}$
Y_4	$>\frac{1}{5}$	$\leqq\frac{5}{24}-\Delta/4$
X_5	$>\frac{1}{6}$	$\leqq\frac{1}{5}$

larger items are available. Thus the notation we have used in Table 1 roughly agrees with the way items are packed by FFD. That is, regular items of type X_i are generally packed by FFD in a bin consisting of the i largest items available when the bin is packed. We call such a bin an X_i bin. Regular items of type Y_i are generally packed with $i-1$ other Y_i items and a (smaller) fallback item. We call such a bin a Y_i bin. (Note that no Y_i bin, $i \geqq 2$, can have more than one fallback item, as the following argument shows. If two fallback items are used, then they combine to fill more than $\frac{1}{3}$ of the bin. In this event, however, the two or more regular items fill less than $\frac{2}{3}$ of the bin, and the smaller regular item has a size less than $\frac{1}{3}$, implying that another regular item would have fit in the bin as well.)

This motivates the range of sizes we have selected for each item type. For example, the sum of the sizes of the two items in an X_2 bin must exceed $1-(\frac{1}{6}+\Delta)$, or else l_{last} would have been used as a fallback item in that bin. Hence, with the exception of items from the first or last X_2 bin, every regular X_2 item must have a size in the range $(\frac{5}{12}-\Delta/2, \frac{1}{2}]$. Similar size restrictions are used to define the other item types as summarized in Table 1. We use these same size ranges to assign a type to each fallback item.

There may also be some bins, which we define as "exceptional" for the FFD packing, that are not packed by FFD with items of the expected sizes. These can only be the first or last bins of a particular type, subject to the following constraints. If the last bin of type Y_i is exceptional (that is, it does not contain i items of type Y_i), then the next bin is an X_{i+1} bin that is not exceptional if there are at least two X_{i+1} bins. Similarly, if the last bin of type X_i is exceptional, then the first bin of type Y_i is not exceptional unless it is also the last Y_i bin.

Consequently, there are at most eight exceptional bins in the FFD packing, including the last bin packed (which contains l_{last}). We define an exceptional item to be one packed in an exceptional bin or one smaller than l_{last}.

We now seek to determine the precise conditions necessary for FFD (L) to exceed $(\frac{6}{5})\,\text{OPT}\,(L)+8$. In this effort, we employ a weighting function w_F: $L \rightarrow \mathbf{R}^+$. We extend w to subsets of L in the obvious fashion. For example, $w_F(B_j)$ denotes $\sum_{l_i \in B_j} w_F(l_i)$. Our intent is to assign each item as small a weight as possible and yet ensure that the weight of any nonexceptional FFD packed bin is at least 1. Table 2 describes our definition of w_F for nonexceptional items.

Recall that fallback items, like regular items, are assigned a type based on their size. We deviate slightly from this definition of w_F for items packed in Y_1 bins. Consider any two Y_1 items a and b, where a precedes b. Since $s(a) \geqq s(b)$, we increase $w(a)$, if necessary, to ensure that $w(a) \geqq w(b)$ and reduce the weight of any item(s) packed with a accordingly. For future reference, we state this formally as follows:

TABLE 2
Weighting function w_F based on FFD *packing.*

Type of nonexceptional items in an FFD-packed bin	Weights assigned
Y_1, any two items	$\frac{3}{5}, \frac{1}{5}, \frac{1}{5}$
Y_1, X_2	$\frac{3}{5}, \frac{2}{5}$
Y_1, Y_2	$\frac{3}{5}, \frac{2}{5}$ if $\exists Y_2$ bin(s), else $\frac{2}{3}, \frac{1}{3}$ if $s(Y_1) \leqq \frac{2}{3} - 2\Delta$, else $\frac{11}{15}, \frac{4}{15}$
Y_1, X_3	$\frac{2}{3}, \frac{1}{3}$
Y_1, Y_3	$\frac{11}{15}, \frac{4}{15}$
Y_1, X_4	$\frac{3}{4}, \frac{1}{4}$
Y_1, Y_4 or smaller item	$\frac{4}{5}, \frac{1}{5}$
X_2, X_2	$\frac{1}{2}, \frac{1}{2}$
Y_2, Y_2, X_3	$\frac{11}{30}, \frac{11}{30}, \frac{4}{15}$
Y_2, Y_2, Y_3 or smaller item	$\frac{2}{5}, \frac{2}{5}, \frac{1}{5}$
X_3, X_3, X_3	$\frac{1}{3}, \frac{1}{3}, \frac{1}{3}$
Y_3, Y_3, Y_3, any item	$\frac{4}{15}, \frac{4}{15}, \frac{4}{15}, \frac{1}{5}$
X_4, X_4, X_4, X_4	$\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}$
any five items	$\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}$

Y_1 weighting rule: If a and b are Y_1 items, and a is packed in a bin before the bin containing b, then $w_F(a) \geqq w_F(b)$.

An exceptional item receives a weight of zero, completing our definition of w_F. For the convenience of the reader, Table 3 provides a listing of the possible weights for each nonexceptional item type.

LEMMA 3.1. *The* FFD *weight of an optimal bin cannot exceed $\frac{6}{5}$ unless the bin contains a Y_1 item or a Y_2 item whose* FFD *weight exceeds $\frac{1}{3}$.*

Proof. Suppose that B^* is a bin of the optimal packing that has weight greater than $\frac{6}{5}$ and B^* contains neither of the items mentioned in the statement of the lemma. Clearly B^* must contain at least 3 items.

Case 1. Suppose $|B^*| = 3$. Then at least one item must have weight greater than $\frac{1}{3}$ and, from the assumptions of the lemma, it can only have type X_2. There cannot be two such items, or else no item larger than l_{last} could fit with them. Thus $w_F(B^*) \leqq \frac{1}{2} + \frac{1}{3} + \frac{1}{3} = \frac{7}{6}$.

TABLE 3
Possible FFD *weights for each nonexceptional item type.*

Type	Weight
Y_1	$\frac{4}{5}, \frac{3}{4}, \frac{11}{15}, \frac{2}{3}, \frac{3}{5}$
X_2	$\frac{1}{2}, \frac{2}{5}$
Y_2	$\frac{2}{5}, \frac{11}{30}, \frac{1}{3}, \frac{4}{15}$
X_3	$\frac{1}{3}, \frac{4}{15}, \frac{1}{5}$
Y_3	$\frac{4}{15}, \frac{1}{5}$
X_4	$\frac{1}{4}, \frac{1}{5}$
X_5 or Y_4	$\frac{1}{5}$

Case 2. Suppose $|B^*| = 4$. If B^* does not contain an X_2 item, then the smallest item packed must have weight exceeding $\frac{1}{5}$, else $w_F(B^*) \leqq 3(\frac{1}{3}) + \frac{1}{5} = \frac{6}{5}$. No item larger than l_{last} can be packed with three items of type X_3, and there cannot be four items greater than $\frac{1}{4}$ in size. Thus there must be one item of weight at most $\frac{1}{4}$ and one other item of type Y_3 or smaller, and $w_F(B^*)$ is at most $\frac{1}{3} + \frac{1}{3} + \frac{4}{15} + \frac{1}{4} < \frac{6}{5}$. Consequently, B^* must contain an X_2 item. The second largest item of B^* must be of type Y_3 or X_4, implying that the remaining two items either (1) are each of type Y_4 or less or (2) contain an item smaller than l_{last}. In either case, $w_F(B^*) < \frac{6}{5}$.

Case 3. Suppose $|B^*| = 5$. B^* must contain an X_3 or Y_3 item since it can contain neither a Y_2 item nor four X_4 items and any item as large as l_{last}. Therefore, the second largest item of B^* must be of type X_4, implying that the remaining three items either (1) are each of type Y_4 or less or (2) contain an item smaller than l_{last}. In either case, $w_F(B^*) < \frac{6}{5}$. □

LEMMA 3.2. *The* FFD *packing of L contains no Y_2 bin.*

Proof. Suppose there is a Y_2 bin. Consider the sorted sublist L' obtained from L by deleting every item that is smaller than $\frac{1}{6} + \Delta$, every item that is larger than $\frac{2}{3} - 2\Delta$, and every item that is placed in a bin with an item larger than $\frac{2}{3} - 2\Delta$ in the FFD packing of L. Clearly, the FFD packing of L' must also have a Y_2 bin. Moreover, since FFD $(L) > (\frac{6}{5})$ OPT $(L) + 8$, it follows that FFD $(L') > (\frac{6}{5})$ OPT $(L') + 8$. (Deleting items smaller than $\frac{1}{6} + \Delta$ does not affect the number of bins used by FFD and cannot increase the number required by OPT. After that, as long as the first item of the list is larger than $\frac{2}{3} - 2\Delta$, it and any other item FFD packs in B_1 can be deleted, reducing the number of bins used by FFD by one and the number needed by OPT by at least one.) Thus, from these observations and the last lemma, it suffices to restrict our attention to L' and an optimal bin B^* that contains z, a Y_1 item or a Y_2 item whose FFD weight exceeds $\frac{1}{3}$, and show that, due to the presence of a Y_2 bin, $w_F(B^*) \leqq \frac{6}{5}$. We assume $w_F(B^*) > \frac{6}{5}$ and consider the possible cases.

Case 1. Suppose z is a Y_1 item.

Suppose $w_F(z) > \frac{3}{5}$. Then the smaller Y_2 item in the Y_2 bin did not fit with z in the FFD packing. Hence $s(z) > 1 - (\frac{5}{12} - \Delta/2) = \frac{7}{12} + \Delta/2$. If $|B^*| = 2$, then the second item can have weight at most $\frac{1}{3}$ and since the weight of z is at most $\frac{4}{5}$, $w_F(B^*) < \frac{6}{5}$. Since $|B^*|$ must be less than 4, we must have $|B^*| = 3$. If the second largest item were at least $\frac{1}{4}$ in size, no third item would fit. If both items are of type Y_4 or X_5, then $w_F(B^*) \leqq \frac{4}{5} + 2(\frac{1}{5}) = \frac{6}{5}$. Thus there must be an X_4 item in B^* and, moreover, it must have weight $\frac{1}{4}$. If FFD packs this X_4 item in a bin with subscript less than that of the bin containing z, then the Y_1 weighting rule implies that its weight is at most $1 - w_F(z)$ and we would get $w_F(B^*) \leqq \frac{6}{5}$. But this X_4 item would fit with z, so the item packed with z by the FFD algorithm is at least as large as an X_4 item. Thus $w_F(z) \leqq \frac{3}{4}$ and $w_F(B^*) \leqq \frac{6}{5}$. (Note that the Y_1 weighting rule cannot cause z to have a weight exceeding $\frac{3}{4}$ unless every X_4 item has a weight less than $\frac{1}{4}$.)

Now suppose that $w_F(z) = \frac{3}{5}$. Then certainly $|B^*| = 3$. No item of size greater than $\frac{1}{3}$ can then be used. If either of the other items had weight less than $\frac{1}{3}$, then $w_F(B^*) \leqq \frac{3}{5} + \frac{1}{3} + \frac{4}{15} = \frac{6}{5}$. However, the only items of weight $\frac{1}{3}$ have size greater than $\frac{1}{4}$, and no two items of size greater than $\frac{1}{4}$ could fit with a Y_1 item. We conclude that z cannot be a Y_1 item.

Case 2. Suppose z is a Y_2 item.

Clearly $|B^*| = 3$ or 4. Suppose $|B^*| = 3$. The only possible problem occurs if B^* contains an X_2 item, a, and an X_3 item, b. In this event, $\Delta > \frac{1}{30}$, or else $s(B^*) > 1$. But then $s(z) + s(b) > \frac{1}{3} + \frac{5}{18} - \Delta/3 > \frac{2}{3} - 2\Delta$, the maximum size for a Y_1 item. Thus a would fit with any Y_1 item. Since it must be the case that $w_F(a) = \frac{1}{2}$, all fallback items in Y_1 bins must be of type X_2. Therefore z is packed by FFD into some Y_2 bin, B_i. Certainly b would fit as the fallback item in B_i, and we conclude that either $w_F(b) = \frac{4}{15}$ or $w_F(z) = 11/30$. In either case, $w_F(B^*) \leqq \frac{6}{5}$.

Suppose $|B^*| = 4$. The second largest item of B^* can only be of type X_3, the third only of type X_4. Thus $w(B^*) < \frac{6}{5}$ unless the smallest item is an X_4 item as well. But this is impossible, since $s(Y_2) + s(X_3) + 2s(X_4) \leqq 1$ implies $\Delta > \frac{1}{30}$ and $s(Y_2) + s(X_3) + 2s$ (any item) $\leqq 1$ implies $\Delta < \frac{1}{30}$. We conclude that z cannot be a Y_2 item.

By definition, $w_F(L') \geqq \text{FFD}(L') - 8$. Lemma 3.1 and the analysis just completed demonstrate that $w_F(L') \leqq (\frac{6}{5})\,\text{OPT}(L')$. Hence we derive $\text{FFD}(L') \leqq (\frac{6}{5})\,\text{OPT}(L') + 8$, contradicting the presumed existence of a Y_2 bin. □.

We state here some important consequences that follow from our analysis of the FFD packing.

COROLLARY 3.1. *If x is a Y_2 item, then $w_F(x) \leqq \frac{1}{3}$. If B^* is any optimal bin not containing a Y_1 item, then $w_F(B^*) \leqq \frac{6}{5}$.*

LEMMA 3.3. *If B^* is any bin of the optimal packing containing an item of size less than $\frac{1}{6} + \Delta$, then $w_F(B^*) \leqq 1$.*

Proof. Suppose B^* contains such an item, a. Then certainly $|B^*|$ must be at least 3, since a is exceptional and therefore $w_F(a) = 0$.

Case 1. Suppose $|B^*| = 3$. Then there must be a Y_1 item, b. The remaining item, c, would fit when b was packed. If it is unavailable, then its weight is at most $1 - w_F(b)$ by the Y_1 weighting rule. If it is available, then the item used in place of c must be at least as large. Since $s(c) < \frac{1}{3}$, there is no way for c to receive more weight than the item packed with b by FFD (see Tables 1, 2, and 3).

Case 2. Suppose $|B^*| = 4$. There must be an item of weight exceeding $\frac{1}{3}$ that, by Lemma 3.2, cannot be of type Y_2. Thus it must be an X_2 item. If each of the remaining items have weight at most $\frac{1}{4}$, then the lemma holds for B^*, so there must be a Y_3 or X_3 item. If both items have size at least $\frac{1}{6} + \Delta$, then $s(B^*) > \frac{5}{12} - \Delta/2 + \frac{1}{4} + \frac{1}{6} + \frac{1}{6} + \Delta > 1$. On the other hand, if there is a second item whose size is less than $\frac{1}{6} + \Delta$, then certainly $w_F(B^*) \leqq 1$.

Case 3. Suppose $|B^*| = 5$. There must be an item of weight exceeding $\frac{1}{4}$, or else $w_F(B^*) \leqq 4(\frac{1}{4})$. It cannot be larger than $\frac{1}{3}$ in size, so it must be of type X_3 or Y_3. There cannot be two items exceeding $\frac{1}{4}$ in size, or else $s(B^*) > 1$. The remaining three items must all have size at least $\frac{1}{6} + \Delta$. If two are less than $\frac{1}{5}$ in size, however, $w_F(B^*) \leqq \frac{1}{3} + \frac{1}{4} + 2(\frac{1}{5}) < 1$. If two are to receive weight $\frac{1}{4}$, however, $s(B^*) > \frac{1}{4} + \frac{5}{12} - \Delta/2 + \frac{1}{6} + \frac{1}{6} + \Delta > 1$.

Thus, in any case, we conclude that $w_F(B^*)$ is at most 1 if B^* contains an item smaller than l_{last}. □

4. A close look at B2F. We now seek to determine the precise conditions necessary for B2F (L) to exceed $(\frac{6}{5})$ OPT $(L) + 8$. In defining the weighting function w_B for the B2F packing, we shall retain the type classification described in § 2. That is, items are still classified strictly according to size as listed in Table 1. Most of our definition for w_B is straightforward and is given in Table 4.

The definition of w_B for items in Y_1 bins is more complicated and is described in the following paragraphs.

We wish to maintain the fact that the sum of the weights of the items in any nonexceptional bin is 1. Thus in any Y_1 bin with only one item, that item has weight 1. (Unlike the FFD packing, such a one-item bin may exist in the B2F packing.) We would also like to keep smaller Y_1 items from having greater weight than larger ones, and we would like the fallback items to have their weight assigned according to their type. The difficulty

TABLE 4
Weighting function w_B for bins not containing an item of size exceeding $\frac{1}{2}$ in B2F *packing.*

Type of nonexceptional items in a B2F-packed bin	Weights assigned
X_2 or Y_2, X_2 or Y_2	$\frac{1}{2}, \frac{1}{2}$
X_2 or Y_2, X_2 or Y_2, any item	$\frac{2}{5}, \frac{2}{5}, \frac{1}{5}$
X_2 or Y_2, X_3 or Y_3, X_3 or Y_3	$\frac{2}{5}, \frac{3}{10}, \frac{3}{10}$
X_2 or Y_2, X_3 or Y_3, X_4 or smaller item	$\frac{1}{2}, \frac{3}{10}, \frac{1}{5}$
X_2 or Y_2, X_4 or Y_4, X_4 or Y_4	$\frac{1}{2}, \frac{1}{4}, \frac{1}{4}$
X_3 or Y_3, X_3 or Y_3, X_3 or Y_3	$\frac{1}{3}, \frac{1}{3}, \frac{1}{3}$
X_3 orY_3, X_3 or Y_3, X_3 or Y_3, X_4 or smaller item	$\frac{4}{15}, \frac{4}{15}, \frac{4}{15}, \frac{1}{5}$
X_3 or Y_3, X_3 or Y_3, X_4 or smaller item, X_4 or smaller item	$\frac{3}{10}, \frac{3}{10}, \frac{1}{5}, \frac{1}{5}$
X_4 or Y_4, X_4 or Y_4, X_4 or Y_4, X_4 or Y_4	$\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}$
any five items	$\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}$

comes with small items (those with size less than or equal to $\frac{1}{3}$), in which case w_B depends on the last such item packed in a Y_1 bin.

Specifically, let h be the index of the last bin in the B2F packing containing a Y_1 item, no X_2 or Y_2 item, and at most one fallback item. All subsequent Y_1 bins contain either two fallback items or one fallback item of type Y_2 or X_2. In either case, the Y_1 item is given weight $\frac{3}{5}$. If there is one fallback item, it is given weight $\frac{2}{5}$; if there are two, each is given weight $\frac{1}{5}$.

If $|B_h| = 1$, then a B_h's Y_1 item and all earlier Y_1 items are assigned weight 1, and all earlier fallback items are assigned weight zero.

If $B_h = \{y, x\}$, where y is of type Y_1 and $s(x) \leqq \frac{1}{3}$, then we determine the weight of x by examining all items of size less than or equal to $s(x)$ that are packed after the last Y_1 item. That is, we set $w_B(x) = \max\{w_B(t) \mid s(t) \leqq s(x), t \text{ not packed in a } Y_1 \text{ bin}\}$. Of all items that are available when x is packed that would fit (no larger item would fit), and that are not packed in Y_1 bins, we choose the one that has maximum weight (using Table 4). If there are no such items, then we set $w_B(x) =$ zero.

Once B_h and $w_B(x)$ have been determined, the rest of w_B is defined as follows. The Y_1 item y in B_h is given weight $w_B(y) = 1 - w_B(x)$. Since $s(x) \leqq \frac{1}{3}$ and the maximum size of any Y_1 item is $\frac{2}{3}$, x must have fit in any preceding bin. Thus each such bin contains either two fallback items, or one fallback item at least as large as x. All Y_1 items preceding B_h are assigned weight $w_B(y)$. If there are two fallback items, each is assigned weight $w_B(x)/2$; if there is only one, it is assigned weight $w_B(x)$.

If B_h does not exist, then $h = 0$ and all Y_1 items are assigned weight $\frac{3}{5}$ with their associated fallback items given weight $\frac{2}{5}$, or $\frac{1}{5}$ each if there are two of them.

The example depicted in Fig. 2 illustrates the role of B_h in determining w_B. Types of items packed in each bin are given on the inside, w_B is listed on the outside. In this example, $h = 4$, and one of the X_4 items in B_i is no larger than the X_4 item in B_4.

DEFINITION. The following bins are exceptional for the B2F packing: the last bin to contain an item of each of the types X_2, Y_2, X_3, Y_3, X_4, Y_4, the last bin containing exactly three X_3 or Y_3 items, and the last bin of the packing.

In general, therefore, the last bin containing an item of a particular type is exceptional, although Y_1 and X_5 items are excluded from this. Note that if an X_2 item is packed with two Y_4 items, there can be no X_2 items left (since any X_2 item is larger than any two Y_4

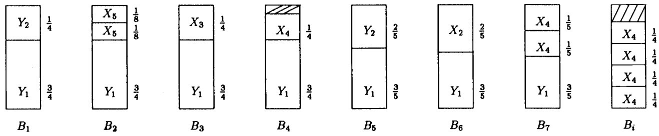


FIG. 2. *The role of B_h in determining w_B. Here, $h = 4$.*

items) and the bin is exceptional. If an X_2 item is packed with an X_4 item and a Y_4 item, there can be no X_4 items left and the bin is exceptional. Similarly, if an X_2 or Y_2 item is packed with an X_4 or Y_4 item and an X_5 item, the bin is exceptional since there can be no more X_4 or Y_4 items available. A bin whose largest item is of type X_3 (Y_3) is configured as described in Table 4 unless there are no more X_3 (Y_3) items available. Also, a bin whose largest item is of type X_4 (Y_4) is configured as described in Table 4 unless there are no more X_4 (Y_4) items left. Finally, the last bin containing three X_3 or Y_3 items and nothing else is classified as exceptional. Although this bin might not otherwise qualify as exceptional, it could cause problems in our proof if its items were each to receive weight $\frac{1}{3}$.

We conclude that there are at most eight exceptional bins in the B2F packing. We define an exceptional item simply as one packed in an exceptional bin. Such an item receives a weight of zero, completing our definition of w_B. For the convenience of the reader, Table 5 provides a listing of the possible weights for each nonexceptional item type.

Before proceeding with the principle results of this section, we first prove some preliminary lemmas that reveal details of the B2F packing. The first of these concerns is the occurrence of items of weight $\frac{1}{3}$, the second the impossibility of a certain configuration containing Y_3 and Y_4 items.

LEMMA 4.1. *If there is an item, x, of* B2F *weight $\frac{1}{3}$, then there must be a bin in the* B2F *packing containing exactly three items, each of which has size no larger than $s(x)$.*

Proof. The only possible types for x are X_3 and Y_3, and the only possible bins for x to be packed in are a three-item bin or one with an item of type Y_1, packed in a bin B_i, where $i \leqq h$. Suppose x is packed with a Y_1 item. From the definition of w_B, it is clear that the fallback item in B_h also has weight $\frac{1}{3}$ and is no larger than x. Without loss of generality, we can assume that x is the fallback item in B_h. If x has weight $\frac{1}{3}$, however, then there must be another item packed after the Y_1 bins that is no larger than x and

TABLE 5
Possible B2F *weights for nonexceptional items in a bin B_i, where i exceeds h.*

Type	Weight
Y_1	$\frac{3}{5}$
X_2	$\frac{1}{2}, \frac{2}{5}$
Y_2	$\frac{1}{2}, \frac{2}{5}$
X_3	$\frac{1}{3}, \frac{3}{10}, \frac{4}{15}, \frac{1}{5}$
Y_3	$\frac{1}{3}, \frac{3}{10}, \frac{4}{15}, \frac{1}{5}$
X_4	$\frac{1}{4}, \frac{1}{5}$
Y_4	$\frac{1}{4}, \frac{1}{5}$
X_5	$\frac{1}{5}$

has weight $\frac{1}{3}$. From Table 4 we know that this item must be packed in a bin containing exactly three items each of weight $\frac{1}{3}$. Moreover, we know from Table 1 that any one of these three items would have been used in place of x if it were larger. Thus we may assume that the lemma holds unless x is an X_3 or Y_3 item packed in a three-item bin. However, the last such bin will contain the three smallest such items (and hence is exceptional). Thus the last three-item bin satisfies the conditions of the lemma. □

LEMMA 4.2. *If there is a Y_4 item of* B2F *weight $\frac{1}{4}$, then there is no Y_3 item of* B2F *weight $\frac{1}{3}$.*

Proof. Suppose there are such Y_3 and Y_4 items. In order to have a Y_3 item of weight $\frac{1}{3}$, there must be a B2F bin, B, containing three Y_3 items and nothing else. (A bin with three items, some of type X_3 and some of type Y_3, would be exceptional since it would contain the last X_3 item, and hence its items would have weight zero.) Of course, a Y_3 item can have weight $\frac{1}{3}$ if it is packed in a Y_1 bin, but even in this case there must be another bin containing three Y_3 items of weight $\frac{1}{3}$. If there were two Y_4 items available when B was packed, then they would have replaced the last Y_3 item since any two Y_3 items and any two Y_4 items will always fit in a single bin. Thus the Y_4 items must have been packed as fallback items in a bin before B was packed. The only way such a fallback item can have weight $\frac{1}{4}$ is if it is packed in a bin containing a Y_2 item and two Y_4 items. (Note that a bin containing an X_2 item and two Y_4 items, or one X_4 item and one Y_4 item, is exceptional.) But if such a bin were to occur before B, then two of the available Y_3 items would have been packed instead with the Y_2 item. Thus we cannot have both items, as specified in the statement of the lemma. □

LEMMA 4.3. *If B^* is a bin of the optimal packing containing a Y_1 item, then $w_B(B^*) \leqq \frac{6}{5}$.*

Proof. Assume otherwise for some bin B^* containing a Y_1 item, a. Since a cannot fit with three or more items in any bin, we must have $|B^*| \leqq 3$. We begin by observing that if a has weight exceeding $\frac{3}{5}$, then we can, without loss of generality, assume that a is the Y_1 item B2F packed in B_h. Otherwise, it would come from a bin preceding B_h in the B2F packing, and would consequently be at least as large as the Y_1 item in B_h. Thus the Y_1 item in B_h would fit in B^* in place of a, and we may as well assume that it is a.

Case 1. Suppose $|B^*| = 2$. Then certainly some item in B^* must have weight exceeding $\frac{3}{5}$, and we can assume that a is packed in B_h. Let b be the other item in B^*. Since b would fit in B_h, either b was packed earlier and thus was not available, or the item packed with a in B_h is at least as large as b. If b is packed by B2F in a Y_1 bin after B_h, then $w_B(b) = \frac{1}{5}$ since b cannot be of type X_2 or Y_2 (if it were, the item packed in B_h could not be of size $\frac{1}{3}$ or less). Thus $w_B(B^*) \leqq \frac{6}{5}$ in this case. If b is packed before a, or if b is packed after the Y_1 bins, $w_B(b) \leqq 1 - w_B(a)$ and so $w_B(B^*)$ is at most 1 in this case. We conclude that if $|B^*| = 2$, $w_B(B^*)$ cannot exceed $\frac{6}{5}$.

Case 2. Suppose $|B^*| = 3$. Let $B^* = \{a, b, c\}$ with $s(b) \geqq s(c)$. Then $s(b) < \frac{1}{3}$ and $s(c) < \frac{1}{4}$, or else $s(B^*) > 1$. Therefore, their weights are at most $\frac{1}{3}$ and $\frac{1}{4}$, respectively. Consequently, we know that $w_B(a)$ must exceed $\frac{3}{5}$ if the lemma is to fail. Thus we can assume that a is the Y_1 item in B_h. We now employ the same argument that we used in Case 1 to prove that the sum of the weights of a and either b or c can be at most 1. If both were available, then B_h would use two fallback items, so either b or c must be packed before a. Then certainly the sum of the weight of a and the weight of that item is at most 1. If one is still available, and it is not packed in a Y_1 bin after a, then it is no larger than the item packed with a by B2F. Consequently, its weight is no greater, and the sum of its weight and that of a is at most 1. If the available item is packed in a Y_1 bin after B_h, then its weight cannot be $\frac{2}{5}$ since its size is at most $\frac{1}{3}$. But if its weight is $\frac{1}{5}$, the weight of

B^* is at most $\frac{6}{5}$. From this we conclude that both b and c must have weight exceeding $\frac{1}{5}$.

Let d be the fallback item in B_h. Thus $s(b) + s(c) > s(d)$, because $s(d) \leqq \frac{1}{3}$ and $s(\mathrm{b}) \geqq s(c) > \frac{1}{6}$. It could not be the case that both b and c were available when d was packed, or else they would have replaced d. Suppose $s(d) > s(b)$. Since $s(d) + s(\text{any } Y_1 \text{ item}) \leqq 1$, and since d is larger than either b or c, whichever of these is packed before d must be one of two fallback items in its bin and hence will have weight less than $\frac{1}{5}$. Suppose $s(b) \geqq s(d) > s(c)$. Then c would have fit with a and d in B_h had it been available. Since it was not used, we conclude that c must be packed before d in a bin with two fallback items, and hence has weight less than $\frac{1}{5}$. The only remaining possibility is that $s(c) \geqq s(d)$. Now, however, any item no larger than d would fit in B_h with a and d. Since none was placed there, none can have been left to be packed after the Y_1 bins, and therefore $w_B(d)$ is zero. In this event, since b and c are packed before d, $w_B(b)$ and $w_B(c)$ are zero as well, contradicting the assumption that $w_B(B^*) > \frac{6}{5}$. □

LEMMA 4.4. *The* B2F *weight of an optimal bin cannot exceed* $\frac{6}{5}$ *unless the bin contains either a* Y_2 *item of* B2F *weight greater than* $\frac{1}{3}$ *or an item of size less than* $\frac{1}{6} + \Delta$.

Proof. To obtain the proof, suppose otherwise for some B^*. We know from Lemma 4.3 that B^* cannot contain a Y_1 item. It is easy to see then that $|B^*| \geqq 3$.

Case 1. Suppose $|B^*| = 3$. Then the only item of weight exceeding $\frac{1}{3}$ can be an X_2 item. Since any two such items and an item of size greater than $\frac{1}{6} + \Delta$ would be too big to fit, there can be at most one item of weight exceeding $\frac{1}{3}$ and $w_B(B^*) \leqq \frac{1}{2} + 2(1/3) < \frac{6}{5}$.

Case 2. Suppose $|B^*| = 4$. Suppose first that the largest item in B^* is an X_2 item. There cannot be another item of size greater than $\frac{1}{4}$, because then the sum of these sizes would exceed $\frac{5}{12} - \Delta/2 + \frac{1}{4} + 2(\frac{1}{6} + \Delta) > 1$. Items of size at most $\frac{1}{4}$ (X_4, Y_4, X_5) can have weight at most $\frac{1}{4}$. If any of these items were to have weight less than or equal to $\frac{1}{5}$, then $w_B(B^*)$ would be at most $\frac{1}{2} + 2(\frac{1}{4}) + \frac{1}{5} = \frac{6}{5}$. Thus all three items besides the X_2 item must be X_4 or Y_4 items of weight $\frac{1}{4}$. But such items have size exceeding $\frac{1}{5}$ and then $s(B^*) > \frac{5}{12} - \Delta + 3(\frac{1}{5})$ which is at least 1 if $\Delta \leqq \frac{1}{30}$. If $\Delta > \frac{1}{30}$, however, $s(B^*) > \frac{5}{12} - \Delta/2 + 3(\frac{1}{6} + \Delta) > 1$. Thus in all cases where $|B^*| = 4$ and B^* contains an X_2 item, $w_B(B^*) \leqq \frac{6}{5}$.

Suppose now that the largest item is a Y_2 item, which has weight less than or equal to $\frac{1}{3}$ by assumption. If there were two additional items of size greater than $\frac{1}{4}$, we would have $s(B^*) > \frac{1}{3} + 2(\frac{1}{4}) + \frac{1}{6} + \Delta > 1$. Thus there must be two items of size less than or equal to $\frac{1}{4}$, and hence of weight at most $\frac{1}{4}$. Since there can be no item of weight exceeding $\frac{1}{3}$, we must have $w_B(B^*) \leqq 2(\frac{1}{3} + \frac{1}{4}) < \frac{6}{5}$.

Since $|B^*| = 4$, there must be at least one item of weight exceeding $\frac{3}{10}$, which must be of type X_3 or Y_3 and of weight $\frac{1}{3}$. If any item has weight less than or equal to $\frac{1}{5}$, $w_B(B^*)$ would be at most $3(\frac{1}{3}) + \frac{1}{5} = \frac{6}{5}$. If there are two items of weight $\frac{1}{4}$, we would still have $w_B(B^*) < \frac{6}{5}$. There cannot be four items of size greater than $\frac{1}{4}$, so there must be an X_4 or Y_4 item of weight $\frac{1}{4}$ and three X_3 or Y_3 items. At least two of the X_3 or Y_3 items must have weight $\frac{1}{3}$, and so there must be a bin in the B2F packing containing three X_3 or Y_3 items of weight $\frac{1}{3}$. In particular, the last three-item bin is exceptional and must contain three items no larger than those in B^*. (Even if the items in B^* are fallback items, there must be such a three-item bin, and the last bin is exceptional.) If the item, x, of weight $\frac{1}{4}$ were still available when this three-item bin was packed, then x and any other item of weight $\frac{1}{4}$ would replace the bin's last item. Thus x must be packed as a fallback item in an earlier bin. The only way to have weight $\frac{1}{4}$ would be in a bin with an X_2 (or Y_2) item and another item of weight $\frac{1}{4}$. In this event, however, x and any of the items in the last three-item bin would fit with the X_2 item since they fit with two other X_3 or Y_3 items. (Note that x cannot be packed as a single fallback item in a Y_1 bin, since any X_3 or Y_3 item would fit and be used instead of x.)

Case 3. Suppose $|B^*| = 5$. If any item has size exceeding $\frac{1}{3}$, $s(B^*)$ would be greater than $\frac{1}{3} + 4(\frac{1}{6} + \Delta) > 1$. Similarly, not all items can have size exceeding $\frac{1}{5}$. If all items are no larger than $\frac{1}{4}$ in size, then the weight of B^* would be at most $\frac{6}{5}$ since none of the items could have weight more than $\frac{1}{4}$ and at least one would have weight at most $\frac{1}{5}$. Thus there must be at least one X_3 or Y_3 item, and at least one X_5 item.

Suppose there is an X_3 item. If there are two additional items of size greater than $\frac{1}{5}$, $s(B^*) > \frac{5}{18} - \Delta/3 + 2(\frac{1}{5}) + 2(\frac{1}{6} + \Delta) > 1$. Thus there is at most one item of weight exceeding $\frac{1}{4}$ and one additional item of size exceeding $\frac{1}{5}$. Hence, $w_B(B^*) \leqq \frac{1}{3} + \frac{1}{4} + 3(\frac{1}{5}) < \frac{6}{5}$.

Suppose finally that the largest item in B^* is of type Y_3. B^* can contain at most one such item, or else $s(B^*) > 2(\frac{1}{4}) + 3(\frac{1}{6} + \Delta) > 1$. Also, B^* cannot contain a Y_3 item and two X_4 items, or else $s(B^*) > \frac{1}{4} + 2(\frac{5}{24} - \Delta/4) + 2(\frac{1}{6} + \Delta) > 1$. However, if B^* contains three items each of weight less than or equal to $\frac{1}{5}$, $w_B(B^*) \leqq \frac{1}{3} + \frac{1}{4} + 3(\frac{1}{5}) < \frac{6}{5}$. There must be at least two items of size (and hence weight) no greater than $\frac{1}{5}$, or else $s(B^*) > \frac{1}{4} + 3(\frac{1}{5}) + \frac{1}{6} + \Delta > 1$. The only remaining possibility is for B^* to contain a Y_3 item, a Y_4 item, an X_4 or Y_4 item, and two X_5 items. By Lemma 4.2, either the Y_3 item has weight less than $\frac{1}{3}$ or the Y_4 item has weight less than $\frac{1}{4}$. Either of these possibilities contradicts the assumption that $w_B(B^*) > \frac{6}{5}$. □

LEMMA 4.5. *There cannot be a Y_1 item a with $w_B(a) \geqq \frac{4}{3}$ if there exist items b and c with $w_B(b) = \frac{1}{3}$, $s(c) > \max(\frac{5}{24} - \Delta/4, \frac{1}{6} + \Delta)$, and $s(a) + s(b) + s(c) \leqq 1$.*

Proof. We shall show that under these conditions no bin of the optimal packing can have a B2F weight exceeding $\frac{6}{5}$. Suppose L contains such items and that, for some optimal bin B^*, $w_B(B^*) > \frac{6}{5}$. As we have seen before, there is no loss of generality in assuming that a is the Y_1 item in B_h. We know from Lemma 4.4 that B^* must contain a Y_2 item of weight greater than $\frac{1}{3}$ or an item of size less than $\frac{1}{6} + \Delta$. If there is a Y_2 item of weight exceeding $\frac{1}{3}$, however, then there must have been such an item available when B_h was packed. Since any such item is smaller than the sum of the sizes of b and c, it would fit with a in B_h, contradicting the definition of B_h. Thus the only possiblility is for B^* to contain an item, d, of size less than $\frac{1}{6} + \Delta$. When a was packed, if d and any other item of size at most that of b were available, they would be used in place of the fallback item in B_h. Since $w_B(b) = \frac{1}{3}$, either b itself must have been available or b must be packed in an earlier Y_1 bin and some item no larger than b must have been available. In either case, there must have been an item no larger than b available when B_h was packed. Thus d must not have been available. But if d is packed in a Y_1 bin before B_h, it cannot be the only fallback item, since there must be an item of weight $\frac{1}{3}$, no larger than b, available that would fit with any Y_1 item. Thus d must have weight no greater than $\frac{1}{6}$. At this point, we must consider the possible configurations for B^*. Certainly B^* must contain at least three items.

Case 1. Suppose $|B^*| = 3$. By Lemma 4.3, B^* cannot contain an item of weight greater than $\frac{1}{2}$. Thus $w_B(B^*) \leqq \frac{1}{2} + \frac{1}{2} + \frac{1}{6} = \frac{7}{6}$.

Case 2. Suppose $|B^*| = 4$. B^* must contain an item of weight greater than $\frac{1}{3}$, which can only be an X_2 item by Lemma 4.3 and by the above arguments focusing on the weight of Y_2 items. Moreover, if there is not a second item of weight greater than $\frac{1}{4}$, then we would have $w_B(B^*) \leqq \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{6} = \frac{7}{6}$. If there is a second item larger than $\frac{1}{4}$ in size, then there must be a second item whose size is less than $\frac{1}{6} + \Delta$, or else $s(B^*) > \frac{5}{12} - \Delta/2 + \frac{1}{4} + \frac{1}{6} + \frac{1}{6} + \Delta > 1$. Since this second small item will also have weight no greater than $\frac{1}{6}$, we again have that $w_B(B^*) \leqq \frac{1}{2} + \frac{1}{3} + 2(\frac{1}{6}) = \frac{7}{6}$.

Case 3. Suppose $|B^*| = 5$. B^* must contain an item of weight greater than $\frac{1}{4}$. There cannot be two such items, or else $s(B^*) > 1$. If any of the remaining items has weight less than or equal to $\frac{1}{5}$, then $w_B(B^*) \leqq \frac{1}{3} + 2(\frac{1}{4}) + \frac{1}{5} + \frac{1}{6} = \frac{6}{5}$. But this means that there must be one item whose size exceeds $\frac{1}{4}$ and three additional items each of whose sizes exceeds $\frac{1}{5}$. Thus $s(B^*) > \frac{1}{4} + 3(\frac{1}{5}) + \frac{1}{6} > 1$. Hence, in all cases, $w_B(B^*) \leqq \frac{6}{5}$.

To complete the proof of Lemma 4.5, we observe that $w_B(L) \geqq$ B2F $(L) - 8$ since each nonexceptional bin has a weight of 1, while $w_B(L) \leqq (\frac{6}{5})$ OPT (L) since each optimal bin has a weight bounded above by $\frac{6}{5}$. Combining these yields B2F $(L) \leqq (\frac{6}{5})$ OPT $(L) + 8$, contradicting the assumption that L was a counterexample for CFB. □

5. Proof of the main result. We shall now employ our weighting function averaging technique to obtain the final result. From Corollary 3.1 and Lemma 4.4 we know the optimal bin configurations that may have "too much" weight from the respective FFD or B2F weighting function, and that since FFD fails to achieve the required bound, any Y_2 item receives an FFD weight of $\frac{1}{3}$. Also, from Lemma 4.5, we know that since B2F fails, any Y_1 item either cannot be packed extremely well or receives a B2F weight of $\frac{3}{5}$. The heart of the proof of the main result is now contained in the following lemma.

LEMMA 5.1. *If B^* is any bin of the optimal packing of L, $w_A(B^*) = (w_F(B^*) + w_B(\mathrm{B}^*))/2 \leqq \frac{6}{5}$.*

Proof. To obtain the proof, suppose otherwise for some optimal bin B^*. Clearly, at least one of the two weighting functions must give B^* a weight exceeding $\frac{6}{5}$.

Case 1. Suppose $w_F(B^*) > \frac{6}{5}$. Then we know that B^* must contain a Y_1 item, a, and that $|B^*| \leqq 3$. If $|B^*| = 2$, then the second item, b, would fit with a when a was packed. If it is unavailable, then the Y_1 packing rule for FFD implies that b cannot have weight exceeding $1 - w_F(a)$. If b is available, then the item packed with a is at least as large as b. If b has weight less than or equal to $\frac{2}{5}$, then $w_F(a) + w_F(b) \leqq \frac{6}{5}$, since a cannot have weight exceeding $\frac{4}{5}$ unless nothing fits with it. Since we already know that there are no Y_2 bins in the FFD packing by Lemma 3.2, b must be an X_2 item. In this case, however, a must also be packed by FFD with an X_2 item. Thus $w_F(a) = \frac{3}{5}$ and $w_F(B^*) < \frac{6}{5}$.

Therefore, we may assume that $B^* = \{a, b, c\}$, where $s(a) > s(b) \geqq s(c)$. It is easy to see that $s(c) < \frac{1}{4}$ and $s(b) < \frac{1}{3}$, or else $s(B^*)$ would exceed 1. Hence $w_F(c) \leqq \frac{1}{4}$ and $w_F(b) \leqq \frac{1}{3}$, implying that $w_F(a)$ must be greater than $\frac{3}{5}$. Let B_i denote the FFD bin containing a. Since b would fit in B_i with a (or any other Y_1 item), $w_F(b) \leqq 1 - w_F(a)$, and $w_F(B^*) \leqq \frac{5}{4}$. Note further that c must be an X_4 item, or else its weight would be $\frac{1}{5}$ and B^* would have weight less than or equal to $\frac{6}{5}$.

This is, for those readers already acquainted with FFD, exactly the kind of situation where one expects FFD to perform poorly. We now show that, in this case, the averaging process with B2F permits our compound algorithm to succeed.

Suppose $w_B(a) = \frac{3}{5}$. Unless $w_B(b) = \frac{1}{3}$ and $w_B(c) = \frac{1}{4}$, we have $w_A(B^*) \leqq (\frac{5}{4} + 23/20)/2 = \frac{6}{5}$. Now $w_B(b) = \frac{1}{3}$ implies the existence of a bin containing three items of type X_3 or Y_3, each of weight $\frac{1}{3}$. There must also be a bin containing three such items each no larger than b, although their weight may be zero if they are exceptional. If c were available when this three-item bin was packed, it and any smaller item would replace the last Y_3 or X_3 item. But if c is the smallest item left, it is either l_{last} and hence is exceptional, or it is a fallback item and has weight at most $\frac{1}{5}$. Therefore, c must not be available. If c were packed in a Y_1 bin, the items of the three-item bin would have been used unless c is packed with a second fallback item. However, it then has weight at most $\frac{1}{5}$. The only remaining possibility would be for c to be packed with another X_4 or Y_4 item and an X_2 or Y_2 item. In this event, however, c and any item from the three-item bin would have fit with any X_2 or Y_2 item and been used instead. Thus we know that it is impossible for a to have weight $\frac{3}{5}$ in the B2F weighting.

Suppose $w_B(a) > \frac{3}{5}$. It must be that a is in a Y_1 bin packed no later than B_h, and as argued before there is no loss of generality in assuming that a is packed in B_h. Let d be the fallback item packed with a in B_h. Thus $s(b) + s(c) > s(d)$, and both b and c could not have been available when d was packed, else they would have replaced d. If $s(d) > s(b)$, then $w_B(d) \geqq \max\{w_B(b), w_B(c)\}$ and whichever of b or c is not available has weight less than or equal to $(\frac{1}{2})w_B(d)$. This causes $w_B(B^*)$ to be at most $1 + (\frac{1}{2})w_B(b)$ no matter where b was packed by B2F. Unless b has weight $\frac{1}{3}$, this quantity is at most $23/20$ and $w_A(B^*)$ would be at most $\frac{6}{5}$. But this is precisely the situation ruled out by Lemma 4.5. If $s(b) \geqq s(d) > s(c)$, then we reach the same conclusion, since it must still be that $w_B(d) \geqq w_B(b)$ and since c is not available implying $w_B(c) \leqq (\frac{1}{2})w_B(d)$. Finally, if $s(c) \geqq s(d)$, then any item no larger than d would fit in B_h along with a and d. Since none was used, $w_B(b)$, $w_B(c)$ and $w_B(d)$ are all zero.

Case 2. Suppose $w_B(B^*) > \frac{6}{5}$ and B^* contains a Y_2 item of B2F weight exceeding $\frac{1}{3}$.

Certainly $|B^*| \leqq 4$, since no bin can contain an item of size greater than $\frac{1}{3}$ and four additional items.

Suppose $|B^*| = 4$. Then there can be no other X_2 or Y_2 items and at most one other item of size exceeding $\frac{1}{4}$, or else $s(B^*) > 1$. If all other items are at most $\frac{1}{4}$ in size, then $w_B(B^*) \leqq \frac{1}{2} + 3(\frac{1}{4}) = \frac{5}{4}$. Since $w_F(B^*) \leqq \frac{1}{3} + 3(\frac{1}{4})$, we would have $w_A(B^*) < \frac{6}{5}$. Therefore there must be an X_3 or Y_3 item.

Suppose B^* contains an X_3 item. Then there must also be either an item of size at most $\frac{1}{5}$ or an item of size less than $\frac{1}{6} + \Delta$. To see this, observe that if all items have size exceeding $\frac{1}{5}$, $s(B^*) > \frac{1}{3} + \frac{5}{18} - \Delta/3 + \frac{2}{5} \geqq 1$ if $\Delta \leqq \frac{1}{30}$. If all items have size greater than or equal to $\frac{1}{6} + \Delta$, $s(B^*) > \frac{1}{3} + \frac{5}{18} - \Delta/3 + 2(\frac{1}{6} + \Delta) \geqq 1$ if $\Delta > \frac{1}{30}$. However, if there is an item of size less than $\frac{1}{6} + \Delta$, then $w_F(B^*) \leqq 2(\frac{1}{3}) + \frac{1}{4} + 0 = 11/12$ while $w_B(B^*) \leqq \frac{1}{2} + \frac{1}{3} + 2(\frac{1}{4}) = \frac{4}{3}$. If, on the other hand, B^* contains an X_5 item, then $w_B(B^*) \leqq \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} = 77/60$ while $w_F(B^*) \leqq 2(\frac{1}{3}) + \frac{1}{4} + \frac{1}{5} = 67/60$. In either case, $w_A(B^*) \leqq \frac{6}{5}$.

Suppose B^* contains a Y_3 item, x. Since $w_F(x) \leqq \frac{4}{15}$, $w_F(B^*) \leqq \frac{1}{3} + \frac{4}{15} + 2(\frac{1}{4}) = 11/10$. Thus $w_B(B^*)$ must be more than $13/10$, or else $w_A(B^*)$ cannot exceed $\frac{6}{5}$. This implies that $w_B(x)$ must be $\frac{1}{3}$. In this event, there must be a three-item bin with three Y_3 items each of weight $\frac{1}{3}$. If there is a Y_2 item of weight $\frac{1}{2}$, it must come from an earlier bin containing exactly two Y_2 items. The second of these items would have been replaced, however, by any two of the Y_3 items, since any Y_2 item will fit with any two Y_3 items. Thus there can be no Y_2 items of weight $\frac{1}{2}$ in the B2F packing, and the weight of the Y_2 item must be $\frac{2}{5}$. Therefore, $w_B(B^*) \leqq \frac{2}{5} + \frac{1}{3} + 2(\frac{1}{4}) < 13/10$.

Suppose now that $|B^*| = 3$. If there is no X_2 item or if there is an item of size less than $\frac{1}{6} + \Delta$, then $w_F(B^*) \leqq 1$ and $w_A(B^*) < \frac{6}{5}$. Thus we may assume that B^* contains an X_2 item and that its remaining item is at least $\frac{1}{6} + \Delta$ in size. Even if the small item, y, is of type X_3, then $w_F(B^*)$ is at most $\frac{1}{2} + 2(\frac{1}{3}) = \frac{7}{6}$. If the Y_2 item, x, has B2F weight less than $\frac{1}{2}$, $w_B(B^*) \leqq \frac{1}{2} + \frac{2}{5} + \frac{1}{3}$ and $w_A(B^*) \leqq \frac{6}{5}$. The only way that x can have weight $\frac{1}{2}$ is to be in a two-item bin, B_j, with another Y_2 item. This means that y must not have been available when B_j was packed, since it would have fit with x and any Y_2 item (it fits in B^* with x and an X_2 item). Thus y cannot have weight $\frac{1}{3}$ unless there is a three-item bin consisting of items no larger than y. These items, however, must have been available when x was packed, and thus y still cannot have weight $\frac{1}{3}$. Therefore, the maximum B2F weight for y is $\frac{3}{10}$.

If y is a Y_3 item, then $w_F(y) = \frac{4}{15}$ and $w_F(B^*) \leqq \frac{1}{2} + \frac{1}{3} + \frac{4}{15} = 11/10$. Thus $w_B(B^*)$ is at most $\frac{1}{2} + \frac{1}{2} + \frac{3}{10} = 13/10$ and $w_A(B^*) \leqq \frac{6}{5}$.

Therefore, y must be an X_3 item. It must also be that Δ exceeds $\frac{1}{30}$, or else $s(B^*) > \frac{5}{12} - \Delta/2 + \frac{1}{3} + \frac{5}{18} - \Delta/3 \geqq 1$. Let B_i be the bin containing the Y_2 item, x, in the FFD packing. We know from Lemma 3.2 that B_i is a Y_1 bin. Let z be the X_2 item in B^*. If z were packed in a Y_1 bin in the FFD packing, its weight would be $\frac{2}{5}$ and $w_A(B^*)$ would be $\leqq \frac{6}{5}$. Thus z must have been available when B_i was packed. Since it was not used in place of x, the size of the Y_1 item in B_i exceeds $1 - s(z)$. But z fits with x and y, so $s(z) < 1 - \frac{1}{3} - (\frac{5}{18} - \Delta/3) = \frac{7}{18} + \Delta/3$. Then $1 - s(z) = 11/18 - \Delta/3$, which is greater than $\frac{2}{3} - 2\Delta$ if $\Delta > \frac{1}{30}$. The weighting function for FFD gives weight at most $\frac{4}{15}$ to a Y_2 item packed with such a large Y_1 item, and again we have $w_A(B^*) \leqq \frac{6}{5}$.

Case 3. Suppose $w_B(B^*) > \frac{6}{5}$ and B^* contains an item a, where $s(a) < \frac{1}{6} + \Delta$.

We know that B^* contains neither a Y_1 item nor a Y_2 item of weight exceeding $\frac{1}{3}$ by Lemma 4.3 and Case 2 above, respectively. We also know that $w_B(a)$ is at most $\frac{1}{4}$ since $s(a) < \frac{1}{4}$. By Lemma 3.3, we know further that $w_F(B^*) \leqq 1$, so that if $w_A(B^*)$ is to exceed $\frac{6}{5}$, we must have $w_B(B^*) > \frac{7}{5}$. Thus $|B^*| > 3$, since any two items with a can each have weight at most $\frac{1}{2}$.

Suppose $|B^*| = 4$. Then there must be an X_2 item, or else $w_B(B^*) \leqq 3(\frac{1}{3}) + \frac{1}{4}$. There can be at most one additional item exceeding $\frac{1}{4}$ in size, or else $s(B^*) > \frac{5}{12} - \Delta/2 + 2(\frac{1}{4}) + \frac{1}{6} > 1$. But then $w_B(B^*) \leqq \frac{1}{2} + \frac{1}{3} + 2(\frac{1}{4}) < \frac{7}{5}$.

Suppose $|B^*| = 5$. There cannot be an X_2 item, or else $s(B^*) > \frac{5}{12} - \Delta/2 + 4(\frac{1}{6}) > 1$. Nor can there be two items of size greater than $\frac{1}{4}$, or else $s(B^*) > 2(\frac{1}{4}) + 3(\frac{1}{6}) = 1$. Finally, if only one item has size exceeding $\frac{1}{4}$, $w_B(B^*) \leqq \frac{1}{3} + 4(\frac{1}{4}) < \frac{7}{5}$. □

THEOREM 5.1. Min $\{\text{FFD}(L), \text{B2F}(L)\} \leqq (\frac{6}{5})\,\text{OPT}(L) + 8$.

Proof. To obtain this inequality, we observe that our presumed counterexample obeys min $\{\text{FFD}(L), \text{B2F}(L)\} - 8 \leqq (\text{FFD}(L) - 8 + \text{B2F}(L) - 8)/2 \leqq (w_F(L) + w_B(L))/2 = w_A(L)$ by our definitions for w_F, w_B, and w_A, while $w_A(L) \leqq (\frac{6}{5})\,\text{OPT}(L)$ by Lemma 5.1. □

6. Remarks. We have limited our analysis to proving that, for any list, either the FFD or the B2F algorithm will asymptotically use within $\frac{6}{5}$ the optimal number of bins. However, we have been unable to find examples that are even close to this bound. In fact, the only examples we have been able to contrive that exceed $\frac{9}{8}$ the optimum depend heavily on the modification that we introduced to B2F to simplify our proof. For these instances, this modification forces the B2F packing to be the same as the FFD packing. If "small" items are not held back, the exact bound might be significantly better (although a proof of this may well be extremely difficult).

Our weighting function averaging technique actually proves that, even if both algorithms produce particularly egregious packings for some list, the average of the number of bins used by FFD and the number used by B2F is asymptotically at most $\frac{6}{5}$ the optimal number of bins for that list. Presumably, the minimum may always be considerably less than this upper bound on the average. Furthermore, we remark that the additive constant we have used (eight) is much higher than necessary. Instead of assigning a weight of zero to every exceptional item, we could assign a weight that agrees with an item's type, and easily reduce this constant. Nevertheless, because we believe that the $\frac{6}{5}$ coefficient is itself inflated, the additive constant appears to be of little significance.

Appendix. Bin packing results for B2F alone. We seek to determine the worst-case behavior of the B2F algorithm. Before doing so, however, we briefly discuss some other aspects of this approach to bin packing.

We could extend the idea of "best 2 fit" to "best j fit," for arbitrary $j > 2$. It seems likely that the expected performance of these more complex algorithms might be better, although the worst-case performance can be shown to be worse, approaching a number greater than 1.3 as j grows without bound. Simple tests using a uniform distribution for item sizes seem to back up the improved expected case, although the run time increases rapidly.

B2F can also be used in the multifit approach to multiprocessor scheduling. Again, its worst-case performance is poorer than that of FFD. In [3], it is shown that B2F's asymptotic worst-case bound is precisely $\frac{6}{5}$, while it has been proved in [4] that FFD can be implemented to ensure a tight bound of 72/61.

Returning to bin packing, Fig. 3 depicts an example illustrating that B2F may require, asymptotically, as many as $\frac{5}{4}$ the optimal number of bins.

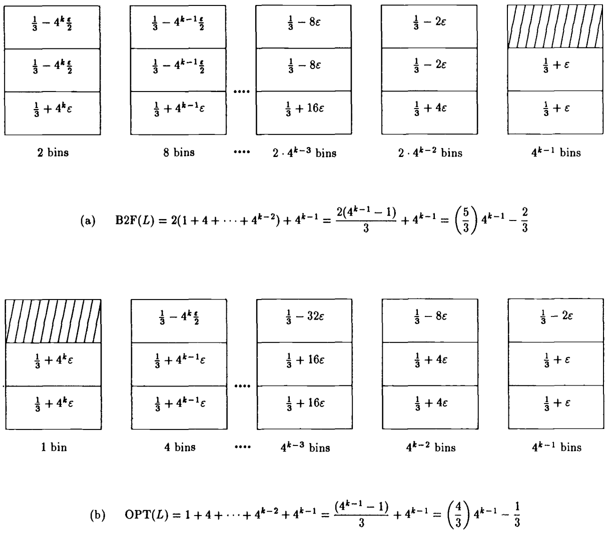
To prove that the $\frac{5}{4}$ ratio cannot be exceeded by B2F, we modify the algorithm slightly in that items less than or equal to $\frac{1}{5}$ the bin size will be held back and packed by the FFD algorithm. This certainly does not affect the example illustrated in Fig. 3, but it allows us to assume that no items of size $\frac{1}{5}$ or less are used in packing L, which we now presume to be minimal counterexample. This reduces the number of cases we must investigate, thereby simplifying our proof (although it probably detracts from the expected performance of the algorithm).

LEMMA A. *Every item in L has less than $\frac{3}{5}$.*

Proof. Let b be the largest item in L and suppose $s(b) \geqq \frac{3}{5}$. Then b is packed in B_1 by the B2F rule. Removing the items of B_1 cannot change the remainder of the packing. Since $s(b) \geqq \frac{3}{5}$, $|B_1| \leqq 2$ and, if $|B_1| = 2$, then B_1 contains the largest item that would fit with b in a bin of size 1. If the item or items of B_1 are removed from L, then both B2F (L) and OPT (L) can easly be reduced by one, contradicting the presumed minimality of L with respect to B2F. □

THEOREM A. B2F $(L) \leqq (\frac{5}{4})$ OPT $(L) + 4$.

Proof. We classify an item, x, by its size so that if $1/(i + 1) < \cdot s(x) \leqq 1/i$, then x is of type X_i. The reasoning above shows that all items are of types X_1, X_2, X_3, or X_4, and items of type X_1 are less than $\frac{3}{5}$ in size. We now define a weighting function w on the items of L based on the B2F packing.


$\frac{1}{3} - 4^k\frac{\varepsilon}{2}$
$\frac{1}{3} - 4^k\frac{\varepsilon}{2}$
$\frac{1}{3} + 4^k\varepsilon$
2 bins
$\frac{1}{3} - 4^{k-1}\frac{\varepsilon}{2}$
$\frac{1}{3} - 4^{k-1}\frac{\varepsilon}{2}$
$\frac{1}{3} + 4^{k-1}\varepsilon$
8 bins
....
$\frac{1}{3} - 8\varepsilon$
$\frac{1}{3} - 8\varepsilon$
$\frac{1}{3} + 16\varepsilon$
$2 \cdot 4^{k-3}$ bins
$\frac{1}{3} - 2\varepsilon$
$\frac{1}{3} - 2\varepsilon$
$\frac{1}{3} + 4\varepsilon$
$2 \cdot 4^{k-2}$ bins
$\frac{1}{3} + \varepsilon$
$\frac{1}{3} + \varepsilon$
4^{k-1} bins
$\frac{1}{3} + 4^k\varepsilon$
$\frac{1}{3} + 4^k\varepsilon$
1 bin
$\frac{1}{3} - 4^k\frac{\varepsilon}{2}$
$\frac{1}{3} + 4^{k-1}\varepsilon$
$\frac{1}{3} + 4^{k-1}\varepsilon$
4 bins
....
$\frac{1}{3} - 32\varepsilon$
$\frac{1}{3} + 16\varepsilon$
$\frac{1}{3} + 16\varepsilon$
4^{k-3} bins
$\frac{1}{3} - 8\varepsilon$
$\frac{1}{3} + 4\varepsilon$
$\frac{1}{3} + 4\varepsilon$
4^{k-2} bins
$\frac{1}{3} - 2\varepsilon$
$\frac{1}{3} + \varepsilon$
$\frac{1}{3} + \varepsilon$
4^{k-1} bins

(a) $\mathrm{B2F}(L) = 2(1 + 4 + \cdots + 4^{k-2}) + 4^{k-1} = \dfrac{2(4^{k-1} - 1)}{3} + 4^{k-1} = \left(\dfrac{5}{3}\right) 4^{k-1} - \dfrac{2}{3}$

(b) $\mathrm{OPT}(L) = 1 + 4 + \cdots + 4^{k-2} + 4^{k-1} = \dfrac{(4^{k-1} - 1)}{3} + 4^{k-1} = \left(\dfrac{4}{3}\right) 4^{k-1} - \dfrac{1}{3}$

FIG. 3. *Worst-case example for* B2F. (B2F (L)/OPT (L)) $= (5(4^{k-1}) - 2)/(4(4^{k-1}) - 1) \rightarrow 5/4$ *as* $k \rightarrow \infty$.

If B is any bin with four items in it, each item is assigned a weight of $\frac{1}{5}$. Suppose B is a bin containing an X_1 item, b. Then if $|B| = 3$, $w(b) = \frac{8}{15}$, and the other two items are each assigned a weight of $\frac{2}{15}$. If $|B| = 2$, then $w(b) = \frac{8}{15}$ and the other item is assigned a weight of $\frac{4}{15}$ if the other item is of type X_2. Otherwise, $w(b) = \frac{3}{5}$ and the remaining item is assigned a weight of $\frac{1}{5}$.

Suppose the largest item in B is of type X_2. Then if $|B| = 2$, each item must be of type X_2, and is assigned a weight of $\frac{2}{5}$, except possibly for the last bin containing an X_2 item. If the last bin containing an X_2 item has only 2 items in it, it will be classified as exceptional (as will its items). All exceptional items are given weight zero. (This is an unnecessarily strict weight reduction, accounting for the constant 4 in the theorem. A more careful analysis using larger weights for the exceptional items could likely reduce this constant to 1.) If $|B| = 3$ and B contains two X_2 items, each is given a weight of $\frac{3}{10}$ and the remaining item is given a weight of $\frac{1}{5}$. If B contains only one X_2 item, then it is given a weight of $\frac{2}{5}$ and the other two are each given a weight of $\frac{1}{5}$. If the largest item is of type X_3, then $|B| = 3$ implies all three items are of type X_3, except possibly for the last such bin (which is also classified as exceptional). All three X_3 items in such a bin are given a weight of $\frac{4}{15}$. One additional exceptional bin shall be identified. If the last X_2 item of size exceeding $\frac{7}{15}$ is packed with an X_2 item of size less than $\frac{7}{20}$, then this bin is classified as exceptional, and its items assigned weights of zero.

The definition of w is summarized in Table 6.

We now show that each bin B^* of the optimal packing must satisfy $w(B^*) \leqq 1$. This, together with the observation that $w(B) = \frac{4}{5}$ for each nonexceptional bin in the B2F packing, will complete the proof of Theorem A.

Suppose B^* is a bin of the optimal packing with $w(B^*) > 1$. Clearly, $|B^*| > 1$. (If B^* contains an exceptional item, then after removing the item $w(B^*)$ would still exceed 1. Thus it is enough to show that $w(B^*) \leqq 1$ for bins not containing exceptional items.)

Case 1. Suppose $|B^*| = 2$.

If neither item has weight greater than $\frac{2}{5}$, then $w(B^*) \leqq \frac{4}{5} < 1$. Thus B^* must contain an item of type X_1. The weight of this item is less than or equal to $\frac{3}{5}$ and the weight of an X_2 item is less than or equal to $\frac{2}{5}$. Since B^* cannot contain two X_1 items, $w(B^*) \leqq \frac{3}{5} + \frac{2}{5} = 1$.

Case 2. Suppose $|B^*| = 3$.

The largest item in B^* must have a weight exceeding $\frac{1}{3}$, and so must be of type X_1 or X_2.

TABLE 6

Weighting function w used in analysis of B2F *alone.*

Nonexceptional bin contents	Weights assigned	
X_1, X_i, X_j	$\frac{8}{15}, \frac{2}{15}, \frac{2}{15}$	$i, j > 2$
X_1, X_2	$\frac{8}{15}, \frac{4}{15}$	
X_1, X_i	$\frac{3}{5}, \frac{1}{5}$	$i > 2$
X_2, X_2	$\frac{2}{5}, \frac{2}{5}$	
X_2, X_2, X_i	$\frac{3}{10}, \frac{3}{10}, \frac{1}{5}$	$i > 2$
X_2, X_i, X_j	$\frac{2}{5}, \frac{1}{5}, \frac{1}{5}$	$i, j > 2$
X_3, X_3, X_3	$\frac{4}{15}, \frac{4}{15}, \frac{4}{15}$	
any four items	$\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}$	

Suppose the largest item is of type X_1, so that $B^* = \{b, c, d\}$, where $s(b) > s(c) \geqq s(d)$. Then neither c nor d can be of type X_2. If both are of type X_4, then $w(B^*) \leqq 1$. Both cannot be of type X_3, since $s(b) + s(c) + s(d)$ cannot exceed 1. Thus c is of type X_3, $w(c) \leqq \frac{4}{15}$, d is of type X_4, and $w(d) \leqq \frac{1}{5}$. If both c and d were available when b was packed, then either b was packed with an X_2 item or with two other items. In either case, $w(b) = \frac{8}{15}$ and $w(B^*) \leqq \frac{8}{15} + \frac{4}{15} + \frac{1}{5} = 1$. Therefore, $w(b)$ must be $\frac{3}{5}$. If c is packed before b, then $w(c) \leqq \frac{1}{5}$ and $w(B^*) \leqq 1$. If d is packed before b, then it must be packed with an X_1 item and another item, since c would have fit and was not used. Hence $w(b) = \frac{2}{15}$ and $w(B^*) \leqq \frac{3}{5} + \frac{4}{15} + \frac{2}{15} = 1$.

Thus the largest item in B^* must be of type X_2. If there is only one X_2 item, $w(B^*) \leqq \frac{2}{5} + 2(\frac{4}{15}) < 1$. Thus $B^* = \{b, c, d\}$ with b and c both of type X_2, where $s(b) \geqq s(c)$. If $w(d) \leqq \frac{1}{5}$, then $w(B^*) \leqq 2(\frac{2}{5}) + \frac{1}{5} = 1$. Thus d is an X_3 item and $w(d) = \frac{4}{15}$. Also, $w(b) = w(c) = \frac{2}{5}$, since otherwise $w(B^*) \leqq \frac{2}{5} + \frac{3}{10} + \frac{4}{15} < 1$. Since $s(d) > \frac{1}{4}$, it must be that $s(c) < \frac{3}{8}$. If d were packed before c, then $w(d)$ would only be $\frac{1}{5}$, so that d must be available. In order for $w(c)$ to be $\frac{2}{5}$, c must be packed by B2F in a bin $B = \{c, x\}$ or $\{c, y, z\}$. If $|B| = 2$, then since d would not fit in B, $s(x) > s(b)$ and b must be in a bin with an X_2 item and one other item, contradicting $w(b) = \frac{2}{5}$. If $|B| = 3$, then neither y nor z can be of type X_2. Since there must be an X_2 item, u, left (or else B would be exceptional) and since u is smaller than c, the B2F rule would have placed c, u, and an X_3 item in B since c, u, and d would have fit. Thus it is impossible to have $w(b) = w(c) = \frac{2}{5}$ while $w(d) = \frac{4}{15}$ and we conclude that, in any event, $w(B^*) \leqq 1$.

Case 3. Suppose $|B^*| = 4$.

B^* cannot contain an X_1 item, since $\frac{1}{2} + 3(\frac{1}{5}) > 1$. Neither can it contain two X_2 items, since $2(\frac{1}{3}) + 2(\frac{1}{5}) > 1$. Similarly, it cannot contain four X_3 items, since each has size greater than $\frac{1}{4}$. However, if it contains three items of type X_3 and one of type X_4, then $w(B^*) \leqq 3(\frac{4}{15}) + \frac{1}{5} = 1$. Thus B^* must contain exactly one X_2 item. If the other three items have weight less than or equal to $\frac{1}{5}$, $w(B^*) \leqq \frac{2}{5} + 3(\frac{1}{5}) = 1$. If there were two X_3 items, $s(B^*) > \frac{1}{3} + 2(\frac{1}{4}) + \frac{1}{5} > 1$. Thus B^* must contain exactly one X_3 item. Let $B^* = \{b, c, d, e\}$, with b of type X_2, and c of type X_3. If $w(b) < \frac{2}{5}$, then $w(B^*) \leqq \frac{3}{10} + \frac{4}{15} + 2(\frac{1}{5}) < 1$. Thus $w(b) = \frac{2}{5}$ and $w(c) = \frac{4}{15}$. This means c must be available when b is packed.

If b is the largest item in some bin B of the B2F packing, then B would contain two X_2 items and another item since $s(b) + s(c) + s(d) + s(e) \leqq 1$ implies that $2s(b) + s(c) < 1$. This cannot happen, however, so it must be that $B = \{x, b\}$ where $s(x) > 1 - 2s(c)$, since b was not replaced by two smaller items. Because $s(c) < 1 - \frac{1}{3} - \frac{2}{5} = \frac{4}{15}$, we know $s(x) > \frac{7}{15}$. Thus B is the third exceptional bin ($s(b) < 1 - \frac{1}{4} - \frac{2}{5} = \frac{7}{20}$) and again $w(B^*) \leqq 1$.

Now, to complete our proof of Theorem A, we note that $w(B) = \frac{4}{5}$ for all but at most four B2F bins (the three exceptional bins and the last bin), so that $\sum_{x \in L} w(x) \geqq (\frac{4}{5})(\text{B2F}(L) - 4)$. At the same time, $w(B^*) \leqq 1$ for all B^* in the optimal packing ensures $\sum_{x \in L} w(x) \leqq \text{OPT}(L)$. Combining these two inequalities yields $\text{B2F}(L) \leqq (\frac{5}{4})\,\text{OPT}(L) + 4$, as desired. □

REFERENCES

[1] W. Fernandez de la Vega and G. S. Lueker, *Bin packing can be solved within* 1 + ε *in linear time*, Combinatorica, 1 (1981), pp. 349-355.

[2] G. N. FREDERICKSON, M. S. HECHT, AND C. E. KIM, *Approximation algorithms for some routing problems*, SIAM J. Computing, 7 (1978), pp. 178–193.

[3] D. K. FRIESEN, *Analysis of a new bin packing algorithm and its application to scheduling*, Computer Science Technical Report, Texas A&M University, College Station, Texas.

[4] D. K. FRIESEN AND M. A. LANGSTON, *Evaluation of a multifit-based scheduling algorithm*, J. Algorithms, 7 (1986), pp. 35–59.

[5] M. R. GAREY AND D. S. JOHNSON, *A* 71/60 *theorem for bin packing*, J. Complexity, 1 (1985), pp. 65–106.

[6] D. S. JOHNSON, *Near optimal bin packing algorithms*, Ph.D. thesis, M.I.T. Cambridge, MA, 1973.

[7] N. KARMARKAR AND R. M. KARP, *An efficient approximation scheme for the one-dimensional bin packing problem*, Proc. 23rd Symp. on Foundations of Computer Science, Chicago, IL, 1982, pp. 312–320.

[8] M. A. LANGSTON, *Interstage transportation planning in the deterministic flowshop environment*, Oper. Res., 35 (1987), pp. 556–564.

[9] A. C. YAO, *New algorithms for bin packing*, J. ACM, 27 (1980), pp. 207–227.

SIAM J. DISC. MATH.
Vol. 4, No. 1, pp. 80–83, February 1991

SEMIKERNELS, QUASI KERNELS, AND GRUNDY FUNCTIONS IN THE LINE DIGRAPH*

H. GALEANA-SÁNCHEZ†, L. PASTRANA RAMÍREZ‡, AND H. A. RINCÓN-MEJÍA‡

Abstract. It is proved that the number of semikernels (quasi kernels) of a digraph D is less than or equal to the number of semikernels (quasi kernels) of its line digraph $L(D)$. It is also proved that the number of Grundy functions of D is equal to the number of Grundy functions of its line digraph $L(D)$ (in the case where every vertex of D has indegree at least one).

Key words. Grundy function, kernel, line digraph, quasi kernel, semikernel

AMS(MOS) subject classification. 05C20

1. Introduction. For general concepts we refer the reader to [1]. Let $D = (X, U)$ be a digraph (also we denote $X = V(D)$ and $U = A(D)$). A set $K \subseteq X$ is said to be a kernel if it is both independent (a vertex in K has no successor in K) and absorbing (a vertex not in K has a successor in K).

This concept was introduced by Von Neumann [10] and it has found many applications [1, p. 304], [2]. Several authors have been investigating sufficient conditions for the existence of kernels in digraphs, namely, Von Neumann and Morgenstern [9], Richardson [11], Duchet and Meyniel [4], [5], and Galeana-Sánchez and Neumann-Lara [7].

In [8] Harminc proved that the number of kernels of a digraph is equal to the number of kernels in its line digraph. In this paper we find similar relations for concepts nearly related to the concept of kernel, and we survey the theorems relating these concepts.

DEFINITION 1.1 [10]. A semikernel S of D is an independent set of vertices such that for every $z \in (V(D) - S)$ for which there exists a Sz-arc there also exists an zS-arc.

DEFINITION 1.2 [3]. A quasi kernel Q of D is an independent set of vertices such that $X = Q \cup \Gamma^{-}(Q) \cup \Gamma^{-}(\Gamma^{-}(Q))$ (where for any $A \subseteq X$, $\Gamma^{-}(A) = \{x \in X \mid x \text{ has a successor in } A\}$).

DEFINITION 1.3 [1, p. 312]. A nonnegative integer function $g(x)$ is called a Grundy function of D if, for every vertex x, $g(x)$ is the smallest nonnegative integer which does not belong to the set $\{g(y) \mid y \in \Gamma^{+}(x)\}$.

This concept, originated by Grundy for digraphs without directed cycles, was extended by Berge and Schützenberger.

The Grundy function can also be defined as a function $g(x)$ such that

(1) $g(x) = k > 0$ implies that for each $0 \leqq j < k$ there is a $y \in \Gamma^{+}(x)$ with $g(y) = j$.

(2) $g(x) = k$ implies that each $y \in \Gamma^{+}(x)$ satisfies $g(y) \neq k$.

THEOREM 1.1 [3]. *Every finite digraph has a quasi kernel. A generalization of this theorem was obtained by Duchet, Hamidoune, and Meyniel* [6].

THEOREM 1.2 [10]. *If D is a digraph such that every induced subdigraph has a nonempty semikernel then D has a kernel.*

THEOREM 1.3 [1, p. 313]. *If D is a digraph such that every induced subdigraph has a kernel then D possesses a Grundy function.*

* Received by the editors April 5, 1989; accepted for publication December 18, 1989.

† Instituto de Matemáticas de la Universidad Nacional Autónoma de México, México, D.F.

‡ Departamento de Matemáticas de la Facultad de Ciencias de la Universidad Nacional Autónoma de México, México, D.F.

COROLLARY 1.1. *If D is a digraph such that every induced subdigraph has a nonempty semikernel then D possesses a Grundy function.*

2. Semikernels, quasi kernels, and Grundy functions in the line digraph.

DEFINITION 2.1. The line digraph of $D = (X, U)$ is the digraph $L(D) = (U, W)$ (we also denote $U = V(L(D))$ and $W = A(L(D))$) with set of vertices the set of arcs of D, and for any $h, k \in U$ there is $(h, k) \in W$ if and only if the corresponding arcs h, k induce a directed path in D, i.e., the terminal endpoint of h is the initial endpoint of k. In what follows we denote the arc $h = (u, v) \in D$ and the vertex h in $L(D)$ by the same symbol. If H is a set of arcs in D, it is also a set of vertices of $L(D)$. When we want to emphasize our interest in H as a set of vertices of $L(D)$ we use the symbol H_L instead of H.

DEFINITION 2.2 [8]. Let $D = (X, U)$ be a digraph. We denote by $\mathcal{P}(X)$ the set of all subsets of the set X, and $f\colon \mathcal{P}(X) \to \mathcal{P}(U)$ will denote the function defined as follows: for each $Z \subseteq X, f(Z) = \{(u, x) \in U \mid x \in Z\}$.

LEMMA 2.1 [8]. *If $Z \subseteq X$ is an independent set of D then $f(Z)_L$ is an independent set in $L(D)$.*

THEOREM 2.1. *If D is a digraph such that every vertex has indegree at least one, then the number of semikernels of D is less than or equal to the number of semikernels of its line digraph $L(D)$.*

Proof. Let $\mathcal{S}$ be the set of all semikernels of D and $\mathcal{S}_1$ be the set of all semikernels of $L(D)$. First we will prove that if S is a semikernel of D then $f(S)_L$ is a semikernel of $L(D)$. Let S be a semikernel of D. It follows from Lemma 2.1 that $f(S)_L$ is an independent set. Let $(s, x) \in W$ be with $s \in f(S)_L$, then in D we have $\{s = (s_1, s_2), x = (s_2, t)\} \subseteq U$, $s_2 \in S$, and since S is a semikernel of D there exists $s_3 \in S$ such that $(t, s_3) \in A(D)$ and then $y = (t, s_3) \in f(S)_L$ with $(x, y) \in A(L(D))$. We will prove that $f'\colon \mathcal{S} \to \mathcal{S}_1$, where f' is the restriction of f to $\mathcal{S}$, is an injective function. Let $S_1, S_2 \in \mathcal{S}$ and $S_1 \neq S_2$. Let us suppose, e.g., that $S_1 - S_2 \neq \varnothing$. Let $v \in S_1 - S_2$. Since indegree of v is at least one, there exists $(x, v) \in A(D)$. Clearly, $(x, v) \in (f(S_1) - f(S_2))$. □

THEOREM 2.2. *If D is a digraph such that every vertex has indegree at least one then the number of quasi kernels of D is less than or equal to the number of quasi kernels of its line digraph $L(D)$.*

Proof. Let Q be the set of all quasi kernels of D and Q_1 the set of all the quasi kernels of $L(D)$. First we will prove that if Q is a quasi kernel of D, then $f(Q)_L$ is a quasi kernel of $L(D)$. Let Q be a quasi kernel of D. It follows from Definition 1.2 and Lemma 1.1 that $f(Q)_L$ is an independent set of $V(L(D))$. Let $x \in (V(L(D)) - f(Q)_L)$; then $x = (x_1, x_2) \in A(D)$ and since $x \notin f(Q)_L$ it follows from Definition 2.2 that $x_2 \in (V(D) - Q)$, and there exists a directed path from x_2 to Q of length at most two. We will analyze the two possible cases:

Case 1. There exists a directed path from x_2 to Q of length one. Let $T = (x_2, u)$ be such a path, then $u \in Q$, $y = (x_2, u) \in f(Q)_L$, and $(x, y) \in A(L(D))$.

Case 2. There exists a directed path from x_2 to Q of length two. Let $T = (x_2, u, w)$ be such a path, then $w \in Q$, $y = (u, w) \in f(Q)_L$, and denoting $z = (x_2, u)$ we have that $T' = (x, z, y)$ is a directed path contained in $L(D)$ with $y \in f(Q)_L$.

In any case we have that there exists a directed path from x to $f(Q)_L$ in $L(D)$ of length at most two, so $f(Q)_L$ is a quasi kernel of $L(D)$.

Now, we will prove that $f''\colon Q \to Q_1$, the restriction of f to Q, is an injective function. Let Q_1 and $Q_2 \in Q$ be such that $Q_1 \neq Q_2$. Let us suppose, e.g., that $Q_1 - Q_2 \neq \varnothing$, and $v \in (Q_1 - Q_2)$. Since indegree of v is at least one, there exists $(x, v) \in F(D)$, clearly $(x, v) \in (f(Q_1) - f(Q_2))$ and then $f(Q_1)_L \neq f(Q_2)_L$. □

Remark 2.1. The hypothesis that each vertex has indegree at least one cannot be omitted in Theorems 2.1 and 2.2. It suffices to consider D with $V(D) = \{u_1, u_2, u_3\}$ and $F(D) = \{(u_1, u_2), (u_2, u_3)\}$.

Remark 2.2. For each $n \in \mathbb{N}$ let us define the digraph D_n as follows: $V(D_n) = \{u, v, w_1, \cdots, w_n\}$, $F(D_n) = \{(u, w_i), (w_i, v) | i \in \{1, \cdots, n\}\}$. The number of semikernels of D_n is two and the number of semikernels of $L(D_n)$ is $2^n - 1$.

Remark 2.3. Let K_3^* to be the complete symmetric directed graph with 3 vertices and H_n the digraph obtained by taking n mutually disjoint copies of K_3^*. The number of quasi kernels of $L(H_n)$ minus the number of quasi kernels of H_n is at least n.

LEMMA 2.2. *Let D be a digraph and $x_0 \in V(D)$. If f_1 and f_2 are Grundy functions of D such that for every $y \in \Gamma^+(x_0)$, $f_1(y) = f_2(y)$ then $f_1(x_0) = f_2(x_0)$.*

Proof. The proof follows directly from Definition 1.3. □

THEOREM 2.3. *If D is a digraph such that each vertex has indegree at least one, then the number of Grundy functions of D is equal to the number of Grundy functions of its line digraph $L(D)$.*

Proof. Let us suppose that f: $V(D) \to \mathbb{N} \cup \{0\}$ is a Grundy function of D and denote $f_L : V(L(D)) \to \mathbb{N} \cup \{0\}$ the function defined as follows: $f_L(x) = f(x_2)$ for each $x = (x_1, x_2) \in V(L(D))$.

OBSERVATION 2.1. *f_L is a Grundy function of $L(D)$.*

(1) $f_L(x) = k > 0$ implies that for each $0 \leqq j < k$, there is a $y \in \Gamma^+_{L(D)}(x)$ with $f_L(y) = j$.

Suppose that $f_L(x) = k > 0$ and $0 \leqq j < k$, then $x = (x_1, x_2) \in A(D)$ and $f(x_2) = k > 0$. Since f is a Grundy function of D and $0 \leqq j < k$, there exists $x_3 \in \Gamma^+_D(x_2)$ such that $f(x_3) = j$ and then $y = (x_2, x_3) \in A(D)$, $(x, y) \in A(L(D))$, and $f_L(y) = j$; i.e., $y \in \Gamma^+_{L(D)}(x)$, with $f_L(y) = j$.

(2) $f_L(x) = k$ implies that each $y \in \Gamma^+_{L(D)}(x)$ satisfies $f_L(y) \neq k$.

Suppose that $f_L(x) = k$ and $y \in \Gamma^+_{L(D)}(x)$; then $x = (x_1, x_2) \in A(D)$, $y = (x_2, x_3) \in A(D)$, $f(x_2) = k$, and $x_3 \in \Gamma^+_D(x_2)$ and since f is a Grundy function of D, it follows that $f(x_3) \neq k$ and $f_L(y) = f(x_3) \neq k$.

OBSERVATION 2.2. *If f^1, f^2 are Grundy functions of D such that $f^1 \neq f^2$ then $f^1_L \neq f^2_L$.*

Suppose that $f^1_L = f^2_L$ and that $x_0 \in V(D)$. Since indegree of x_0 is at least one then there exists an arc $(z, x_0) \in A(D)$. By the hypothesis we have that $f^1_L((z, x_0)) = f^2_L((z, x_0))$, i.e., $f^1(x_0) = f^2(x_0)$.

Let us suppose that g: $V(L(D)) \to \mathbb{N} \cup \{0\}$ is a Grundy function of $L(D)$ and let us denote g_D: $V(D) \to \mathbb{N} \cup \{0\}$ the function defined as follows: for each $x_0 \in V(D)$ let $f = (y, x_0) \in A(D)$ any arc of D with terminal endpoint x_0 (the hypothesis of Theorem 2.3 implies that there exists at least one such arc) and define $g_D(x_0) = g(f)$.

OBSERVATION 2.3. *g_D is well defined.*

Let $x_0 \in V(D)$ and suppose that $f_1 = (y_1, x_0)$ and $f_2 = (y_2, x_0) \in A(D)$.
If $\Gamma^+_D(x_0) = \emptyset$ then $\Gamma^+_{L(D)}(f_1) = \Gamma^+_{L(D)}(f_2) = \emptyset$ and Definition 1.3 implies $g(f_1) = g(f_2) = 0$.

If $\Gamma^+_D(x_0) \neq \emptyset$ then Definition 2.1 implies that $\Gamma^+_{L(D)}(f_1) = \Gamma^+_{L(D)}(f_2)$ and it follows from Definition 1.3 that $g(f_1) = g(f_2)$.

OBSERVATION 2.4. *g_D is a Grundy function of D.*

(1) $g_D(x) = k > 0$ implies that for each $0 \leqq j < k$, there exists a $y \in \Gamma^+_{L(D)}(x)$ with $g_D(y) = j$.

Suppose that $g_D(x) = k > 0$ and $0 \leqq j < k$; the hypothesis and the definition of g_D imply that there exists $f = (z, x) \in A(D)$ with $g(f) = k > 0$ and since g is a Grundy function of $L(D)$ there exists $f' \in \Gamma^+_{L(D)}(f)$ such that $g(f') = j, f' = (x, w)$ for some $w \in V(D)$, $g_D(w) = g(f') = j$. Clearly $w \in \Gamma^+_D(x)$ and take $y = w$.

(2) $g_D(x) = k$ implies that each $y \in \Gamma^+_D(x)$ satisfies $g_D(y) \neq k$.

Suppose that $g_D(x) = k$; then there exists $f = (z, x) \in A(D)$ such that $g(f) = k$ and $y \in \Gamma^+_D(x)$, so $(x, y) \in \Gamma^+_{L(D)}(f)$. Since g is a Grundy function of $L(D)$ it follows that $g((x, y)) \neq k$ and $g_D(y) = g((x, y)) \neq k$.

OBSERVATION 2.5. *If g^1, g^2 are Grundy functions of $L(D)$ such that $g^1 \neq g^2$ then $g^1_D \neq g^2_D$.*

Suppose that $g^1_D = g^2_D$ and let $f = (x, y) \in A(D)$; then $g^1_D(y) = g^2_D(y)$. The definition of g^1_D, g^2_D implies $g^1(f) = g^2(f)$.

REFERENCES

[1] C. BERGE, *Graphs*, North Holland, Amsterdam, New York, 1985.

[2] C. BERGE AND A. RAMACHANDRA RAO, *A combinatorial problem in logic*, Discrete Math., 17 (1977), pp. 23–26.

[3] V. CHVÁTAL AND L. LOVÁSZ, *Every directed graph has a semi-kernel*, in Hypergraph Seminar, Lecture Notes in Math. 411, Springer-Verlag, Berlin, 1974, p. 175.

[4] P. DUCHET, *A sufficient condition for a digraph to be kernel-perfect*, J. Graph Theory, 11 (1987), pp. 81–85.

[5] P. DUCHET AND H. MEYNIEL, *Une généralization du théorème de Richardson sur l'existence de noyaux dans les graphes orientés*, Discrete Math., 43 (1983), pp. 21–27.

[6] P. DUCHET, Y. O. HAMIDOUNE, AND H. MEYNIEL, *Sur les quasi-noyaux d'un graphe*, Discrete Math., 65 (1987), pp. 231–235.

[7] H. GALEANA-SÁNCHEZ AND V. NEUMANN-LARA, *On kernel and semikernels of digraphs*, Discrete Math., 48 (1984), pp. 67–76.

[8] M. HARMINC, *Solutions and kernels of a directed graph*, Math. Slovaca, 32 (1982), pp. 263–267.

[9] J. VON NEUMANN AND O. MORGENSTERN, *Theory of games and economic behavior*, Princeton University Press, Princeton, NJ, 1944.

[10] V. NEUMANN-LARA, *Seminúcleos en una digráfica*, Anales del Instituto de Matemáticas de la Universidad Nacional Autónoma de México, México, 11 (1971), pp. 55–62.

[11] M. RICHARDSON, *Solutions of irreflexible relations*, Ann. Math., 58 (1953), pp. 573–580.

SIAM J. DISC. MATH.
Vol. 4, No. 1, pp. 84–98, February 1991

009

WEAK THREE-LINKING IN EULERIAN DIGRAPHS*

T. IBARAKI† AND S. POLJAK‡

Abstract. Let G be an Eulerian digraph, and a, b, c an ordered triple of its vertices. A polynomial time algorithm of $O(e + n^2)$ time is presented to decide whether G contains three arc disjoint ab-, bc-, and ca-paths, where e and n are the numbers of arcs and vertices, respectively. The algorithm is based on a structural characterization of minimal infeasible instances of the problem.

Key words. Eulerian digraph, disjoint paths, planar graph, polynomial time algorithm, weak three-linking

AMS(MOS) subject classifications. 05C20, 05C38, 05C45, 90B10

1. Introduction. Let $G = (V, E)$ be a digraph and (s_i, t_i), $i = 1, \cdots, k$, be ordered pairs of terminals. A collection of k arc disjoint s_it_i-paths is called a *weak* $(s_1t_1, \cdots, s_kt_k)$*-linking*. When the number of pairs of terminals is restricted to a constant k, the problem of existence of a weak $(s_1t_1, \cdots, s_kt_k)$-linking is called the *weak k-linking* problem. (The term *linking* is used for a collection of vertex disjoint paths, but we do not consider this problem here.)

The digraph $H = (V, F)$ where $F = \{t_is_i: i = 1, \cdots, k\}$ is called the *demand* graph. In this paper we solve the weak 3-linking problem in case $G + H = (V, E \cup F)$ is Eulerian.

Let us recall some related known results on weak linking problems in digraphs. Fortune, Hopcroft, and Wyllie [2] proved that the weak 2-linking problem is NP-complete for a general digraph. However, there are some classes for which the problem is polynomially solvable. If k is fixed, weak k-linking is polynomial for acyclic digraphs [2], and weak 2-linking is polynomial for tournaments [1]. Another result on weak linking in digraphs is due to Frank [3]. He characterized those demand graphs H for which a condition called "directed cut criterion" is necessary and sufficient in case $G + H$ is Eulerian. Another work where the Eulerian condition is involved is [6], where the integral multicommodity flow problem is solved for a class of acyclic planar networks. A recent survey on linking problems, both in directed and undirected graphs, can be found in [4].

To solve the weak 3-linking problem for $G + H$ Eulerian, it is sufficient to consider only the special case when $s_1 = t_3$, $s_2 = t_1$, and $s_3 = t_2$ and G is Eulerian. (Let G and (s_i, t_i), $i = 1, 2, 3$, be an instance of the weak 3-linking problem. Construct G' by adding three new vertices a, b, and c, and arcs as_1, t_1b, bs_2, t_2c, cs_3, and t_3a. Then G has three arc disjoint s_it_i-paths if and only if G' has three arc disjoint ab-, bc-, and ca-paths.)

Let G be an Eulerian digraph, and a, b, c an ordered triple of its vertices. We say that an instance $(G; a, b, c)$ is *feasible*, if there are three arc disjoint ab-, bc-, and ca-paths. Otherwise the instance is *infeasible*. The specified vertices a, b, and c are called the *terminals*. We say that an instance is *minimal infeasible* if it is infeasible, but after contraction of any arc, at least one of whose head and tail are not in $\{a, b, c\}$, we get a graph G' such that $(G'; a, b, c)$ is feasible. We prove the following theorems.

THEOREM 1.1. *Let* $(G; a, b, c)$ *be a minimal infeasible instance. Then G has the following properties*:

* Received by the editors May 1, 1989; accepted for publication (in revised form) March 27, 1990.

† Department of Applied Mathematics and Physics, Faculty of Engineering, Kyoto University, Kyoto 606, Japan.

‡ Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University, Malostranské n. 25, 118 00 Praha 1, Czechoslovakia.
This work was done while the author was visiting Kyoto University and was supported by a grant from the Kyoto University 70th Anniversary Memorial Foundation.

(i) *G is planar 2-connected. The terminals have degree 2, and all other vertices have degree 4.*

(ii) *G has a planar representation in which every face is a directed cycle,* (*or equivalently, the arcs incident to a vertex are alternatively oriented out and in*), *and the terminals lie on a common face which goes through them in the order c, b,* and *a.*

Conversely, any instance (G; a, b, c) *satisfying* (i) *and* (ii) *is infeasible* (*but not necessarily minimal*).

An example of a minimal infeasible instance is given in Fig. 1.

THEOREM 1.2. *There is a polynomial time algorithm to decide whether an instance* (G; a, b, c) *is feasible or infeasible.*

Theorem 1.1 will be proved in the next section where a more general problem, with possibly more terminals, is considered. We introduce a notion of an irreducible infeasible instance and show that every such instance has a certain decomposition, which is called a *series*. The series decomposition of an irreducible instance enables us to reduce the question of feasibility to a collection of subproblems. A polynomial time algorithm, whose existence is stated in Theorem 1.2, will be presented in § 3. There we also show that the weak linking problem, when the number k of terminal pairs is a part of the input, is NP-complete for $G + H$ Eulerian.

We conclude this section with some necessary notation.

Notation. A digraph $G = (V, E)$ consists of a set V of vertices and a set E of directed arcs. For technical reasons, we allow multiple parallel arcs, but loops are excluded. We recall that a digraph is *Eulerian* if it is connected, and the outdegree and indegree of each vertex are equal.

Under a *path* or a *cycle*, we always understand a *directed* path or cycle. Repetition of arcs is not allowed, but we do not require all vertices of a path or a cycle to be distinct. A cycle that visits every arc exactly once is called Eulerian. A path from x to y is called an xy-path. If P is a path, and x and y are two of its vertices, such that x precedes y on P, we denote by P_{xy} the *segment* of P starting at x and terminating at y. Similarly, if C is a cycle and x and y are two of its vertices, then C_{xy} is the part of the cycle from x to y. If $P_1, P_2, \cdots, P_k$ is a collection of arc disjoint paths such that the last vertex of P_i coincides with the initial vertex of P_{i+1} for each $i = 1, \cdots, k - 1$, we denote by $P =$

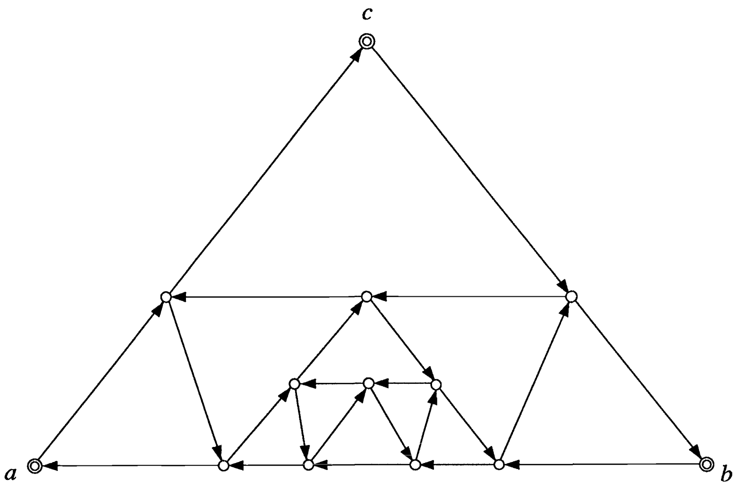


FIG. 1. *A minimal infeasible instance* (G; a, b, c).

$(P_1, P_2, \cdots, P_k)$ the concatenation of the paths. In the following discussion, graph G, path P, or cycle C may sometimes be treated either as a vertex set or an arc set, as far as its meaning is unambiguous from the context. If it is necessary to specify, we use $E(G)$ to mean the arc set, and $V(G)$ to mean the vertex set.

For a subset S of vertices, $\delta^+(S)$ denote the set of arcs from S to $V \backslash S$, $\delta^-(S)$ the set of arcs from $V \backslash S$ to S, and $\delta S = \delta^-(S) \cup \delta^+(S)$. If G is Eulerian, then $|\delta^+(S)| = |\delta^-(S)|$ for every S, where $|A|$ of a set A denotes its cardinality. A set S is called a *k-cut* if $|\delta S|$ is k. (Since we consider Eulerian digraphs only, k must always be even.)

Given two disjoint sets U, $W \subset V$, we denote by $\lambda(U, W)$ the maximum number of arc disjoint paths from U to W. A set S satisfying $S \supset U$, $V \backslash S \supset W$, and $\delta^-(S) = \lambda(U, W)$ is called a minimum (U, W)-cut. It is well known that the minimum (U, W) cuts are closed under union and intersection. Hence, among all minimum (U, W)-cuts, there exists one *largest* cut S with the property that $S \supset S'$ for any other minimum (U, W)-cut S'. The largest minimum (U, W)-cut can be found as a by-product of the Ford–Fulkerson algorithm of maximum flow.

Some further notion, such as degree, planarity and edge, and vertex connectivity properties refers to the unoriented graph $\bar{G}$ obtained from a digraph G by forgetting the arc orientations. In particular, the *degree* of a vertex is the sum of its out- and indegrees, and a digraph is called *connected* if $\bar{G}$ is connected. A subset $U \subset V$ is a *vertex* cut if $G \backslash U$ has more connected components than G. A vertex v is called an *articulation* if $\{v\}$ is a vertex cut. A digraph is *k-connected* if it does not have a vertex cut of size $< k$. A maximal (with respect to set inclusion) 2-connected subgraph of a graph is called a *block*.

2. Structural characterization of infeasibility. In this section we formulate and solve a more general problem. Let $G = (V, E)$ be an Eulerian digraph and $X = (x_1, \cdots, x_m)$, $m \geqq 3$, be an ordered m-tuple of its vertices, which are called *terminals*. We also write $x < x'$ for terminals $x = x_i$ and $x' = x_j$ if $i < j$. We say that an instance $(G; X)$ is *feasible* if there is a triple x_i, x_j, and x_k of terminals such that $x_i < x_j < x_k$ and $(G; x_i, x_j, x_k)$ is feasible in the sense of the previous section. Such a triple is called a *feasible* triple, and a cycle through these terminals is called a *feasible* cycle. Equivalently, $(G; X)$ is infeasible if every Eulerian cycle goes through the terminals in the order $x_m, \cdots, x_1$ (up to a cyclic shift). It follows that a feasible instance $(G; X)$ contains a feasible cycle through x for any terminal $x \in X$.

Let us also remark that, to establish feasibility, it is sufficient to find two arc disjoint paths, say x_ix_j- and x_jx_k-paths with $x_i < x_j < x_k$ (up to a cyclic shift), because the third one always exists since G is Eulerian.

Another sufficient condition for feasibility, which will be used in the proof of Lemma 2.7 (case iiib), is the existence of a disjoint x_1x_3-path P^{13} and x_4x_2-path P^{42}. Since G is Eulerian, there exists also a pair of arc-disjoint paths from $\{x_2, x_3\}$ to $\{x_1, x_4\}$ in digraph $G \backslash E(P^{13} \cup P^{42})$, i.e., either a pair P^{34}, P^{21} or a pair P^{31}, P^{24} (where $P^{i,j}$ stands for a x_ix_j-path). In the former case, $(P^{13}, P^{34}, P^{42}, P^{21})$ forms a feasible cycle through terminals (x_1, x_3, x_4). In the latter case, $C^{13} = (P^{13}, P^{31})$ and $C^{24} = (P^{24}, P^{42})$ are two arc disjoint cycles. Let $u \in V(C^{13})$ and $v \in V(C^{24})$ be a pair of vertices for which the length of the shortest uv-path P is minimum. Then P is arc disjoint with both C^{13} and C^{24}, and since G is Eulerian, there is a vu-path P' in $G \backslash E(C^{13} \cup C^{24} \cup P)$. It is easy to check that the instance $(C^{13} \cup C^{24} \cup P \cup P'; x_1, x_2, x_3, x_4)$ is feasible for any mutual position of u and v on C^{13} and C^{24} (four possibilities), where P and P' may be empty paths. Some other configurations sufficient for feasibility are considered in the proof of Lemma 2.2.

We say that an instance $(G; X)$ is *reducible* if one of the following reductions can be performed.

(1) Let S be a 2-cut and $S \cap X = \emptyset$. Let u be the tail of the arc from $V \backslash S$ to S, and v the head of the arc from S to $V \backslash S$. Delete S and add the arc uv.

(2) Let S be a 2-cut, $|S| \geqq 2$, and S contain exactly one terminal x. Then contract S to x and delete the loops. (Terminal x becomes a vertex of degree 2.)

(3) Let S be a 4-cut such that the subgraph induced on S is connected, $|S| \geqq 2$, and $S \cap X = \emptyset$. Then contract S to a single vertex (of degree 4), and delete the loops.

LEMMA 2.1. *$(G; X)$ is feasible if and only if it is feasible after performing any of reductions* (1), (2), *or* (3).

Proof. It is obvious that if an instance is feasible, then it is feasible also after any of the reductions. The converse is not difficult to see for reductions (1) and (2). For reduction (3), it follows from the following fact. Let s and s' be the heads of the two arcs entering S, and let t and t' be the tails of the arcs leaving S. Since S is connected and G Eulerian, there are both an $(st, s't')$-linking and an $(st', s't)$-linking in S. □

We say that $(G; X)$ is *irreducible* if none of the reductions (1), (2), or (3) can be performed.

LEMMA 2.2. *Let (G, X) be an infeasible irreducible instance. Then each terminal has degree* 2, *and each nonterminal vertex has degree* 4.

Proof. Assume that the degree of some terminal, say x_1, is at least 4. We distinguish two cases.

(i) There are at least two arc disjoint paths from x_1 to $X \backslash x_1$. Let the end vertices of these two paths be x_i and x_j, respectively. Since G is Eulerian, these paths can be completed to two arc disjoint cycles C and C' containing $\{x_1, x_i\}$ and $\{x_1, x_j\}$, respectively. If $x_i \neq x_j$ and $i < j$, then $(x_1, C_{x_1 x_i}, x_i, C_{x_i x_1}, C'_{x_1 x_j}, x_j, C'_{x_j x_1})$ is a feasible cycle through (x_1, x_i, x_j). If $i = j$, let x_k be a terminal distinct from x_1 and x_i. There must exist some third cycle C'' containing x_k and a vertex u of $V(C \cup C')$, and such that C'' is arc disjoint with $C \cup C'$. It is easy to show feasibility for $\{x_1, x_i, x_k)$ using arcs of $C \cup C' \cup C''$.

(ii) If the assumption of (i) does not hold, there is, by the Menger Theorem, a set S with $S \cap X = \{x_1\}$ and $|\delta S| = 2$. Hence reduction (2) can be performed.

Let u be a nonterminal. Clearly, the degree of u is at least 4, otherwise reduction (1) can be performed for $S = \{u\}$. Assume that the degree of u is at least 6. We again apply the Menger Theorem. There are either three arc disjoint paths from u to X, or there is a set S containing u, $S \cap X = \emptyset$ and $|\delta S| \leqq 4$. Clearly, $(G; X)$ is reducible in the latter case. In the former case, the three paths from u lead to distinct members of X, say x_1, x_2, and x_3, since the degrees of terminals are 2. These three paths can be completed to three arc disjoint cycles. It is then easy to see that $(G; X)$ is feasible. □

Let $(G; X)$ be an irreducible instance and let y be a nonterminal vertex of degree 4 which is an articulation of G. Then $G \backslash y$ has exactly two connected components which we denote by U_1 and U_2. Let us say that articulation y *well splits* the terminals, if $X \cap U_1 = (x_j, x_{j+1}, \cdots, x_{k-1})$ and $X \cap U_2 = (x_k, \cdots, x_m, x_1, \cdots, x_{j-1})$ for some $1 \leqq j < k \leqq m$ (the roles of U_1 and U_2 can be interchanged). In this case we define the *1-decomposition* of $(G; X)$ at y as the following pair of instances $(G_1; X_1)$ and $(G_2; X_2)$. For $i = 1, 2$, let G_i be the subdigraph of G induced on vertex set $V_i = U_i \cup \{y\}$ and $X_i = (X \cap U_i) \cup \{y\}$. The terminals in X_i are ordered as in X, and y is added as the last one. (Observe that both G_i are Eulerian and $|X_i| \geqq 3$ (otherwise $|X \cap U_i| \leqq 1$ and G is reducible). Hence the instances $(G_i; X_i)$, $i = 1, 2$ are correctly defined.

LEMMA 2.3. *Let $(G; X)$ be an irreducible instance such that the degrees of all terminals and nonterminals are* 2 *and* 4, *respectively. Let y be an articulation of G. Then $(G; X)$ is infeasible if and only if*

(i) *articulation y well splits the terminals*; *and*

(ii) *$(G_i; X_i)$, $i = 1, 2$, of the* 1*-decomposition of (G, X) at y are both infeasible.*

Proof. Assume that $(G; X)$ is infeasible and let C be an arbitrary Eulerian cycle of G. If we start at terminal x_m, cycle C goes through the terminals in the unique order x_m, $x_{m-1}, \cdots, x_1$ since $(G; X)$ is infeasible. Vertex y is entered by C twice, so the terminals are well split. It is easy to see that feasibility of either of $(G_i; X_i)$, $i = 1, 2$ yields feasibility of $(G; X)$.

The converse follows by an analogous argument. □

Now we present a rather technical lemma which will be used later in the proof of Lemma 2.7.

LEMMA 2.4. *Let F' and F be the digraphs given by either Fig.* 2(a) *or Fig.* 2(b). *Then the existence of weak (s_1t_1, s_2t_2)-linking in F' implies the existence of weak linking in F for the same pairs of terminals, for every choice of (not necessarily distinct) terminals $s_1, s_2 \in S$ and $t_1, t_2 \in T$, where S and T are defined as follows.*

$$S = \{a_1, a_2, y\} \textit{ and } T = \{a_2, y'\} \textit{ for Fig. } 2(\mathrm{a}),$$

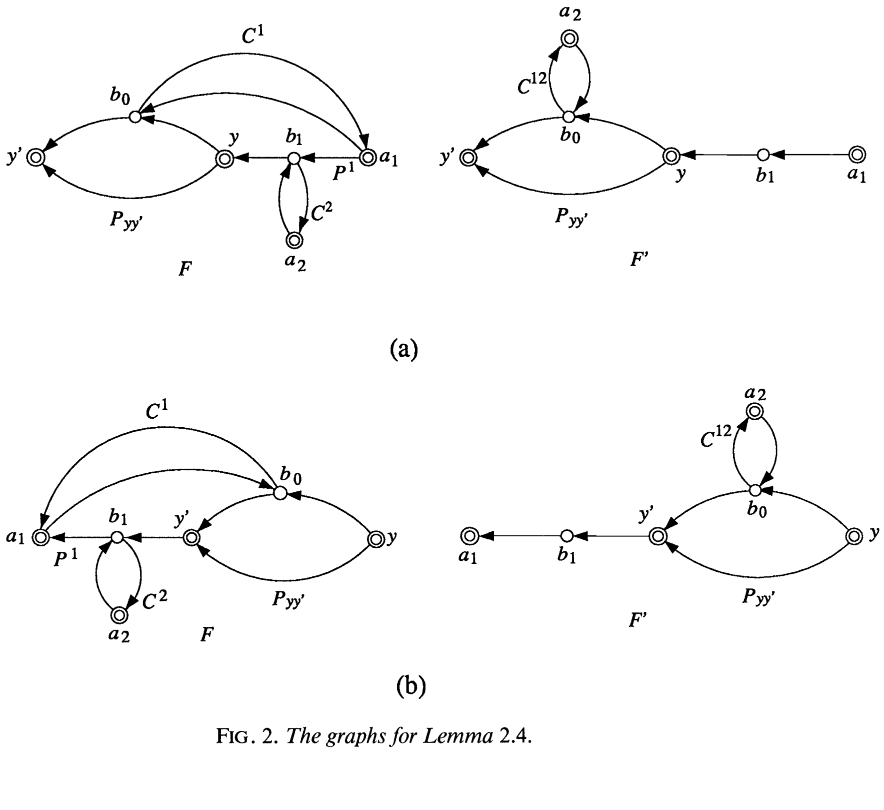
$$S = \{a_2, y\} \textit{ and } T = \{a_1, a_2, y'\} \textit{ for Fig. } 2(\mathrm{b}).$$

Proof. Let F' and F be defined by Fig. 2(a). Then F' has weak (s_1t_1, s_2t_2)-linking for the following seven choices of terminals (s_1t_1, s_2t_2): (a_1y', yy'), (a_1y', ya_2), (a_1a_2, yy'), (a_1y', a_2y'), (a_1a_2, a_2y'), (yy', a_2y'), and (ya_2, a_2y'). It is easy to check that also F has weak (s_1t_1, s_2t_2)-linking in each of these cases. The proof is analogous when F and F' are given by Fig. 2(b). □

Our main result is the following description of irreducible infeasible instances.

THEOREM 2.5. *Let $(G; X)$ be an infeasible irreducible instance. Then*

(i) *G is planar, all terminals have degree* 2 *and all the other vertices have degree* 4; *and*



(a)

(b)

FIG. 2. *The graphs for Lemma* 2.4.

(ii) *G has a planar representation in which every face is a directed cycle* (*or equivalently, the arcs incident to a vertex are alternatively oriented out and in*), *and the terminals lie on the outer face, which is oriented against the order of terminals.*

Conversely, every such instance is infeasible (*but not necessarily irreducible*).

Proof. We show first that every instance $(G; X)$ satisfying the conditions of the theorem is infeasible. We may assume that G is drawn so that the terminals are on the boundary, which is oriented against their order. Assume there is a feasible triple, say x_1, x_2, x_3, of terminals. Let P and P' be arc disjoint x_1x_2- and x_2x_3-paths. Let y and y' be the predecessor and successor of x_2 on the boundary (see Fig. 3). Since x_2 has degree 2, y and y' must lie on the paths P and P', respectively. The boundary of G divides the plane into two regions: inner and outer. The inner region is split by path P' into regions R_1 and R_2, where R_1 contains vertex x_1, and R_2 contains vertices x_2 and y. Let u be the vertex of $V(P) \cap V(P')$ such that P_{x_1u} lies entirely in R_1, and the length of P_{x_1u} is maximum. Then the arc of P entering and leaving u lies in R_1 and R_2, respectively. The degree of u is 4. But then the two arcs leaving u are neighbouring, which contradicts our assumption that each face of G is a directed cycle.

To prove the converse, we need to introduce some additional notation. Let $P = (y_0 = x_1, y_1, \cdots, y_k, y_{k+1} = x_2)$ be an x_1x_2-path. The graph $G\backslash E(P)$ may split into several components. A *component* will always mean a connected component of $G\backslash E(P)$. The collection of these components will be called a *decomposition* given by P. We will classify the components according to the position of terminals. The component containing terminals x_1 and x_2, which belong to the same component because G is Eulerian, will

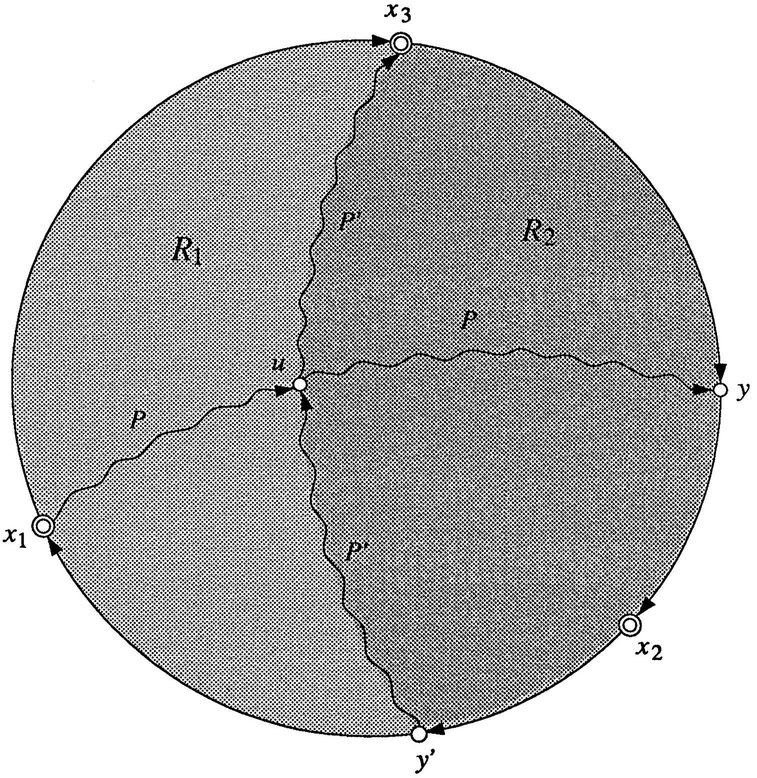


FIG. 3. *The proof of infeasibility in Theorem* 2.5.

be called *basic*, and denoted by B. The components distinct from B that contain some terminals will be called *important*. Finally, the components without terminals will be called *plain*. Also note that each component, different from the basic component, is Eulerian.

We say that a component K_2 is *surrounded* by a component K_1 (or that K_1 *surrounds* K_2) if there are three vertices $v_1 = y_{i_1}$, $v_2 = y_{i_2}$, $v_3 = y_{i_3}$ of P, $i_1 < i_2 < i_3$, such that v_1, $v_3 \in K_1$ and $v_2 \in K_2$. We say that an x_1x_2-path P creates a *series* decomposition of $(G; X)$ if the following are satisfied.

(i) There are no plain components in the decomposition.

(ii) The important components do not surround each other (but they may surround or be surrounded by the basic component).

(iii) The basic component contains only terminals x_1 and x_2.

(iv) Let K_1, K_2, $\cdots$ K_p be the important components in the order in which they are visited by P (cf. (ii)). Then the terminals are distributed in the important components *against* their order, i.e., if $x_i \in K_r$, $x_j \in K_s$ for some $2 < i < j$, then $r \geqq s$.

A series decomposition is depicted in Fig. 4.

An instance $(G; X)$ is called *trivial* if every vertex of G is a terminal. It is not difficult to see that trivial infeasible instances are just simple directed cycles if G is 2-connected.

Now we prove two lemmas before completing the proof of Theorem 2.5.

LEMMA 2.6. *Let $(G; X)$ be an infeasible irreducible instance. Then it has a series decomposition for some x_1x_2-path P.*

Proof. Let us start with an arbitrary x_1x_2-path P. We will modify P until we get a series decomposition.

(i) Let us denote by $\mathscr{S}(P)$ the set of vertices that are in plain components. We must find a decomposition satisfying, among others, $\mathscr{S}(P) = \varnothing$. For this assume that P is chosen so that $|\mathscr{S}(P)|$ is minimum. Assume that a plain component K surrounds some nonplain (i.e., basic or important) component K'. Let v_1, v_2, and v_3 be three vertices of P that lie on P in this order and such that v_1, $v_3 \in K$ and $v_2 \in K'$. Let Q be a v_1v_3-path in K. Define a new x_1x_2-path P' by $P' = (P_{x_1v_1}, Q, P_{v_3x_2})$, and consider the decomposition given by P'. Obviously, we get $\mathscr{S}(P') \subset \mathscr{S}(P) \setminus \{v_1, v_3\}$ which contradicts our assumption on the minimality of $\mathscr{S}(P)$. Therefore assume that a plain component

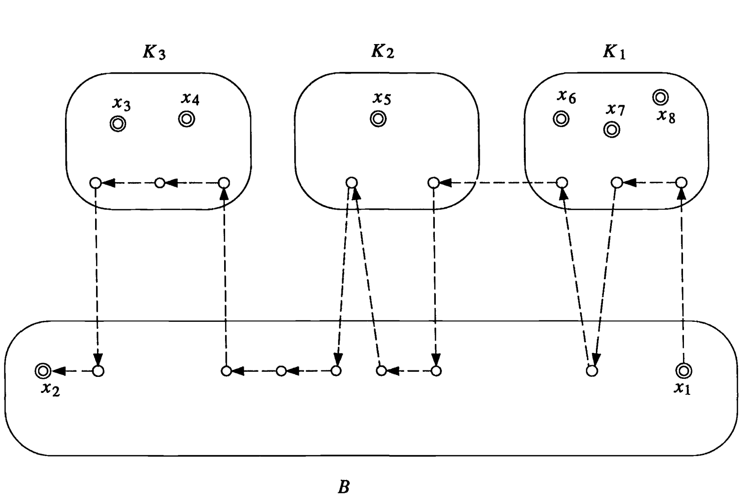


FIG. 4. *A series decomposition (broken arcs denote x_1x_2-path P).*

K does not surround any nonplain component but may surround some other plain components. Let $\mathscr{R}$ be the minimum set of components defined as follows: (a) $K \in \mathscr{R}$ and (b) $K'' \in \mathscr{R}$ whenever K'' is surrounded by some $K' \in \mathscr{R}$. Then $\mathscr{R}$ contains only plain components and $S = \bigcup \mathscr{R}$ is a 2-cut (where δS consists of two arcs of P), which contradicts the irreducibility (cf. property (1)) of $(G; X)$. Thus property (i) of series decomposition is established.

(ii) Assume there are two important components K and K' such that K surrounds K'. Let v_1, v_2, and v_3 be vertices of P as in the definition of surrounding. Let x and x' be terminals contained in K and K', respectively, and C and C' be Eulerian cycles in K and K', respectively. Consider the paths $(P_{x_1v_1}, C, P_{v_1v_2}, C', P_{v_2x_2})$ and $(P_{x_1v_2}, C', P_{v_2v_3}, C, P_{v_3x_2})$. The former path traverses terminals in order x_1, x, x' and the latter one in order x_1, x', x. This shows that instance $(G; X)$ is feasible.

(iii) Assume that the basic component B contains some terminal x_i, $i > 2$. Let Q be a x_2x_i path in B. Then P and Q are arc disjoint, and hence $(G; x_1, x_2, x_i)$ is feasible.

(iv) Assume that property (iv) of series decomposition does not hold. Then it is easy to see that terminals $x_i \in K_r$ and $x_j \in K_s$ can be traversed in the order x_1, x_i, x_j, which shows feasibility of $(G; X)$. □

Let $(G; X)$ be an instance in which G is 2-connected and the degree of each terminal is 2. Let $(B, K_1, \cdots, K_p)$ be a series decomposition of $(G; X)$ with respect to an x_1x_2-path P. Let us call a component K_i *trivial* if it consists of only one vertex, a terminal. Component B is defined to be *trivial* if it consists of only two vertices x_1 and x_2. If every component is trivial, then the instance $(G; X)$ trivially satisfies (i) and (ii) of Theorem 2.5 since it is a directed cycle whose every vertex is a terminal. If a component is nontrivial, then the path P must contain at least two vertices of the component (which are not terminals) by the 2-connectedness of G. The terminals are not on P because they have degree 2 and lie in a nontrivial important component.

We define a new ordered set Y_i of terminals for every nontrivial component K_i as follows. The set Y_i consists of the original terminals from X which belong to K_i in their original order, and followed by the new terminals which are all the vertices of path P belonging to K_i, ordered in the direction of P. We call each instance $(K_i; Y_i)$ a *subproblem* of the instance $(G; X)$. For the basic component B, the subproblem is defined in a slightly different way. We identify in B the terminals x_1 and x_2 into a new vertex x_0 and call the resulting digraph K_0. (Observe that K_0 is Eulerian.) The set Y_0 of terminals consists of all the vertices of P, including x_0, which lie in K_0. The order of terminals is given by the direction of P.

LEMMA 2.7. *Let $(G; X)$ be an infeasible irreducible instance, and let $(B, K_1, \cdots, K_p)$, $p \geqq 1$, be a series decomposition with respect to some x_1x_2-path. Then component K_i is either trivial, or the subproblem $(K_i; Y_i)$ is also irreducible and infeasible for every $i = 0, \cdots, p$.*

Proof. We show first that every subproblem is irreducible. For contradiction, assume that a subproblem $(K; Y)$ is reducible. We distinguish some cases according to which reduction of (1), (2), or (3) can be performed.

(i) If S is a 2- or 4-cut in K and $S \cap Y = \emptyset$, then S is also a 2- or 4-cut in G and $S \cap X = \emptyset$. Hence reduction (1) or (3) can be performed for $(G; X)$.

(ii) Assume S is a 2-cut in K and S contains exactly one terminal t from Y. If $t \notin X$, then t is a vertex of P, and S is a 4-cut in G with $S \cap X = \emptyset$. Then reduction (3) can be performed. If $t \in X$, then S is a 2-cut in G, and reduction (2) can be performed.

Further, we have to show that every subproblem is infeasible. For contradiction, assume that some $(K; Y)$ is feasible. It is not difficult to see that we may assume that a feasible triple contains at least one terminal $x \in X$, since there is a feasible cycle for any terminal from Y as stated in the first paragraph of § 2. Let C be a cycle in K which traverses a feasible triple of terminals. Again, we distinguish several cases.

(i) The feasible triple of terminals consists of three terminals from X. Then the triple is feasible already for $(G; X)$.

(ii) The feasible triple consists of two original terminals and one new terminal, say $(K; x, x', y)$ is feasible where $x, x' \in X$, $x < x'$, and y is a vertex in $P \cap K$. We recall that C is a cycle in K that traverses the terminals in the order y, x, x'. Then the paths $P_1 = (P_{x_1y}, C_{yx})$ and $P_2 = C_{xx'}$ are arc disjoint, and hence $(G; x_1, x, x')$ is feasible.

(iii) The feasible triple is of the form x, y, y' where $x \in X$ and $y, y' \in P \cap K$, $y < y'$. Here we distinguish two subcases: (iiia) $P_{yy'} \not\subset K$, and (iiib) $P_{yy'} \subset K$, where $P_{yy'}$ and K are viewed as sets of vertices.

(iiia) Denote by v a vertex of $P_{yy'} \backslash V(K)$. If K is not the basic component B, then $v \in B$. Let Q be an x_2v-path in B. Then the paths $(P_{x_1y}, C_{yy'}, P_{y'x_2})$ and $(Q_{x_2v}, P_{vy'}, C_{y'x})$ are arc disjoint, which shows that $(G; x_1, x_2, x)$ is feasible. If $K = B$, then $x = x_0$ and the existence of cycle C through x_0, y, y' means existence of an x_2x_1-path Q in B through y and y' (in that order) because terminals x_1 and x_2 have degree one in B. Denote by K' the component containing the vertex v, and let $x' \in X$ be a terminal in K', and W a cycle through v and x' in K'. Then $(Q_{x_2y}, P_{yv}, W_{vx'}, W_{x'v}, P_{vy'}, Q_{y'x_1})$ is a path that proves feasibility of $(G; x_2, x', x_1)$.

(iiib) This case is the crucial one in our analysis. Assume that K is not the basic component. (The case when K is basic is quite similar.) We recall that $P_{yy'}$ is the segment of P from y to y', and that it is entirely contained in K, and that C is a cycle through y, y' and x in K. Let us denote by S_0 the vertex set of $P_{yy'} \cup C_{yy'}$. Let $\bar{P}$ be the maximum segment of P such that it contains $P_{yy'}$ and all its vertices are in K. Let K^* be the digraph obtained from K by reversing the arcs of $C_{y'x}$, and adding the arcs of $P_{x_1y} \cap \bar{P}$ (in their original direction) and the arcs of $P_{y'x_2} \cap \bar{P}$ in the reversed direction. We define the set S as the set of vertices in K^* that are reachable from S_0 by a directed path in K^*. Fig. 5 illustrates these concepts, in which P is indicated by broken arcs, and set S by bold arcs.

We claim that either (b1) S is a 4-cut in G with $S \cap X = \varnothing$, or that (b2) $(G; X)$ is feasible.

Case (b1). Assume that $S \cap X = \varnothing$ and $S \cap P \subset \bar{P}$, where P and $\bar{P}$ are viewed as sets of vertices. We will show that S is a 4-cut in G. Since we also assume $S \cap X = \varnothing$, reduction (3) can be performed with S for $(G; X)$, which contradicts the irreducibility of $(G; X)$.

CLAIM. *The arcs of δS may only belong to either C or P.*

Let U be a connected component of $K \backslash E(C)$. Since K is Eulerian, U is Eulerian as well. Hence U is strongly connected and we have either $V(U) \subset S$ or $V(U) \cap S = \varnothing$ by the definition of S. This proves the claim.

Next we show that $|S \cap E(C)| = 2$, and $|S \cap E(P)| = 2$. By the definition of S, terminal x (lying on C) never belongs to S. Let us denote by u the vertex of S satisfying $u \in V(C_{y'x})$ and $S \cap V(C_{ux}) = \{u\}$ (i.e., u is the "highest" vertex of S on $C_{y'x}$; we have $u := a_2$ in the example in Fig. 5). Let us denote by u' the successor of u on C_{ux} (we have $u' := x$ in Fig. 5). Then, by the definition of S, all vertices of $C_{y'u}$ belong to S, and no vertex of $C_{u'x}$ belongs to S. Hence uu' is the only arc of $C_{y'x}$ which belongs to δS. Quite analogously, δS also contains exactly one arc of each segment C_{xy}, P_{x_1y}, and $P_{y'x_2}$, since S does not contain any vertex of P outside $\bar{P}$ by assumption. This proves that $|\delta S| = 4$.

Case (b2). Assume that either $S \cap X \neq \varnothing$, or S contains a vertex v of $P \backslash \bar{P}$. We show that $(G; X)$ is feasible in either case. Observe that S cannot contain the terminal x, since the terminals have degree 2. So if $S \cap X \neq \varnothing$ then S contains some $x' \neq x$. The

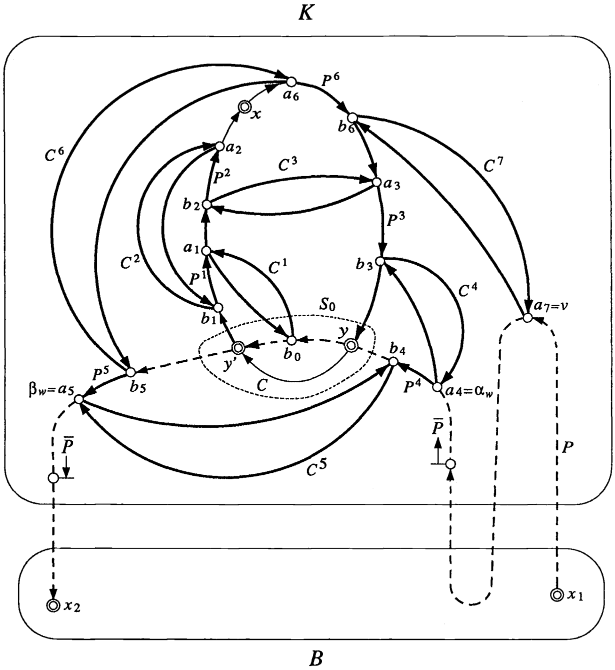


FIG. 5. *Sequence* $W=(b_0, C^1, a_1, P^1, b_1, \cdots, b_6, C^7, a_7=v)$.

assumption implies that there is a vertex $v \in S$ such that $v \in X$ or $v \in P \backslash \bar{P}$, for which there is a sequence

$$W=(b_0, C^1, a_1, P^1, b_1, C^2, a_2, P^2, b_2, \cdots a_{k-1}, P^{k-1}, b_{k-1}, C^k, a_k=v),$$

such that the following are satisfied:

—$b_0 \in S_0$;

—C^i is a segment of a cycle C_i in $K \backslash E(C)$, $i=1, 2, \cdots k$;

—every P^i is a segment of C_{xy}, $C_{y'x}$ or $\bar{P} \backslash E(P_{yy'})$;

—the cycles and paths in the sequence are mutually arc disjoint;

—b_{i-1} and a_i are vertices of C^i, $i=1, \cdots, k$;

—if P^i is a segment of C_{xy} or P_{x_1y} then it is an a_ib_i-path;

—if P^i is a segment of $C_{y'x}$ or $P_{y'x_2}$ then it is a b_ia_i-path.

An example of a sequence W from b_0 to a_7 is given in Fig. 5.

For any sequence W, let us define a pair α_W and β_W of vertices of $\bar{P}$ as follows. Vertex α_W is that a_i which lies on $\bar{P} \cap P_{x_1y}$ and has the highest subscript i; in case there is no such a_i we set $\alpha_W = y$. Similarly, β_W is that a_i which lies on $\bar{P} \cap P_{y'x_2}$ and has the highest subscript i; in case there is no such a_i we set $\beta_W = y'$. Assume that we are given a shortest sequence W (i.e., k is minimum) which starts in a vertex $b_0 \in S_0$ and terminates at vertex $v \in S \cap (X \cup (P \backslash V(\bar{P})))$. Let H be the digraph obtained as the union of the cycle C, the path $\bar{P}_{\alpha_W\beta_W}$, and the cycles $C_1, C_2, \cdots, C_k$. We may assume that $a_i \neq b_i$, and the members of W do not have other common vertices than those required, since otherwise W could be shortened.

CLAIM. *H has both a weak $(\alpha_W x, v\beta_W)$-linking and a weak $(\alpha_W v, x\beta_W)$-linking.*

This claim will be proved by induction, with use of Lemma 2.4. For $k=0$, 1 the statement can be easily verified. Let $k \geqq 2$. Construct an auxiliary digraph H' from H so that the part $W_0 = (b_0, C^1, a_1, P^1, b_1, C^2, a_2)$ is replaced by $W'_0 = (b_0, C^{12}, a_2)$ where C^{12} is a segment of a new cycle C_{12} (not existing in G) containing b_0, a_2 and disjoint from other members of the sequence (old cycles C_1 and C_2 are deleted in H'). Denote the modified sequence by W'. By induction hypothesis, the claim is valid for H' since the length of W' is $k - 1$. For simplicity, let us assume that $\alpha_W = \alpha_{W'}$ and $\beta_W = \beta_{W'}$ (i.e., $a_1 \neq \alpha_W, \beta_W$). The case when $a_1 \in \{\alpha_W, \beta_W\}$ will be discussed later. Let us denote by $L' = (L'_1, L'_2)$ one of the claimed linkings in H' where L'_1 and L'_2 are arc disjoint paths. Now we have to show that $L' = (L'_1, L'_2)$ can be modified to $L = (L_1, L_2)$ which is a weak linking in H for the same pair of terminals as L' in H'. Assume that $a_1 \in C_{xy} \cup P_{x_1y}$. Let F and F' be subdigraphs of H and H' as depicted in Fig. 2(a), which represents the portions of H and H' relevant to the above W_0 and W'_0. (In the case when $a_1 \in C_{y'x} \cup P_{y'x_2}$, the pair from Fig. 2(b) is used. We omit the proof because it is quite analogous to the considered case.) Observe first that each L'_i can enter F' only from a_1, a_2, or y and can exit only from a_2 or y'. Observe also that either of L'_1 and L'_2 can go through F' at most once, because there is no pair of vertex disjoint paths from $\{a_1, a_2, y\}$ to $\{a_2, y'\}$ in F' such that both use some arcs of F'. Assume that both L'_1 and L'_2 go through F'. Let us denote by s_1, t_1 and s_2, t_2 the entrances and exits of L'_1 and L'_2 in F', respectively. Since L'_1 and L'_2 induce a weak (s_1t_1, s_2t_2)-linking in F', by application of Lemma 2.4 there is a weak (s_1t_1, s_2t_2)-linking (Q_1, Q_2) in F. For $i = 1, 2$, let $L_i = (L'_1\backslash(L'_1)_{s_it_i}) \cup Q_i$. Then $L = (L_1, L_2)$ is the required linking in H. If only one of L'_1 and L'_2, say L'_1, goes through F', then set $L_1 = (L'_1\backslash(L'_1)_{s_1t_1}) \cup Q_1$, where Q_1 is an s_1t_1-path in F, and $L_2 = L'_2$. If both L'_1 and L'_2 are disjoint with F', then set $L_1 = L'_1$ and $L_2 = L'_2$.

It remains to discuss the case when $a_1 \in \{\alpha_W, \beta_W\}$, say $a_1 = \alpha_W$. Then $\alpha_{W'} = y$, and the induction hypothesis asserts that one of the initial vertices of $L' = (L'_1, L'_2)$ is $\alpha_{W'} = y$, but we need to start a path in $a_1 = \alpha_W$. In such case we extend it by $\bar{P}_{a_1y}$, because $\bar{P}_{a_1y}$ does not belong to H'. Thus the claim is proved.

Now the claim will be used to finish the proof of the remaining part of case (iiib). Assume $v = x' \in X$ (v is a terminal). Then either $x < x'$ or $x' < x$ hold. Assume $x < x'$, say $x = x_3$ and $x' = x_4$. Let (L_1, L_2) be a weak $(\alpha_W x, v\beta_W)$-linking which exists by the claim. Then $(P_{x_1\alpha_w}, L_1)$ and $(L_2, P_{\beta_wx_2})$ are arc disjoint x_1x_3- and x_4x_2-paths, which proves the feasibility of $(G; X)$, cf. the conditions discussed after the definition of feasibility in § 2. If $x' < x$, then the role of x and $v = x'$ is exhanged, and the weak $(\alpha_W v, x\beta_W)$-linking is applied.

Assume $v \in P_{x_1y}\backslash V(\bar{P})$. Let (L_1, L_2) be a weak $(\alpha_W x, v\beta_W)$-linking, and Q be an x_2z-path in B where $z \in P_{v\alpha_W} \cap B$. Then $(P_{x_1v}, L_2, P_{\beta_Wx_2})$ and $(Q, P_{z\alpha_W}, L_1)$ are arc disjoint x_1x_2- and x_2x-paths. Hence $(G; X)$ is feasible.

Assume $v \in P_{y'x_2}\backslash V(\bar{P})$. Let (L_1, L_2) be a weak $(\alpha_W v, x\beta_W)$-linking, and Q be an zx_1-path in B where $z \in P_{\beta_wv} \cap B$. Then $(P_{x_1\alpha_w}, L_1, P_{vx_2})$ and (L_2, P_{β_wz}, Q) are arc disjoint x_1x_2- and xx_1-paths. Hence $(G; X)$ is feasible. □

Now we can summarize and conclude the proof of Theorem 2.5. Given an infeasible irreducible instance $(G; X)$, consider its series decomposition $(B, K_1, \cdots, K_p)$ with respect to an x_1x_2-path P. By Lemma 2.7, each subproblem is also irreducible and infeasible for a nontrivial component. Using Theorem 2.5 as an induction hypothesis for the subproblems, we obtain that each component is planar, and having the properties formulated in Theorem 2.5. We may draw each component so that the specified face with the terminals is the outer face of the component. Assume that the terminals of a subproblem are placed clockwise on the boundary (i.e., its outer face is oriented anticlockwise) for components $K_1, \cdots, K_p$, and anticlockwise for component K_0 (i.e., its outer face is oriented clockwise). Split terminal x_0 in K_0 back to $x_1\,x_2$. Then interconnect the components by the path P so that the graph remains planar (see Fig. 4). Degrees of terminals from X remain two, and degrees of all other vertices become four. Also, each face of the digraph is oriented. □

Proof of Theorem 1.1. Theorem 1.1 follows from Theorem 2.5 when the number of terminals is restricted to three. Observe that every *minimal infeasible* instance defined in § 1 is *infeasible* and *irreducible* in the sense of § 2, since each of the reductions (1), (2), and (3) can be performed as a sequence of arc contractions. (The converse need not be true.) Finally, an irreducible instance with three terminals must be 2-connected by Lemma 2.3, since 1-decomposition can be defined only if $|X| \geqq 4$. □

3. Complexity results. In this section we show that feasibility of an instance $(G; X)$ can be decided by a polynomial time algorithm, and we prove NP-completeness of the weak linking problem for $G + H$ Eulerian when the number of terminal pairs is not fixed. Although the proof of Theorem 2.5 can be directly turned into a polynomial time algorithm, we prefer to present an algorithm based on the structural characterization of infeasible instances by Theorem 2.5.

LEMMA 3.1. *Let G be a 2-connected digraph, and $(G; X)$ be an infeasible irreducible instance. Then G has a unique plane representation in which all terminals are on the outer face.*

Proof. Let $H = G + z$ be obtained from G by adding a new vertex z and the set of arcs $\{xz \mid x \in X\}$. We claim that H is 3-connected. For contradiction, assume that $G + z$ has a vertex cut $\{u, v\}$. Clearly $z \notin \{u, v\}$, since otherwise G would have an articulation, and is not 2-connected. There must be at least one component S of $G \backslash \{u, v\}$ which does not contain any terminal, otherwise, $H \backslash \{u, v\}$ would be connected. We have $|\delta S| > 4$, otherwise, reduction (1) or (3) can be performed for S in $(G; X)$. Since $|\delta S| > 4$, u and v are not terminals, and since the degrees of u and v are 4, we have $|\delta(S \cup \{u, v\})| = 2$. Thus $(G; X)$ is reducible, and the claim is proved. By the well known Whitney Theorem, a planar 3-connected graph has a unique plane representation. □

Let n and e denote the number of vertices and arcs of a digraph G, respectively.

THEOREM 3.2. *Feasibility of an instance $(G; X)$ can be tested by an $O(e + n^2)$ time algorithm (i.e., $O(n^2)$ time algorithm when G is without multiple edges).*

Proof. Given an instance $(G; X)$ where G is an Eulerian digraph and X an ordered set of terminals, we will perform a test consisting of the following Phases 1, 2, and 3. At each phase, if the instance is found feasible, it terminates the whole test. On the other hand, if the instance passes a phase, the next phase is performed.

```
Phase 1. Transformation to an irreducible instance.
  Label all vertices of $G$ (including the terminals) as "unscanned";

  for every unscanned terminal $x \in X$ do
    begin
      Decide whether $\lambda(x, X \backslash x) \geqq 2$, and if not, find the largest minimum
      $(x, X \backslash x)$-cut $S$ (largest in the sense of $|S|$);
      If $\lambda(x, X \backslash x) \geqq 2$, then the instance is feasible. Stop;
      If $\lambda(x, X \backslash x) = 1$, then perform reduction (2) with $S$, and label terminal $x$
      as scanned.
    end;

  for every unscanned vertex $u \in V \backslash X$ do
```

```
begin
    Decide whether $\lambda(u, X) \geqq 3$, and if not, find the largest minimum $(u, X)$-
    cut $S$ (largest in the sense of $|S|$);
    If $\lambda(u, X) \geqq 3$, then the instance is feasible; stop;
    If $\lambda(u, X) = 2$ and $|S| > 1$, then perform reduction (3) with $S$, and label
    the vertex obtained by contracting $S$ as scanned;
    If $\lambda(u, X) = 2$ and $|S| = 1$, label vertex $u$ as scanned;
    If $\lambda(u, X) = 1$, then perform reduction (1) with $S$.
end.
```

The maximum number $\lambda(u, X)$ of arc disjoint paths from u to X (which is equal to the size ȯf minimum (u, X)-cut) can be computed by Ford–Fulkerson max-flow-min-cut algorithm where all arcs of G receive capacity 1. Given a current flow, the search for an augmenting path requires $O(e)$ time (since the capacities are 1). If no augmenting path exists, the labelling procedure provides a minimum (u, X)-cut S. If the labelling procedure is started from the set X, then $|V \backslash S|$ is minimum, and hence S is the largest minimum (u, X)-cut.

We are not interested in the exact value of $\lambda(u, X)$ in case $\lambda(u, X) \geqq 3$, since then the instance is feasible by (the proof of) Lemma 2.2 (note that, after the first loop, all terminals have degree 2). Thus we have to perform at most three searches for an augmenting path, and thus the complexity of scanning a vertex remains bounded by $O(e)$. Since there are at most n vertices to scan (the number of vertices of G is decreased when performing a reduction 1, 2, or 3), the complexity of Phase 1 is $O(en)$.

This time bound can be further improved by the following technique. Let $\bar{G}$ be the undirected graph obtained from G by neglecting its arc orientation. Since G is Eulerian, $\lambda(U_1, U_2) \geqq k$ if and only if $\lambda_{\bar{G}}(U_1, U_2) \geqq 2k$ for any disjoint subsets U_1 and U_2 of V, where $\lambda_{\bar{G}}$ the edge size of a minimum (U_1, U_2)-cut in $\bar{G}$. For any positive integer k, the method of [7] provides a spanning subgraph $\bar{H}$ of $\bar{G}$ in $O(e)$ time, such that (i) $\bar{H}$ has at most $(n - 1)k$ edges, and (ii) for any disjoint subsets U_1 and U_2 of V, any minimum (U_1, U_2)-cut in $\bar{H}$ is a minimum (U_1, U_2)-cut in $\bar{G}$ if $\lambda_{\bar{H}}(U_1, U_2) < k$ (as proved in [8]). Therefore, construct $\bar{H}$ with $k = 5$ and apply the above test to $\bar{H}$ instead of G. Since the number of edges in $\bar{H}$ is $O(n)$, an augmenting path can be found in $O(n)$ time, and the total time required in Phase 1 becomes $O(e + n^2)$ including $O(e)$ time to construct $\bar{H}$.

We must show that Phase 1 yields an irreducible instance unless it is shown feasible. For a contradiction, assume that the instance still admits a reduction after passing Phase 1. Let S be a set which may be reduced, say it admits reduction (2), and let x be the terminal in S. But it is not possible since the cut, which is largest in the sense of $|S|$, had been taken when x was scanned.

Phase 2. Decomposition into the blocks (i.e., 2-connected components).

{If $|X| = 3$, Phase 2 is not necessary since G must be 2-connected.
The input of Phase 2 is an irreducible instance $(G; X)$ with the degrees of terminals equal to 2, and the degrees of nonterminals equal to 4.
The output of Phase 2 is a collection $(G_i; X_i)$, $i = 1, \cdots, k$, of irreducible instances with the property that $(G; X)$ is feasible if and only if at least one instance $(G_i; X_i)$ is feasible. Moreover, the graphs G_i are 2-connected, since they are the blocks of G.}

Decompose G (as an unoriented graph) into the blocks, and compute the list of articulations. This can be done by the algorithm of Tarjan [10] in $O(n)$ time, since the number of edges of G is linear in n. Let k be the number of blocks, and $V_1, V_2, \cdots, V_k$ be the vertex sets of the blocks of G ordered so that $|V_i \cap \bigcup_{j>i} V_j| = 1$ for every $i = 1, \cdots, k-1$. Let y_i be the articulation for which $V_i \cap \bigcup_{j>i} V_j = \{y_i\}$. Set (formally) $G'_o := G$ and $X'_o := X$.

```
**for** $i = 1$ to $k - 1$ **do**
  **begin**
    Check whether articulation $y_i$ well splits the terminals of $(G'_{i-1};\ x'_{i-1})$. If not, the instance $(G; X)$ is feasible. Stop;
    Perform 1-decomposition of $(G'_{i-1};\ X'_{i-1})$ at $y_i$ into $(G_i;\ X_i)$ and $(G'_i;\ X'_i)$, where $V(G_i) = V_i$ and $G'_i$ is the remaining part of $G'_{i-1}$, i.e., $V(G'_i) = V_{i+1} \cup \cdots \cup V_k$ and $X'_i = X \cap V(G'_i)$.
  **end.**
```

The complexity of Phase 2 is $O(n)$. The correctness follows from Lemma 2.3. Then apply the next phase to each block $(G_i; x_i)$.

Phase 3. Planar representation test.

{The input of Phase 3 is an irreducible instance $(G;\ X)$ with the degrees of terminals equal to 2, and the degrees of nonterminals equal to 4. Moreover, G is 2-connected.
The output is a planar representation of $(G; X)$ satisfying the conditions of Theorem 2.5, unless it is proved that the instance is feasible.}

Use planarity test to decide whether G is planar. If not, $(G; X)$ is feasible. Hence assume G is planar. Use the planarity test again for the graph $G + z$ defined in Lemma 3.1. If $G + z$ is not planar, then $(G; X)$ is feasible, because it does not have a plane representation with terminals on one face. If $G + z$ is planar, test whether its plane representation, which is unique and is also obtained as a by-product of the planarity test, meets the remaining conditions in (ii) of Theorem 2.5. If yes, $(G; X)$ is infeasible, and it is feasible otherwise.

The time complexity of Phase 3 is $O(n)$, since planarity of a graph can be tested in linear time, and also the planar drawing can be obtained at the same time (see [5] and [9]). We must apply Phase 3 repeatedly to instances $(G_i;\ X_i)$, $i = 1, \cdots, k$. However, the total complexity remains bounded by n (the number of vertices of the instance of Phase 2), since $\Sigma |V_i| \leqq 2|V|$.

Therefore the time complexity of the whole algorithm (Phase 1 + Phase 2 + Phase 3 (applied to each block separately)) is $O(e + n^2)$. □

Theorem 1.2 is a corollary of this result.

Let us call the weak k-linking problem with $G + H$ Eulerian the *Eulerian weak k-linking problem*. When the number of terminal pairs is not restricted, we call the weak linking problem with $G + H$ Eulerian the *Eulerian weak linking problem*.

The Eulerian weak 2-linking problem is easy to solve (cf. [3]): a necessary and sufficient condition is that G is (weakly) connected. The Eulerian weak 3-linking problem has been polynomially solved in this paper. We conjecture that the Eulerian weak k-linking problem can be polynomially solved for any fixed k. However, if k is not fixed, the problem becomes NP-complete.

THEOREM 3.3. *The Eulerian weak linking problem is* NP-*complete.*

Proof. The problem obviously belongs to class NP. The NP-completeness will be proved by reducing to it the (general) weak 2-linking problem, which is known to be NP-complete (see [2]).

Given an instance $G = (V, E)$ and (s_i, t_i), $i = 1, 2$, of the weak 2-linking problem, where G is a general digraph, construct the following instance of the Eulerian weak linking problem.

Let us denote by $indeg(v)$ and $outdeg(v)$ the indegree and outdegree of a vertex v in $G + H = (V, E \cup \{t_1s_1, t_2s_2\})$, respectively. Set $G' = (V \cup \{s, t\}, E \cup E')$ where E' consists of $(outdeg(v) - indeg(v))$ parallel arcs sv for each v for which the difference is positive, and $(indeg(v) - outdeg(v))$ parallel arcs vt for each v for which the latter difference is positive. The multiple arcs can be modified to simple arcs by inserting artificial vertices. Let p be the sum of $(outdeg(v) - indeg(v))$ over all v for which the difference is positive. For $i = 3, 4, \cdots p + 2$, define $s_i = s$ and $t_i = t$. Now, G' and (s_i, t_i), $i = 1, \cdots, p + 2$, is an instance of the Eulerian weak linking problem. It is not difficult to see that this instance is feasible if and only if the original instance of weak 2-linking problem was feasible. □

Acknowledgment. We thank the anonymous referee for many detailed improvements of the text.

REFERENCES

[1] J. BANG-JENSEN, *Edge-disjoint in- and out-branchings in tournaments*, preprint 3, Institut for Matematik og Datalogi, Odense Universitat, 1987.

[2] S. FORTUNE, J. HOPCROFT, AND J. WYLLIE, *The directed subgraph homeomorphism problem*, Theoret. Comput. Sci., 10 (1980), pp. 111–121.

[3] A. FRANK, *On connectivity properties of Eulerian digraphs*, in Graph Theory in Memory of G. A. Dirac, Annals of Discrete Mathematics 41, North-Holland, Amsterdam, 1988, pp. 179–194.

[4] ———, *Graph connectivity and network flows*, Report of Institute for Operations Research, University of Bonn, No. 87473-OR (1987); Handbook of Combinatorics, R. Graham, M. Grötschel, and L. Lovász, eds., to appear.

[5] J. E. HOPCROFT AND R. E. TARJAN, *Efficient planarity testing*, J. Assoc. Comput. Mach., 21 (1974), pp. 549–568.

[6] H. NAGAMOCHI AND T. IBARAKI, *Multicommodity flows in certain planar directed networks*, Discrete Appl. Math., 27(1990), pp. 125–145.

[7] ———, *Linear time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph*, Algorithmica, to appear.

[8] H. NAGAMOCHI, Z. SUN, AND T. IBARAKI, *Counting the number of minimum cuts in undirected multigraphs*, Report 89010, Department of Applied Mathematics and Physics, Kyoto University, 1989.

[9] T. NISHIZEKI AND N. CHIBA, *Planar Graphs: Theory and Algorithms*, Annals of Discrete Mathematics 32, North-Holland, Amsterdam 1988.

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

SIAM J. DISC. MATH.
Vol. 4, No. 1, pp. 99–106, February 1991

010

SPANNING TREES WITH MANY LEAVES*

DANIEL J. KLEITMAN† AND DOUGLAS B. WEST‡

Abstract. A connected graph having large minimum vertex degree must have a spanning tree with many leaves. In particular, let $l(n, k)$ be the maximum integer m such that every connected n-vertex graph with minimum degree at least k has a spanning tree with at least m leaves. Then $l(n, 3) \geqq n/4 + 2$, $l(n, 4) \geqq (2n + 8)/5$, and $l(n, k) \leqq n - 3\lfloor n/(k + 1)\rfloor + 2$ for all k. The lower bounds are proved by an algorithm that constructs a spanning tree with at least the desired number of leaves. Finally, $l(n, k) \geqq (1 - b \ln k/k)n$ for large k, again proved algorithmically, where b is any constant exceeding 2.5.

Key words. spanning trees, vertex degrees

AMS(MOS) subject classifications. 05C05, 05C35

1. Introduction. Given a connected simple graph G, suppose we wish to find a spanning tree in G with many leaves. If G is a cycle, we can only guarantee 2 leaves, but we may have better luck if we require that every vertex have degree at least k. To make this precise, let $\mathbf{G}_{n,k}$ denote the collection of connected n-vertex graphs with minimum degree at least k. We wish to determine $l(n, k)$, the maximum m such that every graph in $\mathbf{G}_{n,k}$ has a tree with at least m leaves. Note that $l(n, 2) = 2$.

The question of determining $l(n, k)$ has occurred independently to several researchers. For this investigation, the question was raised by Lovász and Saks [6]. Independently, Payan, Tchuente, and Xuong [7] showed that every 3-regular graph has a tree with at least $n/4$ leaves, and Storer [8] gave the lower bound of $n/4 + 2$ for that case. This was subsequently rediscovered by Linial and Sturtevant [5] and extended to minimum degree 3. Another proof appears in [3]. Storer was motivated by complexity considerations. The problem of finding a spanning tree with maximum number of leaves is NP-complete, even if G is regular of degree 4 [2]. We provide here a simple algorithm to construct a tree with at least $n/4 + 2$ leaves in any $G \in \mathbf{G}_{n,3}$. Extending this approach, we also present an algorithm to construct a tree with at least $(2n + 8)/5$ leaves in any $G \in \mathbf{G}_{n,4}$. Finally, we present a simple family of algorithms that provide lower bounds implying $l(n, k) > (1 - b \ln k/k)n$. In particular, this means that the fraction of the vertices that can be guaranteed to be leaves in the spanning tree with the most leaves approaches 1 as k grows.

For arbitrary k, a simple construction yields a $G \in \mathbf{G}_{n,k}$ with no tree having more than $n - 3\lfloor n/(k + 1)\rfloor + 2$ leaves. When $k \leqq 4$ and $k + 1$ divides n, this achieves the bound. Griggs and Wu [4] have proved optimality for $k = 5$ (and give an alternate proof for $k = 4$). Linial [5] conjectured that this construction is essentially optimal in general, i.e., that $l(n, k) \geqq n - 3n/(k + 1) + c_k$ for each k and an appropriate constant c_k. More generally, Linial suspects that a connected graph with degree sequence $d_1 \geqq d_2 \geqq \cdots \geqq d_n \geqq 2$ has a spanning tree with at least $\Sigma(d_i - 2)/(d_i + 1)$ leaves.

Albertson and Hutchinson [1] have investigated spanning forests. If we seek a forest of c components with many leaves, then the upper and lower bounds presented here still

* Received by the editors September 10, 1988; accepted for publication December 21, 1989. Most of this research was done while the authors visited the Institute for Mathematics and Its Applications at the University of Minnesota, Minneapolis, Minnesota 55455.

† Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139. This research was supported in part by National Science Foundation grant DMS 86-06225 and Air Force Office of Scientific Research grant OSR-86-0076.

‡ Department of Mathematics, University of Illinois, Urbana, Illinois 61801. This research was supported in part by Office of Naval Research grant N00014-85K0570.

hold, with 2 replaced by $2c$. Albertson and Hutchinson were further interested in limiting the diameter of the components, but our methods do not seem relevant to that question.

2. The upper bound construction.

THEOREM 1. $l(n, k) \leqq n - 3\lfloor n/(k+1)\rfloor + 2$.

Proof. We construct $G_{n,k} \in \mathbf{G}_{n,k}$ having no tree with more than $n - 3\lfloor n/(k+1)\rfloor + 2$ leaves. Let $m = \lfloor n/(k+1)\rfloor$ and $r = n - m(k+1)$. Partition the vertex set $V(G)$ into sets $R_0, \cdots, R_{m-1}$, where $|R_i| = k+1$ for $i \neq 0$ and $|R_0| = k+1+r$. Choose $x_i, y_i \in R_i$. Place edges between all pairs of vertices in R_i except x_iy_i. Add the edges $Z = \{x_iy_{(i+1) \bmod m}: 0 \leqq i < m\}$, and let $W = \{x_i\} \cup \{y_i\}$.

It suffices to show that any spanning tree T of $G_{n,k}$ has at most $n - 3m + 2$ leaves. Every pair of edges in Z forms an edge cut so T lacks at most one edge of Z. Suppose first that $x_jy_{j+1} \notin T$; T then contains an x_iy_i-path in R_i for each i. This forces a nonleaf in $R_i - W$ for each i, and each vertex of W must be a nonleaf except $\{x_j, y_{j+1}\}$. On the other hand, if T omits no edge of Z, then T lacks an x_i, y_i-path in R_i for one value of i, say j. This forces at least $3(m-1)$ nonleaves in $V - R_j$, and $k \geqq 2$ forces an additional nonleaf at x_j or y_j. □

Note that $G_{n,k}$ contains many copies of the "almost clique" $K_{k+1} - e$. If this induced subgraph is forbidden, a higher proportion of the vertices must be leaves. In particular, Griggs, Kleitman, and Shastri [3] have shown that every $G \in \mathbf{G}_{n,3}$ that does not contain $K_4 - e$ has a tree with at least $(n+4)/3$ leaves; this was earlier conjectured in [7]. The proof is more difficult than that of the unrestricted result in the next section.

We also note that when k is even there is another class of graphs where the tree with the most leaves has $n - 3\lfloor n/(k+1)\rfloor + 2$ leaves, as shown by a similar argument. The graph can be described as a cyclic sequence of cliques in which each vertex is also joined to every vertex of the clique before and after it. The cliques have sizes $k/2$, $k/2$, 1, $k/2$, $k/2$, 1, $\cdots$. Note that $G_{n,k}$ can also be described in this way with the clique sizes being 1, $k-1$, 1, 1, $k-1$, 1, $\cdots$.

3. The case $k = 3$.

The lower bound for $k = 3$ appeared in [7] and in [8] for 3-regular graphs. We include a short proof of the general result, different from those in [7] and [8], to illustrate the method we will use for $k = 4$. Another proof, similar in spirit to this but phrased also in terms of 3-regular graphs, appears in [3].

This and the later proofs grow the desired spanning tree of G via an iterative algorithm. In each case, we let T denote the current tree with n vertices and l leaves. If x is a leaf of T, then the *out-degree* of x, denoted $d'(x)$, is the number of neighbors it has in $G - T$. The operation of *expansion at* x consists of adding to T the $d'(x)$ edges from x to *all* its neighbors not in T. We grow T by vertex expansion sequences (also called "operations"); this guarantees that all edges from T to $G - T$ are incident to leaves of T.

THEOREM 2. *Every $G \in \mathbf{G}_{N,3}$ has a spanning tree with at least $N/4 + 2$ leaves.*

Proof. A leaf x of T with $d'(x) = 0$ is *dead*; no expansion is possible at a dead leaf, and it must be a leaf in the final tree. Let m be the number of dead leaves in T. An expansion that makes y a dead leaf *kills* y. We call an expansion sequence *admissible* if its effect on T satisfies the "augmentation inequality" $3\Delta l + \Delta m \geqq \Delta n$.

We initialize T to a small subtree and provide a collection of admissible operations to grow T into a spanning tree of G. If G is not 3-regular, we initialize T to be all edges incident to a vertex of maximum degree $\Delta \geqq 4$. If G is 3-regular and every edge belongs to a triangle, then $G = K_4$ and the claim holds. Otherwise G is 3-regular and has an edge in no triangle, and we initialize T to consist of such an edge and the four other edges incident to it.

If T is grown to a spanning tree with L leaves by admissible operations, then all leaves eventually die and summing the augmentation inequality yields $3(L - \Delta) + L \geqq N - \Delta - 1$ if G is not 3-regular, or $3(L - 4) + L \geqq N - 6$ if G is 3-regular. These simplify to $4L \geqq N + 2\Delta - 1 \geqq N + 7$ and $4L \geqq N + 6$, respectively. We can improve this to $4L \geqq N + 8$ by considering the final admissible operation. For this operation, the augmentation inequality is satisfied with an excess of at least two because the operation kills at least two final leaves whose deaths are not usually guaranteed for the operation.

It remains to present a collection of admissible operations of which at least one is always available until T absorbs all vertices, and to verify the statement claimed about the last operation. The three operations we use are illustrated in Fig. 1.

O1: If $d'(x) \geqq 2$ for some current leaf x, then expanding at x yields $\Delta l = \Delta n - 1 \geqq 1$ and $\Delta m \geqq 0$.

O2: If $d'(x) \leqq 1$ for every current leaf x and some vertex outside T has at least two neighbors in T, then expanding at one of them yields $\Delta l = 0$, $\Delta m \geqq 1 = \Delta n$.

O3: If y is the only neighbor of x outside T and y has at least two neighbors not in T, then expanding at x and then y yields $\Delta l = \Delta n - 2 \geqq 1$ and $\Delta m \geqq 0$.

Because $k = 3$, any neighbor not in T of a vertex in T has at least two neighbors in T or at least two neighbors outside T. This implies that one of O1–O3 is available until T becomes a spanning tree. Also, the inequalities they satisfy imply that each is admissible.

Now consider the final operation. Each of the three operations adds (at least one) leaf z to T that did not previously belong to T. That leaf has a neighbor w not appearing in the illustration; since this is the last operation, w must have been a nondead leaf of T. Since z and w both die now, we obtain the needed excess of two dead leaves. □

Before leaving this section, we note that the operations used above also yield the following result.

THEOREM 3. *If every edge of G belongs to a triangle and $G \neq K_3$, then G has a tree with at least $(|V(G)| + 5)/3$ leaves, and this is best possible.*

Proof. We use the same terminology as in the previous proofs, except that now an operation is *admissible* if it satisfies the augmentation inequality $2\Delta l + \Delta m \geqq \Delta n$. Operations O1 and O2 above satisfy this admissibility inequality; we claim they suffice to grow T to a spanning tree. If T does not yet span, then there is an edge xy with $x \in T$, $y \notin T$; xy forms a triangle with some additional vertex z. If $z \notin T$, then O1 applies; if $z \in T$, then O2 applies.

If $G \neq K_3$ and $\Delta(G) < 4$, then $G = K_4$ or $G = K_4 - e$ and the bound holds. Otherwise G has a vertex of degree at least 4 to use as the center of the initial T. If also $\delta(G) \geqq 3$, then again the last operation provides two additional dead leaves, and summing the augmentation inequalities yields $2(L - 4) + L - 2 \geqq N - 5$, or $L \geqq (N + 5)/3$.

If $\delta(G) = 2$, then the last operation may provide only one additional dead leaf if it is O2 to a 2-valent vertex. However, if G has a 2-valent vertex x, then the edge-in-triangle property leads to a vertex w of degree at least 4 within distance 2 of x. If w is adjacent to x, then beginning at w makes x initially a dead leaf and we have the same inequality as above. Otherwise, x and w have two common (adjacent) 3-valent neighbors u, v. If

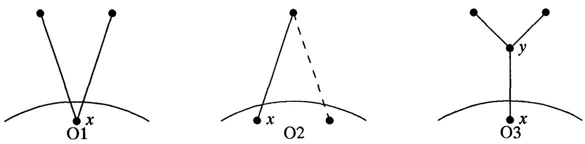


FIG. 1. *Operations used when $k = 3$.*

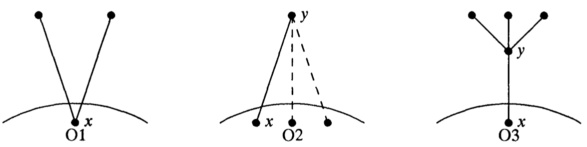


FIG. 2. *Elementary operations used when $k = 4$.*

the initial tree is the star at w plus the edge ux, then we begin with x, v as dead leaves and again get an extra at the end. Now the inequality is $2(L - 4) + L - 3 \geqq N - 6$, or again $L \geqq (N + 5)/3$. □

To show that this is best possible, consider the graph $G_{n+\lfloor n/3 \rfloor,3}$ of § 2, delete one cut edge, and contract the remaining cut edges.

4. The case $k = 4$. For the case $k = 4$ we will use arbitrarily long expansion sequences as operations. We use the same terminology and notation as above, except that now an expansion sequence (or "operation") is *admissible* if it satisfies the augmentation inequality $4\Delta l + \Delta m \geqq 2\Delta n$.

THEOREM 4. *Every $G \in \mathbf{G}_{N,4}$ has a spanning tree with at least $(2N + 8)/5$ leaves.*

Proof. Again we initialize T to be a small subtree and we provide a collection of admissible operations to grow T into a spanning tree of G. If we provide an exhaustive set of admissible operations, summing the augmentation inequalities will yield $4(L - c_1) + (L - c_2) \geqq 2(N - c_3)$, or $L \geqq 2N/5 + c$, where c_1, c_3 are the number of leaves and vertices in the initial tree and c_2 is the number of leaves not counted as dead by summing the general augmentation inequalities. We postpone the discussion of the additive constant.

The first three operations are similar to those used for $k = 3$ and are illustrated in Fig. 2.

O1: If $d'(x) \geqq 2$ for some current leaf x, then expanding at x yields $\Delta l = \Delta n - 1 \geqq 1$ and $\Delta m \geqq 0$.

O2: If $d'(x) \leqq 1$ for every current leaf x and some vertex outside T has at least three neighbors in T, then expanding at one of them yields $\Delta l = 0$, $\Delta m \geqq 2 = 2\Delta n$.

O3: If y is the only neighbor of x outside T and y has at least three neighbors not in T, then expanding at x and then y yields $\Delta l = \Delta n - 2 \geqq 2$ and $\Delta m \geqq 0$.

Each of these operations is admissible. If none of O1–O3 are available, then every nondead leaf of T has out-degree one and its neighbor outside T has two neighbors in T and two neighbors outside T.

The subsequent operations, which involve arbitrarily long expansion sequences, will apply in this case. We consider only *principal* expansion sequences; these expand a single leaf $x = y_0$ of T and then other leaves that do not belong to T before the initial expansion. The *length* r of a principal expansion sequence Y is the number of expansions outside T. A principal expansion sequence is *live* if each expansion after y_0 introduces two new vertices to the tree. Y also denotes the set of vertices expanded.

When O1–O3 are not available, a live sequence almost satisfies the augmentation inequality for admissibility. The expansion at y_0 adds one vertex and kills the other neighbor of y_1 in T. Each subsequent expansion in Y increases l and adds two new vertices. Altogether, $4\Delta l + \Delta m = 4r + 1$ and $2\Delta n = 4r + 2$, leaving a deficiency of one in the augmentation inequality.

O4–O7 rely on various additional conditions that imply admissibility and are illustrated by example in Fig. 3. For specification of O4–O7, let Y be a live sequence of length r and assume O1–O3 are not available. Let W denote the set of leaves introduced by executing Y and let $U = V(G) - (T \cup Y \cup W)$; U is the set of vertices that would remain outside the tree after executing Y.

O4: If some $w \in W$ has a neighbor $u \in T$, then Y is admissible. Executing Y kills u, which increases Δm by one to eliminate the deficiency.

O5: If some $w \in W$ has all its neighbors in $Y \cup W$, then Y is admissible. Executing Y kills w, which increases Δm by one to eliminate the deficiency.

O6: If some $w \in W$ has at least three neighbors in U, then Y followed by (w) is admissible. The final expansion satisfies $4\Delta l - 2\Delta n \geqq 2$, which eliminates the deficiency.

O7: If v is the unique neighbor in U for at least four vertices of W, then Y followed by expansion at one of these vertices is admissible. The final expansion kills (at least) three leaves, yielding $\Delta m - 2\Delta n \geqq 1$, which eliminates the deficiency.

Next we show that some operation of types O1–O7 is always available until T becomes a spanning tree. To prove this, we consider a special class of expansion sequences. A *linear* expansion sequence is a live sequence $Y = (y_0, \cdots, y_r)$ such that, for each $i \geqq 1$, y_{i+1} is one of the two leaves introduced by expanding y_i. The illustrations in Fig. 3 suggest linear sequences although expansion sequences of types O4–O7 need not be linear. For a linear sequence, we let z_i denote the other leaf introduced by expanding y_i and let z_r, w denote the two leaves introduced by expanding y_r. We may refer to w as y_{r+1}. Let $Z = \{z_1, \cdots, z_r\}$ and $W = Y \cup Z \cup \{w\}$. For $1 \leqq i \leqq r$, let $Y_i = (y_0, \cdots, y_i)$ and $Z_i = \{z_1, \cdots, z_i\}$. We use $R \cdot S$ for the concatenation of two vertex sequences, $N(a)$ for the set of neighbors of vertex a, and $N(S)$ for $\cup_{x \in S} N(x)$.

If O1–O3 are unavailable and T does not span G, then any neighbor of T is the end of a linear sequence of length 1; i.e., linear sequences exist. Because G is finite, linear sequences cannot be arbitrarily long. If O1–O7 are unavailable, then for a maximal linear sequence it must be true that each leaf introduced by the last expansion has exactly one neighbor in U.

Suppose O1–O7 are unavailable and let $Y = (y_0, \cdots, y_r)$ be a maximal linear sequence. In addition to y_r and one vertex $v \in U$, w has at least two additional neighbors. Because Y is live, these must appear in Z. Suppose $z_t, z_s \in N(w)$ with $t = \min \{i: z_i \in N(w)\}$, so $t < s \leqq r$.

We claim that z_t must have exactly one neighbor u not in W. Otherwise, Y is of type O5 (killing z_t) or $Y_t \cdot (z_t)$ is of type O6. Furthermore, if $u \neq v$, then $Y_t \cdot (z_t, w)$ is of type O6. Hence we may assume $u = v$. If $s < r$, then $Y_{s-1} \cdot (z_t, y_{r+1}, \cdots, y_{s+2})$ is a type O5 sequence killing y_s. Hence we may also assume $s = r$.

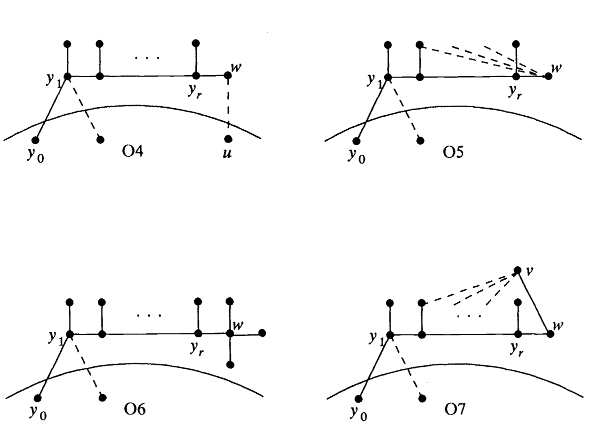


FIG. 3. *Complex operations used when* $k = 4$.

Applying the same arguments to $z_r = z_s = w'$, we obtain a neighbor $z_{t'}$ of w' adjacent to $v' \notin W$ (see Fig. 4). If $t = t'$, then $\{w, w', v, v'\} \subset N(z_t)$ and $Y_t \cdot (z_t)$ is of type O6. If $t \neq t'$ and $v \neq v'$, then $Y_{r-1} \cdot (z_t, z_{t'})$ is a type O5 sequence killing y_r. If $t \neq t'$ and $v = v'$, then v is the only neighbor in U for each of $\{z_t, z_{t'}, w, w'\}$ and $Y \cdot (w)$ is of Type O7.

We have provided an exhaustive set of admissible operations. Now consider the additive constant. Recall that $L = 2N/5 + c$, where $c = (c_2 + 4c_1 - 2c_3)/5$ and c_1, c_2, c_3 are the number of initial leaves, leaves not counted as dead, and initial vertices. As for $k = 3$, each operation illustrated has a leaf incident to another edge not drawn, which again means that the last operation must kill at least two additional leaves (except for O2 and O7, the extra count is always at least four). Since G has minimum degree at least 4, we have $c \geqq (2 + 16 - 10)/5 = 8/5$. □

It should be noted that there are only two known examples of graphs in $\mathbf{G}_{n,4}$ that have no tree with at least $2N/5 + 2$ leaves. These are the 4-regular graph on six vertices and the 4-regular graph on eight vertices around a circle in which each vertex is joined to the four vertices closest to it. The desired bound asks for five and six leaves, respectively. On six vertices, having five leaves would require a 5-valent vertex, and on eight vertices, having six leaves would require two vertices whose neighborhoods include all the vertices. We conjecture that $2N/5 + 2$ is a lower bound except for these two examples. If G has a vertex of degree at least 5, then starting with the edges incident to it yields $c \geqq 2$. If G is 4-regular and has an edge not in a triangle, then starting with its endpoints and their neighbors yields $c_1, c_2, c_3 = 6, 2, 8$ and $c = 2$. Hence any graph that violates this bound is 4-regular and has every edge in a triangle.

5. Larger values of *k*. In general the conjectured lower bound on $l(n, k)$ is $(k - 2)n/(k + 1) + 2$, except possibly for small exceptions. Whenever k is even, there is a small example that slightly violates this bound. Whenever $k > 2$, we can choose n so that $3k/2 + 2 \leqq n < 5(k + 1)/3$ and let G be the graph on n vertices around a circle in which each vertex is adjacent to the k closest vertices, $k/2$ in each direction. Then $(k - 2)n/(k + 1) + 2 > n - 3$, so the bound asks for a tree with $n - 2$ leaves. However, there are no two adjacent vertices whose neighborhoods cover $V(G)$.

The most interesting question, of course, is the coefficient of n in $l(n, k)$. For $k = 5$, Griggs and Wu [4] have proved the conjecture (they also have an alternate proof of the bound for $k = 4$, using a different augmentation inequality for admissibility). For large k we give a short proof that the coefficient approaches 1. The ease of this argument is attributable to the fact that we are not seeking an optimal algorithm for any individual value of k. By considering more operations, i.e., by making the algorithm more complicated, we could improve the rate of convergence.

THEOREM 5. *If k is sufficiently large, then there is an algorithm that constructs a spanning tree with at least* $[1 - b \ln k/k]n$ *leaves in any graph with minimum degree k, where b is any constant exceeding* 2.5.

Proof. We design an algorithm as those above in which the current tree T is expanded at leaves. We will develop an admissibility inequality that has the form $r\Delta l + \Delta M \geqq (r - 1)\Delta n$ where r is a function of k. Here M is a measure of "deadness" for the leaves

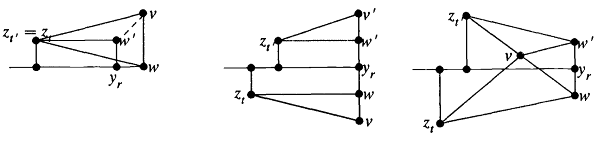


FIG. 4. *Resolution of maximal live sequences when $k = 4$.*

of the current tree. This is not a physical concept. Rather, the final value of M is a multiple counting of the leaves of the final tree, and the individual changes in M are an amortized distribution of this count over the operations.

The statistic we use to measure "deadness" is $M = \Sigma_{i=0}^{r-1} \alpha_i m_i$, where m_i is the number of leaves of T having i neighbors outside T; the coefficients α_i will be chosen shortly. It is natural to think that a leaf is more dead when it has fewer outside neighbors, so we will require $0 = \alpha_{r-1} \leqq \alpha_{r-2} \leqq \cdots \leqq \alpha_0$. This requirement guarantees that expansion at any leaf with out-degree at least r will satisfy $r\Delta l + \Delta M \geqq (r-1)\Delta n$; the net change in M will be nonnegative. Note that it makes sense to assume $r < k$.

If every operation used by the algorithm satisfies $r\Delta l + \Delta M \geqq (r-1)\Delta n$, then beginning with a star at a vertex of degree k and summing the augmentation inequalities yields $r(L-k) + \alpha_0 L \geqq (r-1)(N-k-1)$, or $L \geqq [(r-1)N + (k+1-r)]/(r+\alpha_0) > (1-(\alpha_0+1)/r)N$. We will choose the values of r and $\{\alpha_i\}$ so that the operations are admissible and $(\alpha_0+1)/r < b \ln k/k$, as desired.

For each $i < r$, define an *i-operation* to be an operation that is performed only when the maximum out-degree of current leaves is i. Each i-operation begins by expansion at a vertex x with $d'(x) = i$. This or additional expansions may add to the tree a vertex y that was an outside neighbor of some z in the current tree with $d'(z) = j \leqq i$. The net changes to M for this operation include $-\alpha_i$ for the loss of x as a leaf and $\alpha_{j-1} - \alpha_j$ for the effect of the edge yz on $d'(z)$. It will suffice to consider changes of these types.

Let $c_i = \alpha_{i-1} - \alpha_i$ for all i. If in addition to $\alpha_{r-1} \leqq \cdots \leqq \alpha_0$ we also have $c_{r-1} \leqq \cdots \leqq c_1$, then for any i-operation each edge from a new vertex to an old leaf contributes at least c_i to ΔM. Since we lose the contribution from the leaf expanded to begin the operation and ignore the possible gains for the new vertices, it suffices to show $r\Delta l + c_i q - \alpha_i \geqq (r-1)\Delta n$ for each i-operation, where q is the number of nontree edges from new vertices to old vertices of the tree.

To guarantee the desired properties of the operations we will choose $r = \lfloor k/5 \rfloor$ and $c_i = (r-i)/[i(k-3r)-r]$. Note that this formula for c_i increases as i decreases and that $c_1 < 1$ when $k \geqq 5r$.

Let us now specify the i-operations. Let $i < r$ be the maximum out-degree of current leaves and let x be a current leaf with maximum out-degree. Either we expand at x and stop, which we call Oi, or we expand at x and also at the new neighbor y of x for which the second expansion gives the maximum number of additional leaves; we call the latter Pi. We choose Pi if the number of vertices introduced by the second expansion is more than $3r - i$.

By construction there is always an operation available to grow T until T spans. For the admissibility of Pi, we have $\Delta l = \Delta n - 2$. Ignoring gains due to possible edges from new vertices to old vertices, it suffices to show that $\Delta n \geqq 2r + \alpha_i$. Since $\Delta n > 3r$, this holds when $c_i \leqq 1$ since $\alpha_i = \Sigma_{j=i+1}^{r-1} c_j < rc_i$.

For the admissibility of Oi, suppose that y is an outside neighbor of x and that a second expansion at y would introduce at most $3r - i$ new vertices. Because y also has at most i neighbors among x and the vertices introduced by expanding at x, it has at least $k - 3r$ neighbors in T besides x. This is true for each outside neighbor of x, so $q \geqq i(k-3r)$ for the conditions under which we apply Oi. We have $\Delta l = i - 1$ and $\Delta n = i$, so

$$r\Delta l + c_i q - \alpha_i \geqq r(i-1) + c_i(q-r) \geqq r(i-1) + (r-i) = (r-1)\Delta n.$$

Finally, we study $\Sigma c_i = \alpha_0$. Since $k \geqq 5r$, we have $\Sigma c_i \leqq \Sigma_{i=1}^{r-1}(r-i)/r(2i-1)$. Using calculus we can bound this by $1/r[r - 1 + \int_1^{r-1}(r-x)dx/(2x-1)]$. With the substitution $u = 2x - 1$ we can evaluate the definite integral as

$$\tfrac{1}{4}[(2r-1)\ln(2r-3) - (2r-4)].$$

Putting this all together yields

$$\alpha_0 < \tfrac{1}{4}(r)[4(r-1)-2(r-2)+(2r-1)\ln(2r-3)] < .5+.5\ln 2r.$$

When we replace r by $\lfloor k/5 \rfloor$, we find $1-(\alpha_0+1)/r > 1 - b\ln k/k$ for sufficiently large k as long as $b > 2.5$. □

This constant b can be reduced by choosing c_i and r to make use of some slack in the argument. In particular, the admissibility of Pi requires only $\Delta n \geqq 2r + \alpha_i$, so we can use P$i$ whenever the second expansion introduces more than $2r + \alpha_i - i$ additional vertices. When this fails for all neighbors of x we have $q \geqq i(k - 2r - \alpha_i)$. The admissibility of O$i$ requires only $c_i q - \alpha_i \geqq r - i$, so it suffices to define c_i iteratively with $\alpha_{r-1} = 0$, $c_i = (r - i + \alpha_i)/[i(k - 2r - \alpha_i)]$, and $\alpha_{i-1} = \alpha_i + c_i$. We still wish to keep each c_i small to make α_0 of at most logarithmic size, and for this it suffices to have $k - 2r > \beta r$ (i.e., $r = \lfloor k/(2+\beta) \rfloor$ for some constant $\beta > 0$). The aim is then to bound α_0 by some function $f(\beta)\ln r$, which would lead to the constant $\beta f(\beta)$ in place of b. It does not seem worthwhile to pursue the details of this, since better improvements could be generated by considering a larger variety of operations.

Acknowledgment. The authors are deeply indebted to J. R. Griggs and Mingshen Wu for valuable comments on and corrections to earlier versions of this paper.

REFERENCES

[1] M. O. Albertson and J. P. Hutchinson, *Spanning forests with given radius and few components*, presented at 4th SIAM Conference on Discrete Mathematics, San Francisco, 1988.

[2] M. L. Garey and D. S. Johnson, *Computers and Intractibility: A Guide to the Theory of* NP-*Completeness*, W. H. Freeman, San Francisco, 1979, p. 206.

[3] J. R. Griggs, D. J. Kleitman, and A. Shastri, *Spanning trees with many leaves in cubic graphs*, J. Graph Theory, 13 (1989), pp. 669–695.

[4] J. R. Griggs and M.-S. Wu, *Spanning trees in graphs of minimum degree* 4 *or* 5, Discrete Math., to appear.

[5] N. Linial and D. Sturtevant, *private communication*, 1987.

[6] L. Lovasz and M. E. Saks, *private communication*, 1987.

[7] C. Payan, M. Tchuente, and N. H. Xuong, *Arbres avec un nombre maximum de sommets pendants*, Discrete Math., 49 (1984), pp. 267–273.

[8] J. A. Storer, *Constructing full spanning trees for cubic graphs*, Inform. Process. Lett., 13 (1981), pp. 8–11.

SIAM J. DISC. MATH.
Vol. 4, No. 1, pp. 107-120, February 1991

011

TREE-MATCHINGS IN GRAPH PROCESSES*

TOMASZ ŁUCZAK† AND ANDRZEJ RUCIŃSKI†

Abstract. For a tree T a perfect T-matching in a graph G is a subgraph of G with at least $|G| - |T| + 1$ vertices, each component of which is isomorphic to T. Two properties, $\mathscr{A}$ and $\mathscr{B}$, are introduced where the former is a modification of the fact that the largest component of G has a perfect T-matching and the latter is a suitably chosen necessary condition for $\mathscr{A}$ expressed in terms of forbidden "pendant" subgraphs. We show that in the random graph process $\hat{G}_n$ the hitting times of both above properties coincide. This paper is the first one that deals with the hitting times of nonmonotone graph properties. It extends results of Bollobás and Frieze [*Ann. Discrete Math.*, 28 (1985), pp. 23-46] and Bollobás and Thomason [*Ann. Discrete Math.*, 28 (1985), pp. 47-98].

Key words. random graph process, hitting time, generalized matchings

AMS(MOS) subject classification. 05C80

1. Introduction. Let $G(n, M)$ be a random graph chosen uniformly from the family of all graphs on vertex set $[n] = \{1, 2, \cdots, n\}$ which have M edges, $0 \leqq M \leqq \binom{n}{2}$.

Let $\mathscr{M}$ be the property of having a perfect matching and let $\mathscr{N}\mathscr{T}$ be the property of not containing an isolated vertex. Erdős and Rényi proved in 1966 the following fundamental result.

THEOREM 1 [ER66]. *Let $x_n = M/n - \log n$. Then*

$$\lim_{n\to\infty} \text{Prob}\,(G(2n,M)\in\mathscr{M}) = \lim_{n\to\infty} \text{Prob}\,(G(2n,M)\in\mathscr{N}\mathscr{T})$$

$$= \begin{cases} 0 & \text{if } x_n \to -\infty \\ \exp(-2e^{-x}) & \text{if } x_n \to x \\ 1 & \text{if } x_n \to \infty . \end{cases}$$

We generalize this theorem in three ways. First, similarly as in papers of Bollobás and Thomason [BT85] and Łuczak [Ł87], we ask about the existence of a perfect matching in the largest component of a random graph. Second, we match vertices not into adjacent pairs but into bunches which follow a tree pattern. To be precise, given a connected graph G and another graph H we say that F is a perfect G-matching in H if F is a subgraph of H, every component of F is isomorphic to G and $|H| - |F| \leqq |G| - 1$, where $|K|$ is the number of vertices of a graph K. The last condition allows us not to care about the divisibility of $|H|$ by $|G|$. This is important since we cannot predict the exact size of the largest component of a random graph. We will be interested in perfect T-matchings where T is a tree.

The most important strengthening of Theorem 1 involves graph processes. Let $\tilde{\mathscr{G}}_n$ be the family of all $\binom{n}{2}!$ sequences of graphs on vertex set $[n]$, $\tilde{G}_n = (G_0, G_1, \cdots, G_{\binom{n}{2}})$, where G_i has i edges and contains its predecessor as a subgraph. We turn the family $\tilde{\mathscr{G}}_n$ into a probabilistic space by assigning to each $\tilde{G}_n$ the same probability. Equivalently, we can start with the empty graph and keep selecting edges at random, one by one, in the equiprobable manner. The resulting graph sequence is called a *graph process* and denoted by $\hat{G}_n = (G(n, 0), \cdots, G(n, \binom{n}{2}))$. The Mth stage of the process, $G(n, M)$, coincides with the random graph described above.

* Received by the editors June 6, 1989; accepted for publication (in revised form) January 25, 1990.

† Department of Discrete Mathematics, Adam Mickiewicz University, Poznań, Poland.

Graph processes were introduced by Erdős and Rényi in 1959, who realized that this is the most subtle tool for investigating the evolution of random graphs.

For a graph property $\mathcal{A}$ and a graph sequence $\tilde{G}_n$ we define the trace of $\mathcal{A}$ in $\tilde{G}_n$ as the binary sequence $\underline{a} = \underline{a}(\mathcal{A}, \mathcal{G}_n) = (a_0, \cdots, a_{\binom{n}{2}})$ such that $a_i = 1$ if and only if $G_i \in \mathcal{A}$. The *hitting time* of $\mathcal{A}$ in $\tilde{G}_n$ is defined as $h(\mathcal{A}, G_n) = \min\{i : a_i = 1\} \times (h(\mathcal{A}, \tilde{G}_n) = \binom{n}{2} + 1$ if all a_i are equal zero). If $\mathcal{A}$ is an increasing property then $G_M \in \mathcal{A}$ if and only if $h(\mathcal{A}, \tilde{G}_n) \leqq M$ and so the following reformulation of Theorem 1 is immediate.

THEOREM 1′. *Let* $x_n = M/n - \log n$. *Then*

$$\lim_{n\to\infty} \operatorname{Prob}(h(\mathcal{M}, \hat{G}_{2n}) \leqq n(\log n + x_n))$$

$$= \lim_{n\to\infty} \operatorname{Prob}(h(\mathcal{NT}, \hat{G}_{2n}) \leqq n(\log n + x_n))$$

$$= \begin{cases} 0 & \textit{if } x_n \to -\infty \\ \exp(-2e^{-x}) & \textit{if } x_n \to x \\ 1 & \textit{if } x_n \to \infty. \end{cases}$$

Bollobás and Frieze proved that actually the two hitting times almost surely coincide.

THEOREM 2 [BT85].

$$\lim_{n\to\infty} \operatorname{Prob}(h(\mathcal{M}, \hat{G}_{2n}) = h(\mathcal{NT}, \hat{G}_{2n})) = 1.$$

The meaning of this result is that in most instances vertex degrees tell us whether a graph has a perfect matching. This approach, appropriate whenever an increasing property $\mathcal{A}$ implies $\mathcal{B}$ and $h(\mathcal{A}, \hat{G}_n) = h(\mathcal{B}, \hat{G}_n)$, almost surely, fails for arbitrary $\mathcal{A}$. In principle, it may happen that $\underline{a}(\mathcal{A}, \hat{G}_n)$ has many 1-runs and then the fact that $h(\mathcal{A}, G_n) = h(\mathcal{B}, G_n)$ is useless. We cannot conclude anything about $h(\mathcal{A}, \hat{G}_n)$ from the knowledge of $G(n, M)$ (as we did deducing Theorem 1′) either. It is possible that for each $M = M(n) \operatorname{Prob}(G(n, M) \in \mathcal{A}) \to 0$ whereas $\operatorname{Prob}(G(n, M) \in \mathcal{A}$ for some M, $0 \leqq M \leqq \binom{n}{2}) \to 1$ as $n \to \infty$. (For instance, this is the case of the property that the maximum degree equals $\lfloor n/2 \rfloor$.) One way to overcome these difficulties is by proving that for almost all $\tilde{G}_n \in \tilde{\mathcal{G}}_n$ there is only one 1-run in the trace $\underline{a}(\mathcal{A}, \tilde{G}_n)$. We say that $\tilde{G}_n$ is *$\mathcal{A}$-increasing* if there is exactly one 1-run in $\underline{a}(\mathcal{A}, \tilde{G}_n)$ and $K_n \in \mathcal{A}$, i.e., $a_{\binom{n}{2}} = 1$.

In the next section we introduce two graph properties $\mathcal{M}'_l(T)$ and $\mathcal{N}_l(T)$. The first one is a modification of the fact that the largest component has a perfect T-matching where T is a tree. The latter is a carefully chosen necessary condition for $\mathcal{M}'_l(T)$. Our main result asserts that almost surely the graph process $\hat{G}_n$ is $\mathcal{M}'_l(T)$-increasing and, moreover, $h(\mathcal{M}'_l(T), \hat{G}_n) = h(\mathcal{N}_l(T), \hat{G}_n)$. The asymptotic distribution of $h(\mathcal{N}_l(T), \hat{G}_n)$ will be established by standard methods.

2. Statement of the result. Let $\mathcal{M}(T)$ be the property that a graph has a perfect T-matching, where T is a tree. A necessary condition for the existence of a perfect matching in G, when $|G|$ is even, is nonexistence of a "cherry," i.e., a pair of pendant vertices with a common neighbor. We are going to find a necessary condition for perfect T-matchings in terms of nonexistence of specified branches. We say that B is a branch of G with root v if B is an induced subgraph of G, $v \in V_B$, and for each $u \in V_B - \{v\}$, $\deg_B(u) = \deg_G(u)$. Assume first that $|G|$ is divisible by $|T|$. If G has a branch that is a star S on $|T| + 1$ vertices rooted at the center, then G has no perfect T-matching.

However, for P being a path with at least three edges, already a "cherry" is a constraint for a perfect P-matching. On the other hand, a branch that is a path of any length rooted at the endpoint does not exclude a perfect P-matching. As we can see, we need a detailed analysis of the structure of T in respect to the shapes of its branches. As it will become clear soon (see Lemma 2, § 3), we should be interested in the smallest trees that are not branches of T.

Let $\mathcal{U}_k$ be the family of rooted k-vertex trees. Families $\mathcal{U}_3$ and $\mathcal{U}_4$ are presented in Fig. 1. The roots are indicated by open circles.

Let $a(T)$ be the largest integer k such that for every $U \in \mathcal{U}_k$ there is a branch of T isomorphic to U (the isomorphism has to map the roots on each other). We denote by $\mathcal{A}(T)$ the set of all $U \in \mathcal{U}_{a(T)+1}$ for which there is no branch of T isomorphic to U. A branch of G isomorphic to a member of $\mathcal{A}(T)$ excludes a perfect T-matching with the exception of $T = P_i$, $i = 1, 2$, where P_i is the path of length i. (These two trees are special since for them and only for them $a(T) = |T|$.) But it may happen that a branch of G makes a perfect T-matching impossible even when it is isomorphic to a member of $\mathcal{U}_{a(T)}$. To see this, define the outdegree of a branch B with root v as

$$d^+_{B,G} = \deg_G(v) - \deg_B(v)$$

and set

$$d_{U,T} = \min\{d^+_{B,T} \colon B \text{ is a branch of } T \text{ isomorphic to } U\}.$$

Let B be a branch of T isomorphic to $U \in \mathcal{U}_{a(T)}$ with $d^+_{B,T} = d_{U,T}$. If there is a branch D of G isomorphic to U with outdegree smaller than $d^+_{B,T}$ then, clearly, there is no perfect T-matching in G. Let

$$b(T) = \max\{d_{U,T} \colon U \in \mathcal{U}_{a(T)}\}.$$

If $b(T) \leqq 1$ then, provided G is connected, such a situation cannot happen. Observe that $b(T) = 0$ if and only if $T = P_i$, $i = 1, 2$.

Now we are ready to list the types of branches whose appearance in G contradicts the existence of a perfect T-matching. We say that a branch B of G is r-attached if it has outdegree r. As we will learn from Lemma 3 of § 3, in case $b(T) > 1$, only $(b(T) - 1)$-attached branches of G isomorphic to a member of

$$\mathcal{D}(T) = \{U \in \mathcal{U}_{a(T)} \colon d_{U,T} - b(T)\}$$

will be of critical importance. Hence, we say that a branch B of G is T-excluding if

(i) $T = P_1$ and $B = U_2$, or

(ii) $T = P_2$ and $B \in \{W_2, W_3\}$, or

(iii) $b(T) > 1$ and B is isomorphic to a member of $\mathcal{A}(T)$, or

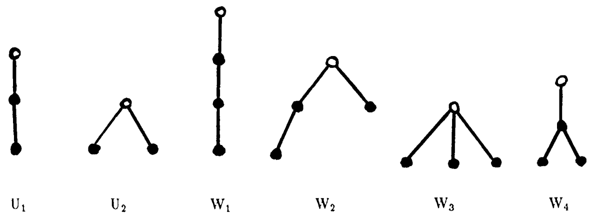


FIG. 1

(iv) $b(T) > 1$, B is $(b(T) - 1)$-attached and B is isomorphic to a member of $\mathcal{D}(T)$.

For better understanding we list below the introduced notions $a(T)$, $b(T)$, $\mathcal{A}(T)$ and $\mathcal{D}(T)$ for paths P_s, stars S_r, and the tree T_0 of Fig. 2.

	$a(T)$	$b(T)$	$\mathcal{A}(T)$	$\mathcal{D}(T)$
$P_1 = S_1 = K_2$	2	0	$\mathcal{U}_3$	$\mathcal{U}_2$
$P_2 = S_2$	3	0	$\mathcal{U}_4$	$\mathcal{U}_3$
$P_s: s \geqq 3$	2	1	$\{U_2\}$	$\{K_2\}$
$S_r: r \geqq 3$	2	$r-1$	$\{U_1\}$	$\{K_2\}$
T_0	3	2	$\{W_1\}$	$\{U_1\}$

(Here S_r stands for the star with r arms and trees U_1, U_2, W_1 and other members of families $\mathcal{U}_3$ and $\mathcal{U}_4$ are given in Fig. 1.)

Let $\mathcal{N}(T)$ be the property that a graph G has no T-excluding branch. Then $\mathcal{N}(T)$ is a necessary condition for $\mathcal{M}(T)$ provided $|G|$ is divisible by $|T|$. To avoid this constraint, let us alter slightly the property $\mathcal{M}(T)$ by imposing the restriction that all unmatched vertices have degrees greater than 1. The new property, denoted by $\mathcal{M}'(T)$, always implies $\mathcal{N}(T)$.

Another obstacle related to the fact that our object of interest is the largest component and not the whole graph is that, at the early stages of the graph process, $\hat{G}_n$, the largest component, may not be unique and, moreover, it "keeps changing places." Eventually, the process stabilizes in the sense that starting from some moment M_0 the vertex sets of the largest components of $G(n, M)$, $M > M_0$, are well defined (i.e., there is just one largest component) and form an increasing sequence of sets. The point M_0 lies somewhere around $n/2$ (see [B85]), so it is safe to assume that we begin to watch the process $\hat{G}_n$ after it acquires $\lceil cn \rceil$ edges for $c > \frac{1}{2}$. The constant c is fixed throughout the paper.

For a graph property $\mathcal{A}$ we denote by $\mathcal{A}_l$ the property that a graph G has more than $c|G|$ edges and the largest component of G has $\mathcal{A}$. By $\mathcal{C}(T)$ we denote those from families $\mathcal{A}(T)$, $\mathcal{D}(T)$ that will be of critical importance for us, namely

$$\mathcal{C}(T) = \begin{cases} \{U_2\} & \text{if } T = P_1 \\ \{W_2, W_3\} & \text{if } T = P_2 \\ \mathcal{A}(T) & \text{if } b(T) = 1 \\ \mathcal{D}(T) & \text{if } b(T) > 2 \end{cases},$$

whereas, when $b(T) = 2$, $\mathcal{C}(T)$ consists of rooted $(k+1)$-vertex trees U such that either $U \in \mathcal{A}(T)$ or $U = U' + v$ where $U' \in \mathcal{D}(T)$ and the additional vertex v, which is the root of U, is adjacent only to the root of U'.

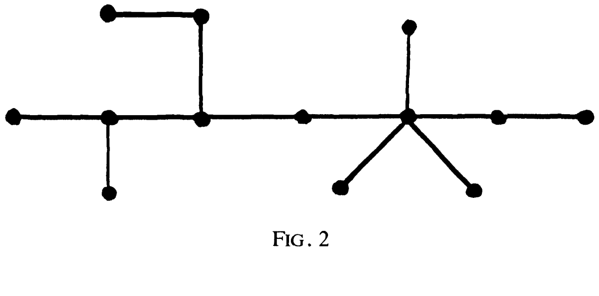


FIG. 2

At last, we are ready to formulate our main result.

THEOREM 3.

(i) *Almost surely, $\hat{G}_n$ is $\mathcal{N}_l(T)$-increasing*;

(ii)
$$\lim_{n\to\infty} \operatorname{Prob}\left(h(\mathcal{N}_l(T),\hat{G}_n)\leqq\frac{n}{2a(T)}(\log n+c(T)\log\log n+d_n)\right)$$
$$=\begin{cases} 0 & \textit{if } d_n\to-\infty,\ |d_n|=o(\log n)\\ e^{-\lambda} & \textit{if } d_n\to d\in(-\infty,\infty)\\ 1 & \textit{if } d_n\to\infty\end{cases},$$

where $c(T)=\max\{a(T),\, a(T)+b(T)-2\}$,

$$\lambda=\frac{e^{-d}}{r!k^{k+r-1}}\sum_{U\in\mathscr{C}(T)}\frac{1}{\operatorname{aut}(U)},$$

aut (U) *is the number of automorphisms of U that fix the root of U*,

$$r=\begin{cases} b(T)-1 & \textit{if } b(T)>1\\ 1 & \textit{if } b(T)\leqq 1,\end{cases}$$

and $k=a(T)$;

(iii) *Almost surely, $\hat{G}_n$ is $\mathcal{M}'_l(T)$-increasing and*

$$h(\mathcal{M}'_l(T),\hat{G}_n)=h(\mathcal{N}_l(T),\hat{G}_n).$$

COROLLARY 1. *For all trees T, almost surely*

(i) $h(\mathcal{M}'_l(T),\, \hat{G}_n) \leqq h(\mathcal{N}_l(P_1),\, \hat{G}_n)$

and

(ii) $h(\mathcal{M}'(T),\, \hat{G}_n) = h(\mathcal{N}\mathscr{T},\, \hat{G}_n)$.

Let $\mathcal{N}\mathscr{T}_k$ be the property that the second largest component has less than k vertices. In 1960 Erdős and Rényi proved that for

$$M=\frac{n}{2k}(\log n+(k-1)\log\log n+d_n)$$

$$(1)\quad \lim_{n\to\infty}\operatorname{Prob}(G(n,M)\in\mathcal{N}\mathscr{T}_k)=\begin{cases} 0 & \text{if } d_n\to-\infty,\ |d_n|=o(\log n)\\ \exp(-k^k e^{-d}/k!) & \text{if } d_n\to d\\ 1 & \text{if } d_n\to\infty.\end{cases}$$

The lemma below supplements that result.

LEMMA 1. *Let $\mathcal{N}\mathscr{T}'_k$ be the property that a graph G has more than c$|$G$|$ edges and $G\in\mathcal{N}\mathscr{T}_k$. Then, almost surely, $\hat{G}_n$ is $\mathcal{N}\mathscr{T}'_k$-increasing and*

$$\lim_{n\to\infty}\operatorname{Prob}\left(h(\mathcal{N}\mathscr{T}'_k(T),\hat{G}_n)\leqq\frac{n}{2k}(\log n+(k-1)\log\log n+d_n)\right)$$

$$(2)\qquad =\begin{cases} 0 & \textit{if } d_n\to-\infty,\ |d_n|=o(\log n)\\ \exp(-k^k e^{-d}/k!) & \textit{if } d_n\to d\\ 1 & \textit{if } d_n\to\infty.\end{cases}$$

Of course, provided the first part of the assertion holds, (2) is just a reformulation of (1). Since $|T| \geqq a(T)$, almost surely, $\hat{G}_n$ hits $\mathcal{N}_l(t)$ at the moment when each component different from the largest one has less than $|T|$ vertices. Thus Theorem 3 can be stated in a slightly stronger form.

THEOREM 3′. *Let $\mathcal{M}'_a(T)$ $[\mathcal{N}_a(T)]$ be the property that a graph G has at least $c|G|$ edges and all components with at least $|T|$ vertices have property $\mathcal{M}'(T)$ $[\mathcal{N}(T)]$. Then, almost surely, $\hat{G}_n$ is $\mathcal{M}'_a(T)$-increasing and*

$$h(\mathcal{M}'_a(T),\hat{G}_n)=h(\mathcal{N}_a(T),\hat{G}_n)=h(\mathcal{N}_l(T),\hat{G}_n).$$

The above theorems and their corollaries immediately imply respective results for the random graph $G(n, M)$.

COROLLARY 2. *For $M = (n/2a(T))(\log n + c(T)\log\log n + d_n)$,*

$$\lim_{n\to\infty} \operatorname{Prob}(G(n,M)\in\mathcal{M}'_l(T))=\lim_{n\to\infty} \operatorname{Prob}(G(n,M)\in\mathcal{N}_l(T))$$

$$=\begin{cases} 0 & \text{if } d_n\to-\infty,\ |d_n|=o(\log n)\\ e^{-\lambda} & \text{if } d_n\to d\in(-\infty,\infty)\\ 1 & \text{if } d_n\to\infty,\end{cases}$$

where λ is given in Theorem 3.

COROLLARY 3. *If $M = (n/4)(\log n + 2\log\log n + d_n)$, $d_n \to \infty$, then, for every tree T, almost surely, the subgraph of $G(n, M)$ induced by vertices of nonzero degrees coincides with the largest component and has a perfect T-matching.*

COROLLARY 4. *If $M = (n/2)(\log n + d_n)$, $d_n \to \infty$, then, for every tree T, almost surely, $G(n, M)$ has a perfect T-matching.*

3. The proof. Part (i) of the theorem is a consequence of the following two lemmas, which will be proved in § 4.

LEMMA 2. *For $U\in\mathcal{U}_{k+1}$, let $\mathcal{B}(U)$ be the property that a graph contains no branch isomorphic to U. Then, almost surely, $\hat{G}_n$ is $\mathcal{B}_l(U)$-increasing and*

$$\lim_{n\to\infty} \operatorname{Prob}\left(h(\mathcal{B}_l(U),\hat{G}_n)\leqq\frac{n}{2k}(\log n+k\log\log n+d)\right)=e^{-\lambda},$$

where $\lambda = (e^d k^k \operatorname{aut}(U))^{-1}$ and $\operatorname{aut}(U)$ is the number of automorphisms of U that fix its root.

LEMMA 3. *For $U\in\mathcal{U}_k$ and $r = 1, 2, \cdots$, let $\mathcal{B}(U, r)$ be the property that a graph contains no r-attached branch isomorphic to U. Then, almost surely, $\hat{G}_n$ is $\mathcal{B}_l(U, r)$-increasing and*

$$\lim_{n\to\infty} \operatorname{Prob}\left(h(\mathcal{B}_l(U,r),\hat{G}_n)\leqq\frac{n}{2k}(\log n+(k+r-1)\log\log n+d)\right)=e^{-\lambda},$$

where $\lambda = (e^d k^{k+r-1} r!\operatorname{aut}(U))^{-1}$.

Part (ii) can be shown using the routine method of moments, similarly to the proof of (5) presented in § 4. To prove part (iii) we will define a set of properties $\mathcal{A}$ which together with $\mathcal{N}_l(T)$ imply $\mathcal{M}'_l(T)$. We will prove that, almost surely, for all $M' < M < n\log n$, $M' = (n/2a(T))\log n$, $G(n,M)\in\mathcal{A}$. Knowing already that, almost surely, $h(\mathcal{N}_l(T)) > M'$ and $\hat{G}_n$ is $\mathcal{N}_l(T)$-increasing, this allows us to conclude that $h(\mathcal{M}'_l(T), \hat{G}_n) = h(\mathcal{N}_l(T), \hat{G}_n)$, almost surely. Moreover, the property that $G \in \mathcal{M}'_l(T)$ and G is connected is increasing. Therefore, keeping in mind that the hitting time of connectivity in $\hat{G}_n$ is approximately $n\log n/2$ (see Lemma 1 above with $k = 1$), $\hat{G}_n$ is, almost surely, $\mathcal{M}'_l(T)$-increasing.

The remainder of this section is devoted to proving that

$$\mathscr{A} \cap \mathscr{N}_l(T) \subseteq \mathscr{M}'_l(T). \tag{3}$$

Despite technical details, the idea of our proof is simple. Let T be a tree with $V(T) = [t]$ and let $V_1, \cdots, V_t$ be a partition of $[n]$ into equal size sets (for a moment assume that n is divisible by t). To show that a graph on vertex set $[n]$ has a perfect T-matching, it is enough to find perfect K_2-matchings in $t-1$ bipartite graphs generated by pairs $\{V_i, V_j\}$ for which $\{i, j\} \in E(T)$. An obvious weakness of this straightforward approach is that we "lose" all edges within the sets V_i. With some more effort it can be refined to gain the required result. To do so, we distinguish "bad" vertices that have few neighbors in at least one of V_i. First we match bad vertices. Then we modify the partition by deleting matched vertices, moving so called "safe" vertices to keep the partition even. Then all bipartite graphs generated by the pairs of sets of the new partition, have large minimum degree and therefore satisfy the Hall's condition.

Now the details come. Let us set $k = a(T)$, $V(T) = [t]$ and partition $[n] = V_1 \cup \cdots \cup V_t$ so that $||V_i| - |V_j|| < 1$ for all i and j. We define $d_i(v) = |N(v) \cap V_i|$, $i = 1, \cdots, t$ and $d(v) = d_1(v) + \cdots + d_t(v)$. We call v bad if, for some $i \in [t]$, $d_i(v) < \log n/50kt$; otherwise, v is called good. A vertex v is small if $d(v) < 5kt^2$; otherwise, v is called large.

DEFINITION (Property $\mathscr{A}$). A graph G on vertex set $[n]$ is said to have property $\mathscr{A}$ if

(i) G has no more than $n/\log^{20t} n$ bad vertices,
(ii) no $2kt$ bad vertices are within distance $10t$ from each other,
(iii) no k small and 1 bad vertices are within distance $10t$ from each other,
(iv) no small vertex lies on a cycle of length less than $3kt$,
(v) for all pairs of disjoint subsets of $[n]$ of size $|S_1| = |S_2| = n(\log\log n)^2/\log n$ there is an edge from S_1 to S_2,
(vi) every subset $S \subset [n]$ with $|S| < 2n(\log\log n)^2/\log n$ contains less than $(\log\log n)^3|S|$ edges,
(vii) there are less than $n/\log^{20t} n$ vertices outside the largest component,
(viii) the maximum degree is smaller than $6 \log n$.

(The quantities $2kt$, $10t$, and so on are quite arbitrary.)

LEMMA 4. *For $M' = (n/2k)\log n$, almost surely, a random graph process $\hat{G}_n$ is such that, for all M satisfying $M' \leqq M \leqq n \log n$, $G(n, M) \in \mathscr{A}$.*

The proof of Lemma 4 is postponed to § 4.

The proof of (3). We ignore all vertices outside L, the largest component of G. Assume that $v_1, \cdots, v_m$ are the bad vertices in L and that $d(v_1) \leqq \cdots \leqq d(v_m)$. Now we describe a procedure matching the bad vertices into copies of T. Our variables are H—the graph induced by yet unmatched vertices and v—a bad vertex in H with the lowest index. At the beginning we set $H = L$ and $v = v_1$. Actually, we show only how the procedure matches bad vertices into branches of T such that either the root or r of its neighbors (when the branch is r-attached in T) are good vertices of G. These branches are immediately extended to a whole copy of T. Such extensions are possible due to property $\mathscr{A}$(ii). Below M stands for the branch and $\bar{M}$ for the copy of T matched at the current stage.

Description of the procedure.

I. Assume first that v is small and choose u to be the nearest large vertex from v. Let P be the shortest path linking v and u. By $\mathscr{A}$(iii), $|P| \leqq k + 1$. Consider a component C of $H - V(P)$ with $|C| < kt$ and a vertex x in C. By $\mathscr{A}$(iv), $d_H(x) \leqq kt$. Moreover, $d_L(x) - d_H(x) < 2kt^2$, since, by $\mathscr{A}$(ii), at most $2kt$ already matched bad vertices were joined to x and, in addition, $N_L(x) - N_H(x)$ could have contained at most $2kt(t-1)$ good vertices. Hence, C contains only small vertices, $|C| < k$, and, by $\mathscr{A}$(iv), C is a branch of H rooted at a vertex of P. Let us denote by B the subgraph of H induced by P and the components of $H - V(P)$ of order at most k. Clearly, B is a tree. Coming back to the choice of u, we assume that u is picked to minimize the order of branch of B hanging at u. In general, by $\mathscr{A}$(iii), $|B| \leqq k + 1$.

I1. However, if u is bad then $|B| \leqq k$.

I1a. $|B| \leqq k - 1$.

Each vertex has at least $4kt^2$ good neighbors in L and at least $2kt^2$ good neighbors in H. Let w be a good neighbor of u in H. Set $M = B \cup \{uw\}$. By the definition of k there is a branch of T isomorphic to M with w as the root.

I1b. $|B| = k$.

B is isomorphic to a branch of T with u as the root. We pick $d = d_{B,T}$ good neighbors of u, $w_1, \cdots, w_d$, and set $M = B \cup \{uw_1, \cdots, uw_d\}$.

I2. Ironically, the case when u is good is more complicated.

I2a. But not if $|B| \leqq k$. Then $M = B$ is ready.

I2b. Assume, hence, that $|B| = k + 1$.

Let w be the neighbor of u in P. For all $x \in V(B) - \{u, w\}$, $d_L(x) = d_H(x) = d_B(x)$ by $\mathscr{A}$(iii).

I2b(i). If $d_L(w) = d_B(u)$ then B is a branch of L and, by $\mathscr{N}_l(T)$, there is a branch of T isomorphic to B with u as the root. Set $M = B$.

I2b(ii). Otherwise, w has a good neighbor different from u. In such case, by the special choice of u, $d_B(u) = 1$ (again, by $\mathscr{A}$(iii)). Thus $B - u$ is a branch of L of order k with w as the root. By $\mathscr{N}_l(T)$ the out-degree of w must be at least $d = d_{B-u,T}$. Set $M = B \cup \{ww_1, \cdots, ww_d\}$, where $w_1, \cdots, w_d$ are good neighbors of w (u is among them).

Once we have gotten through the early phase of the procedure and matched all small vertices there are no difficulties any longer.

II. If v is large, choose u among good neighbors of v. Set $M = B = P = \{vu\}$.

After each round is completed we substitute H for $H - \bar{M}$. Since, by $\mathscr{A}$(ii), each vertex is joined to at most $2kt^2$ (good or bad) already matched vertices, each vertex of L', the subgraph of L induced by yet unmatched vertices, has at least $\log n/50kt - 2kt^2 > \log n/51kt$ neighbors in each set $V'_i = V_i \cap L'$, $i = 1, \cdots, t$. By $\mathscr{A}$(i) and $\mathscr{A}$(vii) $|L'| > n - f$, $f = 2tn/\log^{20t} n$ and so, for all $i, j \in [t]$, $||V'_i| - |V'_j|| < f$. To balance the partition, let us choose a set W of "safe" vertices such that for each $i \in [t]$ $|W \cap V'_i| > tf$ and no two vertices of W are within distance two from each other. The existence of W follows from the fact that $\Delta(L') \leqq \Delta(G) < 6 \log n$. Indeed, W can be defined recursively. After including vertex x to W cross out the set $N^{(2)}$, $|N^{(2)}| < 36 \log^2 n$, of all vertices lying within distance 2 from x and repeat this step. (Let us recall that $|V'_i| \sim |V_i| \sim n/t$.) Now we move "safe" vertices around and, possibly, delete up to $t - 1$ of them to obtain a partition $V''_1, \cdots, V''_t$ of L' satisfying $|V''_i| = \lfloor |L'|/t \rfloor = h$, $i \in [t]$. Let us focus on the bipartite graph F induced in L' by (V''_i, V''_j). By the careful choice of W, $\delta(F) > \log n/51kt - 1 > \log n/52kt$. To finish the proof of (3) we must find a perfect matching in F. Due to Hall's theorem it is enough to check whether, for each $S \subseteq V_i''$, the set $N(S)$ of neighbors of S in V_j'' has at least $|S|$ elements. Suppose there is S with $|N(S)| < |S|$ and consider two cases.

Case 1. $|S| \leqq g = n(\log\log n)^2/\log n$.
Then $|S \cup N(S)| < 2g$ which contradicts $\mathscr{A}(\mathrm{vi})$.
Case 2. $|S| > g$.

Since there is no edge between S and $\overline{N(S)} = V_j'' - N(S)$, $|\overline{N(S)}| \leqq g$ by $\mathscr{A}(v)$. Let $N(\overline{N(S)})$ be the set of neighbors of $\overline{N(S)}$ in V_i''. Arguing as in Case 1, we can show that $|N(\overline{N(S)})| \geqq |\overline{N(S)}|$. Thus $|N(S)| = h - |\overline{N(S)}| \geqq h - |N(\overline{N(S)})| \geqq |S|$, since $S \cap N(\overline{N(S)}) = \varnothing$.

This completes the proof of (3) and therefore the proof of Theorem 3. □

4. The proof of lemmas. In this section we will be frequently using the estimate

$$\binom{A-a}{B-b}\Big/\binom{A}{B} \sim \left(\frac{B}{A}\right)^b \exp\left(-\frac{aB}{A}\right),$$

where A, B, a, b are functions of n for which $B = o(A)$, $b = o(a)$, $a^2 = O(A)$, and $b^2 = o(B)$.

Proof of Lemma 1. Let $M_0 = \lfloor cn \rfloor$, $M_1 = (n/2k)(\log n - \log\log\log n)$, $M_2 = (n/2k)(\log n + (k-1)\log\log n - \log\log\log n)$, $M_3 = n\log n$.

Let X_i be the number of isolated paths P_{k-1} in $G(n, M_i)$, $i = 0, 1$, and let Y_i be the number of those isolated P_{k-1} of $G(n, M_{i-1})$ that are still isolated in $G(n, M_i)$, $i = 1, 2$. We have

$$EX_i = \binom{n}{k}\frac{(k-1)!}{2}\binom{\binom{n-k}{2}}{M_i-k+1}\binom{\binom{n}{2}}{M_i}^{-1}$$

$$\sim \frac{n}{2k}\left(\frac{2M_i}{n}\right)^{k-1}\exp\left(-\frac{2kM_i}{n}\right) \sim \begin{cases} \dfrac{(2c)^{k-1}}{2k}e^{-2kc}n & \text{if } i=0 \\[2ex] \dfrac{1}{2k^k}\log^{k-1} n \log\log n & \text{if } i=1 \end{cases}$$

and

$$E_2X_i = EX_i(X_i-1)$$
$$= \binom{n}{k}\binom{n-k}{k}\left(\frac{(k-1)!}{2}\right)^2\binom{\binom{n-2k}{2}}{M_i-2k+2}\binom{\binom{n}{2}}{M_i}^{-1} \sim (EX_i)^2, \quad \text{for } i=0,1.$$

Hence, by Chebyshev's inequality, almost surely, $X_i > EX_i/2$, for $i = 0, 1$. Clearly,

$$\operatorname{Prob}(Y_i=0) \sim \sum_{l>a} \operatorname{Prob}(Y_i=0 \mid X_{i-1}=l)\operatorname{Prob}(X_{i-1}=l), \quad \text{where } a = \lceil EX_i/2 \rceil.$$

But $\operatorname{Prob}(Y_i = 0 \mid X_{i-1} = l)$ is a decreasing function of l and we must only prove that

$$(4) \qquad \operatorname{Prob}(Y_i=0 \mid X_{i-1}=a) \to 0.$$

We have

$$E(Y_i | X_{i-1}=a)= a \left(\binom{\binom{n-k}{2}-M_{i-1}+k-1}{M_i-M_{i-1}} \right) \left(\binom{\binom{n}{2}-M_{i-1}}{M_i-M_{i-1}} \right)^{-1}$$

$$\sim a \exp\left(-\frac{2k}{n}(M_i-M_{i-1}) \right) = \mu.$$

Similarly, $E_2(Y_i | X_{i-1} = a) \sim \mu^2$, and (4) follows by Chebyshev's inequality. Hence, almost surely, the graph process $\hat{G}_n$ is such that for all $M_0 \leqq M \leqq M_2$ there is a component of order k in $G(n, M)$.

Let $Z = |\{M : M_2 < M < M_3,\ G(n, M)$ has a k-vertex component that was not present in $G(n, M - 1)\}|$. We will show that, almost surely, $Z = 0$. Since a new k-component can only emerge by joining two smaller components,

$$EZ \leqq \sum_{M=M_2}^{M_3} \binom{n}{k} k^{k-2}(k-1) \left(\binom{\binom{n-k}{2}}{(M-1)-(k-2)} \right) \left(\binom{\binom{n}{2}}{M-1} \right)^{-1} \frac{1}{\binom{n}{2}-(M-1)}$$

$$= O(1) \sum_{M} \left(\frac{2M}{n} \right)^{k-2} \left(1 - \frac{2k}{n} \right)^{M} = O(1) \left((M_2' - M_2) f\left(\frac{2M_2}{n} \right) + M_3 f\left(\frac{2M_2'}{n} \right) \right)$$

$$= o(1),$$

where $M_2' = M_2 + (n/2k) \log \log n$ and $f(x) = x^{k-2}e^{-kx}$. (Note that f is decreasing in the interval $(1 - 2/k, \infty)$.)

Since, almost surely, $G(n, M_3)$ is connected (see [ER59]), almost surely, the graph process $\hat{G}_n$ is such that, after moment M_2, no new k-component is created. Hence, for every natural k, almost surely, $\hat{G}_n$ is $\mathscr{N}\mathscr{T}_k^*$-increasing, where $\mathscr{N}\mathscr{T}_k^*$ is the property that a graph G has at least $c|G|$ edges and there is no k-component in G. We already know that, almost surely, $G(n, M_2)$ has no l-component for $l \geqq k + 1$. The first new l-component, $l \geqq k + 1$ after moment M_2 could be created only by joining two components of order at most k and therefore l would be at most $2k$. This is unlikely, since $\hat{G}_n$ is, almost surely, $\mathscr{N}\mathscr{T}_l^*$-increasing, simultaneously for all $l = k + 1, \cdots, 2k$. This implies that, almost surely, after moment M_2 no new l-component, $l > k$, is created and so $\hat{G}_n$ is, almost surely, $\mathscr{N}\mathscr{T}_k'$-increasing. □

Proof of Lemmas 2 *and* 3. Let U be a tree and $W \subset V(U)$, $W \neq \emptyset$ such that each vertex of $V(U) - W$ is joined to a vertex from W. An induced subgraph H of graph G is called a (U, W)-subgraph of G if there is an isomorphism σ between U and H such that for each $x \in W$, $d_H(\sigma(x)) = d_G(\sigma(x))$. If $|W| = |U| - 1$, this coincides with the notion of a branch introduced in § 2. The following lemma is a common generalization of Lemmas 2 and 3. (An r-attached tree-branch B of G with root v can be interpreted as a (U, W)-subgraph with $U = (V(B) \cup N_G(v), E(B) \cup \{\{v, u\} : u \in N_G(v)\})$ and $W = V(B)$.)

LEMMA. *Let $\mathscr{B}(U, W)$ be the property that a graph G has no (U, W)-subgraph. Then, almost surely, $\hat{G}_n$ is $\mathscr{B}_l(U, W)$-increasing and*

$$(5) \qquad \lim_{n \to \infty} \operatorname{Prob}\left(h(\mathscr{B}_l(u, W), \hat{G}_n) < \frac{n}{2|W|}(\log n + (|U| - 1)\log\log n + d)\right) = e^{-\lambda},$$

where $\lambda = (e^d w^{u-1}$ aut $(U, W))^{-1}$ *and* aut (U, W) *is the number of automorphisms σ of U for which $\sigma(W) = W$.*

Proof of lemma. Let $|W| = w$, $|U| = u$, $M_{-1} = (1+c)n/2$, $M_0 = cn$, $M_1 = (n/2w)(\log n - \log\log\log n)$, $M_2 = (n/2w)(\log n + (u-1)\log\log n - \log\log\log n)$, $M_3 = n \log n$, *and*

$$M(d) = \frac{n}{2w}(\log n + (u-1)\log\log n + d).$$

First, we will prove (5) by showing that the number X of (U, W)-subgraphs in $G(n, M(d))$ converges in distribution to a Poisson random variable with expectation λ. By Lemma 1, almost surely, (U, W)-subgraphs of $G(n, M(d))$ belong to the largest component. We have, with $h = \binom{u}{w}(w!(u-w)!/\operatorname{aut}(U, W))$,

$$EX = \binom{n}{u} h \binom{\binom{n-w}{2} + \binom{u-w}{2} + w(u-w)}{M(d)-(u-1)} \binom{\binom{n}{2}}{M(d)}^{-1} \sim \lambda$$

and

$$\begin{aligned} EX(X-1) = {} & \binom{n}{u}\binom{n-u}{u} h^2 \binom{\binom{n-2w}{2} + (u-w)(u-w-1) + 2w(u-w)}{M(d)-2(u-1)} \binom{\binom{n}{2}}{M(d)}^{-1} \\ & + O(1) \sum_{l=1}^{u-w} \binom{n}{u}\binom{n-u}{u-l} \\ & \times \binom{\binom{n-2w}{2} + (u-w-l)(u-w-l-1) + \binom{l}{2} + w(2u-w-l)}{M(d)-2(u-1)+l-1} \\ & \times \binom{\binom{n}{2}}{M(d)}^{-1} \\ & \sim \lambda^2 + O\left(\frac{1}{n}\right), \end{aligned}$$

where l stands for the number of common vertices in two (U, W)-subgraphs. Note that two different (U, W)-subgraphs must be W-disjoint. Similarly, one can prove that $EX(X-1)\cdots(X-r+1) \to \lambda^r$ for $r = 3, 4, \cdots$ and (5) is shown.

Let X_{-1} be the number of isolated U-trees in $G(n, M_{-1})$. By the second moment method one can easily prove that, almost surely, $X_{-1} > \alpha n$ for some constant $\alpha = \alpha(c) > 0$. Moreover, it is known (see [ER60]) that, almost surely, the largest component L_{-1} of $G(n, M_{-1})$ has more than βn vertices, $\beta = \beta(c) > 0$.

Let Y_0 count isolated U-trees of $G(n, M_{-1})$ that are joined by an edge with $V(L_{-1})$ and are (U, W)-subgraphs of $G(n, M_0)$. (For each isolated U-tree of $G(n, M_{-1})$, from among all isomorphic choices of W, we fix them lexicographically first.) Clearly,

$$\begin{aligned} \operatorname{Prob}(Y_0 = 0) & \sim \sum_{x \geq \alpha n} \sum_{l \geq \beta n} \operatorname{Prob}(Y_0 = 0 \mid X_{-1} = x, |L_{-1}| = l) \operatorname{Prob}(X_{-1} = x, |L_{-1}| = l) \\ & \leq \operatorname{Prob}(Y_0 = 0 \mid X_{-1} = \alpha n, |L_{-1}| = \beta n), \end{aligned}$$

since the conditional probability is a decreasing function of x and l. We have

$$E(Y_0 \,|\, X_{-1} = \alpha n,\ |L_{-1}| = \beta n)$$
$$= \alpha n \left(\binom{\binom{n-w}{2} + \binom{u-w}{2} + w(u-w) - M_{-1} + w}{M_0 - M_{-1}} \right.$$
$$- \left(\binom{\binom{n-w}{2} + \binom{u-w}{2} + w(u-w) - M_{-1} + w - (u-w)\beta n}{M_0 - M_{-1}} \right.$$
$$\times \binom{\binom{n}{2} - M_{-1}}{M_0 - M_{-1}}^{-1}$$
$$\sim (u-w)\beta(c-1)\alpha n \binom{\binom{n-w}{2} + \binom{u-w}{2} + M_{-1}}{M_0 - M_{-1}} \binom{\binom{n}{2} - M_{-1}}{M_0 - M_{-1}}^{-1}$$
$$\sim \alpha\beta(c-1)(u-w)\exp(-w(c-1))n.$$

With only some more effort we can compute $E(Y_0(Y_0 - 1) \,|\, X_{-1} = \alpha n,\ |L_{-1}| = \beta n)$ and deduce, using Chebyshev's inequality that, almost surely, $Y_0 > \alpha' n'$ for some $\alpha' > 0$. Note that (U, W)-subgraphs counted by Y_0 are vertex-disjoint. Now, let Y_1 be the number of those (U, W)-subgraphs counted by Y_0 which remain (U, W)-subgraphs of $G(n, M_1)$. Conditioning on Y_0 we can easily show that, almost surely, $Y_1 > 0$. Note that due to Lemma 1, almost surely, every (U, W)-subgraph of $G(n, M_1)$ belongs to the largest component of it. Thus, almost surely, $\hat{G}_n$ is such that for all $M_0 \leqq M \leqq M_1$ the largest component of $G(n, M)$ contains a (U, W)-subgraph. To cover the period (M_1, M_2) we can prove using the same techniques that, almost surely, there are at least $\alpha'' \log\log n$ disjoint (U, W)-subgraphs in $G(n, M_1)$, where $\alpha'' > 0$, and at least one of them remains a (U, W)-subgraph in $G(n, M_2)$. Let $Z = |\{M : M_2 \leqq M \leqq M_3,\ G(n, M)$ has a (U, W)-subgraph that was not present in $G(n, M-1)\}|$. By Lemma 1, almost surely, for $M \geqq M_2$ the largest component cannot "catch" an isolated U-tree, so

$$EZ \leqq o(1) + O(1) \sum_{M=M_2}^{M_3} n^u \binom{\binom{n-w}{2}}{M-1-u+2} \binom{\binom{n}{2}}{M-1}^{-1} \frac{1}{\binom{n}{2} - M + 1}$$
$$= O(1) \sum_M \left(\frac{2M}{n}\right)^{u-2} \exp\left(-\frac{2wM}{n}\right) = o(1).$$

By (5), almost surely, there is no (U', W')-subgraph in $G(n, M_3)$ for all $U' \subseteq U$, $W \subseteq V(U')$. Since this is an increasing property, almost surely, $\hat{G}_n$ is such that for all $M \geqq M_3$ there is no (U, W)-subgraph in $G(n, M)$. Summarizing, we have proved that, almost surely, after moment M_2 no new (U, W)-subgraph emerges. Hence, almost surely, $\hat{G}_n$ is $\mathcal{B}_l(U, W)$-increasing. □

The proof of Lemma 4. Properties $\mathcal{A}$(i), $\mathcal{A}$(v), $\mathcal{A}$(vii), [$\mathcal{A}$(vi), $\mathcal{A}$(viii)] are increasing [decreasing] and therefore it is enough to prove that, almost surely,

$G(n, M')$ $[G(n, n \log n)]$ possesses them. This can be done by showing that the expectation of the number of respective objects is asymptotically small and by then applying Markov's inequality (see [B85], [BF85], [Ł87] for similar proofs). Properties $\mathscr{A}$(ii)–$\mathscr{A}$(iv) are not monotone and the proofs are similar to each other and to those of previous lemmas. We restrict ourselves to present the proof of $\mathscr{A}$(ii).

Let X count sets of $2kt$ bad vertices of $G(n, M')$ that are within distance $10t$ from each other (call them bunches) and let Y_l be the number of l-vertex trees, $2kt \leqq l \leqq 20kt^2$, containing at least $2kt$ bad vertices. Then, clearly,

$$\begin{aligned} EY_l \sim & \binom{n}{l}\binom{l}{2kt} l^{l-2} \sum_{m_1, \cdots, m_{2kt} < \log n/50kt} t^{2kt} \prod_{j=1}^{2kt} \binom{n/t}{m_j} \\ & \times \binom{\binom{n}{2} - 2kt(n/t)}{M' - \sum_i m_i - l + 1} \binom{\binom{n}{2}}{M'}^{-1} \\ & = O(1) n^l \sum_{m_i} \prod_j \left(\frac{en/t}{m_j} \frac{2M'}{n^2}\right)^{m_j} \left(\frac{2M'}{n^2}\right)^{l-1} \exp\left(-\frac{4kM'}{n}\right) \\ & = O(1) n^{-1} (\log n)^{2kt+l-1} (50e)^{\log n/25} \\ & = O(n^{-0.9+0.24}) = o(1), \end{aligned}$$

since the function $f(x) = (c/x)^x$ is increasing for $c > ex$. Therefore,

$$EX = \sum_{l=2kt}^{20kt^2} EY_l = o(1).$$

Let $Z = |\{M : M' \leqq M \leqq n \log n, G(n, M)$ has a bunch that was not present in $G(n, M-1)\}|$. Clearly, the appearance of a new bunch causes the appearance of a new l-vertex tree, $2kt \leqq l \leqq 20kt^2$, containing at least $2kt$ bad vertices. Let Z_l count how many times in the period $(M', n \log n)$ of the process $\hat{G}_n$ such a tree emerges. Then

$$\begin{aligned} EZ_l \leqq & \sum_{M=M'}^{n \log n} \binom{n}{l}\binom{l}{2kt} l^{l-2} \sum_{m_1, \cdots, m_{2kt} < \log n/50kt} t^{2kt} \prod_{j=1}^{2kt} \binom{n/t}{m_j} \binom{\binom{n}{2} - 2kn}{M - 1 - \sum_i m_i - l + 1} \\ & \times \binom{\binom{n}{2}}{M-1}^{-1} \frac{1}{\binom{n}{2} - M + 1} \\ & = O(1) \sum_M n^{l-2} \sum_{m_i} \prod_j \left(\frac{en/t}{m_j} \frac{2M}{n^2}\right)^{m_j} \left(\frac{2M}{n^2}\right)^{l-2} \exp\left(-\frac{4kM}{n}\right) \\ & = O(1)(\log n)^{2kt+l-2} \sum_M \left(\frac{100\, ekM}{n \log n}\right)^{\log n/25} \exp\left(-\frac{4kM}{n}\right). \end{aligned}$$

The last series can be bounded from above by a geometric series with the same first term and the quotient

$$\exp\left(-(1+o(1))\left(\frac{4k}{n} - \frac{\log n}{25M'}\right)\right) < \exp\left(-\frac{3k}{n}\right).$$

Thus

$$\begin{aligned} EZ_l &\leqq O(1)(\log n)^{2k+l-2}\frac{n}{3k}\left(\frac{100\,ekM'}{n\log n}\right)^{\log n/25}\exp\left(-\frac{4kM'}{n}\right)\\ &= O(1)(\log n)^{2kt+l-1}n^{-1}(50e)^{\log n/25}\\ &= O(n^{-0.9+0.24}) = o(1). \end{aligned}$$

This completes the proof of $\mathscr{A}$(ii). □

REFERENCES

[B85] B. Bollobás, *Random Graphs*, Academic Press, New York, 1985.

[BF85] B. Bollobás and A. M. Frieze, *On matchings and Hamiltonian cycles in random graphs*, Ann. Discrete Math., 28 (1985), pp. 23–46.

[BT85] B. Bollobás and A. Thomason, *Random graphs of small order*, Ann. Discrete Math., 28 (1985), pp. 47–98.

[ER59] P. Erdős and A. Rényi, *On random graphs* I, Publ. Math. Debrecen, 6 (1959), pp. 290–297.

[ER60] ———, *On the evolution of random graphs*, MTA Mat. Kut. Int. Közl., 5 (1960), pp. 17–61.

[ER66] ———, *On the existence of a factor of degree one of a connected random graph*, Acta Math. Acad. Sci. Hung., 17 (1966), pp. 359–368.

[Ł87] T. Łuczak, *On matching and Hamiltonian cycles in subgraphs of random graphs*, Ann. Discrete Math., 33 (1987), pp. 171–185.

SIAM J. DISC. MATH.
Vol. 4, No. 1, pp. 121–129, February 1991

THE EXPECTED CAPACITY OF CONCENTRATORS*

NICHOLAS PIPPENGER†

Abstract. The *expected capacity* of a class of sparse concentrators called *modular concentrators* is determined. In these concentrators, each input is connected to exactly two outputs, each output is connected to exactly three inputs, and the *girth* (the length of the shortest cycle in the connexion graph) is large. Two definitions of expected capacity are considered. For the first (which is due to Masson and Morris), it is assumed that a batch of customers arrive at a random set of inputs and that a maximum matching of these customers to servers at the outputs is found. The number of unsatisfied requests is negligible if customers arrive at fewer than one-half of the inputs, and it grows quite gracefully even beyond this threshold. The situation in which customers arrive sequentially is considered, and the decision as to how to serve each is made randomly, without knowledge of future arrivals. In this case, the number of unsatisfied requests is larger but still quite modest.

Key words. communication network, maximum matching, branching process, random packing

AMS(MOS) subject classifications. 68E10, 94C15

1. Batch arrivals.

For the purposes of this paper, a *concentrator* is a bipartite graph $G = (A, B, E)$ comprising a set A of *inputs*, a set B of *outputs*, and a set $E \subseteq A \times B$ of *edges*. The intended interpretation is that the inputs correspond to "customers," the outputs correspond to "servers," and the edges correspond to "channels" or "switches," each capable of providing direct access by a given customer to a given server.

We consider two modes of operation for a concentrator. In the first mode, the operation of the concentrator takes place in "cycles," each of which has two "phases." During the first phase, a subset $X \subseteq A$ of the inputs, called the *requesting inputs*, is chosen. This represents the arrival of a "batch" of customers. During the second phase, a maximum matching $M \subseteq E \cap (X \times B)$ between the requesting inputs and the outputs is chosen. This represents the action of a controller granting access to servers to as many customers as possible. The cardinality $\#X$ is called the *offered traffic*; $\#M$ is called the *carried traffic*; and $\#X - \#M$ is called the *lost traffic*.

The *actual capacity* of a concentrator is the largest k such that the carried traffic is k for all $X \subseteq A$ such that $\#X = k$. The *expected capacity* of a concentrator (which is a function of the offered traffic k) is the expected carried traffic when the requesting inputs are chosen at random, with all sets $X \subseteq A$ such that $\#X = k$ is equally likely.

These definitions of actual and expected capacity were given by Masson and Morris [MM], who investigated their values for "binomial" concentrators. In this paper we study their values for a new class of concentrators that we call "modular" concentrators. The asymptotic behaviour of the expected capacity for modular concentrators can be estimated quite sharply, and it appears quite attractive in view of the sparsity of these concentrators. In particular, the lost traffic is negligible when the offered traffic is less than one-half the number of inputs, and it grows quite gracefully even beyond this threshold. Scheinerman [S] has used the methods of this paper to show that even "random concentrators" have performance only slightly worse than that of modular concentrators.

In the second mode, customers arrive sequentially, and the decision as to how to serve each is made randomly, without knowledge of or dependence on future arrivals. We define this mode of operation in more detail in § 7.

* Received by the editors October 10, 1989; accepted for publication (in revised form) May 8, 1990.

† Department of Computer Science, The University of British Columbia, Vancouver, British Columbia, V6T 1W5 Canada.

2. Modular concentrators. We deal with a class of concentrators for which each input meets exactly two edges and each output meets exactly three edges. For such a concentrator there is a natural number n such that $\#A = 3n$, $\#B = 2n$, and $\#E = 6n$. These concentrators will be called $(3:2)$-*concentrators*.

We begin with the observation that the actual capacity of a $(3:2)$-concentrator is always rather small. By the *cyclomatic number* of a graph with v vertices and w edges, we mean the number $w - v + 1$. If the cyclomatic number of a graph is at most one, then it contains at most one simple cycle, and thus it has at most two independent paths between any two vertices.

THEOREM 2.1. *The actual capacity of a* $(3:2)$-*concentrator is* $O(\log n)$.

Proof. Given the $(3:2)$-concentrator $G = (A, B, E)$, construct the graph $G^* = (B, E^*)$ with vertices B corresponding to the outputs of G and edges $E^* = \{\{b, b'\} : \{a, b\}, \{a, b'\} \in E \text{ for some } a \in A\}$ corresponding to pairs of outputs that are connected to a common input. Let b be any vertex of G^*. Since each vertex meets exactly three edges in G^*, there are exactly $3 \cdot 2^{k-1}$ paths of k steps starting from b. Thus, if $k = \lceil \log_2 (2n + 1) \rceil$, there will be three distinct paths starting from b and ending at a common vertex c. The union U of these three paths has at most $3(k + 1) - 4 = 3k - 1$ vertices, since the beginning and ending vertices are common. But since there are three independent paths from b to c in U, the cyclomatic number of U must be at least two, and thus U must contain more edges than vertices. It follows that there is a set of at most $3k$ inputs in G that are connected only to a smaller number of outputs; thus the actual capacity of G is at most $3k - 1 = O(\log n)$. □

We now turn our attention to a class of $(3:2)$-concentrators for which the expected capacity is much larger than the actual capacity. The *girth* of a graph is the length of the shortest simple cycle in the graph. We construct $(3:2)$-concentrators with girth $\Omega(\log n)$. Our construction follows ideas of Margulis [M1] and Imrich [I].

Let $PSL(2, \mathbf{Z})$ denote the group of two-by-two integer matrices $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ with determinant one ($ad - bc = 1$), where two matrices are considered the same if their corresponding entries are negatives of each other. This group is generated by the matrices $S = \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$ and $R = \left(\begin{smallmatrix} 1 & 1 \\ -0 & 1 \end{smallmatrix}\right)$. We have $S^2 = R^3 = -I$, where I is the identity matrix. Furthermore, these are the only relations satisfied by S and R. Thus $PSL(2, \mathbf{Z})$ is the free product of $\mathbf{Z}/(2)$ (generated by S) and $\mathbf{Z}/(3)$ (generated by R).

Let $q \geqq 5$ be a prime and let $PSL(2, \mathbf{Z}/(q))$ be the quotient group of $PSL(2, \mathbf{Z})$ in which two matrices are considered the same if their corresponding entries differ by multiples of q. There are $(q - 1)q(q + 1)/2$ elements in $PSL(2, \mathbf{Z}/(q))$. The natural homomorphism π from $PSL(2, \mathbf{Z})$ to $PSL(2, \mathbf{Z}/(q))$ reduces entries modulo q.

A word in S and R that is reduced with respect to $S^2 = I$ and $R^3 = I$ must consist of occurrences of S alternating with occurrences of R or $R^2 = R^{-1}$. If such a word is in the kernel of π, it must have norm at least $q - 1$. (By the *norm* of a matrix $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$, we mean the maximum length of the vector $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)\left(\begin{smallmatrix} x \\ y \end{smallmatrix}\right)$, as the vector $\left(\begin{smallmatrix} x \\ y \end{smallmatrix}\right)$ varies over the circle $x^2 + y^2 = 1$. In particular, the norm of a matrix is at least the maximum of the absolute values of its entries.) It follows that a reduced word in the kernel of π must contain at least $\log_\beta (q - 1)$ occurrences of R and R^{-1}, where $\beta = (1 + \sqrt{5})/2$, since the norm of S is 1, the norms of R and R^{-1} are β, and the norm is submultiplicative.

For each prime $q \geqq 5$, let G_q denote the $(3:2)$-concentrator whose edges correspond to the elements of $PSL(2, \mathbf{Z}/(q))$, whose inputs correspond to pairs of elements that differ by a factor of S, and whose outputs correspond to triples of elements that differ by factors of $R^{\pm 1}$. Such a concentrator will be called a *modular* concentrator. Clearly, $n = (q - 1)q(q + 1)/12$. By the argument of the preceding paragraph, any simple cycle in G_q must have length at least $2 \log_\beta (q - 1) = \Omega(\log n)$. Thus we have proved the following lemma.

LEMMA 2.2. *The girth of a modular concentrator is* $\Omega(\log n)$.

The first construction of a $(3:2)$-concentrator with girth $\Omega(\log n)$ is due to Gallager ([G, Appendix C]), in the form of the parity-check matrix of a low-density parity-check code with rate $\frac{1}{3}$ over $GF(2)$. (Gallager's construction can be carried out in "polynomial time," but it is not as explicit as the one given above, which can be carried out in "logarithmic space.") We observe that there are more sophisticated constructions that give $(3:2)$-concentrators with even larger girth than G_q (roughly $(8/3)\log_2 n$ rather than $(2/3)\log_\beta n$) (see Biggs and Hoare [BH], Weiss [W], Margulis [M2], and Chiu [C]). We also observe that we do not need the full strength of Lemma 2.2. If g denotes the girth, it is sufficient that $g \to \infty$ as $n \to \infty$.

3. Hypergeometric and binomial capacities. The expected capacity has been defined *hypergeometrically*, that is, by taking all sets X of inputs with $\#X = k$ to be equally likely. We begin by showing that it is possible to deal instead with a set X of inputs that is defined *binomially*, that is, in which each input appears independently with probability $p = k/2n$.

Let $H(3n, k)$ denote the expected cardinality of a maximum matching when each set of inputs X with $\#X = k$ is equally likely. Let $J(3n, p)$ denote the expected cardinality of a maximum matching when X contains each input independently with probability p.

LEMMA 3.1. *For* $0 < p < 1$, $0 < \varepsilon < \min\{p, 1-p\}$ *and* $3np$ *an integer, we have*

$$J(3n,p-\varepsilon)-\varepsilon^{-2} \leqq H(3n,3np) \leqq J(3n,p+\varepsilon)+\varepsilon^{-2}.$$

Proof. Let X' be a set in which each input appears independently with probability $p+\varepsilon$. We have $\operatorname{Ex}(\#X') = 3n(p+\varepsilon)$ and $\operatorname{Var}(\#X') \leqq 3n$. Thus, by Chebyshev's inequality, we have $\Pr(\#X' < 3np) \leqq 1/3n\varepsilon^2$. If $\#X' \geqq 3np$, then we may delete $\#X' - 3np$ inputs from X' to obtain a set X with exactly $3np$ inputs, in such a way that every set of $3np$ inputs is equally likely. The expected cardinality of a maximum matching for X' is thus at least $H(3n, 3np)$ in this case. We thus have $J(3n, p+\varepsilon) \geqq (1 - 1/3n\varepsilon^2)H(3n, 3np)$. Since $H(3n, 3np) \leqq 3n$, we obtain the right-hand assertion of the lemma. A similar argument yields the left-hand assertion. □

In the following sections we prove the following.

THEOREM 3.2. *For* $0 < p < 1$, *we have*

$$J(3n,p) = 3nh(p) + O(n/(\log n)^{1/2}),$$

where

$$h(p) = \begin{cases} p, & \text{if } 0 < p \leqq \frac{1}{2} \\ p-(2p-1)^3/3p^3, & \text{if } \frac{1}{2} < p < 1. \end{cases}$$

Since $h(p)$ is continuous in p, we may apply Lemma 3.1 with $\varepsilon \to 0$ as $n \to \infty$ to obtain the following corollary.

COROLLARY 3.3. *For rational* $0 < p < 1$ *and n such that* $3np$ *is integral, we have*

$$H(3n,3np) = 3nh(p) + O(n/(\log n)^{1/2}).$$

4. Reduction to small components. We seek to determine the expected number of pairs in a maximum matching when each input is independently requesting with probability p. Let $F(p)$ be the subgraph of G obtained by deleting each input that is not requesting and each edge meeting such an input. Let $F^*(p)$ be the corresponding subgraph of G^*, in which each edge is retained independently with probability p.

LEMMA 4.1. *In an acyclic connected component of* $F^*(p)$, *all but exactly one of the outputs appear in a maximum matching. In a cyclic connected component of* $F^*(p)$, *all of the outputs appear in a maximum matching.*

Proof. If $F^*(p)$ contains a vertex that meets exactly one edge, we may pair the input corresponding to the edge with the output corresponding to the vertex, then find a maximum matching in the graph that remains after this edge and vertex are deleted. This transformation does not change the cyclomatic number of any component. Since an acyclic component that contains an edge must contain a vertex that meets exactly one edge, repeated application of this transformation to an acyclic component must eventually yield an isolated vertex. This proves the first assertion. Repeated application to a cyclic component must eventually yield a graph K^* in which every vertex meets at least two edges. In the corresponding bipartite graph K, every input is connected to exactly two outputs, and every output is connected to at least two inputs, so the marriage theorem ensures the existence of a matching including all of the outputs. This proves the second assertion. □

Let $Z(p)$ denote the expected number of acyclic component in $F^*(p)$. Lemma 4.1 implies that

$$J(3n,p)=2n-Z(p). \tag{4.1}$$

Let g denote the girth of G, and let $Y(p)$ denote the expected number of components of $F^*(p)$ that contain at most $g/8$ edges. A component with at most $g/8$ edges must be acyclic, so $Y(p) \leqq Z(p)$. On the other hand, there are at most $6n/(g/8) = 48n/g$ components with more than $g/8$ edges, so $Z(p) \leqq Y(p) + 48n/g$. Since $g = \Omega(\log n)$, we have

$$Z(p)=Y(p)+O(n/\log n). \tag{4.2}$$

Let $V(p)$ denote the expected number of vertices in $F^*(p)$ in components with at most $g/8$ edges, and let $W(p)$ denote the expected number of edges in such components. Since these components are all acyclic, we have

$$Y(p)=V(p)-W(p). \tag{4.3}$$

Equations (4.1), (4.2), and (4.3) together give the formula

$$J(3n,p)=2n-V(p)+W(p)+O(n/\log n) \tag{4.4}$$

for the expected capacity in terms of the expected numbers of vertices and edges in small components of $F^*(p)$. In the next section we determine the asymptotic behaviour of these expected numbers.

5. Analysis of small components. Let I be an infinite tree in which each vertex meets exactly three edges. Let $I(p)$ be a random subgraph of I in which each edge is independently retained with probability p. Let $v_k(p)$ be the probability that a vertex of I belongs to a component of $I(p)$ with at most k edges. Let $w_k(p)$ be the conditional probability that an edge e of I belongs to a component of $I(p)$ with at most k edges, given that e is retained in $I(p)$. It is clear that

$$V(p)=2nv_{g/8}(p) \quad \text{and} \quad W(p)=3npw_{g/8}(p), \tag{5.1}$$

since a neighbourhood of radius $g/8$ about any vertex or edge in G^* is isomorphic to a corresponding neighbourhood in I, and all quantities in (5.1) are defined in terms of random variables that are independent of events outside these neighbourhoods.

Let $v(p)$ denote the probability that a vertex in I belongs to a finite component of $I(p)$, and let $w(p)$ denote the conditional probability that an edge e of I belongs to a finite component of $I(p)$, given that e is retained in $I(p)$. The theory of branching processes gives the following lemma.

LEMMA 5.1. *We have*

$$v(p) = \begin{cases} 1, & \text{if } 0 < p \leqq \frac{1}{2} \\ (1-p)^3/p^3, & \text{if } \frac{1}{2} < p < 1; \end{cases}$$

and

$$w(p) = \begin{cases} 1, & \text{if } 0 < p \leqq \frac{1}{2}; \\ (1-p)^4/p^4, & \text{if } \frac{1}{2} < p < 1. \end{cases}$$

Proof. Consider a branching process in which the first generation contains a single individual, and each individual in the ith generation independently contributes to the $(i+1)$st generation a number of offspring that is binomially distributed with generating function $(1-p+px)^2$. According to Harris ([H], Chap. I, Thm. 6.1]), the probability of extinction (that is, the probability that the family generated in this way is finite) is the root $q(p)$ of equation $x = (1-p+px)^2$, given by

$$q(p) = \begin{cases} 1, & \text{if } 0 < p \leqq \frac{1}{2}; \\ (1-p)^2/p^2, & \text{if } \frac{1}{2} < p < 1. \end{cases}$$

The probability that a vertex in $I(p)$ belongs to a finite component is simply the probability of extinction when the first generation contains a number of individuals distributed with generating function $(1-p+px)^3$, the generating function for the number of edges incident with the given vertex in $I(p)$. This extinction probability is $(1-p+pq(p))^3$ (which is as given in the statement of the lemma).

Similarly, the conditional probability that an edge e in I belongs to a finite component of $I(p)$, given that e is retained in $I(p)$, is $(1-p+pq(p))^4$ (which is as given in the statement of the lemma), since $(1-p+px)^4$ is the conditional generating function for the number of edges incident with e in $I(p)$, given that e is retained in $I(p)$. □

LEMMA 5.2. *We have*

$$v_k(p) = v(p) + O(k^{-1/2})$$

and

$$w_k(p) = w(p) + O(k^{-1/2}).$$

Proof. Clearly, $v_k(p) \leqq v(p)$. Furthermore, $v(p) - v_k(p)$ is simply the probability that, in the branching process described in the proof of Lemma 5.1 (with the generating function of the initial distribution being $(1-p+px)^3$), extinction occurs after the size of the family exceeds k. According to Harris ([H, Chap. I, Thm 13.1]), the conditional probability that the size of the family is j, given that extinction occurs, is $O(j^{-3/2})$. (The decay is actually much faster than this unless $p = \frac{1}{2}$.) Thus the probability that extinction occurs after the size exceeds k is $\Sigma_{j>k} O(j^{-3/2}) = O(k^{-1/2})$. The proof for $w_k(p)$ and $w(p)$ is analogous. □

Applying Lemmas 2.2 and 5.2 to (5.1) yields

$$V(p) = 2nv(p) + O(n/(\log n)^{1/2})$$

and

$$W(p) = 3npw(p) + O(n/(\log n)^{1/2}).$$

Substitution of these formulae and Lemma 5.1 into (4.4) completes the proof of Theorem 3.2.

6. Extensions for batch arrivals. The concentrators that we have considered are one-stage networks; that is, each edge directly connects an input to an output. It is easy to see, however, that the analysis we have given has immediate application to some multistage networks.

Consider for example the "two-stage $(9:4)$-concentrators" constructed in the following way. Let $q \geqq 5$ and $q' \geqq 5$ be primes (equal or distinct), and set $n = (q-1)q(q+1)/12$ and $n' = (q'-1)q'(q'+1)/12$. Take $3n'$ disjoint copies of G_q and $2n$ disjoint copies of $G_{q'}$, and link each output of each copy of G_q to an input of a copy of $G_{q'}$, with exactly one link between each copy of G_q and each copy of $G_{q'}$. If the inputs of the resulting network are independently requesting, and if appropriate random choices of the maximum matchings in the copies of G_q are made, then the inputs of each copy of $G_{q'}$ will be independently requesting, and the analysis given above can be applied to each stage in turn. (The traffics offered to the various copies of $G_{q'}$ will be dependent, but this does not affect the expected capacity.) The expected capacity will again be piecewise rational, now with breakpoints at $p = \frac{1}{3}$ (the onset of loss in the second stage) and $p = \frac{1}{2}$. The extension to three or more stages should be clear.

It is possible to extend the analysis we have given, with hardly any changes in the arguments, to "$(a:2)$-concentrators with large girth" (for integer $a > 2$). (The construction of such concentrators can be accomplished by the methods of the papers cited in § 2.) It may also be possible to extend Theorem 3.2 (though not Theorem 2.1) to "$(a:b)$-concentrators with large girth" (for integers $a > b > 1$). There seems to be nothing as simple as Lemma 4.1 in this case, but the success of Karp and Sipser [KS] in treating the problem of maximum matchings in sparse random graphs gives hope. For $b = 2$ we prove (and for $b > 2$ it is natural to conjecture) that $v(p)$ is replaced by $q(p)^a$ and $w(p)$ is replaced by $q(p)^{(a-1)b}$, where $q(p)$ is now the appropriate root of the equation $x = (1 - p + px^{b-1})^{a-1}$.

7. Sequential arrivals with random hunting. We now turn to a second mode of operation for concentrators. Consider a concentrator $G = (A, B, E)$. Associate with each input $a \in A$ an *arrival time* τ_a, uniformly distributed in the interval $[0, 1]$, and independent of all other arrival times. The intended interpretation is that the customer corresponding to input a arrives at time τ_a.

Next associate with each input $a \in A$ a *hunting order* β_a, uniformly distributed over the total orders among the outputs connected to a, independent of the hunting orders of other inputs and independent of the arrival times of all inputs. The intended interpretation is that when the customer arrives at input a (at time τ_a), it examines the outputs connected to a in the order prescribed by β_a until it finds one that has not been engaged previously (that is, at a time less than τ_a). If it finds such an output, the output is engaged at time τ_a. If it finds no such output, no action is taken, and the customer remains unserved.

Some comments about this mode of operation are in order. First, the assumption of uniformly distributed arrival times will facilitate calculations, but other independent and identically distributed arrival times would also result in all possible orders of arrival being equally likely, and in the number of arrivals before time t being binomially distributed. (The choice of the arrival-time distribution may be regarded as a choice of the parametrisation of time. An exponential distribution, corresponding to Poisson arrivals, seems the most natural physically.) Second, results concerning the expected number of customers served for this "binomial" arrival process can easily be translated (by the argument given in § 3) into results for the "hypergeometric" arrival process, in which some number k of customers arrive at distinct inputs, with all possible sets of k inputs, as well as all possible orders of arrival, being equally likely.

8. Sequential arrivals for trees. We begin our analysis by looking at some concentrators that are trees. Let C_0 denote the concentrator with a single input that is connected to two outputs, one of which is called the *root* and the other of which is called the *leaf*. For some $k \geqq 1$, suppose that C_{k-1} has been defined. Let C_k denote the concentrator obtained by identifying the leaf of a copy of C_0 with the roots of two copies of C_{k-1} to form an internal output (neither a root nor a leaf); the root of the copy of C_0 becomes the root of C_k, and the leaves of the copies of C_{k-1} (of which there are 2^k) become the leaves of C_k.

For $k \geqq 0$ and $0 \leqq t \leqq 1$, let $Q_k(t)$ denote the probability that the root of C_k is engaged at time t.

LEMMA 8.1. *We have*

$$Q_0(t) = t/2$$

and

$$Q_k(t) = \int_0^t 1 - \frac{1}{2}(1 - Q_{k-1}(s))^2 ds. \tag{8.1}$$

Proof. For the root of C_0 to be engaged at time t, the customer must arrive by time t, which happens with probability t, and must choose the root before the leaf in the hunting order, which happens independently with probability $\frac{1}{2}$. This proves the first assertion. For the root of C_k to be engaged at time t, the customer must again arrive by time t. If the customer arrives at time s, then it will engage the root unless it chooses the leaf of C_0 before the root in the hunting order, and the leaf of C_0 is not engaged by time s. This leaf will be engaged by time s if and only if the root of one of the copies of C_{k-1} would be engaged by time s (with the same arrival times and hunting orders in the copies). These events depend on arrival times and hunting orders for disjoint sets of inputs, so they are independent. This proves the second assertion. □

We now show that the transformation $Q_{k-1} \mapsto Q_k$ has a fixed point; that is, a solution Q of the integral equation

$$Q(t) = \int_0^t 1 - \frac{1}{2}(1 - Q(s))^2 ds. \tag{8.2}$$

To do this, we differentiate (8.2) with respect to t to obtain the differential equation

$$Q'(t) = 1 - \frac{1}{2}(1 - Q(t))^2, \tag{8.3}$$

with the initial condition $Q(0) = 0$. Since (8.3) does not involve t explicitly, it can be solved by quadratures:

$$\int_0^{Q(t)} \frac{dx}{1-(1-x)^2/2} = t, \tag{8.4}$$

where the lower limit of integration has been chosen to satisfy the initial condition. The substitution $y = (1 - x)/\sqrt{2}$ reduces the integral to

$$\sqrt{2}\int_{(1-Q(t))/\sqrt{2}}^{1/\sqrt{2}} \frac{dy}{1-y^2} = \sqrt{2}\tanh^{-1}\frac{1}{\sqrt{2}} - \sqrt{2}\tanh^{-1}\frac{1-Q(t)}{\sqrt{2}}.$$

Thus

$$Q(t) = 1 - \sqrt{2}\tanh\left(\ln(1+\sqrt{2}) - \frac{t}{\sqrt{2}}\right),$$

since $\tanh^{-1} 1/\sqrt{2} = \ln(1 + \sqrt{2})$.

LEMMA 8.2. *We have $Q_k(t) \to Q(t)$ uniformly in t as $k \to \infty$.*

Proof. Set $\Delta_k(t) = Q_k(t) - Q(t)$. Since $0 \leqq Q_0(t), Q(t) \leqq 1$, we have $|\Delta_0(t)| \leqq 1$. Furthermore, (8.1) and (8.2) imply

$$|\Delta_k(t)| = \left| \int_0^t \frac{\Delta_{k-1}(s)(2 - Q_{k-1}(s) - Q(s))}{2} ds \right|.$$

Since $0 \leqq Q_{k-1}(s), Q(s) \leqq 1$, we have $|2 - Q_{k-1}(s) - Q(s)| \leqq 2$, so that

$$|\Delta_k(t)| \leqq \int_0^t |\Delta_{k-1}(s)|\, ds.$$

Thus by induction on k we obtain

$$|\Delta_k(t)| \leqq t^k/k!.$$

This completes the proof. □

For $k \geqq 0$, let D_k denote the concentrator obtained by identifying the roots of three copies of C_k to form the root of D_k; the leaves of the copies of C_k (of which there are $3 \cdot 2^k$) are the leaves of D_k. Letting $R_k(t)$ denote the probability that the root of D_k is engaged at time t, we clearly have $R_k(t) = 1 - (1 - Q_k(t))^3$. Finally, putting $R(t) = 1 - (1 - Q(t))^3$, we see that $R_k(t) \to R(t)$ uniformly in t as $k \to \infty$. Thus we have proved the following proposition.

PROPOSITION 8.3. *As $k \to \infty$, the probability $R_k(t)$ that the root of D_k is engaged at time t tends to*

$$R(t) = 1 - \left(\sqrt{2} \tanh \left(\ln(1 + \sqrt{2}) - \frac{t}{\sqrt{2}} \right) \right)^3,$$

uniformly in t.

9. Sequential arrivals for modular concentrators. Now consider the concentrator G_q and arbitrarily designate one output of this concentrator as the "root." Let N_k denote the subgraph of G_q induced by the inputs of G_q at distance at most $2k + 1$ from the root and the outputs of G_q at distance at most $2k + 2$ from the root. Call the outputs at distance $2k + 2$ from the root the "leaves" of N_k. Set $k = \lfloor (g - 6)/4 \rfloor$, where g is the girth of G_q. Since $g = \Omega(\log q)$ (by Part I, Lemma 2.2), we have $k \to \infty$ as $q \to \infty$. Furthermore, since $4k + 4$ is less than the girth of G_q, N_k is a tree isomorphic to D_k, with root corresponding to root, and leaves corresponding to leaves. Let $S_q(t)$ denote the probability that the root of G_q is engaged at time t.

LEMMA 9.1. *We have $S_q(t) \sim R_k(t)$ uniformly in t as $q \to \infty$ and hence $k \to \infty$.*

Proof. Suppose we wish to determine whether the root of G_q is engaged at some time t. This is determined by the arrival times and hunting orders of the inputs in N_1, unless some input at distance three from the root has an earlier arrival time than the intermediate vertex at distance one; that is, unless there is a path of decreasing arrival times from the root to some leaf of N_2. Even if there is such a path, the engagement of the root is determined by the arrival times and hunting orders of the inputs in N_2, unless there is a path of decreasing arrival times from the root to a leaf in N_3. In general, the engagement of the root is determined by the arrival times and hunting orders of the inputs in N_k, unless there is a path of decreasing arrival times from the root to a leaf in N_k.

Let X_k denote the event "there is a path of decreasing arrival times from the root to a leaf in N_k." We have $\Pr(X_k) \leqq 3 \cdot 2^k/k!$, since there are $3 \cdot 2^k$ paths from the root to a leaf in N_k, and the probability that the arrival times along some such path are decreasing is $1/k!$ (since all $k!$ orders of arrival are equally likely). Furthermore, unless

X_k occurs, the root is engaged in G_q when and only when it is engaged in N_k. Thus we have $|S_q(t) - R_k(t)| \leqq 3 \cdot 2^k / k!$. □

Combining this with Proposition 8.3, we have proved the following theorem.

THEOREM 9.2. *As $q \to \infty$, the probability $S_q(t)$ that an output in G_q is engaged at time t tends to*

$$R(t) = 1 - \left(\sqrt{2} \tanh \left(\ln(1+\sqrt{2}) - \frac{t}{\sqrt{2}} \right) \right)^3,$$

uniformly in t. In particular, the probability that an output is never engaged tends to

$$\left(\sqrt{2} \tanh \left(\ln(1+\sqrt{2}) - \frac{1}{\sqrt{2}} \right) \right)^3 = 0.0145 \cdots.$$

10. Extensions for sequential arrivals. The extensions we have described for batch arrivals all apply to sequential arrivals as well. In particular, for "$(a : b)$-concentrators with large girth," we obtain integral equations that can still be solved by quadratures, though not in general in terms of elementary functions. It is easy, however, to carry out the quadratures numerically and to obtain the fraction of unused servers as a function of time.

When the concentration ratio a/b is an integer, a new possibility arises that does not occur for $(3 : 2)$-concentrators. In this case, it is possible to assign fixed hunting orders to the inputs in such a way that each output is the first choice for a/b inputs, the second choice for another a/b, and so forth. For such an assignment, there can be no unused servers after all customers have arrived. The analysis of this mode of operation leads to differential equations (or systems of differential equations) that cannot be solved by quadratures. It is easy, however, to integrate them numerically, and to obtain the fractions of requests that are served by their first choice, their second choice, and so forth.

REFERENCES

[BH] N. L. BIGGS AND M. J. HOARE, *The sextet construction for cubic graphs*, Combinatorica, 3 (1983), pp. 153–165.

[C] P. CHIU, *Cubic Ramanujan graphs*, Combinatorica, to appear.

[G] R. G. GALLAGER, *Low-Density Parity-Check Codes*, MIT Press, Cambridge, MA, 1963.

[H] T. E. HARRIS, *The Theory of Branching Processes*, Springer-Verlag, New York, 1963.

[I] W. IMRICH, *Explicit constructions of regular graphs without small cycles*, Combinatorica, 4 (1984), pp. 53–59.

[KS] R. M. KARP AND M. SIPSER, *Maximum matchings in sparse random graphs*, IEEE Symp. on Foundations of Computer Science, 22 (1981), pp. 364–375.

[M1] G. A. MARGULIS, *Explicit constructions of graphs without short cycles and low density codes*, Combinatorica, 2 (1982), pp. 71–78.

[M2] ———, *Explicit group-theoretical constructions of combinatorial schemes and their application to the design of expanders and concentrators*, Problems Inform. Transmission, 24 (1988), pp. 39–46.

[MM] G. M. MASSON AND S. B. MORRIS, *Expected capacity of* $\binom{m}{2}$*-networks*, IEEE Trans. Comput., 32 (1983), pp. 649–656.

[P] N. PIPPENGER, *Random sequential adsorption on graphs*, SIAM J. Discrete Math., 2 (1989), pp. 393–401.

[S] E. R. SCHEINERMAN, *On the expected capacity of binomial and random concentrators*, SIAM J. Comput., 19 (1990), pp. 156–163.

[W] A. WEISS, *Girths of bipartite sextet graphs*, Combinatorica, 4 (1984), pp. 241–245.

SIAM J. DISC. MATH.
Vol. 4, No. 1, pp. 130–138, February 1991

013

EDGE-DISJOINT HOMOTOPIC PATHS IN STRAIGHT-LINE PLANAR GRAPHS*

A. SCHRIJVER†

Abstract. Let G be a planar graph, embedded without crossings in the euclidean plane $\mathbb{R}^2$, and let $I_1, \cdots, I_p$ be some of its faces (including the unbounded face), considered as open sets. Suppose there exist (straight) line segments $L_1, \cdots, L_t$ in $\mathbb{R}^2$ so that $G \cup I_1 \cup \cdots \cup I_p = L_1 \cup \cdots \cup L_t \cup I_1 \cup \cdots \cup I_p$ and so that each L_i has its end points in $I_1 \cup \cdots \cup I_p$. Let $C_1, \cdots, C_k$ be curves in $\mathbb{R}^2\backslash(I_1 \cup \cdots \cup I_p)$ with end points in vertices of G. Conditions are described under which there exist pairwise edge-disjoint paths $P_1, \cdots, P_k$ in G so that P_i is homotopic to C_i in $\mathbb{R}^2\backslash(I_1 \cup \cdots \cup I_p)$, for $i = 1, \cdots, k$. This extends results of Kaufmann and Mehlhorn for graphs derived from the rectangular grid.

Key words. edge-disjoint, paths, homotopic, packing, planar

AMS(MOS) subject classifications. 05C10, 05C38

1. Introduction and statement of the theorem. Let $G = (V, E)$ be a planar graph, embedded without crossing edges in the euclidean plane $\mathbb{R}^2$. We identify G with its image in $\mathbb{R}^2$. Let $I_1, \cdots, I_p$ be some of its faces, including the unbounded face, called the *black holes*. (We consider faces as *open* sets.) Moreover, let paths $C_1, \cdots, C_k$ be given with end points in V, not intersecting any black hole. (That is, for each i, C_i is a continuous function $[0, 1] \rightarrow \mathbb{R}^2\backslash(I_1 \cup \cdots \cup I_p)$ with $C(0), C(1) \in V$.)

Motivated by the automatic design of integrated circuits, Mehlhorn posed the following question:

(1) Under which conditions do there exist pairwise edge-disjoint paths $P_1, \cdots, P_k$ in G so that P_i is homotopic to C_i in the space $\mathbb{R}^2\backslash(I_1 \cup \cdots \cup I_p)$ (for $i = 1, \cdots, k$)?

Here a *path* in G is a continuous function P: $[0, 1] \rightarrow G$ with $P(0), P(1) \in V$. Paths $P_1, \cdots, P_k$ are *pairwise edge-disjoint* if the following holds: if $P_i(x) = P_j(y) \notin V$ then $x = y$ and $i = j$. (In particular, if $P_1, \cdots, P_k$ are pairwise edge-disjoint, then each P_i does not pass the same edge more than once.) Two paths P, C: $[0, 1] \rightarrow \mathbb{R}^2\backslash(I_1 \cup \cdots \cup I_p)$ are *homotopic* (*in* $\mathbb{R}^2\backslash(I_1 \cup \cdots \cup I_p)$), denoted by $P \sim C$, if there exists a continuous function Φ: $[0, 1] \times [0, 1] \rightarrow \mathbb{R}^2\backslash(I_1 \cup \cdots \cup I_p)$ so that for all $x \in [0, 1]$: $\Phi(x, 0) = P(x)$, $\Phi(x, 1) = C(x)$, $\Phi(0, x) = P(0)$, $\Phi(1, x) = P(1)$. (In particular, $P(0) = C(0)$ and $P(1) = C(1)$.)

Mehlhorn proposed to study question (1) with the help of the following "cuts." A (*homotopic*) *cut* is a continuous function D: $[0, 1] \rightarrow \mathbb{R}^2\backslash(V \cup I_1 \cup \cdots \cup I_p)$ so that $D(0)$ and $D(1)$ belong to the boundary of $I_1 \cup \cdots \cup I_p$ and so that $|D^{-1}(G)|$ is finite. The *cut condition* (*for* G; $I_1, \cdots, I_p$; $C_1, \cdots, C_k$) is:

$$(2) \qquad (\textit{cut condition}) \text{ for each cut } D\text{: } \operatorname{cr}(G,D) \geqq \sum_{i=1}^{k} \operatorname{mincr}(C_i, D).$$

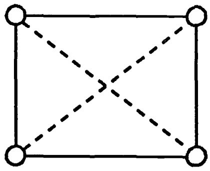
* Received by the editors August 10, 1987; accepted for publication (in revised form) December 6, 1988.

† Mathematical Centre, Kruislaan 413, 1098 SJ Amsterdam, the Netherlands.

Here we use the following notation for curves C, D: $[0, 1] \rightarrow \mathbb{R}^2\backslash(I_1 \cup \cdots \cup I_p)$:

$$\begin{aligned}
&\operatorname{cr}(G,D) := |\{y\in[0,1] \mid D(y)\in G\}|,\\
(3)\qquad &\operatorname{cr}(C,D) := |\{(x,y)\in[0,1]\times[0,1] \mid C(x)=D(y)\}|,\\
&\operatorname{mincr}(C,D) := \min\{\operatorname{cr}(\tilde{C},\tilde{D}) \mid \tilde{C}\sim C, \tilde{D}\sim D \text{ in } \mathbb{R}^2\backslash(I_1\cup\cdots\cup I_p)\}.
\end{aligned}$$

Clearly, the cut condition is a necessary condition for a positive answer to question (1). It is generally not sufficient, not even for quite simple situations. For example, take $k = 2$, $p = 1$, and consider



I_1,

where the straight lines stand for edges of G and where the interrupted lines stand for curves C_1 and C_2.

It turned out that one additional condition, the so-called *parity condition*, can be helpful (cf. § 2 below):

(4) (*parity condition*) for each cut D: $\operatorname{cr}(G, D) \equiv \sum_{i=1}^{k} \operatorname{mincr}(C_i, D) \pmod 2$.

Let us now state our theorem. We say that G; $I_1, \cdots, I_p$; $C_1, \cdots, C_k$ is in the *straight-line case* if

(5) there are line segments $L_1, \cdots, L_t$ in $\mathbb{R}^2$ so that $G \cup I_1 \cup \cdots \cup I_p = L_1 \cup \cdots \cup L_t \cup I_1 \cup \cdots \cup I_p$ and so that each L_j has its end points in $I_1 \cup \cdots \cup I_{p'}$

and

(6) if the aperture at vertex v of G is larger than 180°, then the number of times v occurs as end point of the curves C_i is not larger than the number of edges terminating at v.

Here the *aperture* at vertex v of G is the largest angle that can be made at v so that none of the black holes adjacent to v intersect the interior of the angle. (More formally, let $\rho > 0$ be so that the circle K of radius ρ and centre v does not contain any other vertex of G in its interior and does not intersect any edge except for those adjacent to v. Let $K\backslash(I_1 \cup \cdots \cup I_p)$ have components $K_1, \cdots, K_h$, making angles $\varphi_1, \cdots, \varphi_h$. Then the aperture at v is equal to $\max\{\varphi_1, \cdots, \varphi_h\}$.) Edge $e = \{(1-\lambda)u + \lambda v \mid 0 < \lambda < 1\}$ of G is said to *terminate* at v if for some $\mu > 1$ the set $\{(1-\lambda)u + \lambda v \mid 1 < \lambda < \mu\}$ is contained in $I_1 \cup \cdots \cup I_p$.

THEOREM. *If we are in the straight-line case and the parity condition holds, then there exist pairwise edge-disjoint paths as in* (1) *if and only if the cut condition holds.*

As an illustration, Fig. 1 gives an example of the straight-line case (where the shaded faces, together with the unbounded face, are the black holes, and where the interrupted curves stand for the paths C_i).

The theorem generalizes a result of Kaufmann and Mehlhorn [2] for graphs derived from the rectangular grid in the following way. G is a finite subgraph of the rectangular grid. (That is, V is a finite subset of $\mathbb{Z}^2$ and each edge is a line segment of length 1.) $I_1, \cdots, I_p$ are exactly those faces of G that are not bounded by exactly four edges of G.

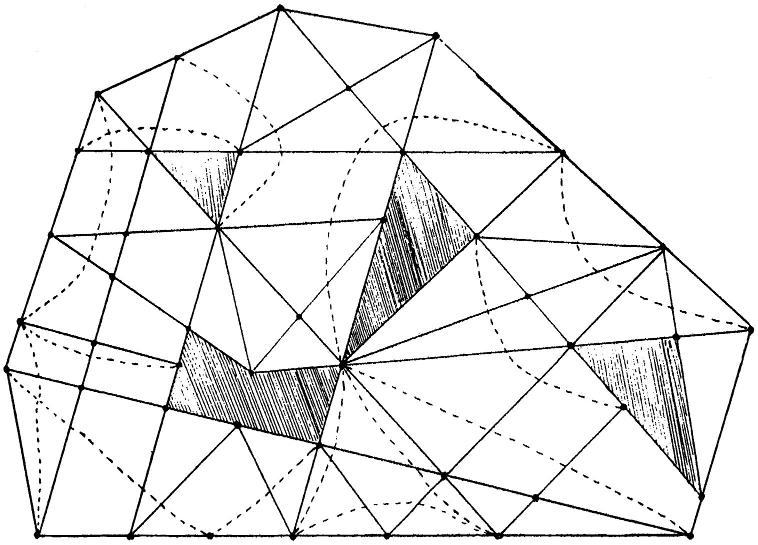


FIG. 1

Moreover, for each vertex v it is required that $\deg(v) + r(v) \leqq 4$, where $\deg(v)$ denotes the degree of v in G, and

$$r(v) := |\{i=1, \cdots, k \mid C_i(0)=v\}| + |\{i=1, \cdots, k \mid C_i(1)=v\}|.$$

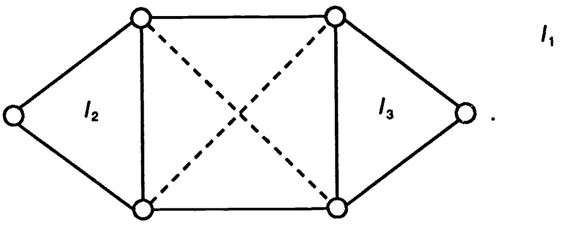
COROLLARY (Kaufmann and Mehlhorn). *If the conditions given in the previous paragraph are satisfied and the parity condition holds, then there exist pairwise edge-disjoint paths as in* (1) *if and only if the cut condition holds.*

In fact, Kaufmann and Mehlhorn found a linear-time algorithm to find these paths, if they exist.

In § 4 we give a proof of our theorem. We make use of a lemma to be proved in § 3 (showing that in the straight-line case we may restrict the cut condition to (almost) straight cuts (analogous to the idea of "1-bend cuts" in [2])), and of results of [4] to be reviewed in § 2.

2. Review of preliminary results. In this section we return to the general case of a planar graph $G = (V, E)$ embedded without crossing edges in the Euclidean plane $\mathbb{R}^2$, with black holes $I_1, \cdots, I_p$ (including the unbounded face) and curves $C_1, \cdots, C_k$. Let each C_i have its end points in vertices on the boundary of $I_1 \cup \cdots \cup I_p$.

It was shown by Okamura and Seymour [3] that if $p = 1$ the cut condition together with the parity condition imply the existence of paths as in (1). (Note that for $p = 1$ two paths P, P' are homotopic if and only if $P(0) = P'(0)$ and $P(1) = P'(1)$.) This was extended by van Hoesel and Schrijver [1] to $p = 2$. It cannot be extended to higher p, as is shown for $p = 3$ by:



However, it was shown in [4] that, for arbitrary p, the cut condition is equivalent to the existence of a "fractional" packing of paths as required, i.e., to the existence of paths $P_1^1, \cdots, P_1^{t_1}, P_2^1, \cdots, P_k^1, \cdots, P_k^{t_k}$ and rationals $\lambda_1^1, \cdots, \lambda_1^{t_1}, \lambda_2^1, \cdots, \lambda_k^1, \cdots, \lambda_k^{t_k} > 0$ such that:

$$(7)\qquad \begin{aligned} &\text{(i) } P_i^j \sim C_i && (i=1,\cdots,k; j=1,\cdots,t_i),\\ &\text{(ii) } \sum_{j=1}^{t_i} \lambda_i^j = 1 && (i=1,\cdots,k),\\ &\text{(iii) } \sum_{i=1}^{k}\sum_{j=1}^{t_i} \lambda_i^j \chi^{P_i^j}(e) \leqq 1 && (e\in E).\end{aligned}$$

Here $\chi^P(e)$ denotes the number of times path P passes edge e.

Another result from [4] to be used below was derived with the theory of simplicial approximations. Let C, D: $[0, 1] \to \mathbb{R}^2\backslash(I_1 \cup \cdots \cup I_p)$ be continuous. Let $C(0)$, $C(1)$, $D(0)$, and $D(1)$ be on the boundary of $I_1 \cup \cdots \cup I_p$, with $\{C(0), C(1)\} \cap \{D(0), D(1)\} = \varnothing$. Let

$$(8)\qquad X := \{(y,z)\in[0,1]\times[0,1] \mid C(y) = D(z)\}$$

be finite, where each (y, z) in X gives a crossing of C and D. For $y, y' \in [0, 1]$ let $C|_y^{y'}$ denote the path from $C(y)$ to $C(y')$ given by:

$$(9)\qquad (C|_y^{y'})(\lambda) := C((1-\lambda)y + \lambda y') \quad \text{for } \lambda\in[0,1];$$

similarly for D. Define for $(y, z), (y', z') \in X$:

$$(10)\qquad (y,z) \approx (y',z') \Leftrightarrow (C|_y^{y'}) \approx (D|_z^{z'}) \quad \text{in } \mathbb{R}^2\backslash(I_1\cup\cdots\cup I_p).$$

We call the classes of the equivalence relation $\approx$ the *classes of intersections* of C and D. Such a class is called *odd* if it contains an odd number of elements. Let odd (C, D) denote the number of odd classes of X. Then

$$(11)\qquad \operatorname{mincr}(C,D) = \operatorname{odd}(C,D).$$

3. A lemma on straight cuts. We call a cut D: $[0, 1] \to \mathbb{R}^2\backslash(V \cup I_1 \cup \cdots \cup I_p)$ a *straight cut* if

(12)
either (i) D is linear,
or (ii) the line segment connecting $D(0)$ and $D(1)$ is contained in G, the functions $D|[0, \frac{1}{2}]$ and $D|[\frac{1}{2}, 1]$ are linear, there is no vertex of G contained in the interior of the triangle $D(0)D(\frac{1}{2})D(1)$, and no edge is intersected more than once by D.

In (ii) we might think of D as being very close to the line segment connecting $D(0)$ and $D(1)$. So a straight cut is determined by its end points, in case (12) (ii) up to "slight" homotopic shifts, which, however, do not change the number of intersections with G.

LEMMA. *In the straight-line case, the cut condition holds if and only if* $\operatorname{cr}(G, D) \geqq \sum_{i=1}^k \operatorname{mincr}(C_i, D)$ *for each straight cut* D.

Proof. Necessity being trivial, we show sufficiency. Let the cut inequality be satisfied by each straight cut. Suppose there exists a cut D: $[0, 1] \to \mathbb{R}^2\backslash(V \cup I_1 \cup \cdots \cup I_p)$ so that

$$(13)\qquad \operatorname{cr}(G,D) < \sum_{i=1}^{k} \operatorname{mincr}(C_i,D).$$

We choose D satisfying (13) so that $t := \mathrm{cr}(G, D)$ is as small as possible. The idea of the proof is to straighten out D as much as possible.

First observe that we may assume that if $D(1)$ is not on the line through the edge containing $D(0)$, then the line segment $\overline{D(0)D(1)}$ does not intersect V (this can be achieved by slightly shifting $D(0)$ along the edge containing $D(0)$). Moreover, we may assume that there exists an $\varepsilon > 0$ so that

(14)
(i) $D|[0, \varepsilon]$ is linear;
(ii) for all $\delta \in (0, \varepsilon]$: $D(\delta)$ does not belong to any line through any pair of vertices of G nor to any line through a pair of points consisting of a vertex of G and an intersection of D and G.

Let $\lambda_1, \cdots, \lambda_t$ be so that $0 = \lambda_1 < \lambda_2 < \cdots < \lambda_{t-1} < \lambda_t = 1$, with $D(\lambda_i) \in G$ for all i. Define

$$(15)\qquad \begin{aligned} p_1 &:= D(\varepsilon), \\ p_i &:= D(\lambda_i) \quad \text{for } i = 2, \cdots, t. \end{aligned}$$

Finally, we may assume that $D|[\varepsilon, \lambda_2]$ and $D|[\lambda_{i-1}, \lambda_i]$ are linear functions ($i = 3, \cdots, t$) (since in the straight-line case each face not in $\{I_1, \cdots, I_p\}$ is convex).

Let $h(D)$ be the smallest index h with $2 \leqq h \leqq t - 1$ so that the angle between $\overline{p_{h-1}p_h}$ and $\overline{p_hp_{h+1}}$ is not 180°. If no such h exists, let $h(D) := t$. We may assume that we have chosen D so that (fixing $t = \mathrm{cr}(G, D)$) $h(D)$ is as large as possible. Let $h := h(D)$.

First consider the case $h < t$. Choose the largest $\lambda \in [0, 1]$ so that the triangle with vertices p_1, p_h, and $p_h + \lambda(p_{h+1} - p_h)$ does not intersect $I_1 \cup \cdots \cup I_p$. Let $p'_h := p_h + \lambda(p_{h+1} - p_h)$. Let D' be the piecewise linear function obtained from D by replacing parts $\overline{p_1p_h}$ and $\overline{p_hp'_h}$ of D by $\overline{p_1p'_h}$.

If $\lambda = 1$, then $p'_h = p_{h+1}$, and hence by (14)(ii) $\overline{p_1p'_h}$ does not intersect any vertex of G. So D' is a cut, with $\mathrm{cr}(G, D') = \mathrm{cr}(G, D)$ (by the conditions (5) and (6) for the straight-line case) and $D' \sim D$. As $h(D') > h(D)$ this contradicts the fact that we have chosen D so that $h(D)$ is as large as possible.

If $\lambda < 1$, then $\overline{p_1p'_h}$ intersects a vertex v of G, on the boundary of $I_1 \cup \cdots \cup I_p$. This vertex is unique by (14)(ii) and has aperture larger than 180°. Consider a circle K with center v, not containing any other vertex of G, and not intersecting any edge of G except for those adjacent to v. Let $K \backslash (I_1 \cup \cdots \cup I_p)$ have components $K_1, \cdots, K_h$. So each K_i is a cut. We may assume that K_1 intersects D' twice. So K_1 is a circular arc of angle larger than 180°. Use the notation A, B, C, E, F for the parts of D' and K_1 as indicated in Fig. 2. Let H denote the part of D from p'_h to p_t. As we have chosen D so that (13) is satisfied with $\mathrm{cr}(G, D)$ as small as possible, we have

$$(16)\qquad \begin{aligned} \mathrm{cr}(G,D) &= \mathrm{cr}(G,EBFH) = \mathrm{cr}(G,EA) + \mathrm{cr}(G,CFH) + \sum_{j=2}^{h} \mathrm{cr}(G,K_j) \\ &\quad + (\text{number of edges terminating at } v) \\ &\geqq \sum_{i=1}^{k} \mathrm{mincr}(C_i,EA) + \sum_{i=1}^{k} \mathrm{mincr}(C_i,CFH) + \sum_{j=2}^{h}\sum_{i=1}^{k} \mathrm{mincr}(C_i,K_j) \\ &\quad + \sum_{i=1}^{k} (\text{number of times } v \text{ is end point of } C_i) \geqq \sum_{i=1}^{k} \mathrm{mincr}(C_i,D) \end{aligned}$$

(using (6)). This contradicts (13).

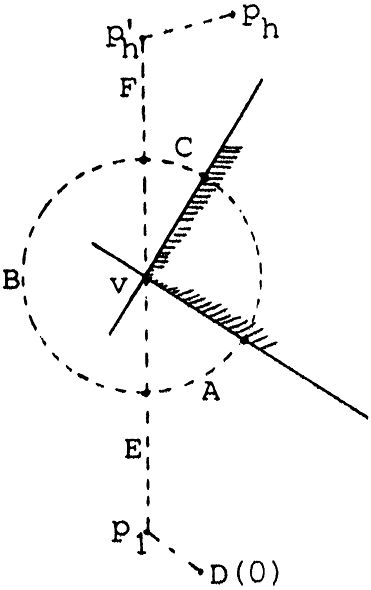


FIG. 2

As $h < t$ leads to a contradiction, we know $h = t$. If the line segment $\overline{D(0)D(1)}$ is not contained in G, then by our assumption this line segment forms a straight cut D', with $\operatorname{cr}(G, D') = \operatorname{cr}(G, D)$ and $D' \sim D$, whence

$$(17) \qquad \operatorname{cr}(G,D) = \operatorname{cr}(G,D') \geqq \sum_{i=1}^{k} \operatorname{mincr}(C_i,D') = \sum_{i=1}^{k} \operatorname{mincr}(C_i,D),$$

contradicting (13). If $\overline{D(0)D(1)}$ is contained in G, then D itself forms a straight cut, contradicting (13). □

4. Proof of the theorem. We now prove our theorem.

THEOREM. *If we are in the straight-line case and the parity condition holds, then there exist pairwise edge-disjoint paths as in* (1) *if and only if the cut condition holds.*

Proof. The proof is by induction on the number of faces not in $\{I_1, \cdots, I_p\}$. If each face belongs to $\{I_1, \cdots, I_p\}$, then the theorem is trivially true. So assume that not all faces belong to $\{I_1, \cdots, I_p\}$.

I. We first consider those situations where the following holds:

(18) G has an edge e_0, connecting vertices u and w, both of degree 2, so that e_0 separates a face in $\{I_1, \cdots, I_p\}$ from a face not in $\{I_1, \cdots, I_p\}$ and so that one of the curves C_i connects u and w following e_0.

Without loss of generality, e_0 separates face I_1 from face $F \notin \{I_1, \cdots, I_p\}$, and C_1 connects u and w following e_0. Moreover, we may assume that none of $C_2, \cdots, C_k$ passes e_0 (we can make detours along the other edges of F). By the parity condition, there exist h, j so that C_h has an end point in u and C_j has an end point in w (possibly $h = j$).

Now let $I_{p+1} := F$. Clearly, $G; I_p, \cdots, I_p, I_{p+1}; C_1, \cdots, C_k$ is again in the straight-line case, in which the parity condition holds. We show

(19) the cut condition holds for $G; I_1, \cdots, I_{p+1}; C_1, \cdots, C_k$.

As the number of faces not in $\{I_1, \cdots, I_{p+1}\}$ is one less than in the original situation, (19) implies by induction that there exist pairwise edge-disjoint paths $P_1 \sim C_1, \cdots, P_k \sim C_k$ in $\mathbb{R}^2\backslash(I_1 \cup \cdots \cup I_{p+1})$. This implies $P_1 \sim C_1, \cdots, P_k \sim C_k$ in $\mathbb{R}^2\backslash(I_1 \cup \cdots \cup I_p)$ as required.

We prove (19). We will refer to $G; I_1, \cdots, I_{p+1}; C_1, \cdots, C_k$ as the *new structure*, and to $G; I_1, \cdots, I_p; C_1, \cdots, C_k$ as the *original structure*. For the new structure we use the notation mincr$'$ instead of mincr.

To show (19) by the lemma, it suffices to prove the cut inequality for straight cuts only. Let D be a straight cut in the new structure. If $D(0)$ and $D(1)$ belong to the boundary of $I_1 \cup \cdots \cup I_p$, then D is also a cut in the original structure, and the cut inequality follows (as $\text{mincr}'(C_i, D) = \text{mincr}(C_i, D)$ for each i). If both $D(0)$ and $D(1)$ belong to the boundary of $I_{p+1} = F$, then $\text{mincr}'(C_i, D) = 0$ for each i (as F is convex), and the cut inequality follows. So we may assume that $D(0)$ belongs to the boundary of $I_1 \cup \cdots \cup I_p$ and $D(1)$ belongs to the boundary of F. We can extend D in $\bar{F}$ to a cut D' ending on e_0. Then D' is a cut in the original structure. Thus we have

$$(20) \qquad \text{cr}(G,D) = \text{cr}(G,D') - 1 \geqq \sum_{i=1}^{k} \text{mincr}(C_i,D') - 1 = \sum_{i=1}^{k} \text{mincr}'(C_i,D),$$

thus showing the cut inequality for D. This proves (19).

II. Now we consider the general case (i.e., we do not assume (18)). As not all faces belong to $\{I_1, \cdots, I_p\}$, there exists an edge, say e_0, separating a face I_h $(1 \leqq h \leqq p)$ from a face F not in $\{I_1, \cdots, I_p\}$. We may assume $h = 1$. Without loss of generality, no path C_i intersects e_0 or F (we can make detours along the boundary of F). Extend G to a graph G' by adding two new vertices, say u and w, on e_0. Let e_0' be the edge connecting u and w. Let C_{k+1} and C_{k+2} be two curves, each connecting u and w via e_0'. We consider two cases.

Case 1. The cut condition holds for $G'; I_1, \cdots, I_p; C_1, \cdots, C_k, C_{k+1}, C_{k+2}$. Now we can apply part I of this proof above, and paths $P_1, \cdots, P_k, P_{k+1}, P_{k+2}$ as required exist.

Case 2. The cut condition does not hold for $G'; I_1, \cdots, I_p; C_1, \cdots, C_k, C_{k+1}, C_{k+2}$. Since also in this new situation we are in the straight-line case, by the lemma there exists a straight cut D so that

$$(21) \qquad \text{cr}(G',D) < \sum_{i=1}^{k+2} \text{mincr}(C_i,D).$$

Since $\text{mincr}(C_{k+1}, D) = \text{mincr}(C_{k+2}, D) \leqq 1$ and since the parity condition holds for $G; I_1, \cdots, I_p; C_1, \cdots, C_k$ we know

$$(22) \qquad \text{cr}(G,D) = \sum_{i=1}^{k} \text{mincr}(C_i,D),$$

and $\text{mincr}(C_{k+1}, D) = \text{mincr}(C_{k+2}) = 1$. Hence D has one of its end points on e_0'.

As the cut condition holds for $G; I_1, \cdots, I_p; C_1, \cdots, C_k$, there exists a "fractional" packing of paths $P_1^1, \cdots, P_1^{t_1}, \cdots, P_k^1, \cdots, P_k^{t_k}$, with coefficients $\lambda_1^1, \cdots, \lambda_1^{t_1}, \cdots, \lambda_k^1, \cdots, \lambda_k^{t_k} > 0$, satisfying (7). By (22), at least one of the P_i^j, say P_1^1, passes edge e_0. So $P_1^1 = R_1 e_0' R_2$ for certain paths R_1 and R_2.

We now show the following claim.

CLAIM. *For each straight cut D' (for G') we have*

$$(23) \quad \text{mincr}(R_1,D') + \text{mincr}(C_{k+1},D') + \text{mincr}(R_2,D') \leqq \text{mincr}(C_1,D') + 2.$$

Proof of the claim. Since

$$(24)\qquad \operatorname{cr}(G,D)=\sum_{i=1}^{k}\operatorname{mincr}(C_i,D)\leqq\sum_{i=1}^{k}\sum_{j=1}^{t_i}\lambda_i^j\cdot\operatorname{cr}(P_i^j,D)\leqq\operatorname{cr}(G,D),$$

and since $\lambda_1^1>0$, we know that $\operatorname{cr}(P_1^1, D)=\operatorname{mincr}(C_1, D)$.

Without loss of generality, $(P_1^1|_0^{1/4})$ coincides with path R_1, $(P_1^1|_{1/4}^{3/4})$ with C_{k+1}, and $(P_1^1|_{3/4}^{1})$ with R_2. Moreover, we may assume that $P_1^1(1/2)=D(0)$.

Let D' be any straight cut. To show (23) we may assume that D and D' intersect each other at most once, and that if D' intersects e_0', then D and D' do not intersect.

Let

$$(25)\qquad X:=\{(x,y)\in[0,1]\times[0,1]\,|\,P_1^1(x)=D'(y)\}.$$

Let $\approx$ be as in (10). So $\operatorname{mincr}(C_1, D')$ is equal to the number of odd classes of $\approx$. We show

(26) if $(x, y), (x', y'), (x'', y''), (x''', y''')\in X$ so that $(x, y)\approx(x', y')$, $(x'', y'')\approx(x''', y''')$, $x, x''\in(0, \frac{1}{2})$ and $x', x'''\in(\frac{1}{2}, 1)$, then D and D' intersect and $(x, y)\approx(x'', y'')$.

Indeed, as $(x, y)\approx(x', y')$, we know $(P_1^1|_x^{x'})\sim(D'|_y^{y'})$. So $(P_1^1|_x^{x'})(D'|_{y'}^{y})$ forms a homotopically trivial cycle K. Since $(P_1^1|_x^{x'})$ passes $D(0)$, D splits K into two homotopically trivial cycles. That is, there is a $\lambda\in(0, 1]$ so that

(27) either (i) $\exists z\in[x, x']$: $(P_1^1|_z^{1/2})(D|_0^{\lambda})$ is a homotopically trivial cycle,
or (ii) $\exists z\in(y, y')$: $(P_1^1|_x^{1/2})(D|_{1/2}^{\lambda})(D'|_z^{y})$ is a homotopically trivial cycle.

Since $\operatorname{cr}(P_1^1, D)=\operatorname{mincr}(P_1^1, D)$, (27) (i) does not occur. So (27) (ii) applies. Hence

$$(28)\qquad (P_1^1|_x^{1/2})\sim(D'|_y^{z})(D|_\lambda^{1/2}).$$

In particular, D and D' intersect, with $D(\lambda)=D'(z)$. We similarly derive from the fact that $(x'', y'')\approx(x''', y''')$ that

$$(29)\qquad (P_1^1|_{x''}^{1/2})\sim(D'|_{y''}^{z})(D|_\lambda^{1/2}).$$

Therefore,

$$(30)\qquad (P_1^1|_x^{x''})\sim(P_1^1|_x^{1/2})(P_1^1|_{1/2}^{x''})\sim(D'|_y^{z})(D|_\lambda^{1/2})(D|_{1/2}^{\lambda})(D'|_z^{y''})\sim(D'|_y^{y''}).$$

So $(x, y)\approx(x'', y'')$. This shows (26).

Now $\operatorname{cr}(C_{k+1}, D')\leqq 1$. If $\operatorname{cr}(C_{k+1}, D')=0$, then the above implies

$$(31)\qquad \operatorname{odd}(P_1^1,D')\geqq(\operatorname{odd}(R_1,D')-1)+(\operatorname{odd}(R_2,D')-1),$$

since by (26) all but at most one class of intersections of R_1 and D' is also a class of intersections of P_1^1 and D'. Similarly for R_2. Equation (31) implies (23).

If $\operatorname{cr}(C_{k+1}, D')=1$, then D and D' do not intersect, by assumption. Hence, by (26), no class of intersections of P_1^1 and D' contains both (x, y) and (x', y') with $x\in(0, \frac{1}{2})$ and $x'\in(\frac{1}{2}, 1)$. Since $\operatorname{cr}(C_{k+1}, D')=1$, there is only one element (x, y) in X with $x\in(\frac{1}{4}, \frac{3}{4})$. Except for the class of intersections of P_1^1 and D' containing this element, all other classes also form a class of intersections of R_1 and D' or of R_2 and D'. Hence

$$(32)\qquad \operatorname{odd}(P_1^1,D')\geqq\operatorname{odd}(R_1,D')+\operatorname{odd}(R_2,D')-1,$$

and (23) follows. □

We next show

(33) the cut condition holds for $G'; I_1, \cdots, I_p; R_1, R_2, C_2, \cdots, C_k, C_{k+1}$.

Suppose not. Since we are again in the straight-line case, by the lemma there exists a straight cut D' so that

$$\text{(34)} \quad \operatorname{mincr}(R_1,D')+\operatorname{mincr}(R_2,D')+\sum_{i=2}^{k+1}\operatorname{mincr}(C_i,D')\geqq \operatorname{cr}(G,D')+2,$$

using the fact that the parity condition holds also for $G'; I_1, \cdots, I_p; R_1, R_2, C_2, \cdots, C_{k+1}$. Since the cut condition does hold for $G'; I_1, \cdots, I_p; C_1, \cdots, C_k$, it follows that

$$\text{(35)} \quad \operatorname{mincr}(R_1,D')+\operatorname{mincr}(R_2,D')+\operatorname{mincr}(C_{k+1},D')>\operatorname{mincr}(C_1,D').$$

Hence

$$\text{(36)} \quad \operatorname{cr}(P_1^1,D')=\operatorname{cr}(R_1,D')+\operatorname{cr}(R_2,D')+\operatorname{cr}(C_{k+1},D')>\operatorname{mincr}(C_1,D').$$

Therefore,

$$\text{(37)} \quad \begin{aligned}\operatorname{cr}(G,D')&\geqq \sum_{i=1}^{k}\sum_{j=1}^{t_i}\lambda_i^j\cdot \operatorname{cr}(P_i^j,D')>\sum_{i=1}^{k}\sum_{j=1}^{t_i}\lambda_i^j\cdot\operatorname{mincr}(C_i,D')\\&=\sum_{i=1}^{k}\operatorname{mincr}(C_i,D').\end{aligned}$$

However, (34) and (37) imply

$$\text{(38)} \quad \begin{aligned}\operatorname{mincr}(R_1,D')+\operatorname{mincr}(R_2,D')+\sum_{i=2}^{k+1}\operatorname{mincr}(C_i,D')&\geqq\operatorname{cr}(G,D')+2\\&>\sum_{i=1}^{k}\operatorname{mincr}(C_i,D')+2,\end{aligned}$$

contradicting the claim.

So (33) holds, and hence by part I of this proof there exist pairwise edge-disjoint paths $Q_1' \sim R_1$, $Q_1'' \sim R_2$, $Q_2 \sim C_2, \cdots, Q_k \sim C_k$, $Q_{k+1} \sim C_{k+1}$. By sticking Q_1', Q_{k+1}, Q_1'' to one path, which is homotopic to C_1, we obtain paths as required. □

REFERENCES

[1] C. VAN HOESEL AND A. SCHRIJVER, *Edge-disjoint homotopic paths in a planar graph with one hole*, J. Combin. Theory Ser. B, 48 (1990), pp. 77–91.

[2] M. KAUFMANN AND K. MEHLHORN, *On local routing of two-terminal nets*, J. Combin. Theory Ser. B, to appear.

[3] H. OKAMURA AND P. D. SEYMOUR, *Multicommodity flows in planar graphs*, J. Combin. Theory Ser. B, 31 (1981), pp. 75–81.

[4] A. SCHRIJVER, *Decomposition of graphs on surfaces and a homotopic circulation theorem*, J. Combin. Theory Ser. B, to appear.

SIAM J. DISC. MATH.
Vol. 4, No. 1, pp. 139-149, February 1991

014

REPRESENTATIONS OF GRAPHS ON A CYLINDER*

ROBERTO TAMASSIA† AND IOANNIS G. TOLLIS‡

Abstract. A complete characterization of the class of graphs that admit a *cylindric visibility representation* is presented, where vertices are represented by intervals parallel to the axis of the cylinder and the edges correspond to pairs of visible intervals. Moreover, linear time algorithms are given for testing the existence of and constructing such a representation. Important applications of cylindric visibility representations can be found in the layout of regular VLSI circuits, such as linear systolic arrays and bit-slice architectures. Also, alternative "dual" characterizations are presented of the graphs that admit visibility representations in the plane and in the cylinder. It is interesting to observe that neither of these two classes is contained in the other, although they have a nonempty intersection.

Key words. visibility graph, visibility representation, design and analysis of algorithms, computational geometry, cylinder, planar graph, caterpillar

AMS(MOS) subject classifications. 68R10, 68U05, 05C10, 05C75

1. Introduction. The concept of *visibility* plays a fundamental role in a variety of geometric applications [10]. Of particular interest are the problems dealing with visibility graphs between parallel *intervals*, which have various applications in VLSI layout [4], [7], [8], [13], [18], motion planning [5], [11], and graph drawing [1], [12], [14], [15]. The combinatorial properties of these graphs have also been extensively investigated [2], [14], [17], [19]. Visibility graphs also arise in the well-known hidden-surface elimination problem for two- and three-dimensional figures.

Let I be a set of parallel intervals in the plane, where an interval is a segment that might or might not contain one or both of its endpoints. Two intervals are said to be *visible* if they can be joined by a line orthogonal to them that does not intersect any other interval of I. The *visibility graph* of I is the graph whose vertices are the intervals of I, and whose edges connect pairs of visible intervals. Conversely, a *visibility representation* for a graph G is a set of intervals whose visibility graph is isomorphic to G. It has been shown that G admits a visibility representation in the plane if and only if there exists a planar embedding for G such that all the cutpoints appear on the boundary of the same face [14], [19]. Furthermore, such a representation can be constructed in linear time [14].

In this paper we consider visibility on a cylindric surface, where vertices are associated with intervals parallel to the axis of the cylinder. Namely, we present a characterization of the class of graphs that admit a visibility representation in the cylinder, and give linear time algorithms for testing the existence of and constructing such a representation. Important applications of *cylindric visibility representations* can be found in the layout of regular VLSI circuits, such as linear systolic arrays and bit-slice architectures [9]. Also, we present alternative "dual" characterizations of the graphs that admit visibility representations in the plane and in the cylinder. It is interesting to observe that neither of these two classes is contained in the other, but they have a nonempty intersection.

* Received by the editors January 16, 1989; accepted for publication (in revised form) October 31, 1989.

† Department of Computer Science, Brown University, Providence, Rhode Island 02912-1910. This research was partially supported by Cadre Technologies, Incorporated.

‡ Department of Computer Science, The University of Texas at Dallas, Richardson, Texas 75083-0688. This research was partially supported by Texas Advanced Research Program grant 3972.

A graph admits a visibility representation in the plane or cylinder if and only if all its connected components do. Hence, in this paper we only consider connected graphs. Let $G = (V, E)$ be a connected graph. A *cutpoint* of G is a vertex whose removal disconnects G. If G has at least one cutpoint, it is said to be 1-*connected*; otherwise, it is said to be 2-*connected*. A *block* of G is a maximal 2-connected subgraph of G. The *block-cutpoint* tree T of G is the graph whose vertices are the blocks and the cutpoints of G, and whose edges connect every block B to the cutpoints contained in B. T can be constructed in $O(|E|)$ time by using depth-first search [3]. A *caterpillar* is a tree whose nonleaf vertices form a path. Figure 1(a) shows a graph whose block-cutpoint tree, given in Fig. 1(b), is a caterpillar. Blocks and cutpoints are represented by large and small circles, respectively.

We show that a graph admits a cylindric visibility representation if and only if it is planar and its block-cutpoint tree is a caterpillar. We present linear time algorithms for testing the existence of and constructing a cylindric visibility representation for a given graph. Also, we characterize visibility representations in the plane and in the cylinder by means of the block-cutpoint trees of the dual graphs. Namely, we show that a planar graph G admits (i) a planar visibility representation if and only if it admits a dual graph G^* whose block-cutpoint tree is a star; and (ii) a cylindric visibility representation if and only if it admits a dual graph G^* whose block-cutpoint tree is a path.

Note that there exist planar graphs whose block-cutpoint tree is a caterpillar and such that in no planar embedding all the cutpoints appear on the boundary of the same face. Indeed, this is the case for the graph of Fig. 1. Conversely, the existence of a visibility representation in the plane does not impose any restriction on the structure of the block-cutpoint tree. For example, Fig. 2(a) shows a graph that admits a visibility representation in the plane, but whose block-cutpoint tree (Fig. 2(b)) is not a caterpillar. Therefore, visibility in the plane does not imply visibility in the cylinder, and vice versa.

In the next section we introduce the concept of *cylindric orientation* and show its relation with cylindric visibility representations. In § 3 we present a linear time algorithm that constructs a cylindric visibility representation for any 2-connected planar graph.

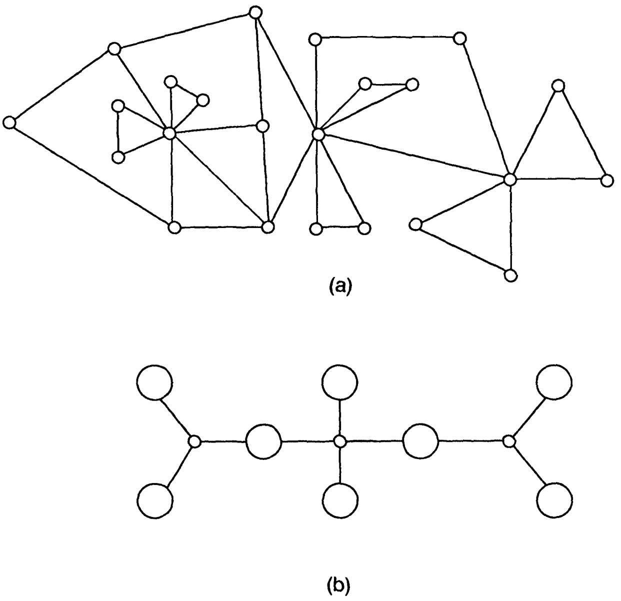


FIG. 1. (a) *A* 1-*connected graph and* (b) *its block-cutpoint tree.*

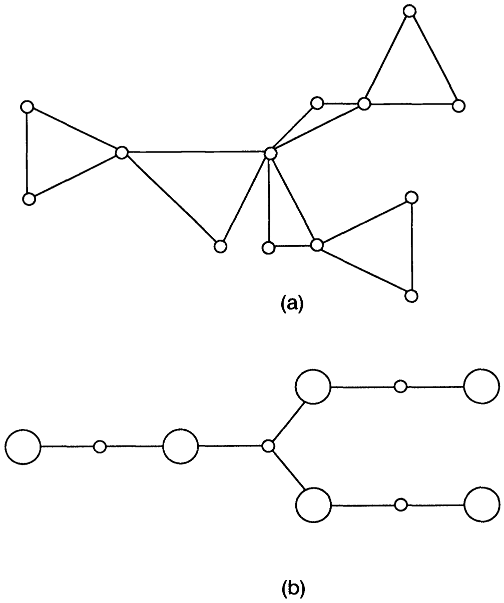


FIG. 2. (a) *A graph that admits a visibility representation in the plane*; (b) *the block-cutpoint tree of the graph in part* (a).

The characterization of cylindric visibility representations is given in § 4. Finally, the dual characterizations are presented in § 5.

2. Cylindric orientations. An (*infinite*) *cylinder* C is the locus of points at the same distance from a straight line, called the *axis* of the cylinder. A cylinder is also the union of an infinite family of circles with their center on the axis and drawn on a plane orthogonal to the axis. Alternatively, C is the union of an infinite family of lines parallel to the axis, and at the same distance from it. The circles and lines of C naturally define a coordinate system, since every point of C is the unique intersection of a line and a circle. Every point of C will be denoted by a pair (x, θ), where x is measured on the axis and θ is the angle with respect to some reference line and some "clockwise" orientation, with $0 \leqq \theta < 2\pi$.

We will consider *cylindric embeddings* of graphs, where vertices are mapped into points of the cylinder, and edges are mapped into nonintersecting Jordan curves on the cylinder. Clearly, a graph admits a cylindric embedding if and only if it is planar. Unlike planar embeddings, cylindric embeddings can have two unbounded faces, which are referred to as the *leftmost face* and *rightmost face* of the embedding.

Let Γ be a cylindric embedding with distinct leftmost and rightmost faces. A cycle γ of Γ is said to *wrap around* cylinder C if it intersects any Jordan curve on the surface of C with endpoints in the leftmost and rightmost faces of Γ, respectively. In other words, the removal of γ disconnects C into two unbounded pieces. A *cylindric orientation* Γ' of Γ is an orientation of the edges of Γ such that:

(1) Γ' has no sources (vertices without incoming edges) and no sinks (vertices without outgoing edges);

(2) every directed cycle wraps around C in the clockwise direction.

The following lemma gives two important properties of cylindric orientations. It can be proved using arguments similar to the ones in the proof of Lemmas 1 and 2 of [14], which give analogous properties for planar *st*-graphs.

LEMMA 1. *Let Γ' be a cylindric orientation. We have*:

(1) *for every vertex v of Γ', the incoming (outgoing) directed edges appear consecutively around v (see Fig.* 3(a));

(2) *the boundary of each internal face of Γ' consists of two directed paths with common origin and destination (see Fig.* 3(b)).

In the following, the concepts of *left* and *right* refer to the orientation of the edges in a cylindric orientation. For example, the face to the left of a directed edge $[u, v]$, denoted $LEFT(u, v)$, is the face containing $[u, v]$ that appears on the left side when traversing $[u, v]$ from vertex u to vertex v. Face $RIGHT(u, v)$ is symmetrically defined. By Lemma 1, for each vertex v, there are two distinct faces that separate the incoming edges from the outgoing edges. These faces are denoted by $LEFT(v)$ and $RIGHT(v)$, where $LEFT(v)$ is the face to the left of the leftmost incoming and outgoing edges, and $RIGHT(v)$ is the face to the right of the rightmost incoming and outgoing edges (see Fig. 3(a)).

An *interval* of cylinder C is a topologically connected subset of a line of C. An *arc* of C is a topologically connected subset of a circle of C. Let I be a set of disjoint intervals of C. Two intervals i_1 and i_2 of I are said to be *visible* if there is an arc a of C with endpoints on i_1 and i_2 that does not intersect any other interval of I. Arc a is said to be a *visibility arc* between i_1 and i_2 and is directed according to the clockwise orientation. In other words, let θ_1 and θ_2 be the angles of i_1 and i_2, respectively, with $\theta_1 < \theta_2$; if a does not intersect the reference line than we direct a from i_1 to i_2; otherwise, we direct a from i_2 to i_1.

A *cylindric visibility representation* K for a graph G is a mapping of vertices of G into disjoint intervals of C, called *vertex intervals*, such that there is an edge (u, v) if and only if the intervals associated with u and v are *visible*. Each edge of G is mapped into either one or two visibility arcs. In the latter case, the union of the two visibility arcs is a circle of C. For simplicity, we use the same name for the vertices of the graph and their corresponding vertex intervals. Figure 4(b) shows a cylindric visibility repre-

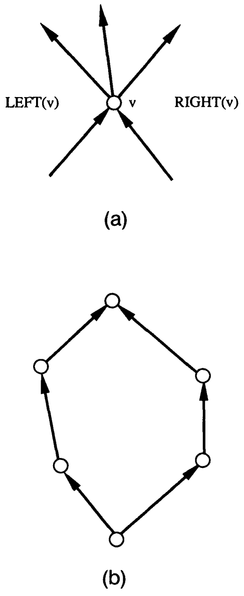


FIG. 3. (a) *Incoming and outgoing directed edges around a vertex*; (b) *directed paths forming the boundary of a face*.

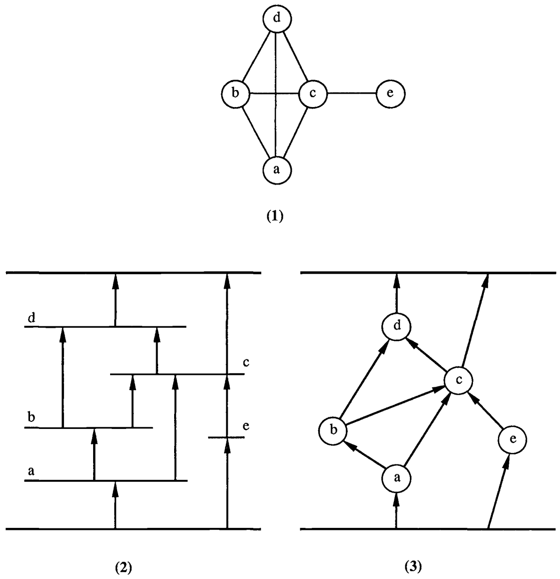


FIG. 4. (a) *A planar graph G*; (b) *a cylindric visibility representation K for G*; *and* (c) *the cylindric orientation associated with K*.

sentation for the graph of Fig. 4(a). Note that the top and bottom heavy lines represent the same line of the cylinder, say the reference line.

From K we can construct a cylindric orientation Γ by shrinking every vertex interval into a point and accordingly deforming the visibility arcs, as shown in Fig. 4(c). Note that the undirected graph obtained from Γ by ignoring the edge directions and the double edges is isomorphic to G. The above construction shows that any graph that admits a cylindric visibility representation must be planar.

Given a planar embedding Π of a 2-connected planar graph G, and two distinct faces f_1 and f_2 of Π, we can construct a cylindric orientation Γ of G with the same topology as Π, leftmost face f_1, and rightmost face f_2.

ALGORITHM *CYL-ORIENT*.

Input: A 2-connected n-vertex planar graph $G = (V, E)$, with $n \geqq 3$. A planar embedding Π for G, and two distinct faces f_1 and f_2 of Π.

Output: A cylindric orientation Γ of G with the same topology as Π, leftmost face f_1, and rightmost face f_2.

(1) Embed Π on the surface of a sphere.

(2) Pierce two holes in the sphere inside faces f_1 and f_2, and deform the pierced sphere into a cylinder. This gives a cylindrical embedding Π'.

(Note that faces f_1 and f_2 might share one or more vertices and/or edges. This, however, does not affect the rest of the algorithm.)

(3) Orient the edges on the boundary of face f_1 in the clockwise direction. Mark face f_1 as "oriented."

(4) Let f be an unmarked face that is adjacent to a marked face. Since G is 2-connected, the edges on the boundary of f can be partitioned into two simple paths, γ_1 and γ_2, where γ_1 contains the oriented edges, and γ_2 contains the unoriented edges. Let u and v be the common endpoints of these two paths such that γ_1 is directed from u to v. We orient all the edges of γ_2 in the direction from u to v. This step is repeated until all faces are marked.

(Note that this process is essentially a visit of the dual graph.)

THEOREM 1. *Algorithm CYL-ORIENT constructs a cylindric orientation Γ for an n-vertex ($n \geqq 3$) 2-connected planar graph G in $O(n)$ time.*

Proof. The algorithm maintains the invariant that after each iteration of step 4 the oriented portion of the graph is a cylindric orientation. If the orientation of γ_2 creates a cycle in the oriented portion that does not wrap around C in the clockwise direction, such a cycle must be the union of γ_2 and of a directed path γ_3 from v to u. This implies that the union of γ_1 and γ_3 is a cycle that does not wrap around C, which violates the invariant. The correctness of the algorithm easily follows by induction. Regarding the time complexity, we observe that $O(1)$ time is spent at each edge. □

Note that the cylindric orientation Γ constructed by the above algorithm does not have double edges. The cylindric orientation for a 2-connected graph with two vertices is a cycle consisting of two symmetrically directed edges, and can be trivially constructed.

3. Construction of cylindric visibility representations. Given an acyclic digraph $D = (V, E)$, a *topological ordering* τ on D maps each vertex v into a nonnegative integer $\tau(v)$ such that $\tau(u) < \tau(v)$ for every directed edge $[u, v] \in E$. A topological ordering can be computed in $O(|V| + |E|)$ time by means of the following recursive formula:

(1) $\tau(s) = 0$, for every source-vertex $s \in V$

(2) $\tau(v) = 1 + \max_{[u,v] \in E} \tau(u)$.

A *face* of a cylindric visibility representation is a maximal topologically connected region of the cylinder delimited by the vertex intervals and the visibility arcs. The faces of a cylindric visibility representation are in one-to-one correspondence with the faces of the associated cylindric orientation. In the rest of this section we restrict our attention to 2-connected graphs, and show how to construct a cylindric visibility representation in linear time.

ALGORITHM *VISIB-2C*.

Input: A 2-connected n-vertex planar graph $G = (V, E)$. A planar embedding Π for G, and two distinct faces f_1 and f_2.

Output: A cylindric visibility representation K for G with leftmost face f_1 and rightmost face f_2.

(1) Construct a cylindric orientation Γ of G with leftmost face f_1 and rightmost face f_2, using algorithm *CYL-ORIENT*.
(The digraph Γ intuitively represents a "circular order" of the vertex intervals in the θ-direction.)

(2) Construct the dual digraph Γ^* of Γ, where dual edges are oriented "from left to right." Namely, the dual of $[u, v]$ is the directed edge $[LEFT(u, v), RIGHT(u, v)]$. Γ^* is acyclic and has exactly one source (f_1) and exactly one sink (f_2).
(A directed edge $[f, g]$ in Γ^* implies that face f will be to the left of face g in the cylindric visibility representation.)

(3) Compute a topological ordering α on the digraph obtained from Γ by removing the edges that intersect a path of Γ^* from f_1 to f_2.
(The ordering α will be used for determining the θ-coordinates of the vertex intervals and the visibility arcs.)

(4) Compute a topological ordering β on Γ^*.
(The ordering β will be used for determining the x-coordinates of the vertex intervals and the visibility arcs.)

```
(5) Let $\theta_0 = 2\pi/n$,
    for each $v \in V$ do
      draw a vertex interval from $(\beta(LEFT(v)), \alpha(v)\theta_0)$ to $(\beta(RIGHT(v)), \alpha(v)\theta_0)$,
      which includes the left endpoint but not the right one;
    endfor;
```

```
(6) For each $[u, v] \in \Gamma$ do
        let $x = \frac{1}{2}(\beta(LEFT(u, v)) + \beta(RIGHT(u, v)))$;
        draw a visibility arc directed from $(x, \alpha(u)\theta_0)$ to $(x, \alpha(v)\theta_0)$;
    endfor;
```

An example of the construction performed by Algorithm *VISIB-2C* is shown in Fig. 5: Fig. 5(a) shows a planar embedding of a 2-connected graph G; Fig. 5(b) shows the cylindric orientation Γ computed in step 1, in heavy lines, and its dual Γ^* computed in step 2; primal and dual vertices are labeled with the value of the corresponding topological ordering, computed in Step 3 or 4; finally, Fig. 5(c) shows the cylindric visibility representation K computed in steps 5 and 6.

Now, we discuss the complexity of the algorithm. Since G is a planar graph, both G and its dual G^* have $O(n)$ vertices and edges. From the results of § 2, steps 1 and 2 take $O(n)$ time. The computation of α and β in steps 3 and 4 can be performed in $O(n)$ time. Finally, steps 5 and 6 take $O(n)$ time. Hence, we have the following theorem.

THEOREM 2. *Algorithm VISIB-2C constructs a cylindric visibility representation K for an n-vertex 2-connected planar graph G in $O(n)$ time.*

4. Cylindric visibility representations and caterpillars. As discussed in the introduction, not every 1-connected planar graph admits a cylindric visibility representation. Here, we provide a necessary and sufficient condition that characterizes the class of graphs that admits such a representation. Before we prove the main theorem of this section, we need some preliminary results.

LEMMA 2. *Let K be a cylindric visibility representation for G, and* γ be a circle of the cylinder that intersects at least three vertex intervals of K. *Then there is a cycle in G that consists of exactly the vertices associated with the vertex intervals intersected by* γ.

Proof. Any two consecutive vertex intervals intersected by γ are visible, and thus the corresponding vertices are adjacent. □

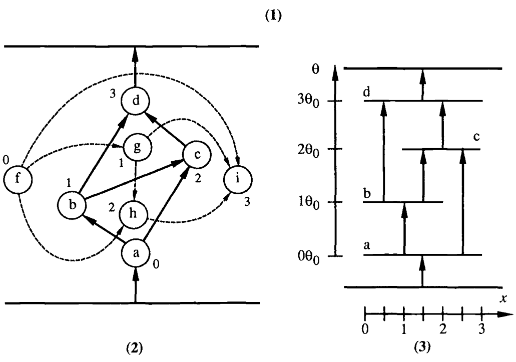


FIG. 5. *A running example for Algorithm* VISIB-2C: (a) *a planar embedding of a* 2*-connected graph G*; (b) *the cylindric orientation* Γ *for G* (*in heavy lines*), *computed in step* 1, *and its dual* Γ^* *computed in step* 2; (c) *the cylindric visibility representation for G computed in steps* 3–6.

We define a *section* of a cylinder C as the portion of C generated by the rotation of an interval i of C around the axis.

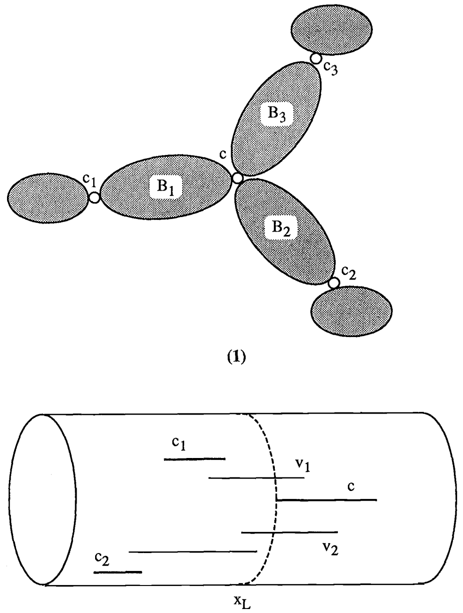
LEMMA 3. *Let K be a cylindric visibility representation for G, and c be a cutpoint of G. Then the vertex interval of any other cutpoint c' of G is not completely contained in the section generated by the vertex interval of c.*

Proof. Assume that there is a cutpoint c', distinct from c, that is completely contained in the section generated by c. There must exist a block B that contains c' but not c. Let v be a vertex of B adjacent to c'. By Lemma 2, there is a cycle of G associated with a circle of the cylinder intersecting c, c', and v. Since every cycle must contain vertices of the same block, we obtain a contradiction. □

The following theorem characterizes the class of graphs that admit a cylindric visibility representation.

THEOREM 3. *A planar graph G admits a cylindric visibility representation if and only if its block-cutpoint tree T is a caterpillar.*

Proof. Necessity. Assume, for a contradiction, that T is not a caterpillar. Then G has either a cutpoint contained in three or more nonleaf blocks, or a block containing three or more cutpoints. We discuss only the first case. The second case is similar. Let c be a cutpoint contained in distinct nonleaf blocks B_1, B_2, and B_3, and let x_L and x_R be the x-coordinates of the left and right endpoints of c, respectively. For $i = 1, 2, 3$, consider a cutpoint c_i in B_i distinct from c. Such cutpoints exist since B_1, B_2, and B_3 are not leaves of T (see Fig. 6(a)). From Lemma 3, at least two of these cutpoints, say c_1 and c_2, are both on the same side of the section generated by c, i.e., either their left endpoints are on the left of x_L or their right endpoints are on the right of x_R. Without loss of generality, assume the first case. There must be vertices v_1 in B_1 and v_2 in B_2 whose vertex intervals intersect the circle of the cylinder at abscissa x_L (see Fig. 6(b)). By Lemma 2, there is a cycle of G associated with this circle. Clearly, such cycle contains vertices from both B_1 and B_2, which is a contradiction to the fact that B_1 and B_2 are distinct blocks.



(1)

(2)

FIG. 6. *Example for the proof of Theorem* 3 (*necessity*).

Sufficiency. We show how to construct a cylindric visibility representation for the graph G from the cylindric visibility representations of its blocks. Denote the nonleaf blocks of G as $B_1, \cdots, B_k$, and the cutpoints of G as $c_0, c_1, \cdots, c_k$, where block B_i contains the cutpoints c_{i-1} and c_i, for $i = 1, \cdots, k$. In the construction of a cylindric visibility representation K_B for a block B of G, we have two cases.

Case 1. B is a leaf of T.

In this case B contains exactly one cutpoint c of G. We construct K_B such that both the leftmost and rightmost faces of K_B contain the cutpoint c. This can be done by selecting faces f_1 and f_2 in Algorithm *VISIB-2C* as two faces containing c. Note that all the vertex intervals of K_B are contained in the section of c.

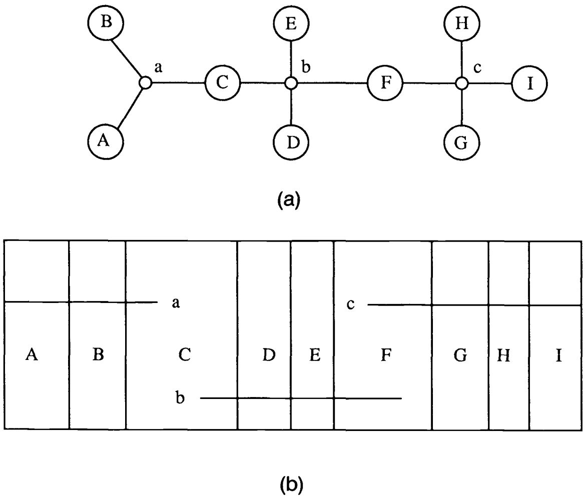
Case 2. $B = B_i$, for some i in $\{1, \cdots, k\}$.

In this case, we construct K_B such that the leftmost face contains cutpoint c_{i-1} and the rightmost face contains cutpoint c_i.

At this point the cylindric visibility representations of all the blocks have been constructed. For each block B, slice K_B at abscissas $\beta(f_1)$ and $\beta(f_2)$. The sections so obtained are then glued together along a common axis, in such a way that, for each $i = 1, \cdots, k-1$, all the sections corresponding to leaf blocks connected to c_i are placed between the sections of blocks B_i and B_{i+1}. The leaf blocks connected to c_0 and c_k are placed before block B_1 and after block B_k, respectively. To complete the construction, the sections must be rotated so that the vertex intervals of different sections corresponding to the same cutpoint become aligned. □

The construction described in the proof of Theorem 3 is illustrated in Fig. 7. Figure 7(a) shows the block-cutpoint tree of a graph, where blocks are denoted by uppercase letters, and cutpoints by lowercase letters. Figure 7(b) shows the arrangement of the sections corresponding to the blocks along the cylinder. Note the leaf blocks A, B, D, E, G, H, and I are contained in the sections of their respective cutpoints. Also, nonleaf blocks C and F have one cutpoint at the left and the other cutpoint at the right, which link them to the rest of the representation.

The time complexity of the construction described in the proof of Theorem 3 is analyzed as follows. Let n be the number of vertices of G. The blocks and cutpoints of G can be computed in time $O(n)$ using depth-first search. By Theorem 2, the cylindric visibility representation of each block B is constructed time $O(m_B)$, where m_B is the



(a)

(b)

FIG. 7. *Construction in the proof of Theorem* 3 (*sufficiency*).

number of edges in B. This sums up to $O(n)$ because each edge belongs to a single block. Finally, combining the sections of the blocks into a unique cylindric visibility representation is easily done in $O(n)$ time.

To test whether a planar graph G admits a cylindric visibility representation, we construct its block-cutpoint tree T, and determine whether it is a caterpillar. This can be done by removing all the leaves from T, and verifying that the resulting graph is a simple path. The above computation takes $O(n)$ time. Hence, we conclude that the following theorem holds.

THEOREM 4. *Given a graph G with n vertices, there are $O(n)$ time algorithms for testing the existence of and constructing a cylindric visibility representation for G.*

5. A dual characterization of visibility representations. In this section we provide "dual" characterizations of the classes of graphs that admit planar visibility representations and cylindric visibility representations.

THEOREM 5. *A planar graph G admits a planar visibility representation if and only if it admits a dual graph G^* whose block-cutpoint tree is a star.*

Proof. G is 2-connected if and only if all of its duals are [6, Ex. 11.4, p. 124]. Hence, the theorem is trivially true for 2-connected graphs. Now, suppose that G is 1-connected. We recall that G admits a planar visibility representation if and only if it has a planar embedding such that all the cutpoints of G are on the same face, say the external face [14]. Hence, if G admits a planar visibility representation there exists a planar embedding of G such that all blocks are embedded in the external face. Let G^* be the dual graph associated with this embedding. G^* has exactly one cutpoint, i.e., the external face.

Conversely, suppose that G admits a dual graph G^* whose block-cutpoint tree is a star. The embedding associated with G^* has a face (the center of the star) containing all blocks, and hence all cutpoints of G. This implies that G admits a planar visibility representation. □

THEOREM 6. *A planar graph G admits a cylindric visibility representation if and only if it admits a dual graph G^* whose block-cutpoint tree is a path.*

Proof. Suppose that G admits a dual graph G^* whose block-cutpoint tree T^* is a path. Let T be the block-cutpoint tree of G. If T is not a caterpillar, then G has either a block containing three or more cutpoints, or a cutpoint contained in three or more nonleaf blocks. We discuss only the first case. The second case is similar. Let B be a block containing distinct cutpoints c_1, c_2, and c_3. Let B_1, B_2, and B_3 be blocks of G distinct from B that contain cutpoints c_1, c_2, and c_3, respectively. Since B^* has at most 2 cutpoints, one of them, denoted f, must contain at least two of these blocks, say B_1 and B_2, which are connected to B only through cutpoints c_1 and c_2. Hence, in T^*, f is adjacent to B^*, B_1^*, and B_2^*, a contradiction. Therefore, T is a caterpillar and, by Theorem 3, G admits a cylindric visibility representation.

Now, suppose that G admits a cylindric visibility representation, and consider the one constructed in the proof of Theorem 3. By construction, for every block B, the rest of the cylindric visibility representation is contained in the leftmost and rightmost faces of K_B. Shrink every vertex segment to obtain a cylindric embedding of G. The above property is preserved by this transformation, and hence the block-cutpoint tree of the dual graph G^* is a simple path. □

6. Extensions and open problems. In the definition of visibility on the cylinder we can exchange the role of intervals and arcs, so that the family I consists of circular arcs, and visibility is defined by intervals parallel to the axis. This new definition induces a different type of cylindric visibility representation, which is topologically equivalent to an *open visibility representation on the sphere*, where the family I consists of open arcs of *parallels* (i.e., entire parallels are not allowed), and visibility is defined by arcs of *meridians* [16]. A characterization of the class of graphs that admit an open visibility representation on the sphere (or, equivalently, the aforementioned variation of cylindric visibility representation) is presented in [16], where it is developed in the framework of a new representation of planar graphs, called *tessellation representation*. We have the following theorem.

THEOREM 7. [16] *Let G be a planar undirected graph with n vertices. G admits an open visibility representation on the sphere if and only if G admits an embedding such that all the cutpoints are on the boundary of at most two faces. Also, there are $O(n)$-time algorithms for testing the existence of and constructing an open visibility representation on the sphere for G.*

It would be interesting to characterize the class of graphs that admit a *toroidal visibility representation*, where the intervals of I wrap around the torus in one way and visibility is defined by arcs wrapping around the torus in the other (orthogonal) way. In this case, the problem appears to be far more difficult, since the graphs that admit such a representation need not be planar.

REFERENCES

[1] G. DI BATTISTA AND R. TAMASSIA, *Algorithms for plane representations of acyclic digraphs*, Theoret. Comput. Sci., 61 (1988), pp. 175–198.

[2] P. DUCHET, Y. HAMIDOUNE, M. LAS VERGNAS, AND H. MEYNIEL, *Representing a planar graph by vertical lines joining different levels*, Discrete Math., 46 (1983), pp. 319–321.

[3] S. EVEN, *Graph Algorithms*, Computer Science Press, Potomac, MD, 1979.

[4] M. GAREY, D. JOHNSON, AND H. SO, *An application of graph coloring to printed circuit testing*, IEEE Trans. Circuits and Systems, CAS-23 (1976), pp. 591–598.

[5] L. J. GUIBAS AND F. F. YAO, *On translating a set of rectangles*, in Advances in Computing Research, vol. 1, F. P. Preparata, ed., JAI Press Inc., Greenwich, CT, pp. 61–77, 1983.

[6] F. HARARY, *Graph Theory*, Addison-Wesley, Reading, MA, 1969.

[7] M. Y. HSUEH AND D. O. PEDERSON, *Computer-aided layout of* LSI *circuit building blocks*, Proc. IEEE Int. Symp. on Circuits and Systems (1979), pp. 474–477.

[8] E. LODI AND L. PAGLI, *A* VLSI *algorithm for a visibility problem*, in VLSI: Algorithms and Architectures, P. Bertolazzi and F. Luccio, eds., North-Holland, Amsterdam, pp. 125–134, 1985.

[9] K. MEHLHORN AND W. RULLING, *Compaction on the torus*, VLSI Algorithms and Architectures, Proc. Aegean Workshop on Computing '88, Corfu, Greece, 1988, Lecture Notes in Computer Science 319, 1988, pp. 212–225. Springer-Verlag, Berlin, New York.

[10] J. O'ROURKE, Art Gallery Theorems and Algorithms, Oxford University Press, Oxford, 1987.

[11] I. RIVAL AND J. URRUTIA, *Representing orders by translating convex figures in the plane*, Order, 4 (1988), pp. 319–339.

[12] P. ROSENSTIEHL AND R. E. TARJAN, *Rectilinear planar layouts of planar graphs and bioplar orientations*, Discrete & Computational Geometry, 1 (1986), pp. 342–351.

[13] M. SCHLAG, F. LUCCIO, P. MAESTRINI, D. T. LEE, AND C. K. WONG, *A visibility problem in* VLSI *layout compaction*, in Advances in Computing Research, vol. 2, F. P. Preparata, ed., JAI Press Inc., Greenwich, CT, 1985, pp. 259–282.

[14] R. TAMASSIA AND I. G. TOLLIS, *A unified approach to visibility representations of planar graphs*, Discrete & Computational Geometry, 1, 1986, pp. 321–341.

[15] ———, Planar grid embedding in linear time, IEEE Trans. Circuits and Systems, CAS-36 (1989), pp. 1230–1234.

[16] ———, *Tessellation representations of planar graphs*, Proc. 27th Annual Allerton Conf., 1989, University of Illinois at Urbana-Champaign, pp. 48–57.

[17] C. THOMASSEN, *Plane representations of graphs*, in Progress in Graph Theory, J. A. Bondy and U. S. R. Murty, eds., Academic Press, New York, 1984, pp. 43–69.

[18] S. WIMER, I. KOREN, AND I. CEDERBAUM, *Floorplans, planar graphs, and layouts*, IEEE Trans. Circuits and Systems, 35 (1988), pp. 267–278.

[19] S. K. WISMATH, *Characterizing bar line-of-sight graphs*, Proc. ACM Symp. on Computational Geometry, Baltimore, MD, 1985, pp. 147–152.

SIAM J. DISC. MATH.
Vol. 4, No. 2, pp. 151-163, May 1991

MOTION PLANNING, TWO-DIRECTIONAL POINT REPRESENTATIONS, AND ORDERED SETS*

FAWZI AL-THUKAIR†, ANDRZEJ PELC‡, IVAN RIVAL§, AND JORGE URRUTIA§

Abstract. Ordered sets are used as a computational model for motion planning problems. Every ordered set has a two-directional point representation using subdivisions. These subdivision points correspond to direction changes along the path of motion.

Key words. motion planning, ordered set, diagram, two-directional point representation, subdivision, tree, cycle

AMS(MOS) subject classifications. 06A10, 52A37, 68E10

How may a robot arm be moved to grasp a delicate object from a crowded shelf without unwanted collisions?

How may a cluster of figures on a computer screen be shifted about to clear the screen without altering their integrity and without collisions?

These questions highlight instances of the recent and rapidly growing theme of "motion planning." Rival and Urrutia (1987) initiated the study of motion planning using a computational model based on the theory of ordered sets. Subsequently, Nowakowski, Rival, and Urrutia (1987) proposed the problem to characterize the ordered sets here called "two-directional orders."

One example of a motion planning problem is the following. Given a finite collection of disjoint figures in the plane, is it possible to assign to each a single direction of motion so that this collection of figures may be separated, through an arbitrarily large distance, by translating each figure one at a time, along its assigned direction? In this model we have considered only convex figures in the plane. Indeed, given a collection of disjoint, convex figures, the separability problem always has a positive solution. Loosely speaking, at least one of the convex figures is on the "outside" or "boundary" of the collection, and therefore it may be removed. Of course, instead of disjoint figures in the plane we can consider robots moving along assigned directions.

To make the mathematical matter more definite, we will here idealize each robot as a point (a circle of negligible radius) on the plane. Suppose that each point is assigned a single direction of motion not necessarily all the same. For points A and B we say that *B obstructs A* if the line joining A to B follows the direction assigned to A. We write $A \twoheadrightarrow B$. More generally, we write $A < B$ if there is a sequence $A = A_1 \twoheadrightarrow A_2 \twoheadrightarrow \cdots \twoheadrightarrow A_k = B$. This relation $<$ is transitive. It is appropriate to call this binary relation $<$ the *blocking* relation. If the blocking relation has no directed cycles then it is antisymmetric, also. In that case the blocking relation $<$ is a (strict) order on the given set of points. If each of the points is assigned the same direction, we call the relation *one-directional*. In that case, any maximal point (with respect to $<$) is on the "outside."

We say that a collection of points, each assigned one of m directions, is an *m-directional point representation* of an ordered set P, if its blocking relation is identical to the ordering of P.

* Received by the editors May 12, 1988; accepted for publication (in revised form) April 5, 1990.

† Department of Mathematics, College of Science, P.O. Box 2454, King Saud University, Riyadh 11341, Saudi Arabia.

‡ Départment D'Informatique, Université du Québec à Hull, Hull J8X 3X7 Canada.

§ Department of Computer Science, University of Ottawa, Ottawa K1N 9B4 Canada.

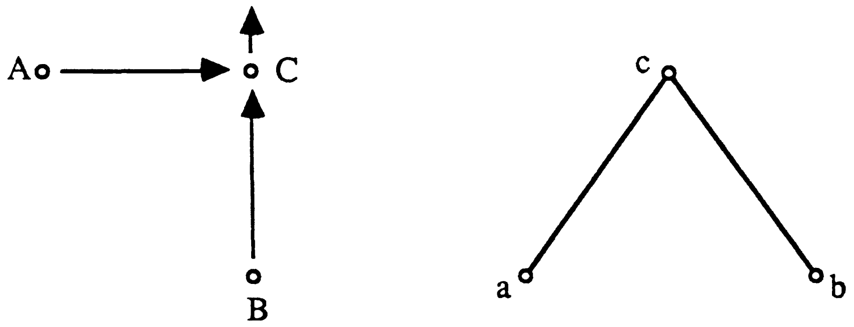
Nowakowski, Rival, and Urrutia (1987) considered ordered sets, which have an m-directional point representation, and called these *m-directional point orders* (see Fig. 1). Indeed, we may even imagine such point representations as models for an assembly line based on a many machine scheduling environment, in which the robots correspond to machines or machine parts.

Nowakowski, Rival, and Urrutia showed *that there are ordered sets with no m-directional point representation, for any positive integer m, yet every finite ordered set has a subdivision with such an m-directional point representation, for some m.* This subdivision consists precisely of the original ordered set with an extra element adjoined along some of the (covering) edges (with just the comparabilities induced, in each case, by just this edge).

Throughout the paper we will use the customary *order diagram* of an ordered set in which the y-coordinate of a point b is larger than that of another point a if $a < b$ and an edge joins them just if b is an upper cover of a. (Say that b is an *upper cover of* a (b *covers* a or a is a *lower cover of* b or a is *covered by* b) if $a < b$ and if $a < c \leqq b$ implies $b = c$.) Thus, an ordered set that contains an element with m lower covers requires at least m directions in its point representation—if it has one. We usually use upper case characters A, B, C, $\cdots$ to stand for the robots in the point representations and lower-case characters a, b, c, $\cdots$ for the elements of ordered sets and the same symbol $<$ for the order relation in both contexts.

An alternative, perhaps more suggestive, interpretation of subdivision is this: Let b cover a and suppose a subdivision point (a, b) is placed along the corresponding covering edge. In a corresponding two-directional point representation a robot A may itself be assigned two directions, in succession, the first followed until a junction corresponding to the subdivision point (a, b) and the second followed from this junction to B (see Fig. 2).

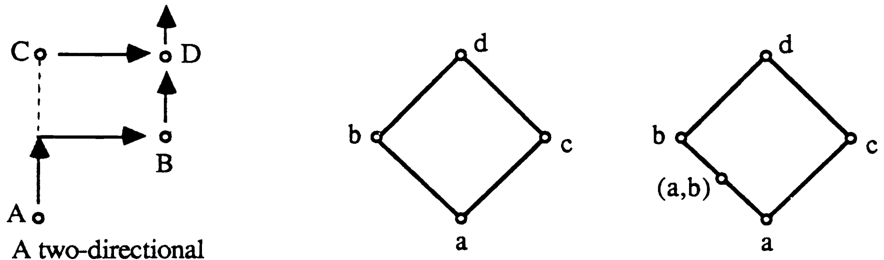
Note that, by transitivity, it may be that $A < B$ and $B < C$, that is, $A < C$, yet C is not "visible" from A along either a horizontal eastward or a vertical upward path. At the same time, although D covers A it may be that B lies along the line of sight from A to D, apparently "obstructing the visibility" between them (see Fig. 3). From the viewpoint of motion planning we may suppose that once B begins to move along its intended direction of motion there is an unobstructed path from A to D. In the interest of continuity



A two-directional point representation of the ordered set {a<b, a<c}.

An order diagram of the ordered set {a<c, b<c}.

FIG. 1



A two-directional point representation of a subdivision of {a<b<d, a<c<d}.

An ordered set {a<b<d, a<c<d}.

A subdivision of it.

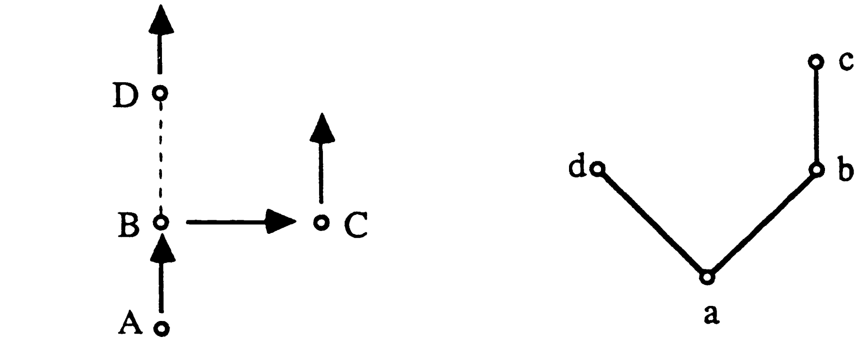
FIG. 2

we will insist, too, that all elements be assigned directions, including, in particular, the maximal elements, even though a maximal element is not constrained to precede any other.

Our leading problem is to characterize two-directional point orders among all orders. Here are our main results. The first highlights a class of ordered sets, each of whose members has a two-directional point representation. Call an ordered set a *tree* if its covering graph contains no cycle (a subset $a_1, a_2, \cdots, a_m$ of distinct points, $m \geqq 4$, such that a_i covers a_{i+1} or a_{i+1} covers a_i for each $i = 1, 2, \cdots, m-1$ and a_1 covers a_m or a_m covers a_1). A *simple cycle* in an ordered set is a cycle $a_1, a_2, \cdots, a_{2k}$, $k \geqq 2$, such that a_{2j} covers a_{2j-1} for each $j = 1, 2, \cdots, k$, and a_{2k} covers a_1. Moreover, we will call a cycle a_1, a_2, a_3, a_4, in which $a_1 < a_2 < a_4$ and $a_1 < a_3 < a_4$ a simple cycle also (cf. Fig. 4).

THEOREM 1. *Every tree in which each element has at most two lower covers has a two-directional point representation, yet an ordered set that contains a simple cycle has no two-directional point representation at all.*

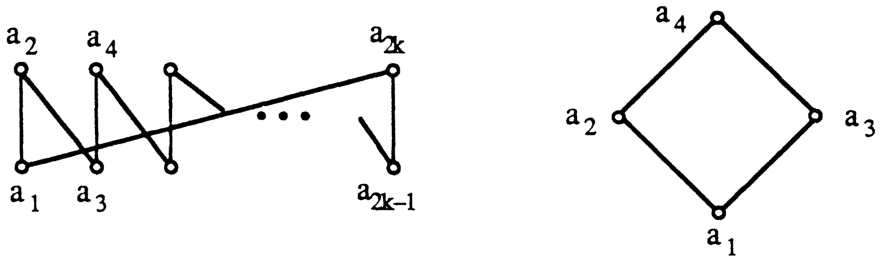
On the positive side we will also show that any "lexicographic sum" of ordered sets, with top and bottom, has a two-directional point representation, provided that both the index set and the blocks do, also.



A two-directional point representation of {a<b<c, a<d}.

An order diagram of {a<b<c, a<d}.

FIG. 3



Simple cycles

FIG. 4

How many subdivision points along any covering edge ensure that an ordered set has a two-directional point representation? Or, in the language of motion planning, how many changes of direction for any robot guarantee that an order has a two-directional point representation?

THEOREM 2. *For any ordered set in which each element has at most two lower covers, at most one subdivision point along some of its covering edges ensures that it has a two-directional point representation.*

In some sense this result is the best possible.

THEOREM 3. *There exist ordered sets in which each element has at most two lower covers such that almost half of its covering edges need be subdivided to ensure a two-directional point representation. Moreover, there are ordered sets in which each element has at most two lower covers with no two-directional point representation if every covering edge is subdivided exactly once.*

Note that while Theorem 2 ensures a two-directional point representation by subdividing some covering edges of P, according to Theorem 3 too many subdivisions may spoil the two-directional point representation.

We are still unable to characterize the ordered sets that have a two-directional point representation. Nevertheless, it seems to us that the solution to the bipartite case would shed light on the general problem.

Trees and cycles. It is easy to see that an ordered set with a two-directional point representation also has one in which the two directions are perpendicular. We will suppose throughout that these directions are northward (**n**) and eastward (**e**).

Our first aim is to show that no simple cycle has a two-directional point representation. Suppose that P is an ordered set with a two-directional point representation. Let a and b be distinct elements of P. If both a and b point northward and lie on the same vertical line in the representation of P, then they must be comparable. For if the y-coordinate of a is below the y-coordinate of b in this representation, then as a points northward, $a < b$; if the y-coordinate of b is below that of a then $b < a$. Now, let a, b be distinct lower covers of c in P. In the representation, c must be located along the "line of sight" of a and of b. Thus, if a and b had the same direction, then each would be along the line of sight of the other and, according to our observation above, a and b would be comparable. Therefore, we may suppose that a points northward and b eastward, say, and that, therefore, c lies at the point of intersection of the northward and eastward lines from these points. It follows, of course, that every element in P has at most two distinct lower covers.

From these preliminary remarks it is an easy matter to deduce that no simple cycle $a_1, a_2, \cdots, a_{2k}$, $k \geqq 3$ and k odd, has a two-directional representation. Suppose one did.

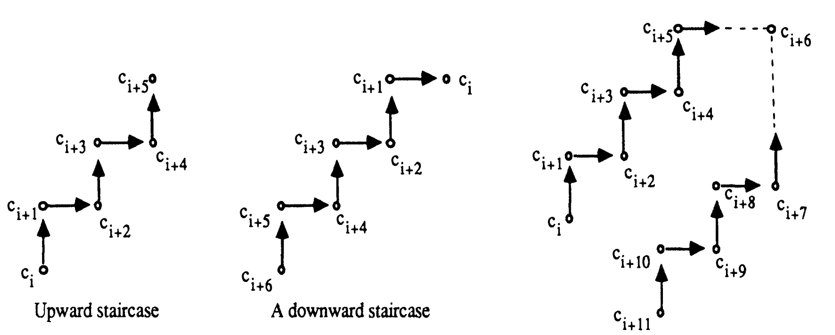
As a_1, a_3 are lower covers of a_2 they have different direction **n**, **e**, say, respectively. Then, a_5 has direction **n**, a_7 **e**, and so on, alternatively, which, of course, is impossible as k is odd.

We claim that no simple cycle at all has a two-directional point representation. The cases $a_1 < a_2 < a_4$ and $a_1 < a_3 < a_4$ as well as $a_1 < a_2, a_4$ and $a_3 < a_2, a_4$ can be checked directly to have none, as a simple longhand effort shows. For the remaining cases another remark is handy. Let $c_1, b_1, c_2, b_2, \cdots, c_m, b_m, c_{m+1}$, be a "zigzag," that is, c_1 covers b_1 and c_i covers b_{i-1} and b_i, for $2 \leqq i \leqq m$, and c_{m+1} covers b_m, and consider a two-directional point representation of it. We may suppose that its minimal elements b_1, $b_2, \cdots, b_m$ alternate in direction **n**, **e**, $\cdots$. As each b_i, $1 \leqq k \leqq m$, is covered by two of the c_j's, then both of them, namely c_{i-1} and c_i, lie along the line of sight of b_i. As c_{i-1} and c_i are noncomparable, neither can be along the line of sight of the other. It follows that in the representation, successive triples of the c_i's follow either an upward staircase pattern or a downward staircase pattern in which an upward staircase may meet a downward staircase with increasing subscript, yet a downward staircase continues only downward (see Fig. 5).

Let $a_1, a_2, \cdots, a_{2k}$, $k \geqq 3$, be an arbitrary simple cycle, that is, a_{2j} covers a_{2j-1}, $j = 1, 2, \cdots, k$, and a_{2k} covers a_1. Suppose that it has a two-directional point representation. Then its maximal elements must follow the staircase pattern indicated above. Since the sequence $a_2, a_4, \cdots$ of maximal elements will repeat following the enumeration of the cycle, at least one portion must be a downward staircase, and, in that case, must continue as a downward staircase throughout—which is impossible. Thus, no simple cycle at all has a two-directional point representation.

We now show by induction on $|P|$ that any ordered set P that is a tree does have a two-directional point representation. Let a be an endpoint of the covering graph of P, that is, either a maximal element of P with precisely one lower cover or else a minimal element with precisely one upper cover. Suppose that a is maximal, that b is its unique lower cover and that a two-directional point representation of $P - \{a\}$ is given. We may assume that b has direction **n**. We will locate a along the vertical from b above it. We may choose its y-coordinate less than any other point already on this vertical yet larger than b, and distinct from the y-coordinate of any other point. Assign a the direction **e**. This constitutes a two-directional point representation of P.

Suppose now that a is minimal with unique upper cover b and that $P - \{a\}$ has a two-directional point representation. There is no loss in generality to assume that b has direction **n**. By hypothesis, b has at most one lower cover c, besides a. Suppose c has



Upward staircase

A downward staircase

An upward staircase meets a downward staircase.

FIG. 5

direction **e**. Then we may locate a on the vertical below b with y-coordinate distinct from the y-coordinate of any other point and above any point already on this vertical yet lower than b. We may assign a the direction **n** to obtain a two-directional point representation of P. Now, suppose that c has direction **n**, in which case c lies on the vertical through b beneath it. Before locating a we make a small change to the representation of $P - \{a\}$ by shifting the location of b just an "epsilon" northward so that its y-coordinate is distinct from the y-coordinate of any other point. In this case we may locate a on the horizontal through b anywhere to the left of it and assign it the direction **e**. This gives a two-directional point representation of P.

Actually we can say somewhat more, for ordered sets constructed as a "lexicographic sum." For an ordered set P and a family $(Q_p \mid p \in P)$ of ordered sets, indexed by P itself, the *lexicographic sum* $\sum_p Q_p$ is the ordered set whose underlying set is the union of the Q_p's and in which $x < y$ if $x, y \in Q_p$, for some $p \in P$, and $x < y$ in Q_p or, if $x \in Q_p$, $y \in Q_r$ and $p < r$ in P.

PROPOSITION. *Let $\sum_p Q_p$ be a lexicographic sum of ordered sets. If P as well as each Q_p has a two-directional point representation, and if each Q_p has a top and a bottom, then $\sum_p Q_p$ itself has a two-directional point representation.*

Proof. Suppose a two-directional point representation of P is given. Let $p \in P$ with coordinates (x, y), let p be directed northward, and suppose that p', with coordinates (x', y'), is the first vertex on this vertical northward path from p. If Q_p is a chain, then we may take a two-directional point representation of it in which each vertex is directed northward. Then if we contract the total vertical distance between the bottom vertex and the top vertex of Q_p to a total distance less than $y' - y$, we may insert this representation of Q_p into the vertical between p and p', replacing p by the bottom of Q_p and avoiding all y-coordinates already occupied by existing points.

Suppose that Q_p is not a chain. In this case we construct another two-directional point representation of P, by shifting each vertex r on the vertical along p by a small horizontal distance $\varepsilon > 0$ to the right less than the horizontal distance between p and any other vertex in its representation. We now contract the region occupied by the representation of Q_p into the ε by $y' - y$ rectangle from p to p', again replacing p by the bottom vertex of Q_p avoiding all y-coordinates already occupied.

In this way we may successively add the blocks to produce a two-directional point representation of the lexicographic sum itself.

It is not clear to us at this writing how we may naturally extend the class of ordered sets with a two-directional point representation. Lattices with at most two lower covers, even planar ones, need not have a two-directional point representation (e.g., the simple cycle $\{a < b < d, a < c < d\}$).

Elsewhere (cf. Czyzowicz, Pelc, and Rival) we have studied ordered sets, and especially lattices, with a diagram using only two different slopes for its edges. For instance, the 4-element cycle lattice can, of course, be drawn using only two slopes, yet it does not have a two-directional point representation. On the other hand, there are ordered sets (see Fig. 6) with no two-slope diagram (for nontrivial reasons) yet, which have a two-directional point representation (see Fig. 7). Still, there is an obvious connection between two-slope diagrams and point representations. If each vertex is allowed not just one of two directions, but both of the two directions, then it is easy to verify that there is a two-slope diagram. The converse, too, is obviously true.

Subdivision. Let P be an ordered set in which each element has at most two lower covers. Even if P itself has no two-directional point representation, we will show that there is an ordered set obtained from P by subdividing some edges of the diagram of P at most once that, in turn, has a two-directional point representation.

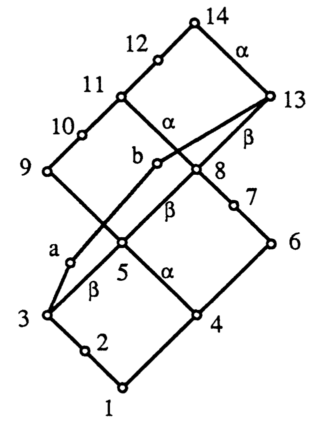


FIG. 6

Before we do this, let us record a rather simple transparent construction that, however, proves less. We show that there is an ordered set P' constructed from P by adjoining at most two subdivision points along every covering edge which itself has a two-directional point representation.

An example of this construction is illustrated in Fig. 8.

Let L be a linear extension of P and arrange the elements of P as points at unit intervals along the $y = x$ line on the plane in the same increasing order as they occur in L. We proceed by induction on the *height* of an element in L (that is, the size of the longest chain in L from it to the bottom of L) to assign it successive directions, changing at most twice, to produce a two-directional point representation. Suppose that the elements of L labelled $A_1, A_2, \cdots, A_{m-1}$ are already directed. Suppose that A_n is an upper cover of A_m. As A_n has at most two lower covers in P either the eastward direction to A_n is available or else the northward direction to A_n is available. Suppose then that the eastward direction is available and is chosen from a single subdivision point on the (A_m, A_n) edge

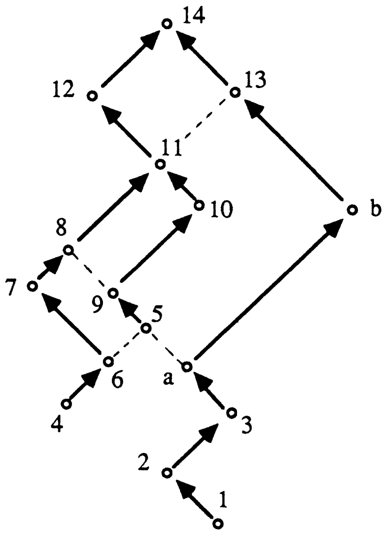
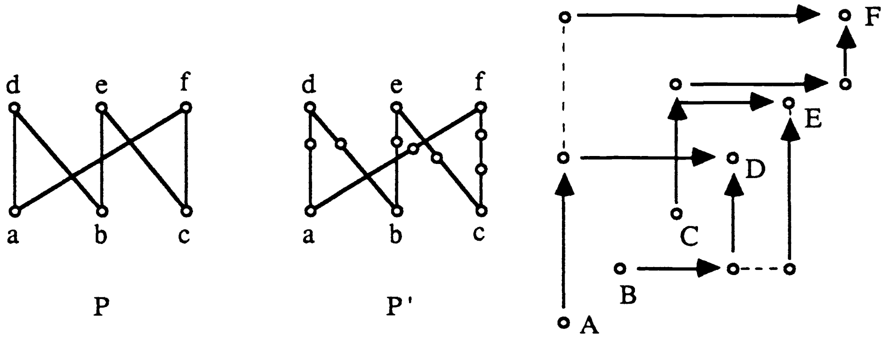


FIG. 7



A two-directional point representation of P'.

FIG. 8

(cf. Fig. 9). In fact, for any upper cover $A_{n'}$ of A_m for which this eastward direction is available we may choose a single subdivision point and direct A_m northward, as before, and direct the subdivision point eastward. Now, let C be an upper cover of A_m for which there already exists a point directed eastward toward it. In this case two subdivision points along the (A_m, C) edge suffice: the first located north of A_m at a point whose y-coordinate is distinct from the y-coordinate of any other point already constructed; the second located along the horizontal east from the first subdivision point and along the vertical below C. Then direct the first eastward and the second northward. The same construction can be carried out for any upper cover D of A_m whose incoming northward direction is available.

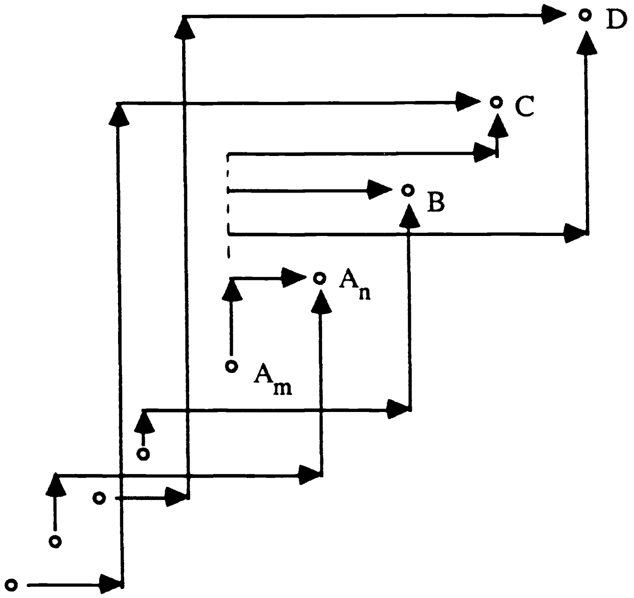


FIG. 9

We turn now to the proof of Theorem 2. We first treat the special case that every chain in P has at most two elements, that is, P has "height" at most two. Moreover, let us assume that P has a quite specific structure. Indeed, suppose that $P = P(G)$ is constructed from a graph G on the n vertices $v_1, v_2, \cdots, v_n$ with the minimal elements of P corresponding to these n vertices of G and the maximal elements of P corresponding precisely to those pairs $w_{ij} = (v_i, v_j)$ of vertices of G, joined by an edge in G. Then put $v_i < w_{ij}$ and $v_j < w_{ij}$. Evidently each element of $P(G)$ has at most two lower covers.

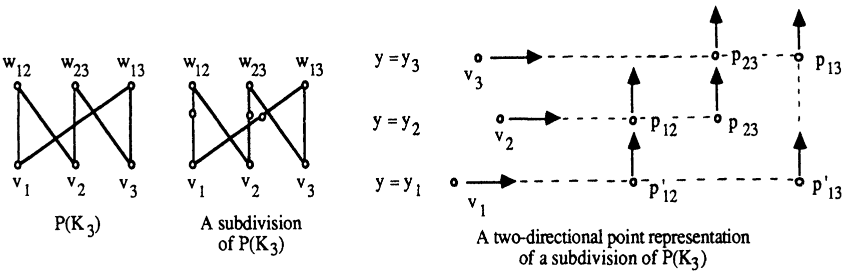
We will now make P even more particular. Let $P = P(K_n)$, where K_n stands for the complete graph on n vertices, that is, every pair of vertices is joined by an edge. We will show that there is an ordered set obtained from $P(K_n)$ by subdividing at most half of its edges that has a two-directional point representation. To begin, select locations $p_1, p_2, \cdots, p_n$ for the vertices $v_1, v_2, \cdots, v_n$ on n horizontal lines with equations $y = y_1, y = y_2, \cdots, y = y_n$, say p_i has coordinates (x_i, y_i), $i = 1, 2, \cdots, n$. We locate the vertices w_{ij} beyond (that is, to the right of) the vertical line $x = \max\{x_i | i = 1, 2, \cdots, n\}$. For each w_{ij} satisfying $i < j$, choose a location p_{ij} on $y = y_j$ with coordinates (x_{ij}, y_j) and define another location p'_{ij} on $y = y_i$ at (x_{ij}, y_i). We may suppose that all of these x-coordinates x_{ij} are distinct. Now, for each p_i assign it the horizontal direction to the right and, for each p_{ij} and p'_{ij} assign the vertical upward direction. The vertices p'_{ij} correspond to subdivisions of the corresponding edges from v_i to w_{ij} (see Fig. 10). In this way half of the edges of $P(K_n)$ are subdivided and this resulting subdivision has a two-directional point representation.

It is an easy consequence that, actually, for any graph G, some subdivision of the ordered set $P = P(G)$ also has a two-directional point representation. To see this, just erase the points p_{ij}, p'_{ij} from the representation of the above described subdivision of $P(K_n)$, n being the number of vertices of G, whenever v_i, v_j are not joined by an edge in G.

We may now extend this idea to supply a two-directional point representation of some subdivision of any ordered set P in which each maximum chain has at most two elements. Indeed, just like the case for $P(K_n)$, subdividing at most half of the edges is enough. Locate the minimals of P, each on a different horizontal line. For each maximal element with two lower covers we proceed as for the representation of $P(G)$. In fact, if all the maximals of P have two lower covers, then $P = P(G)$, where possibly G has some multiple edges (see Fig. 11).

If, on the other hand, there are maximals with just one lower cover, then it suffices to locate these on the horizontal line corresponding to its unique lower cover and direct it upward (see Fig. 12).

For this "bipartite" case, we have consistently directed the minimals horizontally and the maximals, together with all subdivision points, vertically. Of course, we could

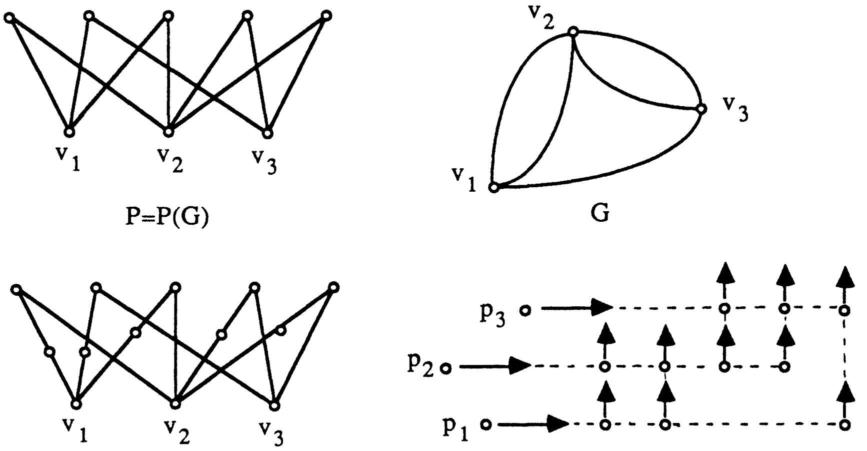


$P(K_3)$

A subdivision of $P(K_3)$

A two-directional point representation of a subdivision of $P(K_3)$

FIG. 10



P=P(G)

G

A subdivision of P

A two-directional point representation of a subdivision of P=P(G)

FIG. 11

have interchanged the two directions, with an appropriate change to all locations (a reflection along the diagonal $y = x$).

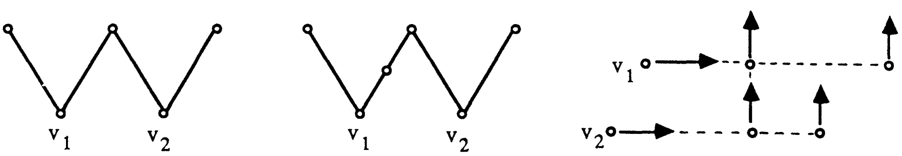
We are now ready to treat the general case. First, we partition P into "levels":

$$L_1 = \min(P);$$

$$\text{for } i > 1, \quad L_i = \min\left(P - \bigcup_{j<i} L_j\right),$$

where min (P) stands for the minimals of P. Note that consecutive pairs L_i, L_{i+1} determine bipartite orders, each of which does have a two-directional point representation. In fact, as long as there are no covering relations between pairs of elements $x \in L_i$, $y \in L_j$, $j \geqq i + 2$, then we may successively locate positions for the elements of the levels, alternating directions for the levels. Thus, if for $L_1 \cup L_2$ all vertices associated with L_1 are directed horizontally, then the vertices of L_2, as well as subdivision points, are directed vertically. At the next step in $L_2 \cup L_3$ each vertex in L_3 is directed horizontally just as the subdivision points in $L_2 \cup L_3$, and so on. Note that not all edges are subdivided; for instance, no edge associated with the lower cover of an element with only one lower cover is itself subdivided.

Let a two-directional point representation of a subdivision of the ordered set $(L_1 \cup L_2) \cup (L_3 \cup L_4) \cup \cdots$ be given. We suppose now that there are, however, covering edges joining elements in levels two or more apart. To this end let us suppose that $x \in$



P

A subdivision of P

A two-directional point representation of a subdivision of P

FIG. 12

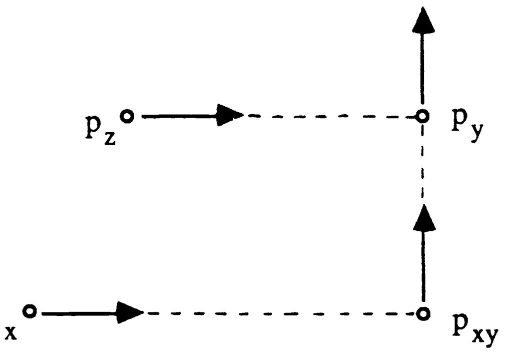


FIG. 13

L_i, i odd say, and $y \in L_j, j \geqq i+2$, are such elements. Let p_x, p_y stand for the corresponding points in the representation of $(L_1 \cup L_2) \cup (L_3 \cup L_4) \cup \cdots$. Then each coordinate of p_y is larger than the corresponding coordinate of p_x. By hypothesis, y can have precisely one other lower cover $z \neq x$ and $z \in L_{j-1}$, and, by construction, the covering edge y above z is not subdivided. Now, p_z is directed either horizontally or vertically. If horizontally, like p_x, then we insert a point p_{xy}, directed upward, at the intersection of the horizontal through p_x and the vertical through p_y (see Fig. 13). Now, if there is a point p_c already located on the vertical between p_{xy} and p_y, it cannot be comparable to p_y. As no point in the representation of the subdivision can be directed upward to p_c, we may shift p_c slightly to the right. This results in a representation of the subdivision again, along with the required comparability of $x < y$ using a single subdivision. Otherwise, p_z is directed upward. As z is not itself a subdivision point, p_y must be directed horizontally. Then move p_y slightly to the right, say a distance $\varepsilon > 0$. Insert a point p_{zy}, directed horizontally, at the intersection of the vertical through p_z and the horizontal through p_y, that is, at the former location of p_y itself. Also insert a point p_{xy}, directed vertically at the intersection of the horizontal through p_x and the vertical through p_y, (now shifted a distance $\varepsilon > 0$ horizontally). We may suppose that no other points lie on the segment between p_{xy} and p_y (otherwise, shift it horizontally by a small distance) (see Fig. 14). In this fashion we can produce a two-directional point representation of a subdivision of P. This completes the proof of Theorem 2.

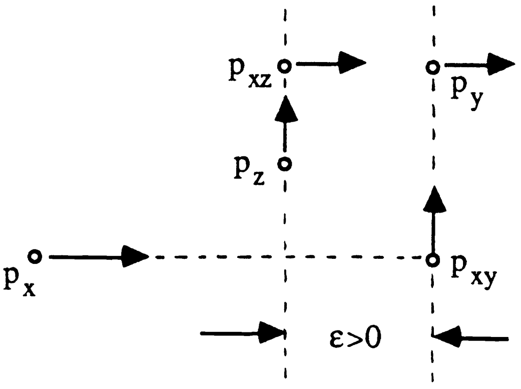


FIG. 14

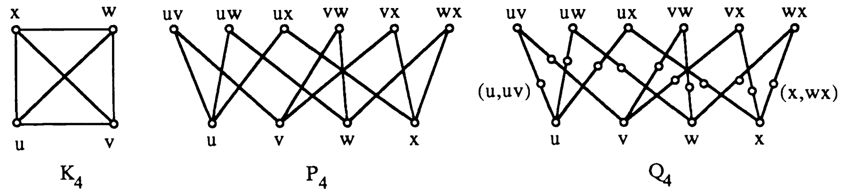
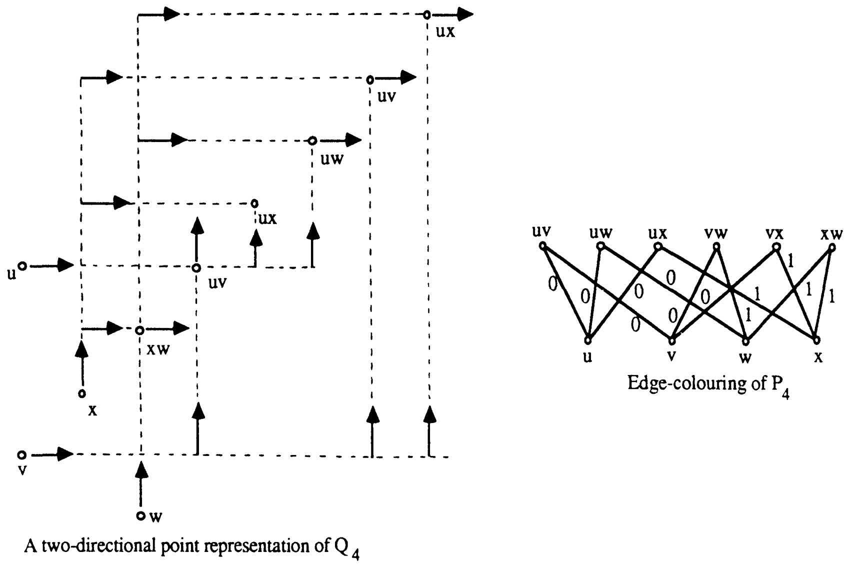


FIG. 15

We now turn to the proof of Theorem 3 and first prove the second part. We construct a family $(P_n | n \geqq 7)$ of ordered sets, each member of which has no two-directional point representation. We will also prove that if the diagram of P_n is subdivided by adding exactly one point along each covering edge, the ordered set Q_n thus obtained has no two-directional point representation. Indeed, let $P_n = P(K_n)$ (see Fig. 15). Of course, this bipartite order P_n itself has no two-directional point representation, for any $n \geqq 3$, as it certainly contains a simple cycle. We will show that adding exactly one subdivision point along each edge of P_n, $n \geqq 7$, cannot produce an ordered set with such a representation. Suppose, for a contradiction, that every ordered set Q_n, $n \geqq 7$, obtained from P_n by adding precisely one subdivision point (u, uv) along every covering edge u to uv, does have a two-directional point representation.

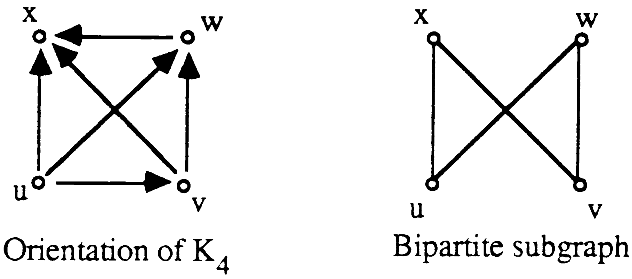
Our aim is to construct a particular two-colouring of the edges of P_n based on the representation of Q_n. Let u be an arbitrary vertex of K_n. Note that, in the representation of Q_n, all but at most one of the upper covers of u have a direction different from that assigned to u. Colour the edge from u to uv 1 if (u, uv) is directed eastward, otherwise colour the edge 0. Note that the two incident edges of each maximal vertex uv of P_n carry distinct colours (see Fig. 16). On the other hand, among the n incident edges of each minimal vertex u all, but at most one, receive the same colour. Now orient the edges of K_n according to this rule: $u \rightarrow v$ if the edge u to uv in P_n has colour 1; $v \rightarrow u$ if this edge u to uv in P_n has colour 0 (see Fig. 17). Then for any vertex u of K_n, either all



Edge-colouring of P_4

A two-directional point representation of Q_4

FIG. 16



Orientation of K_4

Bipartite subgraph

FIG. 17

but at most one of the edges are directed away from u or, all but at most one of the edges are directed into u. For each vertex u delete from K_n, $n \geqq 4$, the minority edge, if it exists. Then, for every vertex u of the induced subgraph, either every edge is directed away from u or every edge is directed into u, that is, the induced subgraph must be bipartite. In summary, we have shown that the removal of at most n edges from K_n produces a bipartite graph. If $n \geqq 7$ then one of the two parts of the bipartition contains at least four vertices whose six edges must have been removed, according to the construction. This is impossible if only n edges are removed in all, each one incident to a distinct vertex.

To prove the first part of Theorem 3, we will show that if P is a subdivision of $P(K_n)$ which has a two-directional representation, then there are at most $n - 1$ vertices $w_{i,j}$ such that neither of the two original edges v_i to $w_{i,j}$ or v_j to $w_{i,j}$ is subdivided in P. For contradiction, suppose that there are n such vertices $w_{i,j}$. These n vertices together with the $2n$ incident edges and all n minimal vertices of $P(K_n)$ form a bipartite order on $2n$ vertices with $2n$ edges. Such an order must contain a cycle, which contradicts Theorem 1.

REFERENCES

J. CZYZOWICZ, A. PELC, AND I. RIVAL, *Drawing orders with few slopes*, Discrete Math., 82 (1990), pp. 233–250.

R. J. NOWAKOWSKI, I. RIVAL, AND J. URRUTIA, *Representing orders on the plane by translating points and lines*, Discrete Math., to appear.

I. RIVAL AND J. URRUTIA, *Representing orders by translating convex figures in the plane*, Order, 4 (1987), pp. 319–339.

SIAM J. DISC. MATH.
Vol. 4, No. 2, pp. 164-171, May 1991

002

ON THE COMPLETE WEIGHT ENUMERATOR OF REED-SOLOMON CODES*

IAN F. BLAKE† AND KHUN KITH†

Abstract. The complete weight enumerator of a code enumerates the code words by the number of symbols of each kind contained in each code word. As for the ordinary weight enumerators, the complete weight enumerators for linear codes satisfy a duality theorem. These weight enumerators are studied here for certain realizations of Reed–Solomon codes of dimensions two, three, and four over a field of characteristic two. Some applications of these results are considered.

Key words. error correcting codes, complete weight enumerators

AMS(MOS) subject classification. 94B05

1. Introduction. The complete weight enumerator (cwe) of a code enumerates the code words by the number of symbols of each kind contained in each code word. As for ordinary weight enumerators for linear codes, they satisfy a duality theorem. One application for such a cwe is in determining the weight and distance structure of certain kinds of concatenated codes. As an example it is sometimes of interest to obtain a binary code from a q-ary code, $q = 2^m$, by replacing each field symbol with its binary representation in some basis of F_q over F_2, where F_q is the finite field of q elements. The binary weight enumerators of such binary image codes, obtained from certain representations of Reed–Solomon codes, are considered by Kasami and Lin [2] and some of the references contained therein. Such results can also be obtained from the more general approach using cwe's considered here. The problem addressed here is noted as research problem 11.2 in [4].

The next section reviews some results on Reed–Solomon codes and cwe's that will be of use in the sequel. The following three sections derive the cwe of particular realizations of a Reed–Solomon code of dimensions two, three, and four, respectively. By the duality theorem for cwe's, these can be also obtained for dimensions $n - 2$, $n - 3$, and $n - 4$. Only Reed–Solomon codes over fields of characteristic two are considered in this work, although many of the results hold for the more general case [3]. Section 6 considers applications of the results derived.

2. Preliminaries. Throughout, let $q = 2^m$ and $n = q - 1$, and let α be a primitive element of F_q. Denote by $RS_b(n, k)$ the cyclic Reed–Solomon code of length n and dimension k with generator polynomial

$$g_b(x) = (x - \alpha^b)(x - \alpha^{b+1}) \cdots (x - \alpha^{b+n-k-1}).$$

The dual of $RS_b(n, k)$, $RS_b^{\perp}(n, k)$, is $RS_{n-b+1}(n, n - k)$. Let $ERS_b(q, k)$ be the extended code obtained from $RS_b(n, k)$ by adding an overall parity check. Then $ERS_b^{\perp}(q, k) = ERS_{n-b+1}(q, q - k)$. In like manner, the generator and parity check matrices for these codes may be described. Let

$$\theta_i = [1, \alpha^i, \alpha^{2i}, \cdots, \alpha^{i(q-2)}].$$

* Received by the editors March 19, 1990; accepted for publication (in revised form) May 8, 1990.

† Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1. This work was supported by Natural Sciences and Engineering Research Council of Canada grant 7382.

Then the parity check matrix for $RS_b(n, k)$, H_b, has as rows the vectors $\theta_b, \theta_{b+1}, \cdots, \theta_{n-k+b-1}$ and the generator matrix G_b has as rows the vectors $\theta_{n-b+1}, \cdots, \theta_{n-1}, \theta_0, \cdots, \theta_{k-b}$.

Using a notation similar to that of Kasami and Lin [2], let

$$V(f(x)) = (f(1), f(\alpha), \cdots, f(\alpha^{q-2})) \in F_q^n$$

for $f(x)$ a polynomial over F_q and

$$V_e(f(x)) = (f(0), f(1), \cdots, f(\alpha^{q-2})) \in F_q^q.$$

It is readily established that

$$RS_b(n,k) = \{V(x^{n-b+1} f(x)), \text{degree} f < k\}$$

and in particular that

$$RS_1(n,k) = \{V(f(x)), \text{degree} f < k\}, \text{ and } ERS_1(q,k) = \{V_e(f(x)), \text{degree} f < k\}.$$

That the extended code is of this form follows immediately from the fact that

$$\sum_{i=0}^{q-2} \alpha^{ik} = \frac{1-\alpha^{(q-1)k}}{1-\alpha^k} = 0.$$

Reed–Solomon codes are examples of maximum distance separable (MDS) codes for which the minimum distance is $d = n - k + 1$. The (ordinary) weight enumerator for such codes is uniquely determined by the code parameters and is given by

$$A_l = (q-1)\binom{n}{l} \sum_{i=0}^{l-d} (-1)^i \binom{l-1}{i} q^{l-d-i}, l \geq d.$$

The cwe of a code enumerates code words according to the number of times each element of the field appears in each code word. It will be convenient to use cwe for both the code and individual code words. Denote the field elements by $F_q = \{\alpha^j, j \in B\}$, where $B = \{*, 0, 1, \cdots, q-2\}$ and by convention $\alpha^* = 0$. Also let F_q^* denote $F_q \backslash \{0\}$ and $B^* = B \backslash \{*\}$. For $u = (u_1, \cdots, u_n) \in F_q^n$, let $w[u]$ be the cwe of u defined as

$$w[u] = z_*^{s_*} z_0^{s_0} \cdots z_{q-2}^{q-2} = \prod_{j \in B} z_j^{s_j},$$

where s_j is the number of components of u equal to α^j, $\sum_B s_j = n$. The cwe of a code $\mathscr{C}$ is then

$$W_{\mathscr{C}}(z_*, z_0, z_1, \cdots, z_{q-2}) = W_{\mathscr{C}}(\underline{z}) = \sum_{u \in \mathscr{C}} w[u].$$

Let χ_1 denote the character of F_q given by $\chi_1(\beta) = (-1)^{\beta_0}$, where β_0 is the zeroth component of $\beta \in F_q$ with respect to the polynomial basis generated by α. Then the MacWilliams theorem for the cwe for a linear code [4] is

$$W_{\mathscr{C}^\perp}(z_*, z_0, \cdots, z_{q-2}) = \frac{1}{|\mathscr{C}|} W_{\mathscr{C}}\left(\sum_{s \in B} \chi_1(\alpha^* \alpha^s) z_s, \cdots, \sum_{s \in B} \chi_1(\alpha^{q-2} \alpha^s) z_s\right).$$

Unlike the ordinary weight enumerator, the cwe for Reed–Solomon codes depends on the particular generator polynomial used. Some useful properties of code word cwe's follow from their definition. If $w[u] = \prod_{j \in A} z_j^{s_j}$, $A \subset B$ then the cwe of the scalar multiple of u, $\alpha^k u$, is obtained as

$$(1) \qquad w[\alpha^k u] = \prod_{j \in \{k+A\}} z_j^{s_j},$$

where the addition in $\{k+A\}$ is modulo $q-1$ and $k + * = *$. For convenience, denote $w[\alpha^k u]$ as $w[u]^{(k)}$. Note also that $V(f(\alpha^h x))$ is a cyclic shift by h positions of $V(f(x))$ and so

$$w[V(f(\alpha^h x))] = w[V(f(x))].$$

Let $T(\beta)$ denote the trace of $\beta \in F_q = F_{2^m}$, $T(\beta) = \sum_{i=0}^{m-1} \beta^{2^i}$, of F_q over F_2, and let $B_0 = \{i, T(\alpha^i) = 0\}$ and $\bar{B}_0 = B \backslash B_0$, where $|B_0| = 2^{m-1}$. Recall that B_0 is a union of cyclotomic cosets and that $T(x+y) = T(x) + T(y)$. Note also that the quadratic $x^2 + a_1 x + a_0 = 0$ has solutions over F_q if and only if $T(a_0/a_1^2) = 0$ and that the cubic $x^3 + a_2 x^2 + a_1 x + a_0 = 0$ has a unique solution if and only if $T(\{(a_0 + a_1 a_2)/(a_1 + a_2^2)^{3/2}\}^{-1}) \neq T(1)$ [1].

3. The complete weight enumerator for $RS_1(n, 2)$ and $ERS_1(q, 2)$. The ordinary weight enumerator for $RS_1(n, 2)$ is

$$A_0 = 1, \quad A_{n-1} = (q-1)^2, \quad A_n = 2(q-1)$$

and for $ERS_1(q, 2)$ is

$$A_0 = 1, \quad A_{q-1} = q(q-1), \quad A_q = (q-1).$$

The cwe's of the code words of various weights are easily arrived at using the polynomial description of $RS_1(n, 2)$. For fixed a_0, $a_1 \in F_q$ consider the values of the polynomial $a_0 + a_1 x$ as x runs through F_q^*. If $a_1 = 0$ the corresponding code word contains a_0 in each coordinate position, and the total contribution to the cwe is

$$\sum_{j \in B} z_j^{q-1};$$

this is the cwe of $RS_1(n, 1)$. If $a_1 \neq 0$ then as x runs through F_q^*, $a_0 + a_1 x$ runs through all values of F_q except a_0. If $a_0 = \alpha^i$ then the contributions of such terms to the cwe are of the form

$$\prod_{j \in B, j \neq i} z_j = \gamma / z_i, \qquad \gamma \triangleq \prod_{j \in B} z_j$$

and each such term appears $(q-1)$ times corresponding to the nonzero value of a_1. The cwe of $RS_1(n, 2)$ is then

$$W_{RS_1(n,2)}(\underline{z}) = \sum_{j \in B} z_j^{q-1} + (q-1)\gamma \sum_{i \in B} \frac{1}{z_i}.$$

The argument for $ERS_1(q, 2)$ is simpler. For $a_1 = 0$ the q code words corresponding to the values of $a_0 \in F_q$ are constant and contribute the terms

$$\sum_{i \in B} z_i^q$$

to the cwe. For $a_1 \neq 0$, as x runs through F_q the polynomial $a_0 + a_1 x$ runs through F_q. The $q(q-1)$ terms of the cwe are then $q(q-1)\gamma$, and the cwe for $ERS_1(q, 2)$ is

$$W_{ERS_1(q,2)}(\underline{z}) = \sum_{i \in B} z_i^q + q(q-1)\gamma;$$

thus the first term is $W_{ERS_1(q,1)}(\underline{z})$.

4. The complete weight enumerator of $RS_1(n, 3)$ and $ERS_1(q, 3)$. The ordinary weight enumerator for $RS_1(n, 3)$ is

$$A_0 = 1, \quad A_{n-2} = (q-1)\binom{q-1}{2}, \quad A_{n-1} = 3(q-1)^2, \quad A_n = q(q-1)^2/2 + 3(q-1)$$

and for $ERS_1(q, 3)$

$$A_0 = 1, \quad A_{q-2} = (q-1)\binom{q}{q-2}, \quad A_{q-1} = 2q(q-1), \quad A_q = (q-1)(q^2-q+2)/2.$$

It is clear that $RS_1(n, 3)$ contains $RS_1(n, 2)$, and it is sufficient to consider code words corresponding to polynomials of the form $a_2x^2 + a_1x + a_0$, $a_2 \neq 0$. It is possible to identify the added code words (from $RS_1(n, 2)$ to $RS_1(n, 3)$) in terms of their properties. For example, the $(q-1)\binom{q}{q-2}$ words of weight $q-2$ correspond to the polynomials of the form $a(x + \alpha_1)(x + \alpha_2)$, $\alpha_1 \neq \alpha_2$. However, such observations do not appear to be useful in determining the corresponding code word cwe's.

From (1) it is sufficient to consider the case of monic polynomials ($a_2 = 1$) and consider first the case of $a_1 \neq 0$. In this case, transform the variable by $x = a_1 y$ so that $x^2 + a_1 x + a_0$ becomes $a_1^2(y^2 + y + (a_0/a_1^2))$ and it is sufficient to consider the polynomials of the form $y^2 + y + a_0$. Scalar multiples will be considered later. To determine the number of times the symbol η will appear in the code word corresponding to the polynomial $y^2 + y + a_0$, we require the number of solutions to $y^2 + y + a_0 = \eta$. This will have two solutions in F_q if and only if $T(a_0) = T(\eta)$ and, if $a_0 = \eta$, the solutions are 0 and 1. Thus if we define

$$\beta_0 = \prod_{j \in B_0} z_j^2 \text{ and } \bar{\beta}_0 = \prod_{j \in \bar{B}_0} z_j^2$$

then

$$w[V(y^2 + y + \alpha^i)] = \begin{cases} \beta_0 / z_i \, i \in B_0 \\ \bar{\beta}_0 / z_i \, i \in \bar{B}_0 \end{cases}.$$

To consider scalar multiples of the code words corresponding to these polynomials, it is convenient to define translations of the sets B_0 and $\bar{B}_0$ as $B_k = \{k + B_0\}$ and $\bar{B}_k = \{k + \bar{B}_0\}$, where addition is modulo $q-1$ and $j + * = *$. Further define

$$\beta_k = \prod_{j \in B_k} z_j^2 \quad \text{and} \quad \bar{\beta}_k = \prod_{j \in \bar{B}_k} z_j^2, \qquad k \in B.$$

The contributions to the cwe of the scalar multiples are then

$$w[\alpha^k V(y^2 + y + \alpha^i)] = \begin{cases} \beta_k / z_{i+k}, & i \in B_0 \\ \bar{\beta}_k / z_{i+k}, & i \in \bar{B}_0 \end{cases}.$$

Each such contribution to the cwe appears $(q-1)$ times and hence of $q(q-1)^2$ polynomials under consideration their contribution to the cwe is

$$(q-1)\sum_{k\in B^*}\left\{\sum_{i\in B_0}\frac{\beta_k}{z_{i+k}}+\sum_{i\in \bar{B}_0}\frac{\bar{\beta}_0}{\bar{z}_{i+k}}\right\}.$$

For the case where $a_1 = 0$ it is clear that it is sufficient to consider polynomials of the form $a_2(x+a_0)^2$. As x runs through F_q^* the polynomial assumes all values except $a_2a_0^2$ and again, since squaring in fields of characteristic two is an automorphism, the contribution to the cwe of these $q(q-1)$ polynomials is

$$(q-1)\gamma\left\{\sum_{j\in B}\frac{1}{z_j}\right\}.$$

The cwe of $RS_1(q, 3)$ is then

$$W_{RS_1(n,3)}(\underline{z}) = W_{RS_1(n,2)}(\underline{z}) + (q-1)\left\{\sum_{k\in B^*}\left[\sum_{i\in B_0}\frac{\beta_k}{z_{i+k}}+\sum_{i\in\bar{B}_0}\frac{\bar{\beta}_k}{z_{i+k}}\right]+\gamma\left[\sum_{j\in B}\frac{1}{z_j}\right]\right\}.$$

For $ERS_1(q, 3)$ the same arguments are used, and the expressions simplify when the summations are allowed to run over F_q rather than F_q^*. The resulting cwe is

$$\text{(2)}\qquad W_{ERS_1(q,3)}(\underline{z}) = W_{ERS_1(q,2)}(\underline{z}) + ((q-1)q/2)\left\{\sum_{k\in B^*}(\beta_k+\bar{\beta}_k)\right\}+q(q-1)\gamma.$$

5. The complete weight enumerator for $ERS_1(q, 4)$. The complete weight enumerator for the four-dimensional case is a little more involved and, for the sake of simplicity, is found here for the extended code only. Some computation in F_q is required to resolve this case. The ordinary weight enumerator for $ERS_1(n, 4)$ is given by

$$A_{q-3} = (q-1)\binom{q}{q-3},\quad A_{q-2} = 3(q-1)\binom{q}{q-2},\quad A_{q-1} = q(q-1)(q^2-q+6)/2,$$

$$A_{q-4} = q(q-1)\left[q^3-\binom{q-1}{1}q^2+\binom{q-1}{2}q-\binom{q-1}{3}\right],$$

and again it is an easy matter to identify the contributions to the various weight classes of the various types of polynomials. The cwe's of code words corresponding to monic polynomials $x^3+a_2x^2+a_1x+a_0$ of degree three are first found and four subcases are considered. Consider first the subcase where $a_2 = 0$ and $a_1 \neq 0$. By making the substitution $x = a_1^{1/2}z$, this is equivalent, up to scalar multiples, to determining the cwe of z^3+z+a_0. Denote by $u_\eta(\underline{z})$ the cwe of $V(z^3+z+\eta)$ and although some observations on the form of these cwe's may be made, it appears they must largely be determined by computation. For example, it will be seen that only one such cwe from each conjugacy class need be determined. Also note that the total contribution of such polynomials to the cwe is

$$(q-1)\sum_{\eta\in F_q}u_\eta(\underline{z}).$$

For the case $a_2 = a_1 = 0$ the polynomials are of the form x^3+a_0 and let

$$v_\lambda(\underline{z}) = W[V_e(x^3+\lambda)].$$

If m is even then $3 | 2^m - 1$ and let $I = \{3i, i = 0, 1, \cdots, (2^m - 4)/3\}$. Then

$$v_0(\underline{z}) = z_* \prod_{i \in I} z_i^3$$

and if m is odd

$$v_0(\underline{z}) = \gamma = v_\lambda(\underline{z}).$$

The contribution to the cwe of terms of this form is

$$\sum_{\lambda \in F_q} v_\lambda(\underline{z}).$$

For the case when $a_2 \neq 0$, by the substitution $x = a_2 y$, $x^3 + a_2 x^2 + a_1 x + a_0$ can be expressed as a scalar multiple of a polynomial of the form $y^3 + y^2 + a_1 y + a_0$. Thus in the case when $a_2 \neq 0$ and $a_1 \neq 1$, the further change of variable $y = z(a_1 + 1)^{1/2} + 1$ is made to give the polynomial, up to scalar multiple, $z^3 + z + a$, $a = (a_0 + a_1)/(a_1 + 1)^{3/2}$. It is easily checked that as (a_0, a_1) runs through the $q(q - 1)$ allowable values, the coefficient a runs through each value of F_q, $q - 1$ times. The contribution of these $q(q - 1)^2$ polynomials is then

$$(q-1)^2 \sum_{\eta \in F_q} u_\eta(\underline{z}).$$

Finally, for the case where $a_2 \neq 0$ and $a_1 = 1$, the polynomial $y^3 + y^2 + y + a_0 = (y + 1)^3 + (a_0 + 1)$, which, by transformation of the variable $z = y + 1$, gives $z^3 + \lambda$. The contribution of the $q(q - 1)$ such polynomials to the cwe is of the form

$$(q-1) \sum_{\lambda \in F_q} v_\lambda(\underline{z}).$$

The contribution of the four cases is then

$$r(\underline{z}) = q(q-1) \sum_{\eta \in F_q} u_\eta(\underline{z}) + q \sum_{\lambda \in F_q} v_\lambda(\underline{z}).$$

Finally, the set of all scalar multiples of the polynomials considered is included as

$$(3) \qquad \sum_{j \in B^*} [r(\underline{z})]^{(j)} = q(q-1) \sum_{\eta \in F_q} \sum_{j \in B^*} [u_\eta(\underline{z})]^{(j)} + q \sum_{\lambda \in F_q} \sum_{j \in B^*} [v_\lambda(\underline{z})]^{(j)}$$

and the cwe for $ERS_1(q, 4)$ is then

$$(4) \qquad W_{ERS_1(q,4)}(\underline{z}) = W_{ERS_1(q,3)}(\underline{z}) + \sum_{j \in B^*} [r(\underline{z})]^{(j)}.$$

A property of the cwe of a code word that will be computationally useful is as follows. If $w[u] = \prod_{j \in A} z_j^{s_j}$, denote by

$$w[u]_{(k)} = \prod_{j \in A} z_{2^k j}^{s_j},$$

where again arithmetic is modulo $q - 1$, $2^k * = *$ and $2^k 0 = 0$. Let $f(x) = \sum_{j=0}^r a_j x^j$ and $f_i(x) = \sum_{j-0}^r a_j^{2^i} x^j$. Then $w[V(f_i(x))] = w[V(f(x))]_{(i)}$. This follows since if $f(\alpha^l) = \alpha^j$ then $f_i(\alpha^{2^i l}) = \alpha^{2^i j}$. In particular, suppose coefficients a_j of $f(x)$ are in F_2, $j = 1, 2, \cdots, r$ and $a_0 \in F_{2^m}$. Then the cwe's of polynomials obtained by allowing a_0 to range over its conjugacy class $\{a_0^{2^i}\}$, $i = 1, 2, \cdots, s$ are easily obtained as

$w[V(f(x))]_{(i)}$, $i = 1, 2, \cdots, s$. Thus, as mentioned, for polynomials of this form only one cwe from each conjugacy class need be determined.

6. Applications. Two applications of the cwe are discussed in this section and the cwe of $ERS_1(8, 4)$ found as a small example of the techniques of the previous section.

The problem discussed in [2] is to determine the binary weight enumerator of the code obtained from the Reed–Solomon code by replacing each element of F_{2^m} by its binary m-tuple with respect to some basis. Similar results can be obtained by the cwe approach. Let element α^i have a binary weight w_i with respect to the chosen basis. If $\mathscr{C}$ is a code over F_q with cwe $W_{\mathscr{C}}$ and $\mathscr{C}_b$ is the binary code obtained in the above manner, then the (binary) weight enumerator of $\mathscr{C}_b$ is given by

$$W_{\mathscr{C}_b}(x) = W_{\mathscr{C}}(\underline{z})\,|_{\{z_i = x^{w_i}, i \in B\}}.$$

The results of [2], using results more tuned to that specific problem, give more detailed information on the weight structure of $\mathscr{C}_b$. With further analysis the cwe approach would yield the same results.

As a second application of cwe's, the problem of using Reed–Solomon codes with M-ary phase shift keying is considered, where $M = q$. For convenience we use complex notation and associate the symbol $0 \in F_q$ to $1 \in \mathscr{C}$ and map the symbol α^i to the phase ζ^{i+1}, $i = 0, 1, \cdots, q-2$, $\zeta = e^{2\pi i/q}$. There are, of course, other ways of mapping the symbols. The theoretical performance of such a system depends on the set of Euclidean distances between code words, although to achieve this performance would require a so-called soft decision decoding, which is at present unknown for such codes. The problem is nonetheless of interest and may be approached using the cwe. If $W_E(x)$ is the distance enumerator of the Euclidean code obtained from $\mathscr{C}$ by the above mapping, whose coefficient of x^{d^2} gives the number of code words at Euclidean distance squared d^2 from any given code word, then

$$W_E(x) = W_{\mathscr{C}}(\underline{z})\,|_{\{z_i = x^{|\zeta(i+1)-1|^2}\}}.$$

To illustrate the techniques of the previous sections, the cwe of $ERS_1(8, 4)$ is determined. Note first that since $3 | q - 1 = 7$ then $v_\lambda(\underline{z})^{(j)} = \gamma$, for all $j \in B$. It is easily verified that $B_0 = \{*, 1, 2, 4\}$ and $\bar{B}_0 = \{0, 3, 5, 6\}$ and

$$\beta_k = \prod_{j \in \{k + B_0\}} z_j \qquad \bar{\beta}_k = \prod_{\{j \in \bar{B}_0\}} z_j.$$

Equations (2), (3), and (4) give

$$W_{ERS_1(8,4)}(\underline{z}) = \sum_{i \in B} z_i^8 + 504\gamma + 28\left\{\sum_{k \in B} (\beta_k + \bar{\beta}_k)\right\}$$
$$+ 56 \sum_{j \in B^*} \sum_{\eta \in F_q} u_\eta(\underline{z})^{(j)}.$$

The cwe's $u_\eta(\underline{z})$ are shown in Table 1 and the $u_\eta(\underline{z})^{(j)}$ are easily computed from these. For example,

$$u_{\alpha^5}(\underline{z}) = z_* z_1 z_2 z_4^3 z_5^2 \text{ and } u_{\alpha^5}(\underline{z})^{(2)} = z_* z_0^2 z_3 z_4 z_6^3.$$

Thus all q^4 terms of the cwe of $ERS_1(8, 4)$ are readily obtained. Note that

$$u_{\alpha^2}(\underline{z}) = u_\alpha(\underline{z})_{(2)} \text{ and } u_{\alpha^{2^2}}(\underline{z})\, u_\alpha(\underline{z}))_{(2^2)}$$

TABLE 1
Complete weight enumerators $u_\eta(z)$.

$j, \eta = \alpha^j$	$u_\eta(z) = w[V_e(x^3 + x + \eta)]$
*	$z_*^2 z_0^3 z_3 z_5 z_6$
0	$z_*^3 z_0^2 z_1 z_2 z_4$
1	$z_0 z_1^2 z_3^3 z_5 z_6$
2	$z_0 z_2^2 z_3 z_5 z_6^3$
3	$z_* z_1^3 z_2 z_3^2 z_4$
4	$z_0 z_3 z_4^2 z_5^3 z_6$
5	$z_* z_1 z_2 z_4^3 z_5^2$
6	$z_* z_1 z_2^3 z_4 z_6^2$

and

$$u_{\alpha^6}(\underline{z}) = u_{\alpha^3}(\underline{z})_{(2)} \text{ and } u_{\alpha^{5^2}}(\underline{z}) = u_{\alpha^3}(\underline{z})_{(2^2)};$$

i.e., it is only necessary to compute one cwe from each conjugacy class of F_q. This observation is particularly useful for larger fields.

7. Comments. The cwe's of certain low dimensional Reed–Solomon codes have been determined and, by duality, the corresponding high dimensional codes have also been determined. It was pointed out in the previous section how such cwe's might be useful in certain applications and an example of the computation of the cwe of $ERS_1(8, 4)$ was given. For the four dimensional codes the arguments considered several cases as the polynomial coefficients assumed certain values, and it seems unlikely that similar arguments for higher dimensions will be of interest to pursue. Yet the cwe's of the Reed–Solomon codes show considerable structure, which raises the question as to whether another approach might be more successful. Clearly, the cwe possesses symmetries that might be exploited to obtain more information on their structure. It is possible that the cwe's of doubly and triply extended codes might be obtained with the techniques described, but this was not pursued.

REFERENCES

[1] E. R. BERLEKAMP, H. RUMSEY, AND G. SOLOMON, *On the solution of algebraic equations over finite fields*, Inform. and Control, 10 (1967), pp. 553-564.

[2] T. KASAMI AND S. LIN, *The binary weight distribution of the extended* $(2^m, 2^m - 4)$ *code of the Reed–Solomon code over* $\mathrm{GF}(2^m)$ *with generator polynomial* $(x - \alpha)(x - \alpha^2)(x - \alpha^3)$, Linear Algebra Appl., 98 (1988), pp. 291-307.

[3] KHUN KITH, *Complete weight enumeration of Reed-Solomon codes*, Master's thesis, Department of Electrical and Computing Engineering, University of Waterloo, Waterloo, Ontario, Canada, 1989.

[4] F. J. MACWILLIAMS AND N. J. A. SLOANE, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1977.

SIAM J. DISC. MATH.
Vol. 4, No. 2, pp. 172-181, May 1991

003

NEW RESULTS ON SERVER PROBLEMS*

M. CHROBAK†, H. KARLOFF‡§, T. PAYNE†, AND S. VISHWANATHAN‡

Abstract. In the *k-server problem*, one must choose how k mobile servers will serve each of a sequence of requests, making decisions in an online manner. An optimal deterministic online strategy is exhibited when the requests fall on the real line. For the *weighted-cache problem*, in which the cost of moving to x from any other point is $w(x)$, the weight of x, an optimal deterministic algorithm is also provided. The nonexistence of competitive algorithms for the asymmetric two-server problem and of memoryless algorithms for the weighted-cache problem is proved. A fast algorithm for offline computing of an optimal schedule is given, and it is shown that finding an optimal offline schedule is at least as hard as the assignment problem.

Key words. online algorithm, offline algorithm, server problem, competitive analysis

AMS(MOS) subject classification. 68Q20

1. Introduction. The k-server problem can be stated as follows. We are given a metric space M, and k *servers* that move among the points of M, each occupying one point of M. Repeatedly, a *request* (a point $x \in M$) appears. To *serve* x, each server moves some distance, possibly zero, after which the point x must be occupied by one of our servers. The *cost* incurred is the sum of the k distances moved. We must serve this request by considering only the current and past requests: the decisions are made *online*. The server problem encompasses many interesting problems as special cases, for example: heuristics for linear search [14], paging [9], [13], font caching in printers [13], and motion planning for 2-headed disks [4].

Let us call a sequence of k not necessarily distinct points of a metric space a *configuration*. If M is a metric space, we call an online strategy $\mathcal{S}$ *c_k-competitive for M* if for every initial configuration $R = (r_1, r_2, \cdots, r_k)$, there is a real a such that the following property holds. Let σ be an arbitrary request sequence for M and let $\mathrm{OPT}_R(\sigma)$ be the optimal (offline) cost to serve σ, when initially the ith server occupies r_i. Then the total cost incurred by $\mathcal{S}$ on σ, when its ith server starts in r_i, is at most

$$c_k \mathrm{OPT}_R(\sigma) + a.$$

Here, the benchmark is the minimum cost needed to serve the request sequence, minimized over all possible ways of serving this sequence. Thus for $\mathcal{S}$ to be c_k-competitive, its cost must not exceed a more than c_k times the cost of the optimal offline algorithm. We say $\mathcal{S}$ is *competitive for M* if it is c_k-competitive for M for some c_k. Strategy $\mathcal{S}$ is *c_k-competitive* or *competitive* if the respective definitions hold for all metric spaces. It is known [10] that no c_k-competitive strategy exists if $c_k < k$. Also, no generality is lost in assuming, if desired, that the initial locations r_i are distinct.

Yet a more general model, that of *task systems*, was studied by Borodin, Linial, and Saks [2], who gave an optimal online algorithm in their model. However, in their approach the "competitive ratio" is allowed to depend on the cardinality of the metric space. Our problems are less general but sometimes permit stronger results.

At the moment, for no $k \geqq 3$ is any competitive (deterministic) algorithm known for all metric spaces. Several important results were obtained by Manasse, McGeoch,

* Received by the editors October 17, 1989; accepted for publication (in revised form) July 12, 1990.

† Department of Mathematics and Computer Science, University of California, Riverside, California 92521.

‡ Department of Computer Science, University of Chicago, Chicago, Illinois 60637.

§ This research was supported by National Science Foundation grant CCR 8807534.

and Sleator [10], [11]. They presented a 2-competitive algorithm for the 2-server problem and an $(n-1)$-competitive algorithm for the $(n-1)$-server problem on n-point metric spaces. It was also Manasse, McGeoch, and Sleator [10] who showed that for any metric space with $k+1$ or more points, no c_k-competitive strategy exists for that metric space if $c_k < k$. Recently, Irani and Rubinfeld [8] proved that a version of a balancing algorithm is 10-competitive for two servers.

One approach to the problem is to seek a randomized strategy for which the expected value of the cost does not exceed a plus c_k times the optimal cost. The lower bound mentioned above collapses in the randomized model. In fact, for the *paging* problem—the allowable metric spaces are those with all unit distances—Fiat et al. [7] presented a strategy for paging that is $2H_k$-competitive, where H_k is the kth harmonic number. (H_k is asymptotic to ln k.) They also proved that for the paging problem no strategy can be c_k-competitive unless $c_k \geqq H_k$. Thus, the H_k-competitive algorithm presented by McGeoch and Sleator [12] is optimal.

In § 2, we consider the case of k servers on a line. The simple algorithm we present is k-competitive and hence optimal. This is the first competitive deterministic algorithm for k servers in a metric space with unboundedly large distances. Our algorithm is, in addition, *memoryless* in that the algorithm can be specified by a function $f : M^{k+1} \rightarrow M^k$. When in configuration $(p_1, p_2, \cdots, p_k)$ with a request at Q, the servers move to $f(p_1, p_2, \cdots, p_k, Q) = (p'_1, p'_2, \cdots, p'_k)$, the ith moving from p_i to p'_i (of course $Q \in \{p'_1, \cdots, p'_k\}$). Such an algorithm keeps no record of the past other than the configuration itself. The importance of memoryless algorithms has been emphasized by Raghavan and Snir [13]. (One should be aware of the possibility that in certain infinite metric spaces one might be able to store the entire history of the computation in the location of one or more servers.)

Section 3 proves that a simple balancing algorithm is k-competitive—and thus optimal—for the *weighted-cache problem*, in which associated with each point x is a positive weight $w(x)$ and the distance to x from any other point is $w(x)$. Thus, we allow the "metric" to be asymmetric and apply the definitions above. (We say an *asymmetric metric space* is a space for which $d(x, y)$ is nonnegative and $d(x, y) = 0$ implies $x = y$. Distances must obey the triangle inequality, yet need not be symmetric.) This problem was proposed by Manasse, McGeoch, and Sleator [11], and a randomized k-competitive algorithm was given by Raghavan and Snir [13]. In the same section, we also present a new memoryless algorithm for the unweighted case.

In § 4, we first show that for the general asymmetric problem, there can be no competitive algorithm, even for two servers. This stands in contrast to the conjecture "that among all k-server problems, the ones that have the highest competitive factor are the symmetric ones," made by Manasse, McGeoch, and Sleator in [11]. We then show that there is no deterministic memoryless algorithm for the weighted-cache problem.

In Section 5, we consider offline algorithms for the server problem. We give a $O(kn^2)$-time algorithm to find an optimal schedule for k servers and n requests. Following this result is a proof that when $k = n/2$ this problem is at least as hard as the assignment problem on n-node bipartite graphs. Last, we show that on the line (where the input size is n), the time needed to find an optimal schedule in the algebraic computation tree model is $\Omega(n \log n)$.

By virtue of the triangle inequality, an optimal strategy can be assumed never to move more than one server while serving a request (see [10]). However, with concurrent motion of the servers, the algorithm for the line is memoryless, and the proofs are simpler. When we simulate our algorithm with one that moves only one server at a time, this new algorithm will need memory to store the virtual positions of our servers.

Often, our goal will be to show that an online algorithm we construct is competitive. To do this, we will conceptually create an *adversary* who, with his own k servers, must serve the same sequence of requests as we serve, starting from the same configuration, but who knows—indeed, who chooses—the entire request sequence in advance. If we can prove that our cost is no more than a plus k times the adversary's cost on the same request sequence, despite the adversary's foresight, then our online algorithm must be k-competitive.

2. A memoryless algorithm for the line.

In this section, we present an algorithm for k servers on a line. The algorithm is simple, memoryless, and achieves the optimal ratio: k. Each request is specified by a real number, the location on the line of the request. Here is the algorithm.

Algorithm DOUBLE-COVERAGE. Where s denotes our server closest to the request P, and d is the distance from s to P, serve the request with s. Then, if we have any servers on the "side" of P opposite to s (e.g., to P's left if s is to P's right), move the closest one d units toward P. Thus, we actually move one or two servers in response to a request, moving one if and only if all of our servers are on the same side of P.

We imagine that in response to a request, first the adversary serves the request, all of our servers remaining stationary. After the adversary has served the request he stays stationary while we serve the request. S^* and A^* denote the total cost we and the adversary incur over the whole sequence of requests, respectively. We label our k servers $s_1, \cdots, s_k$; the adversary's k servers are labeled $x_1, \cdots, x_k$. We will also use s_i and x_j to denote real numbers giving the current locations of servers s_i and x_j on the real line. By relabeling the servers as they move along the line, if necessary, we will always assume that $s_1 \leqq s_2 \leqq \cdots \leqq s_k$ and $x_1 \leqq x_2 \leqq \cdots \leqq x_k$.

LEMMA 1. *Suppose that Φ is a nonnegative potential function such that*

(1) *While the adversary serves a request, Φ cannot increase by more than k times the distance moved by the adversary, and*

(2) *When we serve a request, Φ decreases by at least the cost we incur in serving the request.*

Then $S^* \leqq kA^* + \Phi_0$, where Φ_0 is the initial value of Φ.

The proof of this lemma, standard in the study of amortized time analysis, proceeds by a simple summation over the whole sequence of requests. The details are omitted.

Our potential function is $\Phi = \Psi + \Theta$, where

$$\Psi = k \sum_{i=1}^{k} |x_i - s_i|$$

and

$$\Theta = \sum_{i<j} (s_j - s_i),$$

the potential function of [6] specialized to our problem. That this simple potential function (as opposed to our more complicated formulation of the same potential function) indeed works for DOUBLE-COVERAGE was pointed out to us by Borodin [3].

THEOREM 1. *$S^* \leqq kA^* + \Phi_0$, that is,* DOUBLE-COVERAGE *is k-competitive.*

Proof. We will prove the theorem by showing that conditions (1) and (2) from Lemma 1 hold. Lemma 1 then implies that $S^* \leqq kA^* + \Phi_0$. ($\Phi_0 = 0$ if all $2k$ servers initially coincide.) We may assume that as a server moves, it does not pass any other servers, neither ours nor the adversary's. Otherwise, we can divide the motion of a server into phases, within each one of which the order of the servers remains unchanged. (Of course, at the end of a phase if the moving server "overtakes" another server, we relabel them, if necessary. Doing so leaves Φ unchanged.)

Suppose that x_i moves a distance d within a phase. Then it is clear that Ψ can increase by at most kd, while Θ remains unchanged, and therefore condition (1) holds.

Condition (2) deals only with the second half of the move, after the adversary has served the request. There are two cases: either all of our k servers are on the same side of the request P, or they are not. If all are on one side, say, the right—the other case is similar—our closest server is s_1 and, since the adversary has already served the request, he has a server, say, x_j, at P. Clearly $s_1 \geqq x_j \geqq x_1$. Moving s_1 to the left d units decreases Ψ by kd and increases Θ by $(k - 1)d$, decreasing Φ in total by d. Thus, condition (2) holds in this case.

Now, the second case, in which we incur a cost of $2d$. Suppose s_i is our nearest server to P's left, s_{i+1} then being our nearest server to P's right, and let the adversary server on the request site be, say, x_j.

First, we analyze the effect of the motion of our servers on Ψ. Only the ith and $(i + 1)$st terms can change. If the adversary's server x_j at the request site satisfies $j \leqq i$, then the ith term of Ψ will decrease by kd while the $(i + 1)$st can increase by at most kd. If $j \geqq i + 1$, the $(i + 1)$st term decreases by kd while the ith can increase by at most kd. In either case, Ψ cannot increase.

Now, we study Θ. The change in Θ due to the movement of our two servers is

$$d[-(k-i)+(i-1)-(i)+(k-(i+1))]=-2d.$$

It follows that (2) holds. □

3. Cache problems. We first present a k-competitive algorithm for the weighted-cache problem, a version of the server problem in which each point x is assigned a positive *weight*, $w(x)$. Upon serving a request at x, a server is charged 0 or $w(x)$, according to whether it occupied x or did not occupy x just before serving the request, respectively. This situation is exactly the server problem on the asymmetric metric space in which the distance to x from any other point is $w(x)$. Initially, our ith server and the adversary's ith coincide at some point r_i; we assume the r_i's are distinct.

Let S^* and A^* be the total cost incurred by us and by the adversary, respectively, over the entire request sequence. We label our servers $s_1, s_2, \cdots, s_k$ arbitrarily and let S_i always denote the current total of the costs charged to s_i. In this notation, our strategy can be stated as follows.

Algorithm BALANCE. If we have a server at the requested point, we serve the request with that server. Otherwise, we use any server s_a for which $S_a = \min\{S_1, \cdots, S_k\}$.

We assume without loss of generality that once an occupied point becomes unoccupied (by both us and the adversary) it remains so—a request for it may be replaced by a request for a new point of equal weight. We also assume that there are no requests for points we currently occupy—omitting such a request and the corresponding adversary move leaves our cost unchanged and will, by the triangle inequality, never increase the adversary's cost. We assume, finally, that at the end of the game we and the adversary occupy the same set of points, called the *final* points—subsequent requests for points occupied by unopposed adversary servers would run up our charges and cost the adversary nothing.

Let W_{NF} denote the sum of the weights of the nonfinal points, and let W_F denote the sum of the weights of the final points. Note that $A^* = W_{NF} + W_F$, since each requested point is occupied by the adversary exactly once.

Let T always denote the value of $\min\{S_1, \cdots, S_k\}$ *just before the most recent service*. We assume its initial value to be zero and let T^* denote its final value. By $\|s_i\|$ we denote the weight of the point currently occupied by s_i.

LEMMA 3. *$S_i - \|s_i\| \leqq T$, for $i = 1, \cdots, k$.*

Proof. Suppose that we served the most recent request with s_j, and choose i arbitrarily. If s_i has not served any requests so far, then $S_i = 0$ and, hence, $S_i - \|s_i\| \leqq T$. Otherwise, consider the time just before s_i moved most recently. At that time, $S_i \leqq S_j$, because BALANCE chose to use s_i to serve the request. Currently, S_i is exactly $\|s_i\|$ larger than it was then, and T is at least as large now as S_j was then. Thus, now, $S_i - \|s_i\| \leqq T$. □

LEMMA 4. *$T^* \leqq W_{NF}$.*

Proof. Zero initially, T cannot increase until all k of our servers have moved; in fact, T cannot increase from zero until one has moved twice. It follows that if T increases while s_j occupies y, then at some point in the past s_j served a request at y.

Consider the period of time between time t, just after a server s_j most recently served a request on a nonfinal point y, and time t', just after s_j has vacated y in order to serve some other request. At time t, T is equal to $S' - w(y)$, where S' is the value of S_j at time t. At time t', just after s_j serves a new request, $T = S'$. We conclude that T increases by exactly $w(y)$ during a visit of our server to a nonfinal point y.

We assume that the adversary always moves first to serve a request. Therefore, just after the adversary's move, there is a point that is occupied only by one of our servers. The lemma is proven by allocating the increases in T to such points.

When s_a increases T by serving a request, we do the following. We choose any point y that is occupied, just before s_a serves the request, only by one of our servers, say s_j, and allocate to y the increase in T due to serving the request by s_a. (Perhaps $s_a = s_j$.) Previously, s_j must have served a request at this nonfinal y. While s_j occupies y, T increases by $w(y)$, and thus the total increase of T allocated to y during this visit of s_j is at most $w(y)$ (it may be less, since some increases in T may be allocated to other points). Since y will never again be requested after s_j leaves y, the total increase of T allocated to y is at most $w(y)$.

All increases of T can be allocated in this way to nonfinal points. Thus, the final value T^* of T is the sum of the increases allocated to nonfinal points, and this is at most W_{NF}, by the previous paragraph. □

We now prove that BALANCE is k-competitive. From [10], we infer that even for the "unweighted" cache problem in which all weights are one (and hence the metric is symmetric), there can be no c_k-competitive algorithm unless $c_k \geqq k$. Thus if $c_k < k$ no online algorithm can be c_k-competitive for all instances of the weighted-cache problem.

THEOREM 5. *$S^* \leqq kA^*$ if no two of our servers coincide at the start.*

Proof. By Lemma 3 we infer that at the end of the game

$$\sum_{i=1}^{k} (S_i - T^*) \leqq \sum_{i=1}^{k} \|s_i\| = W_F.$$

Using this inequality and Lemma 4 we derive

$$\begin{aligned} S^* = \sum_{i=1}^{k} S_i &= kT^* + \sum_{i=1}^{k} (S_i - T^*) \\ &\leqq kW_{NF} + W_F \leqq k(W_{NF} + W_F) \\ &= kA^*. \end{aligned}$$

□

Note. Chlebus [5] proved independently, using a different technique, that BALANCE is k-competitive.

Next we consider the unweighted-cache problem, for which several k-competitive algorithms are known: FIFO, LRU, and those of [9] and [13]. Most are not memoryless. We present a new k-competitive memoryless algorithm. Our algorithm, unlike the memoryless algorithm Flush–When–Full [9], keeps the cache full at all times.

For convenience, we view the metric space for the unweighted-cache problem as the real interval $I = (0, 1]$ by mapping the metric space into I arbitrarily. Initially, the k servers occupy any k distinct points. Always, each of our k servers occupies a position given by a real number in I (each such position is the location of some previous request or an initial location). In each time step, a request (a real number $v \in I$) is specified by the adversary. If we have a server at v, we move no servers and pay nothing. Otherwise, we choose one of our k servers and move it to v, incurring *unit* cost. (No two of our servers ever occupy the same point simultaneously. Without loss of generality, neither do two of the adversary's.)

We label our k servers $s_1, s_2, \cdots, s_k$ arbitrarily, and also use s_i to denote the location of our ith server. We use Algorithm ROTATE.

Algorithm ROTATE. In response to a request at a point $v \in (0, 1]$ unoccupied by any of our servers, if there is an $s_i < v$, choose a maximum such s_i; if no $s_i < v$ exists, choose a maximum s_i. Serve the request at v with server s_i.

Intuitively, our servers move rightward "around" the interval. ROTATE clearly is memoryless. We will show that ROTATE is k-competitive.

For $0 < a, b \leqq 1$, we will use $(a, b]$ to mean $\{x | a < x \leqq b\}$ if $a \leqq b$, and to mean $\{x | a < x \leqq 1 \text{ or } 0 < x \leqq b\}$ if $a > b$.

THEOREM 6. ROTATE *is k-competitive.*

Proof. Label the adversary's k servers $x_1, x_2, \cdots, x_k$ arbitrarily, using x_i also to denote the location of the adversary's ith server. Let $c_{i,p}$ denote the number of the adversary's servers in $(s_i, x_p]$ and, where π is a permutation of $\{1, 2, \cdots, k\}$, let $c(\pi) = \sum_{i=1}^{k} c_{i,\pi(i)}$. Our potential function is

$$\Phi = \min_{\pi} c(\pi),$$

the minimum being taken over all permutations π of $1, 2, \cdots, k$. In other words, Φ is the weight of the minimum weight perfect matching in the bipartite graph with edges $\{s_i, x_p\}$ of weight c_{ip}. We say that π *realizes* Φ if $\Phi = c(\pi)$.

We assume that in response to each request the adversary moves first. We prove the theorem by showing that inequalities (1) and (2) from Lemma 1 hold.

Inequality (1) can be proven as follows. Clearly, we may focus on the case in which the adversary incurs unit cost, by moving x_p (say) to the request site v. We imagine that he moves x_p "rightward" to v, wrapping around the end if necessary. Φ can change only when x_p passes a point occupied only by one of our servers, a point occupied only by one of the adversary's servers, or a point occupied by both. When x_p passes a point containing only one of our servers, Φ increases by at most one. If x_p passes a point occupied only by one of the adversary servers, say x_q, Φ remains unchanged (we may interchange p and q in a permutation π realizing Φ). If x_p passes a point occupied by s_l and x_q, and, say, $\pi(i) = p$, $\pi(j) = q$, then it is not hard to verify that if τ is the same as π except that $\tau(i) = q$, $\tau(j) = p$, then $c(\tau) \leqq c(\pi) + 1$. Thus, because we have only k servers, as x_p moves to v the total increase in Φ is at most k.

Before proving (2), let us look at permutations π realizing Φ. Call two intervals $(s_i, x_p]$, $(s_j, x_q]$ *independent* if neither is a subset of the other. If no two s_i's are identical, and no two x_p's are identical, then there is always a permutation π realizing Φ for which pairs of distinct intervals $(s_i, x_{\pi(i)}]$ are independent. For if $(s_i, x_{\pi(i)}]$ is a subset of $(s_j, x_{\pi(j)}]$, let τ be the same permutation as π except that $\tau(j) = \pi(i)$ and $\tau(i) = \pi(j)$; $c(\tau) \leqq c(\pi)$. Where the length of $(u, v]$ is defined in the obvious way, the sum of the squares of the lengths of $(i, \tau(i)]$ is strictly less than the sum of the squares of the lengths of $(i, \pi(i)]$, so this process must terminate.

Now we prove (2) from Lemma 1. If the request site is occupied by one of our servers, both sides are zero. Otherwise, we incur a cost of one. Suppose that the request site is occupied by x_p, and ROTATE specifies that the request is to be served by s_i. Choose a permutation π realizing Φ for which pairs of distinct intervals are independent.

If $\pi(i) = p$, as s_i moves rightward, $c(\pi)$ decreases by at least one.

Thus we may assume that $\pi(i) \neq p$. If $x_p \notin (s_i, x_{\pi(i)}]$, then, since s_i is the first server of ours to the "left" of x_p, $(s_i, x_{\pi(i)}]$ is a subset of $(s_{\pi^{-1}(p)}, x_p]$, a contradiction. Thus $x_p \in (s_i, x_{\pi(i)}]$ and, therefore, as s_i moves rightward to serve the request, $c(\pi)$ decreases by at least one. □

4. Nonexistence theorems. First we prove that in the asymmetric case in general there can be no competitive algorithm, even if there are only two servers.

THEOREM 7. *There is no competitive algorithm for the* 2*-server problem on asymmetric metric spaces.*

Proof. Where K is positive, consider a $3K$-node digraph H_K on vertex set $\{x_0, y_0, z_0, x_1, y_1, z_1, \cdots, x_{K-1}, y_{K-1}, z_{K-1}\}$ with $4K$ arcs $\{(x_i, z_i), (y_i, z_i), (z_i, x_{i+1}), (z_i, y_{i+1}) | 0 \leqq i \leqq K-1\}$. Define M_K to be the asymmetric metric space for which $d(x, y)$ is the length of the shortest $x \rightarrow y$ path in H_K. Then $d(x_i, y_i) = d(y_i, x_i) = 2K$ while $d(z_i, x_i) = d(z_i, y_i) = 2K - 1$.

If an online algorithm is purportedly c_k-competitive, choose K so that $(2K-1)/4 \geqq c_k + 1$ and place all the servers initially at z_0 in M_K. Let a be such that $S^* \leqq c_k A^* + a$ for all request sequences whose servers start at z_0.

The adversary's request sequence consists of a sequence of *phases*. At the beginning of the ith phase $(i \geqq 1)$, the adversary has two servers on z_{i-1}; the online algorithm's servers are anywhere. (Where necessary within this proof, we do arithmetic modulo K.) The adversary starts by requesting x_i and then y_i, and uses two servers to serve the requests. If after the two requests the online algorithm does not have two servers at x_i and y_i, the adversary simply alternately requests x_i and y_i, incurring no cost himself, until the online algorithm puts his two servers there (or until the cost of the online algorithm exceeds a more than c times the offline cost to date). Now the adversary requests z_i. If the online algorithm serves the request with the server at x_i, the adversary uses his server at y_i, and *vice versa*. In the first case, the adversary then requests x_i, incurring no cost but costing the online algorithm at least $2K-1$. Next, the adversary requests z_i and moves his second server to z_i. (The second case is similar.) This completes the ith phase.

In one phase, the adversary incurs a cost of 4, while the algorithm incurs a cost of at least $2K-1$. Since $(2K-1)/4 \geqq c_k + 1$, if we run enough phases eventually $S^* > c_k A^* + a$. □

Our algorithm BALANCE for the weighted-cache problem is not memoryless. Raghavan and Snir [13] presented a randomized memoryless algorithm for the weighted-cache problem. The results of § 3 show that for the unweighted-cache problem, a memoryless optimal algorithm does indeed exist. The theorem below shows that using either randomization or memory is indeed necessary when arbitrary weights are allowed.

THEOREM 8. *There is no memoryless competitive algorithm for the k-server weighted-cache problem.*

Proof. We will show that for each c_k, there is an assignment of positive weights to any finite set of $n \geqq k + 1$ points so that no c_k-competitive memoryless algorithm can exist for the associated instance of the weighted-cache problem. Let the vertex set be $\{1, 2, \cdots, n\}$, $n \geqq k + 1$. There are no more than n^k configurations. Let K be positive and assign weight $w(v) = K^v$ to node v. The adversary chooses any strategy that always places the next request on an unoccupied node. Eventually, one configuration C appears a second time (after at most n^k requests, in fact). The adversary now modifies his strategy: from now on, he simply repeats the requests in the cycle between the first and second occurrence of C. The memorylessness of the algorithm ensures that the cycle will be repeated *ad infinitum*. Where m is the highest-numbered node that is requested in the cycle, each iteration through the cycle costs the algorithm at least K^m. The adversary simply uses two servers: one sits permanently on m while the other serves all requests to nodes other than m, thereby incurring a cost of at most $K^{m-1}n^k$ for the cycle. Over an extremely long request sequence, the costs incurred before reaching the cycle can be ignored. Thus, in the limit, the ratio between the cost we incur to that incurred by the adversary is at least $K^m/(K^{m-1}n^k) = K/n^k$. A suitable choice of K makes this ratio exceed c_k. □

Now we again consider the k-server problem on the line. The algorithm we presented in § 2 often moved two servers to serve a request. In fact, the following theorem states that no memoryless competitive algorithm on the line always moves at most one server in response to a request. Similar to the previous proof, it uses points K^i on the real line in place of weights K^i. We omit the details.

THEOREM 9. *There are no memoryless competitive algorithms for k servers on the line that move only one server at a time.*

5. Offline problems. We study the problem of finding an optimal strategy to serve a sequence of n requests with k servers, if the request sequence is given in advance. We assume that the k servers initially occupy one point, the *origin* (but the algorithm can easily be modified if they do not). When there are n requests, the inputs to our problem are the superdiagonal entries of an $(n + 1) \times (n + 1)$ matrix, whose $(0, j)$ entry is the distance from the origin to the location of request j, $j = 1, 2, \cdots, n$, and whose (i, j) entry is the distance from the location of request i to the location of request j, $1 \leqq i < j \leqq n$.

The dynamic programming algorithm of Manasse, McGeoch, and Sleator [10] is especially suited for the case in which the number of requests dramatically exceeds the number m of points in the metric space. Its running time is $O(nm\binom{m}{k})$ and space usage at least $\binom{m}{k}$ for a request sequence of length n in an m-point metric space.

THEOREM 10. *There is a $O(kn^2)$-time offline algorithm to find an optimal schedule for k servers to serve a sequence of n requests.*

Proof. We reduce the k-server problem to the problem of finding a minimum cost flow of maximum value in an acyclic network. If we have k servers $s_1, \cdots, s_k$ and n requests $r_1, \cdots, r_n$, we build this $(2 + k + 2n)$-node acyclic network: the vertex set is

$$V = \{s, s_1, s_2, \cdots, s_k, r_1, r'_1, r_2, r'_2, \cdots, r_n, r'_n, t\}.$$

Nodes s and t are the source and sink, respectively. Each arc has capacity one. There is an arc of cost 0 from s to each s_i, an arc of cost 0 from each r'_j to t, as well as an arc to t from each s_i, of cost 0. From each s_i, there is an arc to r_j of cost equal to the distance from the origin to the location of r_j. For $i < j$ there is an arc from r'_i to r_j of cost equal to the distance between r_i to r_j. Furthermore, from r_i to r'_i there is an arc of cost $-K$, where K is an extremely large real.

The value of the maximum flow in this network is k. Because all capacities are integral, and because the network is acyclic, we can use minimum-cost augmentation [15] to find an integral min-cost flow of value k in time $O(kn^2)$. An integral $s \twoheadrightarrow t$ flow of value k can be decomposed into k arc-disjoint $s \twoheadrightarrow t$ paths, the ith one passing through s_i. Because $-K$ is so small, an integral min-cost flow of value k saturates all of the (r_j, r'_j) arcs, and hence corresponds to an optimal schedule for serving the requests, the ith server serving exactly those requests contained in the $s \twoheadrightarrow t$ path that passes through s_i. □

In the next theorem we prove that finding an optimal offline strategy for metric spaces is at least as hard as finding a minimum-weight perfect matching in a complete bipartite graph. This is true even though the metric space distances must satisfy the triangle inequality.

THEOREM 11. *If there is an algorithm that computes optimal offline strategies for k-server request sequences of length 2k and runs in time $g(k)$, then there is an algorithm that finds minimum weight perfect matchings in 2k-node complete bipartite graphs in time $g(k) + O(k^2)$.*

Proof. If G is a complete $2k$-node bipartite graph with k boys and k girls, with an edge of weight w_{ij} from boy i to girl j, build a server problem on $2k + 1$ points $s, x_1, x_2, \cdots, x_k, y_1, y_2, \cdots, y_k$. Initially, all k servers occupy s. The length of edge $\{x_i, y_j\}$ is $K + w_{ij}$, where again K is extremely large. The edges $\{s, x_i\}$ are of length K, while all remaining edges are of length $2K$. For K sufficiently large, the triangle inequality holds.

Consider the request sequence $x_1, x_2, \cdots, x_k, y_1, y_2, \cdots, y_k$. If K is large enough, an offline strategy that traverses even one edge of length $2K$ is suboptimal. Thus, the first k requests, those to $x_1, \cdots, x_k$, must be served by k different servers. The same goes for the last k requests. Thus, an optimal offline algorithm is one that pairs up the boys and girls so as to minimize the length of the edges traversed, thereby solving the assignment problem. □

Where $n = 2k$, our algorithm runs in $O(n^3)$ time, the best time known for the assignment problem.

If the metric space is the line, where specifying n request locations requires only n reals, is there a faster algorithm than the one above? Possibly there is. But there is no linear-time algorithm to find an optimal offline strategy on the line.

THEOREM 12. *In the algebraic computation tree model, any algorithm that finds an optimal offline strategy for n requests on the line takes $\Omega(n \log n)$ time.*

Proof. Let $k = n/2$ (for even n) and assume that all k servers are initially at the origin. Given reals $a_1, a_2, \cdots, a_k$, determining if $a_i \in X = \{3, 3^2, \cdots, 3^k\}$ for all i is known to require $\Omega(n \log n)$ time in the algebraic computation tree model [1]. Consider the request sequence $3^k, 3^{k-1}, \cdots, 3, a_1, a_2, \cdots, a_k$, of length n. We claim that the cost of an optimal offline strategy for this request sequence is $N = \sum_{i=1}^{k} 3^i$ if and only if $\{a_1, \cdots, a_k\} \subseteq X$. If the latter holds, there is a schedule of cost N: serve the first k requests with k distinct servers, and then incur a cost of 0 for the last k requests. If the first k requests are not served by different servers, then the schedule's cost exceeds N. If the first k requests *are* served by k different servers, the optimal cost can be N only if $\{a_1, \cdots, a_k\} \subseteq X$. Thus, the optimal cost is N if and only if $\{a_1, \cdots, a_k\} \subseteq X$. □

Acknowledgments. We would like to thank Piotr Berman, Bogdan Chlebus, Gábor Tardos, Ketan Mulmuley, and Janos Simon for many valuable discussions related to our research.

REFERENCES

[1] M. Ben-Or, *Lower bounds for algebraic computation trees*, Proc. 15th ACM Symposium on Theory of Computing (1983), pp. 80–86.

[2] A. Borodin, N. Linial, and M. Saks, *An optimal online algorithm for metrical task systems*, Proc. 19th ACM Symposium on Theory of Computing (1987), pp. 373–382.

[3] A. Borodin, personal communication.

[4] A. R. Calderbank, E. G. Coffman, and L. Flatto, *Sequencing problems in two-server systems*, Math. Oper. Res., 10 (1985), pp. 585–598.

[5] B. Chlebus, University of California, Riverside, CA, personal communication.

[6] D. Coppersmith, P. G. Doyle, P. Raghavan, and M. Snir, *Random walks on weighted graphs, with applications to on-line algorithms*, Proc. 22nd ACM Symposium on Theory of Computing, Baltimore, (1990), pp. 369–378.

[7] A. Fiat, R. Karp, M. Luby, L. A. McGeoch, D. Sleator, and N. E. Young, *Competitive paging algorithms*, Technical report CMU-CS-88-196, Computer Science Department, Carnegie-Mellon University, 1988.

[8] S. Irani and R. Rubinfeld, *A competitive 2-server algorithm*, Inform. Process Lett., submitted.

[9] A. Karlin, M. Manasse, L. Rudolph, and D. Sleator, *Competitive snoopy caching*, Algorithmica, 3 (1988), pp. 79–119.

[10] M. Manasse, L. A. McGeoch, and D. Sleator, *Competitive algorithms for online problems*, Proc. 20th ACM Symposium on Theory of Computing (1988), pp. 322–333.

[11] ———, *Competitive algorithms for server problems*, J. Algorithms, 11 (1990), pp. 208–230.

[12] L. McGeoch and D. Sleator, *A strongly competitive randomized paging algorithm*, Algorithmica, to appear.

[13] P. Raghavan and M. Snir, *Memory versus randomization in online algorithms*, Technical Report, IBM Research Report RC 15622, March 1990.

[14] D. Sleator and R. E. Tarjan, *Amortized efficiency of list update and paging rules*, Comm. ACM, 28 (1985), pp. 202–208.

[15] R. Tarjan, *Data Structures and Network Algorithms*, BMS-NSF Regional Conference Series in Applied Mathematics, Vol. 44, 1983, pp. 109–111.

SIAM J. DISC. MATH.
Vol. 4, No. 2, pp. 182-195, May 1991

004

TOWARDS A LARGE SET OF STEINER QUADRUPLE SYSTEMS*

TUVI ETZION† AND ALAN HARTMAN‡

Abstract. Let $D(v)$ be the number of pairwise disjoint Steiner quadruple systems. A simple counting argument shows that $D(v) \geqq v - 3$. In this paper it is proved that $D(2^k n) \geqq (2^k - 1)n$, $k \geqq 2$, if there exists a set of $3n$ pairwise disjoint Steiner quadruple systems of order $4n$ with a certain structure. This implies that $D(v) \geqq v - o(v)$ for infinitely many values of v. New lower bounds on $D(v)$ for many values of v that are not divisible by 4 are also given, and it is proved that $D(v) \geqq 2$ for all $v \equiv 2$ or $4 (\mathrm{mod}\ 6)$, $v \geqq 8$.

Key words. Steiner quadruple system, orthogonal array, one-factorization, disjoint Steiner systems

AMS(MOS) subject classifications. 51E10, 05B30, 05B15

1. Introduction. A *Steiner quadruple system* (SQS) is a pair (Q, q) where Q is a finite set of *points* and q is a collection of 4-element subsets of Q called *blocks* such that every 3-element subset of Q is contained in exactly one block of q. The number of points in Q is the order of the SQS, and it is well known that an SQS of order v, denoted SQS(v), has $b_v = \frac{1}{4}\binom{v}{3}$ blocks. Hanani [5] proved that Steiner quadruple systems of order v exist if and only if $v \equiv 2$ or $4 \ (\mathrm{mod}\ 6)$. Two SQS (Q, q_1) and (Q, q_2) are disjoint if $q_1 \cap q_2 = \emptyset$. A coloring of an SQS is a partition of the set of points into color classes such that no block is properly contained in any color class. An SQS is k-chromatic if it can be k-colored, but no proper coloring having fewer than k color classes exists. A set of p pairwise disjoint SQSs (PDQs) is mutually 2-chromatic if the same partition of Q is a 2-coloring of all the PDQs. If (Q, q) is a 2-chromatic SQS$(2v)$, with 2-coloring A, B, then by Doyen and Vandensavel [3], $|A| = |B| = v$, and the number of blocks n_1, n_2, n_3 that meet A in 1, 2, and 3 points, respectively, is

$$n_1 = n_3 = \binom{v}{3}, \qquad n_2 = \binom{v}{2}.$$

Hence, the maximum size of a set of disjoint mutually 2-chromatic SQS$(2v)$ is v since the number of 4-subsets intersecting A in 3 points is $\binom{v}{3}v$.

Let $D(v)$ denote the maximum number of PDQs. Since $\binom{v}{4} = (v - 3)b_v$, we have that $D(v) \leqq v - 3$. A set of $v - 3$ PDQs of order v is called a *large set*. An application of sets of PDQs is in the construction of constant weight codes with distance 4 [1].

It is well known that $D(4) = 1$, $D(8) = 2$, and Kramer and Mesner [11] proved that $D(10) = 5$. Phelps [17] proved that $D(2.5^t) \geqq 5^t$, Phelps and Rosa [19] proved that $D(2.5^a \cdot 13^b \cdot 17^c) \geqq 5^a \cdot 13^b \cdot 17^c$, for all $a, b, c \geqq 0$, $a + b + c > 0$, and Lindner [12] proved that $D(2v) \geqq v$ for $v \equiv 2$ or $4 \ (\mathrm{mod}\ 6)$, $v \geqq 8$. Recently, Phelps [18] has shown that $D(22) \geqq 11$. All the PDQs of these four constructions are mutually 2-chromatic. Lindner [13] proved that $D(4v) \geqq 3v$ for $v \equiv 2$ or $4 \ (\mathrm{mod}\ 6)$, $v \geqq 8$ by using his v PDQs of order $2v$ [12].

In § 2 we use a construction with a similar structure to the one of Lindner [13] to obtain $D(4v) \geqq 3p$, where $v \equiv 1$ or $5 (\mathrm{mod}\ 6)$ and a set of p mutually 2-chromatic PDQs of order $2v$ exists. If $p = v$ then our set of $3p$ PDQs is maximal.

In § 3 we use the PDQs of Lindner [13], our PDQs of § 2, and a set of $2^k - 1$ Boolean SQSs of order 2^k to obtain $D(2^k v) \geqq (2^k - 1)v$ for $v \equiv 2$ or $4 (\mathrm{mod}\ 6)$, $v \geqq 8$,

* Received by the editors October 6, 1989; accepted for publication (in revised form) July 13, 1990.

† Computer Science Department, Technion, Haifa, Israel.
‡ IBM Israel, Science and Technology and Scientific Center, Haifa, Israel.

for $v = 5^a \cdot 13^b \cdot 17^c$, for all $a, b, c \geqq 0$, $a + b + c > 0$, and also for $v = 11$. This result implies, for example, $D(5.2^t) \geqq 5.2^t - 5$, and this is almost best possible, since if there exist $5.2^t - 4$ PDQs of order 5.2^t, the unused quadruples must form an additional disjoint system, and thus a large set exists. No large set of PDQs has been constructed yet.

In § 4 we give various recursive bounds on $D(v)$ in order to improve the state of knowledge when v is not divisible by 4. We also prove the long standing conjecture, due to Lindner and Rosa [14], that $D(v) \geqq 2$ for all $v \equiv 2$ or 4 (mod 6) with $v \geqq 8$.

2. The construction for orders 4v, v odd. An *orthogonal array* $OA(t, k, n)$ is an $n^t \times k$ matrix M, with entries from the set $\{0, 1, \cdots, n-1\}$, such that the submatrix generated by any t columns contains each ordered t-tuple exactly once as a row. If $M(i,0) = j_0$, $M(i,1) = j_1, \cdots, M(i, k-1) = j_{k-1}$, we can also write $(j_0, j_1, \cdots, j_{k-1}) \in M$ or $\{(j_0, 0), (j_1, 1), \cdots, (j_{k-1}, k-1)\} \in M$.

Two orthogonal arrays M_1 and M_2 are disjoint if they have no row in common.

A set $M_0, M_1, \cdots, M_{n-1}$ of n disjoint $OA(t, k, n)$ is said to have property X if the first $t + 1$ columns of the arrays cover each ordered $(t + 1)$-tuple exactly once. Note, that an orthogonal array $OA(t + 1, k + 1, n)$ implies the existence of a set with property X, but the contrary does not follow.

LEMMA 1. *For $n \equiv 1$ or 5 (mod 6) there exists an $OA(3, 5, n)$.*

Proof. Raghavarao [20] proved the existence of $OA(3, 5, p)$ for all primes p such that $p \geqq 5$. The proof then follows from the direct product construction for orthogonal arrays, since the smallest prime factor of n is at least 5.

LEMMA 2. *If there exists an $OA(3, 5, n)$ then there exists a set of n disjoint $OA(3, 5, n)$ with property X.*

Proof. Given an $OA(3, 5, n)$, M_0, the rows of M_r, $0 \leqq r \leqq n-1$ are defined by $(a + r, b, c, d, e)$, where $(a, b, c, d, e) \in M_0$ and $a + r$ is taken modulo n. It is obvious that each M_r, $0 \leqq r \leqq n-1$ is an $OA(3, 5, n)$. Property X follows from the construction.

We now construct a set of PDQs of order $4v$ using as input, a set of mutually 2-chromatic PDQs of order $2v$, an orthogonal array, a near one-factorization of K_v, and a fixed partition of a set of size 4. Let $v \equiv 1$ or 5(mod 6).

- Let D_k, $0 \leqq k \leqq p-1$, be the block sets of p mutually 2-chromatic PDQs of order $2v$, whose point set is $Z_v \times Z_2$ and whose color classes are $Z_v \times \{i\}$, $i = 0, 1$.
- Let M be an $OA(3, 5, v)$.
- Let $F = \{F_0, F_1, \cdots, F_{v-1}\}$ be a near-one-factorization of K_v such that in F_i vertex i is isolated.
- Let $\pi = \{(i, j), (s, t)\}$ be a fixed partition of Z_4 into two ordered pairs.

The point set of our quadruple systems is $Z_v \times Z_4$ and we construct the block sets $S_k(M, F, \pi)$ with $0 \leqq k \leqq p-1$ as the union of the blocks of Types A, B, and C defined below.

Type A.

$$[(a,i),(b,i),(c,j),(d,j)], \qquad [(a,s),(b,s),(c,t),(d,t)],$$

where $$[(a,0),(b,0),(c,1),(d,1)] \in D_k,$$

$$[(a,i),(b,j),(c,j),(d,j)], \qquad [(a,s),(b,t),(c,t),(d,t)],$$

where $$[(a,0),(b,1),(c,1),(d,1)] \in D_k,$$

$$[(a,i),(b,i),(c,i),(d,j)], \qquad [(a,s),(b,s),(c,s),(d,t)],$$

where $$[(a,0),(b,0),(c,0),(d,1)] \in D_k.$$

There are $2b_{2v}$ blocks of this type, which consist of two isomorphic copies of D_k—one on the point set $Z_v \times \{i,j\}$ and the other with point set $Z_v \times \{s,t\}$.

Type B.

$$[(a,i),(b,j),(e,s),(f,s)], \qquad [(a,i),(b,j),(g,t),(h,t)],$$

where $\{e,f\} \in F_c, \quad \{g,h\} \in F_d, \quad \{(a,i),(b,j),(c,s),(d,t),(k,4)\} \in M,$

$$[(a,s),(b,t),(e,i),(f,i)], \qquad [(a,s),(b,t),(g,j),(h,j)],$$

where $\{e,f\} \in F_c, \quad \{g,h\} \in F_d, \quad \{(c,i),(d,j),(a,s),(b,t),(k,4)\} \in M.$

There are $4v^2(v-1)/2$ blocks of Type B, and each of the blocks of Type B intersect three of the sets $Z_v \times \{x\}$ with $x \in Z_4$.

Type C.

$$[(a,i),(b,j),(c,s),(d,t)],$$

where $\{(a,i),(b,j),(c,s),(d,t),(k,4)\} \in M.$

There are v^2 blocks of Type C, and each of the blocks of Type C intersect all four of the sets $Z_v \times \{x\}$ with $x \in Z_4$.

Each $S_k(M, F, \pi)$, $0 \leqq k \leqq p-1$, has $2b_{2v} + 4v^2(v-1)/2 + v^2 = b_{4v}$ blocks.

To show that each $S_k(M, F, \pi)$ is indeed an SQS we note that any 3-subset T of $Z_v \times \{0, 1, 2, 3\}$ is contained in a unique block by the following arguments:

1) If $T \subset Z_v \times \{i,j\}$ or $T \subset Z_v \times \{s,t\}$ then T is contained in a unique block of Type A.
2) If T is of the form $\{(x,i),(y,i),(z,s)\}$, then $\{x,y\}$ is contained in a unique factor F_c and there is a unique row of M containing (c,i), (z,s), $(k,4)$—and thus T is in a unique block of Type B.
3) If T is of the form $\{(x,i),(y,j),(z,s)\}$ then, either (x,i), (y,j), (z,s), $(k,4)$ is contained in a row of M—in which case T is in a block of Type C. Otherwise, the row containing (x,i), (y,j), $(k,4)$ contains (w,s) with $w \neq z$. In this case a unique pair in F_w contains z—and thus T is in a unique block of Type B.

All the other triples T are covered by the symmetries of the construction and analogous arguments.

The fact that all the p SQSs are disjoint follows from:

1) The disjointness of the D_k, for blocks contained in $Z_v \times \{i,j\}$ or $Z_v \times \{s,t\}$.
2) The disjointness of the $OA(2, 4, v)$ induced by the rows of M containing the element $(k,4)$, for blocks of Types B and C.

Now let $\pi_0 = \{(0,1),(2,3)\}$, $\pi_1 = \{(0,2),(1,3)\}$, and $\pi_2 = \{(0,3),(1,2)\}$. Let M_0, M_1, M_2 be three $OA(3, 5, v)$ from a set of v $OA(3, 5, v)$ with property X. We construct the following block sets of $3p$ SQSs, $S_k(M_0, F, \pi_0)$, $S_k(M_1, F, \pi_1)$, and $S_k(M_2, F, \pi_2)$, with $0 \leqq k < p$. We denote this set of SQSs by $B((D_k), (M_i), F)$.

We claim that this is a set of PDQs, and to prove this it only remains to show that $S_k(M_i, F, \pi_i) \cap S_m(M_j, F, \pi_j) = \varnothing$, for any $i \neq j$. But this holds since the only possible blocks in common are those of Type C, and these are disjoint by property X. Hence, we have the following theorem.

THEOREM 1. *If there exists a set of p mutually* 2*-chromatic* PDQs *of order* $2v$ *then* $D(4v) \geqq 3p$.

Phelps and Rosa [19] proved that there exists a set of n mutually 2-chromatic PDQs of order $2n$, for $n = 5^a.13^b.17^c$, $a, b, c \geqq 0$, and $a + b + c > 0$. Phelps [18] has recently proved the same result for $n = 11$. Hence, we have the following corollary.

COROLLARY 1. *$D(4.5^a \cdot 13^b \cdot 17^c) \geqq 3.5^a \cdot 13^b \cdot 17^c$, for all $a, b, c \geqq 0$ and $a + b + c > 0$. Furthermore, $D(44) \geqq 33$.*

Before moving on to our next construction we list some of the properties of the set $B((D_k), (M_i), F)$ of PDQs.

- No block in any of the systems is entirely contained in any of the sets $Z_v \times \{i\}$, $i \in Z_4$, so these sets are a 4-coloring of each of the systems.
- The blocks in $S_k(M_0, F, \pi_0)$ that are entirely contained in $Z_v \times \{0, 1\}$ form the block sets of p mutually 2-chromatic PDQs of order $2v$ with 2-coloring $Z_v \times \{0\}$, $Z_v \times \{1\}$. Likewise, those blocks induced on the set $Z_v \times \{2, 3\}$ also form a set of mutually 2-chromatic PDQs. Furthermore, there are no other bichromatic blocks in any of the $S_k(M_0, F, \pi_0)$.
- Analogous statements can be made about the sets of block sets $S_k(M_1, F, \pi_1)$ with color classes 0, 2 and 1, 3, and also the sets $S_k(M_2, F, \pi_2)$ with color classes 0, 3 and 1, 2.
- For fixed k and all i there is an isomorphism between the quadruple system D_k and the subsystems of order $2v$ described above. Thus if $(a_0, a_1) \in \pi_i$ then the induced SQS on $Z_v \times \{a_0, a_1\}$ is isomorphic to the subdesign induced on $Z_v \times \{0, 1\}$. Furthermore, the isomorphism is given by $(x, a_0) \rightarrow (x, 0)$ and $(x, a_1) \rightarrow (x, 1)$.
- The block sets $S_k(M_1, F, \pi_1)$, and $S_k(M_2, F, \pi_2)$, with $0 \leqq k < p$ form a set of $2p$ mutually 2-chromatic PDQs of order $4v$ with two coloring $Z_v \times \{0, 1\}$, $Z_v \times \{2, 3\}$.
- If $p = v$ then our $3v$ PDQs are maximal (nonextendable).

3. The construction for orders $2^k v$, $k \geqq 3$. Our construction for $(2^k - 1)v$ PDQs of order $2^k v$ uses a set of Boolean Steiner quadruple systems, which are defined below. An order of Z_2^k is induced by identifying $x \in Z_2^k$ with the integer $i < 2^k$ whose binary representation is x.

We define a set of $2^k - 1$ SQSs all with point set Z_2^k and block sets B_i with $i \in Z_2^k - \{0\}$. These will be called the Boolean Steiner quadruple systems. The block set B_i, $i \in Z_2^k - \{0\}$ is defined to be the union of the blocks of types (B.1) and (B.2) specified below:

$$[x, y, z, w], \tag{B.1}$$

where

$$x + y + z + w = i \quad \text{and} \quad |\{x, y, z, w\}| = 4,$$

$$[x, y, z, w], \tag{B.2}$$

where

$$x + y = z + w = i \quad \text{and} \quad |\{x, y, z, w\}| = 4.$$

To show that (Z_2^k, B_i) is an SQS, let $T = \{q, r, s\}$ be a 3-subset of Z_2^k, and let $t = q + r + s + i$.

- If $t \notin \{q, r, s\}$ then $[q, r, s, t]$ is the unique block of type (B.1) containing T. Furthermore, no block of type (B.2) contains T, since if any 2-subset of T has sum i this implies that the third member equals t, a contradiction.
- If $t \in T$, say $t = q$, then we have $r + s = i$. We also observe that $q + i \notin T$, since $q + i = q$ is impossible, and $q + i = r$ implies $q = s$ contradicting the cardinality of T. Hence, $[q, r, s, q + i]$ is the unique block of type (B.2) containing T, and it is easy to see that no block of type (B.1) contains T.

The number of blocks of type (B.2) is $\binom{2^{k-1}}{2}$ since the number of pairs of distinct members of Z_2^k which sum to i is 2^{k-1}. Hence, the number of blocks of type (B.1) is $b_{2^k} - \binom{2^{k-1}}{2}$.

Note that each block of (B.2) has zero sum. Also note that $B_i \cap B_j$ contains the quadruple $[q, r, s, t]$ if and only if $q + r + s + t = 0$, and

$$\{i,j\} \subset \{q+r, q+s, q+t\}.$$

Hence, each zero sum 4-subset of Z_2^k is contained in precisely three of the Boolean SQSs, and each quadruple with nonzero sum is contained in a unique Boolean system.

The other main ingredient in our construction for $(2^k - 1)v$ PDQs of order $2^k v$ is a set of $3v$ PDQs of order $4v$ with a series of additional properties. These properties are satisfied by the set $B((D_j), (M_i), F)$ constructed in Corollary 1, and also by Lindner's set of PDQs constructed in [13].

Let D_{ij}, $i \in Z_3$, $j \in Z_p$, be the block sets of $3p$ PDQs with point set $Z_v \times Z_4$. We will refer to the subsets $Z_v \times \{x\}$ of the point set as color classes. We will say that the set D_{ij} has *Property Y*, if the following properties hold.

Property Y.1. None of the D_{ij} contains a monochromatic quadruple.

Property Y.2. Each set D_{0j} contains two 2-chromatic subdesigns of order $2v$, one with colors 0 and 1, and the other with colors 2 and 3. Furthermore, no other bichromatic quadruples are contained in D_{0j}.

Each set D_{1j} contains two 2-chromatic subdesigns of order $2v$, one with colors 0 and 2, and the other with colors 1 and 3. Furthermore, no other bichromatic quadruples are contained in D_{1j}.

Each set D_{2j} contains two 2-chromatic subdesigns of order $2v$, one with colors 0 and 3, and the other with colors 1 and 2. Furthermore, no other bichromatic quadruples are contained in D_{2j}.

Property Y.3. Let $c(0, 1) = c(2, 3) = 0$, $c(0, 2) = c(1, 3) = 1$, and $c(0, 3) = c(1, 2) = 2$. For each $j \in Z_p$ and all $m, n \in Z_4$ with $m < n$, the mapping that sends $(x, 0) \rightarrow (x, m)$ and $(x, 1) \rightarrow (x, n)$ is an isomorphism between the bichromatic quadruples in D_{0j} with colors 0 and 1, and the bichromatic quadruples in $D_{c(m,n)j}$ with colors m and n. In other words,

$$[(x,0),(y,0),(z,0),(t,1)] \in D_{0j} \text{ iff } [(x,m),(y,m),(z,m),(t,n)] \in D_{c(m,n)j},$$

$$[(x,0),(y,0),(z,1),(t,1)] \in D_{0j} \text{ iff } [(x,m),(y,m),(z,n),(t,n)] \in D_{c(m,n)j},$$

$$[(x,0),(y,1),(z,1),(t,1)] \in D_{0j} \text{ iff } [(x,m),(y,n),(z,n),(t,n)] \in D_{c(m,n)j}.$$

If $p = v$ then the sets of bichromatic quadruples in D_{0j} with colors 0 and 1 must form a set of v mutually 2-chromatic PDQs of order $2v$. Thus $3p = 3v$ is the maximum possible size of a set satisfying Property Y.

We now define a set of $(2^k - 1)p$ SQSs with point set $Z_v \times Z_2^k$. Let D_{ij} be a set of $3p$ PDQs of order $4v$ on the point set $Z_v \times Z_4$ with Property Y, and let B_i be the block set of the ith Boolean SQS of order 2^k. We define the block set S_{ij}, $i \in Z_2^k - \{0\}, j \in Z_p$, to be the union of the blocks of Types A and B defined below.

Type A.

$$[(x,q),(y,r),(z,s),(w,t)],$$

where

$$x+y+z+w \equiv j \pmod v \quad \text{and} \quad q+r+s+t = i, \qquad |\{q,r,s,t\}| = 4.$$

There are $v^3(b_{2^k} - \binom{2^{k-1}}{2})$ blocks of this type, i.e., v^3 blocks for each block of type (B.1) in B_i.

Type B. Let $[q, r, s, t]$ be a block of type (B.2) in B_i with $q < r < s < t$, and define $h(r) = 0$, $h(s) = 1$, and $h(t) = 2$. Since we have a block of type (B.2), we must have $q + i \in \{r, s, t\}$. Now define $g(0) = q$, $g(1) = r$, $g(2) = s$, and $g(3) = t$. For each block $[q, r, s, t]$ of type (B.2) in B_i, and for each block $[(x, a), (y, b), (z, c), (w, d)]$ in $D_{h(q+i)j}$ construct the block

$$[(x,g(a)),(y,g(b)),(z,g(c)),(w,g(d))].$$

There are $\binom{2^{k-1}}{2}(b_{4v} - 2b_{2v}) + 2^{k-1}b_{2v}$ blocks of Type B in S_{ij}, since the number of quadruples in $D_{h(q+i)j}$ is b_{4v}, but the 2-chromatic subsystems on $Z_v \times \{x, x+i\}$, $x \in Z_2^k$, (having b_{2v} blocks) only occur once, because of Property Y.3.

A little algebraic manipulation shows that each S_{ij} has $b_{2^k v}$ blocks.

Note that the blocks of Type B are the blocks of SQSs of order $4v$ on the points $Z_v \times \{q, r, s, t\}$, where $[q, r, s, t]$ is a block in (B.2) of the $(q + r)$th Boolean Steiner quadruple system. As noted above, $[q, r, s, t]$ is also a block of the $(q + s)$th and $(q + t)$th system, but in these three systems the values taken by $h(r)$, $h(s)$, and $h(t)$ are distinct members of Z_3.

To show that each S_{ij} is indeed an SQS we note that any 3-subset $T = \{(x, q), (y, r), (z, s)\}$ of $Z_v \times Z_2^k$ is contained in a unique block by the following arguments:

1) If $|\{q, r, s, q + r + s + i\}| = 4$ then $\{q, r, s\}$ is contained in a unique (B.1) block of the ith Boolean SQS and the block $[(x, q), (y, r), (z, s), (w, t)]$, where $x + y + z + w \equiv j (\text{mod } v)$ and $q + r + s + t = i$ is the unique block of Type A in S_{ij}, which contains T. Furthermore, no block of Type B contains T since no pair of members of $\{q, r, s\}$ sums to i.
2) If $|\{q, r, s\}| = 3$ and $q + r + s + i \in \{q, r, s\}$, say $q + r + s + i = s$ (which implies that $q + r = i$) then $\{q, r, s\}$ is contained in a unique (B.2) block of the ith Boolean SQS, namely $[q, r, s, s + i]$. Let t be the minimum element of this block. Now there is a unique block in $D_{h(t+i)j}$ that contains the 3-subset $\{(x, g^{-1}(q)), (y, g^{-1}(r)), (z, g^{-1}(s))\}$ and this generates a unique block of Type B in S_{ij} that contains T. Furthermore, no block of Type A contains T since $|\{q, r, s, q + r + s + i\}| < 4$.
3) If $|\{q, r, s\}| = 2$, say $r = s$, and $q + r \neq i$, then there is a unique (B.2) block of the ith Boolean SQS, namely $[q, q + i, r, r + i]$, that contains the pair $\{q, r\}$. Let t be the minimum element of this block. Now there is a unique block in $D_{h(t+i)j}$ that contains the 3-subset $\{(x, g^{-1}(q)), (y, g^{-1}(r)), (z, g^{-1}(r))\}$ and this generates a unique block of Type B in S_{ij}, which contains T. Furthermore, no block of Type A contains T since $|\{q, r, s\}| < 3$.
4) If $|\{q, r, s\}| = 2$, say $r = s$, and $q + r = i$, then there are $2^{k-1} - 1$ (B.2) blocks of the ith Boolean SQS that contain the pair $\{q, r\}$. However, each of these blocks generates only a single subdesign of order $2v$ on $Z_v \times \{q, r\}$ by property Y.3. This subdesign contains a unique block whose pre-image in D_{0j} under one of the isomorphisms contains the 3-subset $\{(x, 0), (y, 1), (z, 1)\}$ (in the case where $q < r$) or $\{(x, 1), (y, 0), (z, 0)\}$ (in the case where $q > r$). This generates a unique block of Type B in S_{ij}, which contains T. Furthermore, no block of Type A contains T since $|\{q, r, s\}| < 3$.
5) If $|\{q, r, s\}| = 1$, then the argument is similar to the previous case, considering the (B.2) blocks of the ith Boolean SQS that contain the pair $\{q, q + i\}$.

We argue the disjointness of distinct S_{ij}s as follows:

1) Two distinct S_{ij} can only intersect in blocks of the same type, since if $[(x, q), (y, r), (z, s), (w, t)]$ is a block of Type B with $|\{q, r, s, t\}| = 4$ then $q + r + s + t = 0$ and no block of Type A has this property. Furthermore, all blocks of Type A have four distinct second coordinates.
2) The disjointness of $\{x, y, z, w\}$, defined by $x + y + z + w \equiv j (\mathrm{mod}\, v)$, for different j's, and the disjointness of $\{q, r, s, t\}$, defined by $q + r + s + t = i$, for different i's guarantees the disjointness of the blocks of Type A.
3) The sets of blocks of Type B are disjoint since, by construction, no block is monochromatic. If $u = [(x, q), (y, r), (z, s), (w, t)]$ is a 2-chromatic block of Type B and say, $q \neq r$, then u can only be a member of S_{ij}, where $i = q + r$. This follows from the fact that all 2-chromatic blocks in any of the S_{ij} have color classes that sum to i. The uniqueness of the subscript j is then implied by the disjointness of the $D_{0j}, j \in Z_p$. Property Y.2 ensures that the only blocks of this form come from the induced subsystems of order $2v$.

   If $|\{q, r, s, t\}| > 2$ with say, $|\{q, r, s\}| = 3$, then u can only be a member of S_{ij}, where $i = q + r$ or $q + s$ or $r + s$. Again the uniqueness of the system S_{ij} containing u is guaranteed by the disjointness of the D_{nj}, $n \in Z_3, j \in Z_p$.

Hence, we have the following theorem.

THEOREM 2. *If there exists a set of* $3p$ PDQs *of order* $4v$ *with Property* Y *then* $D(2^k v) \geqq (2^k - 1)p$ *for all* $k \geqq 2$.

Since the $3v$ PDQs constructed in the previous section, and also those of Lindner [13], satisfy property Y, we have the following corollary.

COROLLARY 2. $D(2^k v) \geqq (2^k - 1)v$ *for* $v \equiv 2$ *or* $4 \pmod 6$, $v \geqq 8$, *for* $v = 5^a.13^b.17^c$, $a, b, c \geqq 0$, *and* $a + b + c > 0$, *for* $v = 11$, *and for all* $k \geqq 2$.

Finally, we note that if the set of $3v$ PDQs of order $4v$ is maximal then the set of $(2^k - 1)v$ PDQs of order $2^k v$ is also maximal.

4. Other recursive constructions. In this section we show that some of the recursive constructions for SQS can be utilized to give lower bounds on $D(v)$. The bounds we obtain are not very good, but aside from the constructions of Phelps [17], [18], and Phelps and Rosa [19] no other constructions are known that give nontrivial bounds on $D(v)$ when v is not divisible by 4.

The first construction we give is a version of the tripling construction in [7]. A quadruple system of order v with a *hole* of order s, denoted by SQS(v:s), is a triple (X, S, q), where X is a set of size v, S is a subset of X of size s, and q is a set of 4-subsets of X, called blocks, such that every 3-subset $T \subset X$ with $|T \cap S| < 3$ is contained in a unique block, and *no* 3-subset $T \subset S$ is contained in any block. Two systems (X, S, q_1) and (X, S, q_2) are disjoint if $q_1 \cap q_2 = \emptyset$. Let $D(v{:}s)$ denote the number of pairwise disjoint quadruple systems of order v with a hole of order s. (Note that $D(v{:}s) = D(v)$ since a 2-subset S contains no 3-subsets.)

Let $v = 6n + 2s$, let $q_1, q_2, \cdots, q_{D(v)}$ be a set of PDQs of order v, and let $r_1, r_2, \cdots, r_{D(v:s)}$ be a set of pairwise disjoint systems of order v with a hole of size s.

For $x \in Z_n = \{0, 1, \cdots, n - 1\}$, we define the notation $|x| = \min(x, n - x)$. Let $F_1, F_2, \cdots, F_{4n+s-1}$ be a one-factorization of the graph with vertex set Z_{6n+s} and edge set containing all pairs $\{x, y\}$ such that $x \equiv y \pmod 2$ or $|x - y| \geqq 2n + 1$. (This graph has a one-factorization by a result of Stern and Lenz [21], as do all other graphs that we factorize in this section.) Finally, let $\alpha(i, j)$ be a Latin square of order $4n + s - 1$ with symbol set $\{1, 2, \cdots, 4n + s - 1\}$, and let $S = \{\infty_0, \infty_1, \cdots, \infty_{s-1}\}$.

We now construct a set of $\min(D(v), D(v{:}s), (2v - s - 3)/3)$ PDQs of order $3v - 2s$ with point set $S \cup (Z_{6n+s} \times Z_3)$. (Note that $(2v - s - 3)/3 = 4n + s - 1$.) The block set of the jth SQS is constructed as follows.

Type 1a. Construct the blocks of q_j on the point set $S \cup (Z_{6n+s} \times \{0\})$.

Type 1b. Construct the blocks of r_j on the point sets $S \cup (Z_{6n+s} \times \{i\})$ for each $i = 1, 2$.

Type 2. For each $i = 0, 1, \cdots, s - 1$, and each $a, b, c \in Z_{6n+s}$ such that $a + b + c \equiv 6n + i + j \pmod{6n + s}$, construct the block

$$[\infty_i, (a,0), (b,1), (c,2)].$$

Type 3. For each $i = 0, 1, 2$ and each $a, b, c \in Z_{6n+s}$ such that $a + b + c \equiv 2ni + j \pmod{6n + s}$ and each $k = 0, 1, \cdots, n - 1$, construct the block

$$[(a+k,i), (a+2n-1-k,i), (b,i+1), (c,i+2)].$$

Type 4. For each $i = 0, 1, 2$ and each $k = 1, 2, \cdots, 4n + s - 1$ and each $\{a, b\} \in F_k$ and each $\{c, d\} \in F_{\alpha(j,k)}$ construct the block

$$[(a,i), (b,i), (c,i+1), (d,i+1)].$$

To prove that we have indeed constructed a set of PDQs of order $3v - 2s$ we note that the systems constructed can only intersect in blocks of the same type. Furthermore, as we verify that each system has a unique block containing each triple, we verify that this block is different for distinct values of j. This will establish the disjointness of the systems. We remark that the number of disjoint sets of blocks of Type 4 is limited by the number of rows in a Latin square of side $4n + s - 1$.

We now verify that the jth system has a unique block containing each 3-subset, T, of its point set.

1) If $T \subset S \cup (Z_{6n+s} \times \{0\})$ then T is contained in a unique block of Type 1a in q_j.
2) If $T \subset S \cup (Z_{6n+s} \times \{i\})$ with $i = 1$ or 2, and $T \not\subset S$, then T is contained in a unique block of Type 1b in r_j.
3) If $T = \{\infty_i, (a, k), (b, k + 1)\}$ then T is contained in a unique block of Type 2 whose fourth point is $(6n + i + j - a - b, k + 2)$.
4) If $T = \{(a, 0), (b, 1), (c, 2)\}$ then
   a) If $a + b + c = 6n + j + x$ with $0 \leqq x \leqq s - 1$, then T is contained in a unique block of Type 2 whose fourth point is ∞_x.
   b) If $a + b + c \neq 6n + j + x$ with $0 \leqq x \leqq s - 1$, then T is contained in a unique block of Type 3 whose fourth point is (d, i) where i is given by the interval $[j + 2ni, j + 2ni + 2n - 1]$ that contains $a + b + c$. Now we compute d as follows. Say $i = 0$ and $a + b + c = j + k$ $(0 \leqq k < n)$ then $d = a + 2n - 1 - 2k$. If $a + b + c = j + 2n - 1 - k$ then $d = a + 2k + 1 - 2n$. A similar argument works for $i = 1, 2$.
5) If $T = \{(a, i), (b, i), (c, i + 1)\}$ then
   a) If $|b - a| = 2k + 1$ for some $0 \leqq k < n$ then T is contained in a unique block of Type 3 whose fourth point is $(j + 2ni - c - a + k, i + 2)$, where $b - a = 2k + 1$.
   b) If $|b - a| \neq 2k + 1$ for some $0 \leqq k < n$ then T is contained in a unique block of Type 4 whose fourth point $(d, i + 1)$ is computed by noting that $\{a, b\}$ is in a unique one-factor, say F_m, and d is the second point of the unique edge containing c in the factor $F_{\alpha(j,m)}$.

6) If $T = \{(a, i), (b, i), (c, i - 1)\}$ then the argument is similar to the previous case.

This proves the following theorem.

THEOREM 3. *If* $s \equiv 2$ *or* $4 \pmod 6$ *and* $v \equiv 2s \pmod 6$ *then*

$$D(3v-2s) \geqq \min(D(v), D(v{:}s), (2v-s-3)/3).$$

When $s = 2$ this gives the inequality $D(3v - 4) \geqq \min(D(v), (2v - 5)/3)$ for all $v \equiv 4 \pmod 6$.

Using systems with holes in place of the blocks of Type 1a yields the inequality $D(3v - 2s{:}s) \geqq \min(D(v{:}s), (2v - s - 3)/3)$. Omitting the blocks of Type 1a gives the same lower bound for $D(3v - 2s{:}v)$.

We now show that there exists a pair of PDQs of order 38, each of which contains three subsystems of order 14 that intersect in two common points. This construction is vital to our proof of the Lindner and Rosa conjecture.

To simplify the discussion we introduce the notation $G(n, L)$ to denote the graph with vertex set Z_n and edges $\{x, y\}$ for all $|x - y| \in L$.

Example 1. We construct two disjoint SQS(38) with point set $\{\infty_0, \infty_1\} \cup (Z_{12} \times Z_3)$. The block set of the jth SQS(38) ($j = 0, 1$) is constructed as follows:

Type 1. Let q_0 and q_1 be two disjoint SQS(14). (It is known [1] that $D(14) \geqq 4$.) Construct the blocks of q_j on the point sets $\{\infty_0, \infty_1\} \cup (Z_{12} \times \{i\})$ for each $i \in Z_3$.

Type 2. Let $X(0) = (0, 3)$ and let $X(1) = (3, 6)$. For each $i = 0, 1$ and each $a, b, c \in Z_{12}$ such that $a + b + c \equiv x \pmod{12}$, and x is the ith coordinate of $X(j)$, construct the block

$$[\infty_i, (a, 0), (b, 1), (c, 2)].$$

Type 3. Let $P(i, 0) = \{\{6 + i, 9 + i\}\}$ and $P(i, 1) = \{\{8 + i, 11 + i\}\}$ for $i = 0, 1, 2$. For each $i = 0, 1, 2$ and each $a, b, c \in Z_{12}$ such that $a + b + c \equiv 0 \pmod{12}$, and $\{x, y\} \in P(i, j)$, construct the block

$$[(a+x, i), (a+y, i), (b, i+1), (c, i+2)].$$

Type 4. We first define two one-factorizations F^j_k of the graphs $G(12, \{1, 5, 6\})$ (for $j = 0$) and $G(12, \{2, 4, 6\})$ (for $j = 1$). Let F^0_0, F^0_1 be the one-factorization of $G(12, \{1\})$, let F^0_2, F^0_3 be the one-factorization of $G(12, \{5\})$, and let F^0_4 be the edge set of $G(12, \{6\})$. Let F^1_0, F^1_1, F^1_2 be the one-factorization of $G(12, \{4, 6\})$, defined by $F^1_i = \{\{3\varepsilon + i, 3\varepsilon + i + 4\}, \{3n + i + 2, 3n + i + 8\} : \varepsilon = 0, 1, 2, 3, n = 0, 1\}$, and let F^1_3, F^1_4 be either of the two one-factorizations of $G(12, \{2\})$.

For each $i = 0, 1, 2$ and each $k = 0, 1, \cdots, 4$ and each $\{a, b\} \in F^j_k$ and each $\{c, d\} \in F^j_{k+2}$ (addition mod 5 in the subscript) construct the block

$$[(a, i), (b, i), (c, i+1), (d, i+1)].$$

Type 5. Let $H_0 = \{\{1, 5\}, \{2, 4\}\}$ and $H_1 = \{\{2, 7\}, \{4, 5\}\}$. For each $i = 0, 1, 2$ and each $a \in Z_{12}$ and each $\varepsilon = 0, 1, 2, 3$ and each $\{x, y\} \in H_j$ construct the block

$$[(a, i+1), (a+3\varepsilon, i+2), (x-2a-3\varepsilon, i), (y-2a-3\varepsilon, i)].$$

Type 6. Let $D_0 = \{2, 4\}$ and $D_1 = \{1, 5\}$. For each $i = 0, 1, 2$ and each $a \in Z_{12}$ and each $\varepsilon = 0, 1, 2, 3$ and each $d \in D_j$ construct the block

$$[(a, i), (a+d, i), (a+3\varepsilon, i+1), (a+d+3\varepsilon, i+1)].$$

Verification that the systems constructed are both SQS(38)s is similar to the previous construction, and full details can be found in [8]. The disjointness of the systems is a little more complicated since there is the possibility of conflicts between blocks of Types 4 and 6. These conflicts are avoided by our careful construction of the one-factorizations and their ordering in the Type 4 quadruples.

We have obtained a generalization of this example using the tripling construction of [8] and obtained a proof of the following theorem.

THEOREM 4. *If* $s \equiv 2$ *or* $4 \pmod 6$ *and* $v \equiv s \pmod 6$ *then* $D(3v-2s) \geqq \min(D(v), D(v{:}s), 2)$.

This theorem is not necessary for our proof of the Lindner and Rosa conjecture, and since the proof is messy, we omit it. Theorem 4 can probably be strengthened to give the inequality $D(3v-2s) \geqq \min(D(v), D(v{:}s), f(v,s))$ for $v \equiv s \pmod 6$, and some function satisfying $(v-s)/6 \leqq f(v,s) \leqq (v-s)/3$. However, the proof of this inequality would be extremely tedious.

Colbourn and Hartman [2] have used the construction of Theorem 4 (with $v = 10$, $s = 4$, and omitting the blocks of Type 1a) to construct a pair of SQS(22:10) designs that intersect in precisely two blocks. Hartman and Yehudai [10] were able to modify this construction to produce a pair of disjoint SQS(22:10)s. The importance of this construction is that Colbourn and Hartman proved that if $D(22{:}10) \geqq 2$ then $D(v) \geqq 2$ for all $v \equiv 10 \pmod{12}$ with $v \geqq 46$. Thus, we have the following theorem.

THEOREM 5 (Colbourn, Hartman, and Yehudai). $D(v) \geqq 2$ *for all* $v \equiv 10 \pmod{12}$ *with* $v \geqq 46$.

We turn now to the quadrupling construction (construction 3.5 of [5]), and we will show that $D(4v-6) \geqq \min(D(v), (v-2)/2)$. Throughout this section we let $v = 2f+2$ and we will construct $\min(D(v), f)$ PDQs with point set $\{\infty_0, \infty_1\} \cup (Z_{2f} \times Z_4)$. A major ingredient of the construction is a partition of the edge set of K_{2f} into $2f$ parts $G_0, G_1, \cdots, G_{f-1}$ and $H_0, H_1, \cdots, H_{f-1}$ with the property that each H_i is a one-factor, and $G_i \cup \{\{2i, 2i+1\}\}$ is a one-factor for all $0 \leqq i < f$. These partitions were constructed by Hanani for all $f \geqq 1$ in [5]. The other ingredient is a standard one-factorization $F_0, F_1, \cdots, F_{2f-2}$ of K_{2f}. Let $q_1, q_2, \cdots, q_{D(v)}$ be a set of PDQs of order v. The block set of the jth SQS is constructed as follows:

Type 1. Construct the blocks of q_j on the point sets $\{\infty_0, \infty_1\} \cup (Z_{2f} \times \{i\})$ for each $i \in Z_4$.

Type 2. The other blocks containing ∞_0 and ∞_1 are the following:

$$\{[\infty_0,(a,0),(b,1),(c,2)]: a+b+c \equiv 2j \pmod{2f}\ (a,b,c) \equiv (0,0,0) \pmod 2\},$$

$$\{[\infty_0,(a,0),(b,1),(c,2)]: a+b+c \equiv 2j+1 \pmod{2f}\ (a,b,c) \equiv (1,1,1) \pmod 2\},$$

$$\{[\infty_0,(a,0),(b,1),(c,3)]: a+b+c \equiv 2j+2 \pmod{2f}\ (a,b,c) \equiv (1,0,1) \pmod 2\},$$

$$\{[\infty_0,(a,0),(b,1),(c,3)]: a+b+c \equiv 2j+1 \pmod{2f}\ (a,b,c) \equiv (0,1,0) \pmod 2\},$$

$$\{[\infty_0,(a,0),(b,2),(c,3)]: a+b+c \equiv 2j \pmod{2f}\ (a,b,c) \equiv (0,1,1) \pmod 2\},$$

$$\{[\infty_0,(a,0),(b,2),(c,3)]: a+b+c \equiv 2j+1 \pmod{2f}\ (a,b,c) \equiv (1,0,0) \pmod 2\},$$

$$\{[\infty_0,(a,1),(b,2),(c,3)]: a+b+c \equiv 2j+2 \pmod{2f}\ (a,b,c) \equiv (1,0,1) \pmod 2\},$$

$$\{[\infty_0,(a,1),(b,2),(c,3)]: a+b+c \equiv 2j+1 \pmod{2f}\ (a,b,c) \equiv (0,1,0) \pmod 2\},$$

$$\{[\infty_1,(a,0),(b,1),(c,2)]: a+b+c \equiv 2j \pmod{2f}\ (a,b,c) \equiv (1,1,0) \pmod 2\},$$

$$\{[\infty_1,(a,0),(b,1),(c,2)]: a+b+c \equiv 2j+1 \pmod{2f}\ (a,b,c) \equiv (0,0,1) \pmod 2\},$$

$$\{[\infty_1,(a,0),(b,1),(c,3)]: a+b+c \equiv 2j+2 \pmod{2f}\ (a,b,c) \equiv (0,1,1) \pmod 2\},$$

$\{[\infty_1,(a,0),(b,1),(c,3)]:a+b+c\equiv 2j+1\,(\mathrm{mod}\,2f)(a,b,c)\equiv(1,0,0)(\mathrm{mod}\,2)\}$,

$\{[\infty_1,(a,0),(b,2),(c,3)]:a+b+c\equiv 2j(\mathrm{mod}\,2f)(a,b,c)\equiv(0,0,0)(\mathrm{mod}\,2)\}$,

$\{[\infty_1,(a,0),(b,2),(c,3)]:a+b+c\equiv 2j+1\,(\mathrm{mod}\,2f)(a,b,c)\equiv(1,1,1)(\mathrm{mod}\,2)\}$,

$\{[\infty_1,(a,1),(b,2),(c,3)]:a+b+c\equiv 2j+2\,(\mathrm{mod}\,2f)(a,b,c)\equiv(1,1,0)(\mathrm{mod}\,2)\}$,

$\{[\infty_1(a,1),(b,2),(c,3)]:a+b+c\equiv 2j+1\,(\mathrm{mod}\,2f)(a,b,c)\equiv(0,0,1)(\mathrm{mod}\,2)\}$.

Type 3. Construct the following blocks using the Hanani factorization:

$\{[(a,0),(b,1),(x,2),(y,2)]:a\equiv b(\mathrm{mod}\,2)a+b+2c\equiv 2j(\mathrm{mod}\,2f)\{x,y\}\in G_c\}$,

$\{[(a,0),(b,1),(x,3),(y,3)]:a\not\equiv b(\mathrm{mod}\,2)a+b+2c\equiv 2j+1\,(\mathrm{mod}\,2f)\{x,y\}\in G_c\}$,

$\{[(a,2),(b,3),(x,0),(y,0)]:a\equiv b(\mathrm{mod}\,2)a+b+2c\equiv 2j(\mathrm{mod}\,2f)\{x,y\}\in G_c\}$,

$\{[(a,2),(b,3),(x,1),(y,1)]:a\not\equiv b(\mathrm{mod}\,2)a+b+2c\equiv 2j+1\,(\mathrm{mod}\,2f)\{x,y\}\in G_c\}$,

$\{[(a,0),(b,1),(x,2),(y,2)]:a\not\equiv b(\mathrm{mod}\,2)a+b+2c\equiv 2j+1\,(\mathrm{mod}\,2f)\{x,y\}\in H_c\}$,

$\{[(a,0),(b,1),(x,3),(y,3)]:a\equiv b(\mathrm{mod}\,2)a+b+2c\equiv 2j(\mathrm{mod}\,2f)\{x,y\}\in H_c\}$,

$\{[(a,2),(b,3),(x,0),(y,0)]:a\not\equiv b(\mathrm{mod}\,2)a+b+2c\equiv 2j+1\,(\mathrm{mod}\,2f)\{x,y\}\in H_c\}$,

$\{[(a,2),(b,3),(x,1),(y,1)]:a\equiv b(\mathrm{mod}\,2)a+b+2c\equiv 2j(\mathrm{mod}\,2f)\{x,y\}\in H_c\}$.

Type 4. Let $\alpha(i,j)$ be a Latin square of order $2f-1$ over the symbol set $\{0, 1, \cdots, 2f-2\}$, and construct the following blocks using the one-factorization:

$$\{[(a,0),(b,0),(x,1),(y,1)]:\{a,b\}\in F_i\{x,y\}\in F_{\alpha(i,j)}0\leqq i<2f-1\}$$

$$\{[(a,2),(b,2),(x,3),(y,3)]:\{a,b\}\in F_i\{x,y\}\in F_{\alpha(i,j)}0\leqq i<2f-1\}.$$

The full details of verification that the j = 0th system is actually an SQS are contained in Hanani's paper, and the verification for $j > 0$ is almost identical. The disjointness of the systems is guaranteed by the dependence of the constructions on the parameter j and can be easily verified. To assist the reader we give the verification argument for a few representative cases and note that the symmetries of the construction make the argument for the remaining cases a simple exercise.

If T is a 3-subset of $\{\infty_0, \infty_1\} \cup (Z_{2f}\times\{i\})$ for some $i \in Z_4$ then T is contained in a unique block of Type 1.

If $T = \{\infty_0, (a, 0), (b, 1)\}$ then according to the parities of a and b there is a unique block of Type 2 containing T.

If $T = \{(a, 0), (b, 1), (c, 2)\}$, then

1) If $a \equiv b \equiv i\,(\mathrm{mod}\,2)$ then
   a) If $a + b + c \equiv 2j\,(\mathrm{mod}\,2f)$ then the fourth point of the block of Type 2 containing T is ∞_i.
   b) If $a + b + c \equiv 2j + 1\,(\mathrm{mod}\,2f)$ then the fourth point of the block of Type 2 containing T is ∞_{1-i}.
   c) If $a + b + c \not\equiv 2j$ or $2j + 1\,(\mathrm{mod}\,2f)$ then the fourth point, $(d, 2)$, of the block of Type 3 containing T is computed as follows: Let x be the unique solution to the equation $a + b + x \equiv 2j\,(\mathrm{mod}\,2f)$. Now x is even, and there is a unique edge $\{c, d\}$ in $G_{x/2}$ that contains c since, by assumption $c \neq x$ or $x + 1$.

2) If $a \not\equiv b \pmod 2$ then the fourth point, $(d, 2)$, of the block of Type 3 containing T is given by the unique edge $\{c, d\}$ in $H_{x/2}$ that contains c, where x is the unique solution to the equation $a + b + x \equiv 2j + 1 \pmod{2f}$.

If $T = \{(a, 0), (b, 0), (c, 2)\}$ then the edge $\{a, b\}$ is contained either in G_x or H_x for some x, and the fourth point $(d, 3)$, of the block of Type 3 containing T is given by the solution to $2x + c + d \equiv 2j \pmod{2f}$, (if $\{a, b\} \in G_x$) or $2x + c + d \equiv 2j + 1 \pmod{2f}$, (if $\{a, b\} \in H_x$).

If $T = \{(a, 0), (b, 0), (c, 1)\}$ then the edge $\{a, b\}$ is contained in a unique one-factor F_x and the fourth point $(d, 1)$ of the block of Type 4 containing T is given by the unique edge $\{c, d\}$ in $F_{\alpha(x,j)}$ that contains c.

We have thus indicated the proof of the following theorem.

THEOREM 6. *If* $v \equiv 2$ *or* $4 \pmod 6$ *then* $D(4v - 6) \geqq \min (D(v), (v - 2)/2)$.

Theorem 6 can certainly be generalized to give the inequality $D(4v - 3s) \geqq \min (D(v), D(v{:}s), (v - s)/2)$ but, again, the proof is tedious.

We now apply the singular direct product construction for quadruple systems (Proposition 8 of [6]) to obtain other recursive bounds on $D(v)$ as follows.

THEOREM 7. *If* $n \equiv 1$ *or* $3 \pmod 6$ *and* $v \equiv 4 \pmod 6$ *then* $D(n(v - 2) + 2) \geqq \min (D(v), (2v - 5)/3)$.

Proof. Let $(Z_n \cup \{\infty\}, q)$ be an SQS of order $n + 1$. We construct the jth quadruple system on the point set $\{\infty_0, \infty_1\} \cup (Z_{v-2} \times Z_n)$ as follows: For each point $x \in Z_n$ construct the blocks of the jth quadruple system in a set of PDQs of order v on the point set $\{\infty_0, \infty_1\} \cup (Z_{v-2} \times \{x\})$. For each block $[\infty, x, y, z] \in q$ that contains ∞ use the construction of Theorem 3 on the point set $\{\infty_0, \infty_1\} \cup (Z_{v-2} \times \{x, y, z\})$, omitting the blocks of Type 1. For each block $[x, y, z, t] \in q$ that does not contain ∞ construct the blocks:

$$\{[(a,x),(b,y),(c,z),(d,t)]: a+b+c+d \equiv j \pmod{v-2}\}.$$

Theorem 7 implies, for example, that $D(142) \geqq 11$ and $D(302) \geqq 11$ using $v = 22$ and $n = 7$ and 15, respectively.

A similar proof also gives the following generalization of Construction 3.6 of [5].

THEOREM 8. *If* $n \equiv 1$ *or* $3 \pmod 6$ *then* $D(12n + 2) \geqq 2$.

The proof of Theorem 8 is identical to the proof of the previous theorem, using the systems of order 38 constructed in Example 1 in place if the systems constructed in Theorem 3.

We are now in a position to prove the Lindner and Rosa conjecture.

THEOREM 9. $D(v) \geqq 2$ *for all* $v \equiv 2$ *or* $4 \pmod 6$, $v \geqq 8$.

Proof. If $v \equiv 4$ or $8 \pmod{12}$ and $v \geqq 16$ then Lindner [12] proved that $D(v) \geqq v/2$; furthermore, it is well known that $D(8) = 2$. If $v \equiv 10 \pmod{12}$ and $v \geqq 46$, the result follows from Theorem 5; the remaining values $v \in \{10, 22, 34\}$ are covered by [11], [18], and [19]. If $v \equiv 2 \pmod{24}$ then $8 \leqq (v + 6)/4 \equiv 2 \pmod 6$ and the result follows from Theorem 6 and the induction hypothesis. If $v \equiv 14$ or $38 \pmod{72}$ then $v = 12n + 2$ for some $n \equiv 1$ or $3 \pmod 6$ and the result follows from Theorem 8. Finally, if $v \equiv 62 \pmod{72}$ then $22 \leqq (v + 4)/3 \equiv 4 \pmod 6$ and the result follows from Theorem 3 with $s = 2$, and the induction hypothesis.

One final result for improving the known bounds on $D(v)$ uses the notion of an $H(m, g, k, t)$ design. An $H(m, g, k, t)$ design is a triple $(X, \mathbf{G}, \mathbf{B})$, where X is a set of points whose cardinality is mg, and $\mathbf{G} = \{G_1, G_2, \cdots, G_m\}$ is a partition of X into m sets of cardinality g; the members of $\mathbf{G}$ are called *groups*. A *transverse* of $\mathbf{G}$ is a subset of X that meets each group in at most one point. The set $\mathbf{B}$ contains k-element transverses of **G**, called *blocks*, with the property that each t-element transverse of **G** is contained in precisely one block.

When $g = 1$ then an $H(m, 1, k, t)$ is just a Steiner system $S(t, k, m)$ and when $k = m$ then an $H(k, g, k, t)$ is equivalent to an $OA(t, k, g)$.

The technique for enlarging the group size of an H-design gives sets of (block) disjoint designs as follows.

LEMMA 3. *If there exists an* $H(m, g, 4, 3)$ *then there exist* n *disjoint* $H(m, ng, 4, 3)$ *designs.*

Proof. If $(X, \mathbf{G}, \mathbf{B})$ is an $H(m, g, 4, 3)$ design then form the new designs on the point set $X \times Z_n$, with groups $G_i \times Z_n$ and, for each block $\{x, y, z, t\} \in \mathbf{B}$, form the n^3 blocks $[(x, a), (y, b), (z, c), (t, d)]$, where $a + b + c + d \equiv j \pmod n$. Letting j range over Z_n gives n disjoint designs.

This lemma is used in the proof of the following theorem.

THEOREM 10. *If there exists an* $H(m, g, 4, 3)$ *and* $ng \equiv 2$ *or* $4 \pmod 6$ *then* $D(nmg) \geqq \min(D(ng), n)$.

Proof. Let $\mathbf{B}_1, \mathbf{B}_2, \cdots, \mathbf{B}_n$ be the block sets of n disjoint $H(m, ng, 4, 3)$ designs, let $q_1, q_2, \cdots, q_{D(ng)}$ be the block sets of PDQs of order ng, and let $F_1, F_2, \cdots, F_{ng-1}$ be a one-factorization of K_{ng}. Finally, let $\alpha(j, k)$ be a Latin square of side $ng - 1$. (In fact, it is sufficient to have an $(ng - 1) \times D(ng)$ Latin rectangle.)

The block set of the j-th SQS consists of $\mathbf{B}_j$, and a copy of q_j on each of the groups of the $H(m, ng, 4, 3)$. A final group of blocks is given by constructing a copy of the one-factorization on each of the groups, and for each pair of distinct groups G_x and G_y $(x < y)$ forming the blocks $\{a, b, c, d\}$, where $\{a, b\}$ is a member of F_k on G_x and $\{c, d\}$ is a member of $F_{\alpha(j,k)}$ on G_y.

We can apply Theorem 10 to give new bounds on $D(v)$ if, for example, we have an $H(m, 2, 4, 3)$ with $m \equiv 1$ or $5 \pmod 6$. Hartman, Mills, and Mullin [9] have shown

TABLE 1
The best known lower bounds for $D(v)$.

v	$D(v) \geq$	Reference	v	$D(v) \geq$	Reference	v	$D(v) \geq$	Reference
4	1	trivial	38	2	Example 1	70	5	Thm. 10
8	2	folklore	40	35	Thm. 2	74	9	Thm. 6
10	5	[11]	44	33	Thm. 1	76	40	[4]
14	4	[1]	46	2	Thm. 5	80	75	Thm. 2
16	8	[12]	50	25	[17]	82	10	Thm. 6
20	15	Thm. 1	52	39	Thm. 1	86	2	Thm. 8
22	11	[18]	56	42	[13]	88	77	Thm. 2
26	13	[19]	58	7	Thm. 6	92	48	[4]
28	18	[4]	62	11	Thm. 3	94	2	Thm. 5
32	24	[13]	64	56	Thm. 2	98	12	Thm. 6
34	17	[19]	68	51	Thm. 1	100	75	Thm. 1

that there exists an $H(m, 2, 4, 3)$ for all $m \equiv 1$ or $7 (\mathrm{mod}\ 18)$. So we have $D(70) \geqq 5$ using $m = 7$, $g = 2$, and $n = 5$ in Theorem 10.

Mills [16] has also constructed the designs $H(11, 2, 4, 3)$ and $H(13, 2, 4, 3)$. Using the techniques of [9] and [15] together with an $H(13, 2, 4, 3)$ one can show the existence of $H(m, 2, 4, 3)$ for all $m \equiv 1 \ (\mathrm{mod}\ 6)$. Thus Theorem 10 is applicable to all orders $v = 2nm$ with $m \equiv 1 \ (\mathrm{mod}\ 6)$ and $n > 1$. The cases where $m \equiv 5 (\mathrm{mod}\ 6)$ are more complicated, since no $H(5, 2, 4, 3)$ exists.

Conclusions. Two major open problems, originally posed by Lindner and Rosa [14] in 1978, have been tackled here. The first problem is the construction of a large set of PDQs of some order v. We have shown that one can get $v - 5$ PDQs of order $v = 5.2^n$ $n \geqq 1$. We have also shown that $D(v) \geqq (1 - \varepsilon)v$ for infinitely many v and any $\varepsilon > 0$. Unfortunately, the existence of a large set still remains an open problem.

The second problem is to show that $D(v) \geqq 2$ for all admissible values of $v \geqq 8$. We have solved this problem, and in many cases we have given even better lower bounds on $D(v)$. The state of the art for $v \leqq 100$ is given in Table 1.

REFERENCES

[1] A. E. Brouwer, J. B. Shearer, N. J. A. Sloane, and W. D. Smith, *A new table of constant weight codes*, IEEE Trans. on Inform. Theory, IT-36 (1990), pp. 1334–1380.

[2] C. J. Colbourn and A. Hartman, *Intersections and supports of quadruple systems*, Annals of Discrete Math., to appear.

[3] J. Doyen and M. Vandensavel, *Non-isomorphic Steiner quadruple systems*, Bull. Soc. Math. Belg., 23 (1971), pp. 393-410.

[4] T. Etzion, *Partitions for quadruples*, preprint.

[5] H. Hanani, *On quadruple systems*, Canad. J. Math., 12 (1960), pp. 145-157.

[6] ———, *On some tactical configurations*, Canad. J. Math., 15 (1963), pp. 702-722.

[7] A. Hartman, *Tripling quadruple systems*, Ars Combin., 10 (1980), pp. 255-309.

[8] ———, *A general recursive construction for quadruple systems*, J. Combin. Theory Ser. A, 33 (1982), pp. 121-134.

[9] A. Hartman, W. H. Mills, and R. C. Mullin, *Covering triples by quadruples: an asymptotic solution*, J. Combin. Theory Ser. A, 41 (1986), pp. 117-138.

[10] A. Hartman and Z. Yehudai, *Intersections of Steiner quadruple systems*, Discrete Math., to appear.

[11] E. S. Kramer and D. M. Mesner, *Intersections among Steiner systems*, J. Combin. Theory Ser. A, 16 (1974), pp. 272-285.

[12] C. C. Lindner, *A note on disjoint Steiner quadruple systems*, Ars Combin., 3 (1977), pp. 271-276.

[13] ———, *On the construction of pairwise disjoint Steiner quadruple systems*, Ars Combin., 19 (1985), pp. 153-156.

[14] C. C. Lindner and A. Rosa, *Steiner quadruple systems—A survey*, Discrete Math., 22 (1978), pp. 147-181.

[15] W. H. Mills, *On the covering of triples by quadruples*, Congr. Numer., 10 (1974), pp. 563-581, and in Proc. 5th South Eastern Conference of Combinatorics Graph Theory and Computing, Boca Raton, 1974.

[16] ———, personal communication.

[17] K. T. Phelps, *A construction of disjoint Steiner quadruple systems*, Congr. Numer., 19 (1977), pp. 559-567, and in Proc. 8th South Eastern Conference on Combinatorics Graph Theory and Computing, Baton Rouge, 1977.

[18] ———, *A class of* 2-*chromatic* SQS(22), Discrete Math., to appear.

[19] K. T. Phelps and A. Rosa, 2-*Chromatic Steiner quadruple systems*, European J. Combin., 1 (1980), pp. 253-258.

[20] D. Raghavarao, *Constructions and Combinatorial Problems in the Design of Experiments*, John Wiley, New York, 1971.

[21] G. Stern and H. Lenz, *Steiner triple systems with given subspaces; another proof of the Doyen-Wilson theorem*, Boll. Un. Mat. Ital. A(5), 17 (1980), pp. 109-114.

SIAM J. DISC. MATH.
Vol. 4, No. 2, pp. 196–199, May 1991

MAXIMAL INDEPENDENT SUBSETS IN STEINER SYSTEMS AND IN PLANAR SETS*

ZOLTÁN FÜREDI†

Abstract. A set of points is independent if there are no three on a line. It is proved that $\Omega(\sqrt{n \log n}) < \alpha(n) < o(n)$, where $\alpha(n)$ denotes the maximum α such that every planar set of n points with no four on a line contains an independent subset of size α.

Key words. planar subsets, Steiner systems, linear hypergraphs, polarized set mappings

AMS(MOS) subject classifications. 51A20, 05B40, 05C55

1. Independent planar sets.

1. Independent planar sets. A set of points on the Euclidean plane is called *independent* if there are no three on a line among them. Denote the size of the largest independent subset of the set S by $\alpha(S)$, and let $\alpha(n) = \min\{\alpha(S) : |S| = n$ and S does not contain more than three points on a line$\}$. These sets S are briefly called 3-*independent*.

Erdös [E] proposed in several places the problem to determine or to give bounds for $\alpha(n)$. An old result of Erdös and Hajnal [EH] (or in other words, the greedy algorithm) implies that $\alpha(S) \geqq \lfloor \sqrt{2n} \rfloor$ for every n-element 3-independent set S. Obviously, the sequence $\alpha(n)/n$ is monotone decreasing, $\lim \alpha(n)/n$ exists. Erdös remarked that every known construction S contains at least $|S|/3$ independent points. The aim of this note is to improve both estimates using deep combinatorial theorems.

THEOREM 1.1. *There is a positive constant c such that* $c\sqrt{n \log n} < \alpha(n)$ *holds for all n. On the other hand* $\lim \alpha(n)/n = 0$ *whenever n tends to infinity.*

1.1. A construction giving $\alpha(n) = o(n)$. Let $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_t$ be distinct unit vectors linearly independent over the rationals with the property that if the directions of two integer linear combinations are the same, then the two vectors are essentially the same, i.e.,

$$\text{(P)} \qquad \sum a_i\mathbf{v}_i = \gamma\left(\sum b_i\mathbf{v}_i\right) \quad \text{and} \quad a_i, b_i \in \mathbf{N} \text{ implies } \gamma \in \mathbf{Q}.$$

Note that the linear independence of the system $\{\mathbf{v}_i\}$ implies that $a_i = \gamma b_i$. Define S^t as the set of 3^t integer linear combinations of $\{\mathbf{v}_i\}_{i=1}^t$ with coefficients 0, 1, or 2. Property (P) implies that there are no four points of S^t on a line, and the density version of the Hales–Jewett theorem, recently proved by Fürstenberg and Katznelson [FK], implies that $\alpha(S^t) = o(3^t)$ whenever t tends to infinity.

For completeness we recall the theorem applied above. For every positive ε there exists a $t(\varepsilon)$ such that if $I \subset \{0, 1, 2\}^T$, $|T| = t > t(\varepsilon)$, and $|I| > \varepsilon 3^t$, then we can find three sequences $s_i \in I$ $(i = 0, 1, 2)$ and a partition of the coordinate set T, $T = C \cup V$ such that $(s_i)_v = i$ in all coordinates $v \in V$, and $(s_0)_c = (s_1)_c = (s_2)_c$ holds for $c \in C$.

2. Independent sets in Steiner triple systems. A *hypergraph*, $\mathbf{H}$, is a pair $(V, \mathscr{E})$, where $\mathscr{E}$ is a family of subsets of V. The elements of V are called *vertices*, the $E \in \mathscr{E}$ are called hyperedges. A hypergraph is called *linear* or (almost disjoint) if $|E \cap E'| \leqq 1$ holds for all distinct $E, E' \in \mathscr{E}$. A *cycle* of length k is a sequence of distinct vertices and edges $x_1, E_1, x_2, E_2, \cdots x_k, E_k$ $(x_i \in V, E_i \in \mathscr{E})$ such that $\{x_i, x_{i+1}\} \subset E_i$ $(x_{k+1} = x_1$, by definition). $(V, \mathscr{E})$ has *girth* at least g if it has no cycles of length $2, 3, \cdots, g-1$.

[*] Received by the editors December 2, 1988; accepted for publication December 18, 1989.

[†] Mathematical Institute of the Hungarian Academy of Sciences, 1364 Budapest, P. O. B. 127, Hungary.
This research was partially supported by the Hungarian National Science Foundation grant 1812.

Namely, the girth is at least 3 if and only if the hypergraph is linear. For the $x \in V$ we set $\mathscr{E}[x] = \{E : x \in E \in \mathscr{E}\}$. The *degree*, deg $(\mathbf{H}, x)$ or briefly deg (x), is the cardinality of $\mathscr{E}[x]$. The average degree $\bar{d}$ is the mean value of the degrees, i.e.,

$$\bar{d} = \frac{1}{|V|} \sum_{x \in V} \deg(x).$$

A set $I \subset V$ is *independent* if I contains no hyperedges $E \in \mathscr{E}$. $\alpha(\mathbf{H})$ denotes the maximum cardinality of an independent set. The *restriction* of $(V, \mathscr{E})$ to a subset W, denoted by $\mathbf{H} | W$, is given by $(V \cap W, \mathscr{E} | W)$, where $\mathscr{E} | W = \{E \in \mathscr{E} : E \subset W\}$.

A linear hypergraph $\mathbf{S}$ having 3-element hyperedges is called a partial Steiner triple system. $\mathbf{S}$ is a Steiner system, $S(|V|, 3, 2)$, if every pair of its vertices is contained in a (unique) three-tuple. The following result is due to Komlós, Pintz, and Szemerédi [KPSz]. There exists a positive constant c_3 such that if $\mathbf{S}$ is a partial Steiner family of girth at least 5 over n vertices, with average degree $\bar{d} \leqq d < n^{0.2} (d > c_4)$, then

$$\alpha(S) > c_3 n \sqrt{\frac{\log d}{d}}. \tag{2.1}$$

As a corollary of (2.1) Phelps and Rödl [PR] obtained a solution of a problem of Erdös and Hajnal ([EH66], see also, e.g., [E69, Prob. 19]), the true order of magnitude of the size of an independent set in a partial Steiner triple system. Namely, they proved that for every partial Steiner triple system $S = (V, \mathscr{E})$ with $|V| = n$ there exists a subset $I \subset V$ of size

$$|I| \geqq c_5 \sqrt{n \log n} \tag{2.2}$$

containing no edges from $\mathscr{E}$. Here c_5 is an absolute constant not depending on n. The proof of (2.2) utilizes the probabilistic method.

2.1. Large independent subsets of a planar set. Here we prove the lower bound in (1.1). Suppose that S is 3-independent planar set. Define $\mathscr{S}$ as the family of triples of S whose 3 points lie on a line. Obviously, $|A \cap B| \leqq 1$ holds for all distinct $A, B \in \mathscr{S}$, i.e., $\mathscr{S}$ is a partial Stiener triple system. Theorem 2.2 says that every partial Steiner system $\mathscr{S}$ on S contains a set $I \subset S$, $|I| > c_5 \sqrt{n \log n}$ such that $A \not\subset I$ for all $A \in \mathscr{S}$. Then the points of I are independent.

2.2. A remark on polarized set mappings. We can think that the above application of the Komlós, Pintz, Szemerédi theorem also answers the following question of Erdös and Hajnal ([EH58], see also [E69 Prob. 20]). Let V be the set of the first n positive integers. Let f be a function from the pairs of V to V such that $f(P) \notin P$. A set $I \subset V$ is said to be independent if for any $i \in I, j \in I, f(i, j) \notin I$. Denote by $g(n)$ the minimum of the largest independent set where the minimum is taken over all functions $f(i, j)$. Erdös and Hajnal proved that

$$c_6 n^{1/3} < g(n) < c_7 \sqrt{n \log n}.$$

Here the lower bound is obtained by the greedy algorithm, and the upper bound is the mean value of $g(f)$ whenever $f(i, j)$ is chosen independently and uniformly from $V \setminus \{i, j\}$.

Although the system of triples $\{(i, j, f(i, j)) : i, j \in V\}$ looks very much alike a Steiner family, the next theorem shows that the true order of the magnitude of $g(n)$ is not $\sqrt{n \log n}$. Hence this yields another example for the necessity of the constraint of the large girth in (2.1).

THEOREM 2.3.

$$\frac{2\sqrt{3}}{9}\sqrt{n} < g(n) < 2\sqrt{n}.$$

Proof. The lower bound is a special case of a theorem of Spencer [S], which says that there exists an absolute constant c_8 such that every 3-hypergraph $\mathbf{H}$ over n vertices with average degree d contains an independent set of size at least

$$\alpha(\mathbf{H}) > c_8 \frac{n}{\sqrt{d}}. \tag{2.4}$$

This inequality is a weaker but more general version of (2.1).

The upper bound is given by the following example f. Let $V_1 \cup \cdots \cup V_a$ be a partition of V, where $a = \lceil \sqrt{n} \rceil$, and $V_i = \{(i, 1), (i, 2), \cdots, (i, b_i)\}$, where $\lfloor \sqrt{n} \rfloor = b_a \geqq b_{a-1} \geqq \cdots \geqq b_1$. For $i < j$, $x \neq y$, $y \leqq b_j$ let

$$f((i,x),(j,y)) = (j,x),$$

and otherwise define f arbitrarily. Let I be an f-independent set and let us denote the projection of $V_i \cap I$ by I_i, i.e., $I_i = \{x : (i, x) \in I\}$. Then $I_i \cap I_j \neq \varnothing$, $|I_j| > 1$ is impossible, implying $|I| \leqq \lceil \sqrt{n} \rceil + \lfloor \sqrt{n} \rfloor - 1$. □

3. Problems. More generally, for $2 \leqq i < k$ let $\alpha_k^{(i)}(n) = \min\{\alpha^{(i)}(S) : |S| = n, |S \cap l| \leqq k$ for all lines $l\}$, where $\alpha^{(i)}(S)$ denotes the size of the largest i-independent subset. Define $\beta_k^{(i)}(n)$ as the smallest integer s such that every partial Steiner k-family over n elements has an i-independent set of size s. Clearly $\beta \leqq \alpha$.

The above proof, with a simple generalization of Spencer's theorem [S] (i.e., for every k-hypergraph has an i-independent set of size at least $O(n/\sqrt[i]{d})$) and the general Fürstenberg's theorem, give that for all fixed k we have

$$\beta_k^{(2)}(n) \leqq \beta_k^{(3)}(n) \leqq \cdots \beta_k^{(k-1)}(n) \leqq a_k^{(k-1)}(n) = o(n),$$

$$\Omega(n^{(i-1)/i}) < \beta_k^{(i)}(n).$$

$$\Omega(\sqrt{n \log n}) < \beta_k^{(2)}(n).$$

A theorem of Ajtai, Komlós, Pintz, Spencer, and Szemerédi [AKPSSz] implies that the second inequality is not sharp, i.e., $\lim \beta_k^{(i)}(n) n^{-(i-1)/i} = \infty$, and probably their methods give that

$$\Omega(n^{(i-1)/i}(\log n)^{1/i}) \leqq \beta_k^{(i)}(n). \qquad (?)$$

We can conjecture that the true order of magnitude of $\alpha(n)$ is much closer to the upper bounds because it is very difficult to realize a Steiner triple system on the plane (although there are 3-independent n-sets with $(n^2/6) - O(n)$ collinear triples. See [BGS] or an elementary construction in [FP].).

It also seems interesting to investigate the higher dimensional versions of this problem.

Acknowledgment. The author is indebted to Noga Alon for his valuable remarks.

REFERENCES

[AKPSSz] M. AJTAI, J. KOMLÓS, J. PINTZ, J. SPENCER, AND E. SZEMERÉDI, *Extremal uncrowded hypergraphs*, J. Combin. Theory Ser. A, 32 (1982), pp. 321–335.

[BGS] S. BURR, B. GRÜNBAUM, AND N. J. A. SLOAN, *The orchard problem*, Geom. Dedicata, 2 (1974), pp. 397–424.

[E69] P. Erdös, *Some unsolved problems in graph theory and combinatorial analysis*, in Combinatorial Mathematics and Its Applications (Proc. Conf., Oxford, 1969), Academic Press, London, New York, 1971, pp. 97–109.

[E] P. Erdös, private communication.

[EH58] P. Erdös and A. Hajnal, *On the structure of set mappings*, Acta Math. Acad. Sci. Hungar., 9 (1958), pp. 111–131.

[EH66] ———, *On chromatic number of graphs and set-systems*, Acta Math. Acad. Sci. Hungar., 17 (1966), pp. 61–99.

[FP] Z. Füredi and I. Palásti, *Arrangements of lines with a large number of triangles*, Proc. Amer. Math. Soc., 92 (1984), pp. 561–566.

[FK] H. Fürstenberg and Y. Katznelson, *A density version of the Hales–Jewett theorem for k = 3*, Discrete Math., 75 (1989), pp. 227–241.

[KPSz] J. Komlós, J. Pintz, and E. Szemerédi, *A lower bound for Heilbronn's conjecture*, J. London Math. Soc., 25 (1982), pp. 13–24.

[PR] K. T. Phelps and V. Rödl, *Steiner triple systems with minimum independence number*, Ars Combin., 21 (1986), pp. 167–172.

[S] J. Spencer, *Turán's theorem for k-graphs*, Discrete Math., 2 (1972), pp. 183–186.

SIAM J. DISC. MATH.
Vol. 4, No. 2, pp. 200–206, May 1991

LYAPUNOV FUNCTIONALS FOR AUTOMATA NETWORKS DEFINED BY CYCLICALLY MONOTONE FUNCTIONS*

ERIC GOLES† AND SERVET MARTINEZ†

Abstract. For automata networks with dynamics given by $x(t+1)=f(Ax(t)-b)$, where A is symmetric and f is cyclically monotone, it is proved that there exists a Lyapunov functional $(H(x(t)))_{t\geqq 1}$ that increases with t along the orbit. This allows one to study the finite limit cycles (which are of length one or two), the transient behavior in the finite case, and also to write down explicit integral functionals. The existence of $(H(x(t)))$ is important for a physical approach to the study of these automata networks. The condition that f be cyclically monotone in $\mathbb{R}$ is equivalent to f increasing and is usually used in applications to computer sciences.

Key words. automata networks, Lyapunov functionals, cyclically monotone functions, transient behavior, periods

AMS(MOS) subject classifications. 68Q80, 68R99

1. Introduction. Automata networks were originally introduced by von Neumann [12] and McCulloch [7] for modeling self-reproducing systems and neural activity. Subsequently, they have been used as models in various fields, including computer sciences [2], [5], [8] and physics [1], [13]. In this context some interesting applications have been developed recently in pattern recognition, associative memories, and the spin glass model of condensed matter (for recent reviews see [1]).

Since an automata network is a discrete dynamical system (in time and space), it is very difficult to give characterizations of its dynamics, and so mathematical results are scarce, and usually they have been obtained in very particular cases (regular arrays, extremely simple local rules, etc.) by nongeneralizable combinatorial arguments.

The class of networks we will study is defined, roughly speaking, by symmetric cellular spaces and local functions that are cyclically monotone. More precisely, its evolution is given by $x(t+1)=f(Ax(t)-b)$, with $x(t)\in\mathbb{R}^n$, $b\in\mathbb{R}^n$, A a symmetric matrix, f cyclically monotone, and t an integer. This kind of network has applications in problems discussed above. Its periodic behavior has been studied in [9] and [10] by a rather combinatorial algebraic approach.

Here we will take a physical approach to the problem. We will prove the existence of Lyapunov functionals $(H(x(t)))_{t\geqq 1}$ for the above class of networks (Theorem 1). From a mathematical point of view the proof of the existence of the functional $(H(x(t)))$ supplied in this work is an extension of the same result we have found in [3] and [4] for "positive" local function f (which means $\langle f(x)-f(y), x\rangle \geqq 0$ for all x, y). This extension allows us to study a wide class of dynamical systems; it means in the real case that instead of the sign type functions $f(x) = h(\text{sign}\,(x))$, where $h(-1)\leqq h(0)\leqq h(1)$ (which are the "positive" functions in $\mathbb{R}$), we can now deal with any increasing function f (which are the cyclically monotone functions in $\mathbb{R}$; see Lemma 2). That is, we are able to explain the dynamics of a wide class of systems, which include some of those defined on a continuous state space. For instance, $f(x)=\tanh(x)$ for $x\in[-a, a]$, $\tanh(a)$ if

* Received by the editors February 10, 1987; accepted for publication (in revised form) June 27, 1990. This research was partially financed by Fondo Nacional de Ciencias 554-88, 553-88, and by grants TWAS/8748 and TWAS/8773.

† Departamento de Ingeniería Matemática, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, Casilla 170/3, correo 3, Santiago, Chile.

$x < a$, $-\tanh(a)$ if $x < -a$, which is used in applications concerning smoothing techniques [4], [6].

In Theorems 2–4 we characterize the limit cycle lengths T when the local function f is strictly cyclically monotone, and we prove $T \leqq 2$. When A is positive definite we show there are only fixed points, $T = 1$. If A is negative definite and $f(-b) = 0$, we prove $T = 2$, except for the only fixed point $x = 0$. In Corollaries 3–5, we establish these results for real networks where the strictness condition on f has been weakened.

When the state space $Q = f(\mathbb{R}^n)$ is finite, we deduce in Corollary 6 a bound for the transient length from the existence of $(H(x(t)))$. Furthermore, for algorithms that use this class of networks [2], [5]–[8] the existence of increasing functionals allows us to study convergence properties.

We must remark that the existence of Lyapunov functionals is relevant from the physical point of view. Moreover, when we write our functional for continuous local function f we get results analogous to those obtained in [6] (see Corollary 2).

2. Definitions and preliminary results.

2.1. The cyclically monotone property. The potential. In this section we define the class of local functions that will be used in the automata network. Characterizations and examples will be given.

Let X be a real vector space with scalar product $\langle , \rangle : X \times X \to \mathbb{R}$ and let $f: X \to X$. We say f is *cyclically monotone* (C.M.) if and only if

$$(1) \qquad \sum_{i=0}^{n-1} \langle u_i - u_{i+1}, f(u_i) \rangle \geqq 0, \quad \forall n \geqq 2, \quad \forall (u_0, \cdots, u_{n-1}, u_n = u_0) \in X^{n+1}.$$

If f is C.M., then it is monotone, that is:

$$(2) \qquad \langle f(u) - f(v), u - v \rangle \geqq 0 \quad \forall u, v \in X.$$

To see this, it is sufficient to take $n = 2$ and the sequence (u, v, u).

Furthermore, we will say that f is *strictly cyclically monotone* (S.C.M.) if and only if it is C.M. and strictly monotone; this last means that it is monotone and $\langle f(u) - f(v), u - v \rangle = 0$ if and only if $f(u) = f(v)$.

From Rockafellar [11] we have that f is C.M. if and only if there exists a proper *convex* function $g: X \to \mathbb{R}$ such that f is a subgradient of g; i.e.,

$$(3) \qquad g(u) \geqq g(v) + \langle f(v), u - v \rangle \forall u, v \in X.$$

In such a case we say that *g is a potential* of f, and we denote the pair of subgradients and (one of) its potential by (f, g).

Another tool from convex analysis that we will use is the polar of a function. If $g: X \to \mathbb{R}$, we define its polar function by $g^*: X \to \mathbb{R} \cup \{-\infty, +\infty\}$ such that:

$$(4) \qquad g^*(v) = \sup_{u \in X} (\langle u, v \rangle - g(u)).$$

Now if g is a potential of f we have:

$$\langle v, f(v) \rangle - g(v) \geqq \langle u, f(v) \rangle - g(u), \text{ so } \langle v, f(v) \rangle - g(v) = \sup_{u \in X} \langle u, f(v) \rangle - g(u)$$

and, therefore,

$$(5) \qquad g^*(f(v)) + g(v) = \langle v, f(v) \rangle.$$

For more details see [11].

2.2. The real case. Now in the real case we can characterize the C.M. and the S.C.M. functions.

LEMMA 1. *A function $f: \mathbb{R} \to \mathbb{R}$ is cyclically monotone if and only if it is strictly cyclically monotone, and this property holds if and only if f is not decreasing.*

Proof. Let us show the equivalence between C.M. and nondecreasing functions. The C.M. condition is (1) and the nondecreasing property in $\mathbb{R}$ is (2). If in general we have (1) $\Rightarrow$ (2), we must only prove (2) $\Rightarrow$ (1), and we will do it by induction over $n \geqq 2$. The case $n = 2$ is verified because it is equivalent to (2). Suppose (1) is verified for $n' \leqq n$; let us show it for $n + 1$.

If the sequence $(u_0, \cdots, u_n)$ is completely ordered, i.e., $u_0 \geqq u_1 \geqq \cdots \geqq u_n$ or $u_0 \leqq u_1 \cdots \leqq u_n$, we get

$$\sum_{i=0}^{n} (u_i - u_{i+1}) f(u_i) \geqq f(u_n) \sum_{i=0}^{n} (u_i - u_{i+1}) = 0,$$

and the result follows.

If there exists $0 \leqq k \leqq n$ such that $u_{k-1} \geqq u_k \leqq u_{k+1}$ (we note $u_{-1} = u_n$) with at least one strict inequality, we obtain

$$(u_{k-1} - u_k) f(u_{k-1}) + (u_k - u_{k+1}) f(u_k) \geqq (u_{k-1} - u_{k+1}) f(u_{k-1});$$

hence, $\sum_{i=0}^{n} (u_i - u_{i+1}) f(u_i) \geqq \sum_{i=0}^{n-1} (\hat{u}_i - \hat{u}_{i+1}) f(\hat{u}_i)$, where $\hat{u}_i = u_i$ if $i \leqq k - 1$, $\hat{u}_i = u_{i+1}$ if $k \leqq i \leqq n - 1$. By the induction hypothesis the last sum is nonnegative, and the result holds.

The S.C.M. property condition follows because $(f(u) - f(v))(v - u) = 0$ implies $f(u) = f(v)$. □

In the real case the potential can be specified. Let f be a C.M. function in $\mathbb{R}$. Let $\delta_f(u) = f(u^+) - f(u^-)$ be the jump of f at u. As f is increasing the set $D(f) = \{u : \delta_f(u) > 0\}$ is countable. Let us suppose f is right continuous and the set of jump coordinates $D(f)$ is discrete. For a fixed $\alpha \in \mathbb{R}$ and $D(f)$ we write

$$(6) \qquad f_d(x) = \sum_{\alpha < u \leqq x} \delta_f(u) \text{ if } x \geqq \alpha, f_d(x) = - \sum_{x < u \leqq \alpha} \delta_f(u) \text{ if } x < \alpha,$$

which is an increasing pure jump function such that $f_d(\alpha) = 0$ (if f is an increasing pure jump function we get $f(x) = f(\alpha) + f_d(x)$). To explicitly obtain the potential associated to f_d, let us define $D^+(f) = \{u_n : n > 0\}$ and $D^-(f) = \{u_n : n < 0\}$ to be the set of ordered jump coordinates greater than α and smaller than α, respectively. If $x > \alpha$, let $n(x) = \sup\{n > 0 : u_n < x\}$; if $x < \alpha$ we write $n(x) = \inf\{n < 0 : u_n > x\}$ ($n(x) = 0$ if the sets are void). Now the potential associated to f_d is

$$g_d(x) = \sum_{1 \leqq n < n(x)} (u_{n+1} - u_n) \delta_f(u_n) + (x - u_{n(x)}) \delta_f(u_{n(x)}) \quad \text{if } n(x) \geqq 1,$$

$$(7) \qquad g_d(x) = \sum_{n(x) < n \leqq -1} |u_{n-1} - u_n| \delta_f(u_n) + |x - u_n(x)| \delta_f(u_{n(x)}) \quad \text{if } n(x) \leqq -1,$$

$$g_d(x) = 0 \quad \text{if } n(x) = 0.$$

Now take

$$(8) \qquad f_c(x) = f(x) - f_d(x),$$

which is an increasing continuous function such that $f_c(\alpha) = f(\alpha)$ (if f is an increasing continuous function we get $f_c = f$). The potential associated to f_c is

$$(9) \qquad g_c(x) = \int_\alpha^x f_c(u)\,du.$$

Then we get the known decomposition

$$(10) \qquad f = f_c + f_d, \text{ and its potential is } g = g_c + g_d.$$

The polar function satisfies

$$g^*(f(v)) = \langle v, f(v)\rangle - g(v) = (vf_c(v) - g_c(v)) + (vf_d(v) - g_d(v)).$$

We now give an explicit form for the polar function when f is a strictly increasing continuous function. In this case its inverse function f^{-1} is defined. The polar function satisfies: $g^*(f(v)) = vf(v) - \int_\alpha^v f(u)\,du$. From the equality

$$\int_\alpha^v (f(u) - f(\alpha))\,du = (v-\alpha)(f(v) - f(\alpha)) - \int_{f(\alpha)}^{f(v)} (f^{-1}(\xi) - \alpha)\,d\xi$$

we deduce

$$(11) \qquad \begin{aligned} g^*(f(v)) &= \int_{f(\alpha)}^{f(v)} f^{-1}(\xi)\,d\xi + \alpha f(\alpha), \quad \text{or} \\ g^*(x) &= \int_{f(\alpha)}^{x} f^{-1}(\xi)\,d\xi + \alpha f(\alpha). \end{aligned}$$

3. Automata networks. Let A be a linear operator in the real vector space X with inner product $\langle\, ,\, \rangle$, $b \in X$, and $f \colon X \to X$. We take the space state $Q = f(X)$ and let the dynamics on Q be defined by

$$(12) \qquad x(t+1) = f(Ax(t) - b).$$

This automaton is denoted by $\mathscr{A} = (Q, f, A, b)$.

Recall that a functional $H : Q \to \mathbb{R}$ is said to be a *Lyapunov functional* if and only if $(H(x(t)))_{t \geqq 1}$ increases with $t \geqq 1$.

THEOREM 1. *Let* $\mathscr{A} = (Q, f, A, b)$ *be an automaton where* A *is symmetric and* f *cyclically monotone. Let* g *be a potential of* f *and* g^*, *the polar associated to* g. *Then*

$$(13) \qquad H(x) = -g^*(x) + g(Ax - b) - \langle b, x\rangle$$

is a Lyapunov functional; in fact, $(H(x(t)))_{t \geqq 1}$ *increases with* $t \geqq 1$. *Now let*

$$(14) \qquad \hat{H}(x, x') = -\langle x, Ax'\rangle + g(Ax - b) + g(Ax' - b).$$

Then $(\hat{H}(x(t+1), x(t)))_{t \geqq 0}$ *increases with* $t \geqq 0$, *and we have the equality*

$$(15) \qquad H(x(t+1)) = \hat{H}(x(t+1), x(t)).$$

Proof. Note that the increasing property of $(H(x(t)))_{t \geqq 1}$ will follow from the increasing property of $(\hat{H}(x(t+1), x(t)))_{t \geqq 0}$ and (15).

Let us denote by $\Delta_t \hat{H} = \hat{H}(x(t+1), x(t)) - \hat{H}(x(t), x(t-1))$ for $t \geqq 1$. Since A is symmetric we have

$$\Delta_t \hat{H} = -\langle x(t), Ax(t+1) - Ax(t-1)\rangle + g(Ax(t+1) - b) - g(Ax(t-1) - b).$$

Let us define $u_t = Ax(t) - b$, then

$$\Delta_t \hat{H} = -\langle f(u_{t-1}), u_{t+1} - u_{t-1} \rangle + g(u_{t+1}) - g(u_{t-1}).$$

Since g is a potential of f we conclude $\Delta_t \hat{H} \geqq 0$.

Now from definition of g^* we have

$$g(Ax(t) - b) = -g^*(x(t+1)) + \langle x(t+1), Ax(t) \rangle - \langle x(t+1), b \rangle,$$

so $\hat{H}(x(t+1), x(t)) = H(x(t+1))$. □

For the main applications we will deal with product spaces. Let

$$\underline{X} = X^r = \{\underline{x} = (x_1, \cdots, x_r) : x_i \in X, i = 1, \cdots, r\}$$

be endowed with the inner product $[\underline{x}, \underline{x'}] = \sum_{i=1}^r \langle x_i, x_i' \rangle$. Let $\underline{A} = (A_{ij} : 1 \leqq i, j \leqq r)$ be a matrix acting on $\underline{X}$ such that $(\underline{Ax})_i = \sum_{j=1}^r A_{ij} \underline{x}_j$. We assume that the operator $\underline{A}$ is symmetric (with respect to [,]), which occurs if and only if $A_{ij}^T = A_{ji}$. Now let $f_i : X \to X$ be cyclically monotone for $i = 1, \cdots, r$. Then it is easy to prove that $\underline{f}$ defined by $(\underline{f}(\underline{x}))_i = f_i(x_i)$ is also cyclically monotone. If $g_i : X \to \mathbb{R}$ is a potential associated to f_i, the function $\underline{g} : \underline{X} \to \mathbb{R}$ defined by $\underline{g}(\underline{x}) = \sum_{i=1}^r g_i(x_i)$ is a potential associated to $\underline{f}$ and $\underline{g}^*(\underline{v}) = \sum_{i=1}^r g_i^*(v_i)$ is the polar associated to $\underline{g}$. If every f_i is strictly cyclically monotone, then $\underline{f}$.

Now, for the automaton $\underline{\mathscr{A}} = (\underline{Q}, \underline{f}, \underline{A}, \underline{b})$, where $\underline{Q} = \underline{f}(\underline{X})$, $\underline{b} \in \underline{X}$, we have the following corollary.

COROLLARY 1. *Let* $\underline{\mathscr{A}} = (\underline{Q}, \underline{f}, \underline{A}, \underline{b})$ *satisfy the above properties. Then*

$$\underline{H}(\underline{x}) = \sum_{i=1}^r \left(-g_i^*(x_i) + g_i\left(\sum_{j=1}^r A_{ij}x_j - b_i \right) - \langle b_i, x_i \rangle \right) \tag{16}$$

is a Lyapunov functional. Furthermore,

$$\underline{\hat{H}}(\underline{x}, \underline{x'}) = \sum_{i=1}^r \left(\sum_{j=1}^r -\langle x_i, A_{ij}x_j' \rangle + g_i\left(\sum_{j=1}^r A_{ij}x_j - b_i \right) + g_i\left(\sum_{j=1}^r A_{ij}x_j' - b_i \right) \right) \tag{17}$$

has the property that $\underline{\hat{H}}(\underline{x}(t+1), \underline{x'}(t))_{t \geqq 0}$ *is increasing and* $H(t+1) = \hat{H}(\underline{x}(t+1), \underline{x}(t))$.

Now let us apply this result for $X = \mathbb{R}$.

COROLLARY 2. *Let* $\underline{\mathscr{A}} = (\underline{Q}, \underline{f}, \underline{A}, \underline{b})$ *be such that* $\underline{A} = (a_{ij} : 1 \leqq i, j \leqq r)$ *is symmetric and* $(f_i : i = 1, \cdots, r)$ *is a sequence of strictly increasing continuous real functions. Then*

$$\underline{H}(\underline{x}) = \sum_{i=1}^r \left(-\int_{f_i(\alpha_i)}^{x_i} f_i^{-1}(\xi)\, d\xi + \int_{\alpha}^{\sum_{j=1}^r a_{ij}x_j - b_i} f_i(u)\, du - b_i x_i - \alpha_i f_i(\alpha_i) \right) \tag{18}$$

is a Lyapunov functional; that is $(\underline{H}(\underline{x}(t)))_{t \geqq 1}$ *is increasing with* $t \geqq 1$, *where* $(\alpha_i)_{i=1}^r$ *are fixed.*

Proof. This follows from Corollary 1 and (6) and (7). □

Recall that (18) is an expression similar to that obtained in [6]. Now for the length of the limit cycles we deduce the following result.

THEOREM 2. *Let* $\mathscr{A} = (Q, f, A, b)$ *be such that* A *is symmetric and* f *strictly cyclically monotone. Then the length* T *of any limit cycle satisfies* $T \leqq 2$.

Proof. Let $(x(0), \cdots, x(T-1)$ be a limit cycle; that is $x(T) = x(0)$ and $x(t) \neq x(t')$ for $0 \leqq t < t' < T$. As $(\hat{H}(x(t+1), x(t)))_{t \geqq 0}$ is increasing and $x(t + t') =$

$x(t+t'(\text{mod}\,(T)))$ we deduce $\hat{H}(x(t+1)), x(t)) =$ constant. Then with above notation

$$\Delta_{t+1}\hat{H} = g(u(t+2)) - g(u(t)) - \langle f(u(t)), u(t+2) - u(t)\rangle = 0.$$

By the strict condition $u(t+2) = u(t)$. Hence $f(u(t+2)) = f(u(t))$, that is, $x(t+3) = x(t+1)$, and the result follows. □

For many applications [1], [2], [6] we must consider the automaton given by $\underline{X} = \mathbb{R}^r$, $\underline{A} = (a_{ij} : 1 \leqq i, j \leqq r)$ a symmetric matrix, $\underline{f} = (f_1, \cdots, f_r)$ where $f_i : \mathbb{R} \to Q_i$ is a real cyclically monotone function.

COROLLARY 3. *Let* $\mathscr{A} = (\underline{Q}, \underline{f}, \underline{A}, \underline{b})$, *where* $\underline{f} = (f_1, \cdots, f_r)$ *is a set of increasing real functions. Then any limit cycle has length* $T \leqq 2$.

Proof. From Lemma 1 any f_i is strictly cyclically monotone, so $\underline{f}$. Theorem 2 implies the result. □

There exist some cases in which we can determine exactly the length of the cycles.

THEOREM 3. *Let* $\mathscr{A} = (Q, f, A, b)$ *be such that* A *is a symmetric positive definite operator and* f *is strictly cyclically monotone. Then any limit cycle is a fixed point, that is,* $T = 1$.

Proof. Let $(x(0), x(1))$ be the limit cycle (we know $T \leqq 2$). Let

$$\begin{aligned}\gamma &= \langle x(0) - x(1), A(x(0) - x(1))\rangle \\ &= \langle x(0), (Ax(0) - b) - (Ax(1) - b)\rangle + \langle x(1), (Ax(1) - b) - (Ax(0) - b)\rangle.\end{aligned}$$

We have

$$\begin{aligned}&\langle x(t), (Ax(t) - b) - (Ax(t-1) - b)\rangle \\ &= \langle f(Ax(t-1) - b), (Ax(t) - b) - (Ax(t-1) - b)\rangle \leqq g(Ax(t) - b) - g(Ax(t-1) - b).\end{aligned}$$

Then $\gamma \leqq 0$. Being A positive defined, we obtain $x(1) = x(0)$. □

Note that we have only used the cyclically monotone property of f, the positive definite property of A, and that any limit cycle is of length $T \leqq 2$. So we obtain the following corollary.

COROLLARY 4. *Let* $\mathscr{A} = (\underline{Q}, \underline{f}, \underline{A}, \underline{b})$, *where* $\underline{f} = (f_1, \cdots, f_r)$, *be a set of nondecreasing real functions. If* $\underline{A}$ *is symmetric positive definite, any limit cycle is a fixed point.*

THEOREM 4. *Let* $\mathscr{A} = (Q, f, A, b)$ *be such that* A *is a symmetric negative definite operator and* f *is strictly cyclically monotone with* $f(-b) = 0$. *Then* $x = 0$ *is the only fixed point and any other limit cycle is of length* $T = 2$.

Proof. Let $(x(0), x(1))$ be a limit cycle. As g is a potential of f,

$$g(-b) - g(Ax(0) - b) - \langle f(Ax(0) - b), -b - (Ax(0) - b)\rangle \geqq 0.$$

Let $x(0)$ be a fixed point, $f(Ax(0) - b) = x(0)$, then

$$g(-b) \geqq g(Ax(0) - b) - \langle x(0), Ax(0)\rangle.$$

Since $f(-b) = 0$ and g is its potential, $g(-b)$ is a global minimum of g (see [11]). As A is negative definite, $-\langle x(0), Ax(0)\rangle \geqq 0$. Then we must necessarily have $x(0) = 0$, which is a fixed point. □

We also deduce Corollary 5 from the above proof.

COROLLARY 5. *Let* $\mathscr{A} = (\underline{Q}, \underline{f}, \underline{A}, \underline{b})$, *where* $\underline{f} = (f_1, \cdots, f_r)$, *be a set of nondecreasing real functions. If* $\underline{A}$ *is negative definite and* $f_i(-b_i) = 0$ *for* $i = 1, \cdots, r$, *then the only fixed point is* $\underline{x} = 0$ *and any other limit cycle is of length* 2.

In estimating the transient length we will assume that the local C.M. function f takes only a finite number of values; that is, the state space of the automaton $Q = f(X)$ is finite. In the real case this means that f is a jump function.

COROLLARY 6. *Let $\mathscr{A} = (Q, f, A, b)$ be a finite automaton. Let*

$$\Delta_1 \hat{H}_m = \min_{x(0) \in Q, x(0) \neq x(2)} (\hat{H}(x(2), x(1)) - \hat{H}(x(1), x(0))).$$

Then the transient length L of any trajectory is bounded by

$$(19) \qquad \begin{aligned} &L = 0 \text{ } if \, \Delta_1 \hat{H}_m = 0 \\ &L \leqq \frac{\max\limits_{u,v \in Q} \hat{H}(u,v) - \min\limits_{u,v \in Q} \hat{H}(u,v)}{\Delta_1 \hat{H}_m} \quad if \, \Delta_1 \hat{H}_m > 0. \end{aligned}$$

Proof. If $\Delta_1 \hat{H}_m = 0$, obviously $L = 0$. Let us suppose $\Delta_1 \hat{H}_m > 0$. Let $x(0), x(1), \cdots, x(L), x(L+1), x(L)$ be a trajectory with the limit cycle $(x(L), x(L+1))$. Since Q is finite we have

$$\min_{u,v \in Q} \hat{H}(u,v) + t \Delta_1 \hat{H}_m \leqq \hat{H}(x(t+1), x(t)) \leqq \max_{u,v \in Q} \hat{H}(u,v), \forall t \leqq L.$$

Hence, we obtain the bound of expression (19). □

Acknowledgments. We are grateful to R. Cominetti and A. M. Odlyzko for helpful comments and valuable suggestions. We also thank the referee for important corrections made in the final version of the paper.

REFERENCES

[1] E. BIENENSTOCK, F. FOGELMAN, AND G. WEISBUCH, EDS., *Disordered Systems and Biological Organization*, NATO ASI Series F, Springer-Verlag, New York, Berlin, 1986.

[2] E. GOLES, *Dynamics of positive automata networks*. Theoret. Comput. Sci., 41 (1985), pp. 19–32.

[3] E. GOLES AND S. MARTÍNEZ, *Properties on positive functions associated to automata networks*, Discrete Appl. Math., 18 (1987), pp. 39–46.

[4] ———, *Neural and automata networks: dynamical behaviour and applications*, Collect. Math. Appl., 58, Kluwer Academic, Hingham, MA, 1990.

[5] J. J. HOPFIELD, *Neural networks and physical systems with emergent collective computational abilities*, Proc. Nat. Acad. Sci., USA, 79 (1982), pp. 2554–2558.

[6] J. J. HOPFIELD AND D. W. TANK, *Collective Computation with Continuous Variables in Disordered Systems and Biological Organization*, E. Bienenstock, F. Fogelman, and G. Weisbuch, eds., NATO ASI Series F, Comp. and Systems Sciences, 20, Springer-Verlag, New York, Berlin, 1986, pp. 153–170.

[7] W. MCCULLOCH AND W. PITTS, *A logical calculus of the ideas immanent in nervous activity*, Bull. Math. Biophys. 5 (1943), pp. 115–133.

[8] A. M. ODLYZKO AND D. J. RANDALL, *On the periods of some graph transformations*, Complex Syst., 1 (1987), pp. 203–210.

[9] S. POLJAK AND D. TURZIK, *On systems, periods and semipositive mappings*, Res. Rep. Tech. University of Thakurova, Czechoslovakia, 1984.

[10] ———, *On an application of convexity to discrete systems*, Discrete Appl. Math., 13 (1986), pp. 27–32.

[11] T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[12] J. VON NEUMANN, *Theory of Self-Reproducing Automata*, W. Burks, ed., University of Illinois Press, Chicago, IL, 1966.

[13] S. WOLFRAM, *Theory and Applications of Cellular Automata*, World Scientific, Teaneck, NJ, 1986.

SIAM J. DISC. MATH.
Vol. 4, No. 2, pp. 207-222, May 1991

TIGHT BOUNDS ON MINIMUM BROADCAST NETWORKS*

MICHELANGELO GRIGNI† AND DAVID PELEG‡

Abstract. A *broadcast graph* is an n-vertex communication network that supports a broadcast from any one vertex to all other vertices in optimal time $\lceil \lg n \rceil$, given that each message transmission takes one time unit and a vertex participates in at most one transmission per time step. This paper establishes tight bounds for $B(n)$, the minimum number of edges of a broadcast graph, and $D(n)$, the minimum maxdegree of a broadcast graph. Let $L(n)$ denote the number of consecutive leading 1's in the binary representation of integer $n - 1$. It is shown that $B(n) = \Theta(L(n) \cdot n)$ and $D(n) = \Theta(\lg \lg n + L(n))$, and for every n we give a construction simultaneously within a constant factor of both lower bounds. For all n, graphs with $O(n)$ edges and $O(\lg \lg n)$ maxdegree requiring at most $\lceil \lg n \rceil + 1$ time units to broadcast are constructed. These broadcast protocols may be implemented with local control and $O(\lg \lg n)$ bits overhead per message.

Key words. broadcasting, graphs, networks, Fibonacci numbers

AMS(MOS) subject classifications. 94C15, 68M10

1. Introduction.

This paper deals with graphs suitable for performing broadcasts efficiently. We represent a communication network by a connected graph G, where the vertices of G represent processors and the edges represent bidirectional communication channels. We assume communication has the following constraints:

1) Messages may be sent directly only between neighbors in the graph.
2) Each message transmission takes one unit of time.
3) A vertex may participate in at most one message transfer at a time.

That is, if u sends a message to v, neither u nor v may send or receive another message on that step. A *broadcast protocol* for G allows any *originator* vertex to send a message to all other vertices in the network. This broadcast model is studied in several papers [BHLP1]-[FHMP], [HHL], [HL], [L], [MH], [P], [RL]-[Wa].

Given G and vertex $v \in G$ let $b(v, G)$ be the minimum time needed to broadcast from v. Let $b(G) = \max_v b(v, G)$, the *broadcast radius* of G. Since the number of vertices knowing the message may at most double on each step, $b(G) \geqq \lceil \lg n \rceil$ for any n-vertex graph G (lg denotes $\log_2$). A *broadcast graph* is an n-vertex graph G with $b(G) = \lceil \lg n \rceil$.

We consider three cost measures for broadcast graphs and their protocols. The first, which is often the most significant cost measure in network design, is the number of edges. Let $B(n)$ denote the minimum number of edges of any n-vertex broadcast graph. A *minimum broadcast graph* is a broadcast graph with $B(n)$ edges; a number of previous papers have dealt with determining values of $B(n)$ and finding minimum or near-minimum broadcast graphs. The values of $B(n)$ were determined precisely for $n \leqq 19$ [FHMP], [MH], [Wa]. For general n it was shown that $B(n) = O(n \lg n)$ [F1] and that $B(n) = \Omega(n)$ (more precisely, $n - 1$ is a stated lower bound in [F1], [L], and $B(n) \geqq n$ for $n > 3$ is implied by the discussion in [F2]). For n a power of two, $B(n) =$

* Received by the editors August 11, 1987; accepted for publication (in revised form) June 27, 1990.

† Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139. This material is based upon work supported by a National Science Foundation Graduate Fellowship and was done partly at AT&T Bell Laboratories, Murray Hill, New Jersey 07974. This work was further supported by National Science Foundation grant 8912586CCR and Air Force Office of Scientific Research grant AFOSR-89-0271.

‡ Department of Applied Mathematics, The Weizmann Institute, Rehovot 76100, Israel. Part of this work was carried out while the author was visiting Stanford University and was supported in part by a Weizmann fellowship, contracts ONR N00014-85-C-0731 and ONR N00014-88-K-0166 from the Office of Naval Research, and by a grant from the Stanford Center for Integrated Systems.

$1/2\, n \lg n$ [FHMP], realizable by the hypercube graph. However, for n not a power of 2 the behavior of $B(n)$ was not precisely determined.

The second cost measure that we consider is the maximum degree of broadcast graphs. This measure is not as well studied as the previous one in the context of broadcast graphs, but it is no less important due to current limitations in networking technology. For vertex $v \in G$ let $d(v)$ denote its degree, so the maxdegree of G is $\Delta(G) = \max_v d(v)$. Let $D(n)$ be the minimum maxdegree of any n-vertex broadcast graph. Several previous papers concentrated on broadcasting on bounded-degree graphs [BHLP1], [LP].

The final cost measure that we consider is the message overhead needed to implement the broadcast protocol under local control. We assume the broadcast messages may carry along extra control bits, and we bound the maximum number of extra bits needed on any message sent in the protocol. We assume that processors know the size of the graph and their own identity in the graph, as well as local information such as the identities of their neighbors. We assume also that processors know on which edge an incoming message arrives; in some situations this is all the information the processor needs. We use a synchronous model where all messages take unit time, although we do not assume processors have access to a global clock.

For an integer $n > 1$ let $L(n)$ denote the number of leading 1's in the binary representation of $n - 1$; for example $L(14) = L(1101_2 + 1) = 2$. Then $1 \leqq L(n) \leqq \lceil \lg n \rceil$. $L(n)$ is monotone increasing in the range $2^{t-1} < n \leqq 2^t$ for any $t \geqq 1$. For n in such range we have $L(n) = t - \lceil \lg (2^t - n + 1) \rceil$. Note that $L(n)$ grows slowly in this interval; in particular, it equals 1 over the first half of the interval $(2^{t-1} < n \leqq 2^{t-1} + 2^{t-2})$, 2 over the next quarter of the interval and so on. More generally, for all $n, l \geqq 1$,

$$\frac{|\{i : 1 < i \leqq n, L(i) > l\}|}{n} < 2^{-l},$$

so $L(n)$ is bounded by a constant for "most" values of n.

We show $B(n) = \Theta(L(n) \cdot n)$ and $D(n) = \Theta(\lg \lg n + L(n))$, and we construct graphs meeting both bounds simultaneously. Since $L(n) = o(\lg \lg n)$ for most n, this implies that most minimum broadcast graphs must be irregular, since they have $O(L(n))$ (constant) average degree but $\Omega(\lg \lg n)$ maxdegree.

Furthermore, we give protocols that may be implemented with $O(\lg \lg n)$ bit overhead per message in the synchronous model. In the asynchronous model, where there is no guarantee on message transmission time, the same graphs need $O(\lg n)$ bits overhead per message to avoid message collisions. These asynchronous protocols are tree-shaped: there are exactly $n - 1$ messages sent, one to each processor besides the originator.

In view of the practical significance of keeping $B(n)$ and $D(n)$ as small as possible, it may sometimes be desirable to allow a slight increase in broadcast time in order to allow a decrease in these cost parameters. This has led to the following relaxation of the problem [F1], [L]. A *relaxed broadcast graph* G has $b(G) \leqq \lceil \lg n \rceil + 1$. Let $B'(n)$ and $D'(n)$ denote the minimum number of edges and maxdegree required for an n-vertex relaxed broadcast graph. In [F1] it is noted that $B'(n)$ may be significantly less than $B(n)$ when n is equal to or slightly less than a power of 2. They demonstrate this fact by considering $n = 16$ (where the minimum time requirement is four steps, while the relaxed requirement is five steps) for which $B(n) = 32$ and $B'(n) = 19$. We construct relaxed broadcast graphs with $O(n)$ edges and $O(\lg \lg n)$ maxdegree, both within a constant factor of optimal. (Again a priori these must be irregular graphs.)

Although we have made our definitions for undirected graphs, our constructions use directed graphs, where messages may only travel in the direction of the edge. This leads to the analogous definitions of broadcast digraphs, their minimum edge number

$\vec{B}(n)$, and their minimum relaxed edge number $\vec{B}'(n)$. Clearly, $B(n) \leqq \vec{B}(n) \leqq 2 \cdot B(n)$ and $B'(n) \leqq \vec{B}'(n) \leqq 2 \cdot B'(n)$, so edge counting results in either model are equivalent up to a factor of 2. In the directed model we will let $d_{\text{out}}(v)$ and Δ_{out} refer to outdegree while $d_{\text{in}}(v)$ and Δ_{in} refer to indegree. Let $d = d_{\text{out}} + d_{\text{in}}$ and $\Delta(G) = \max_v d(v)$, then $\vec{D}(n)$ is the minimum of $\Delta(G)$ over all n-vertex broadcast digraphs G, $D(n) \leqq \vec{D}(n) \leqq 2 \cdot D(n)$. Similarly, define $\vec{D}'(n)$ as the minimum maxdegree of relaxed broadcast digraphs.

The paper is organized as follows. In § 2 we derive lower bounds. In §§ 3 and 4 we construct preliminary graphs, serving as building blocks for our main constructions given in § 5. Section 6 discusses a generalization of the model allowing "conference calls." Finally, in § 7 we offer some related problems and open questions.

2. Broadcast tree lower bounds. For a vertex v in graph (or digraph) G, a *broadcast tree* T is a time-labeled directed subgraph describing a broadcast originated by v, by the following rules:

1) T is spanning in G rooted at v, directed toward the leaves.

2) Each vertex u is labeled with an integer $t(u)$, where $t(v) = 0$.

3) Whenever u is a parent of w in T, $t(u) < t(w)$.

4) Whenever u and w are siblings in T, $t(u) \neq t(w)$.

Given such a T, interpret label $t(u)$ as the step when u receives the message originated by v; the conditions guarantee that the parent of u has the message and is free to send it to u on step $t(u)$. Define $t(T) = \max_u t(u)$; we say T is a k-step broadcast tree when $k = t(T)$. A collection of such trees, one rooted at each $v \in G$, defines a broadcast protocol for G.

For example, let H_r be the r-dimensional hypercube, with vertices given the usual binary numbering $0, \cdots, 2^r - 1$. Broadcast from 0 by sending along the sth dimension on step s, for $1 \leqq s \leqq r$: each vertex v that knows the message sends it to $v + 2^{s-1}$. The edges used by this protocol define the *Boolean broadcast tree* T_r [F1]. (See Fig. 1.)

Not every broadcast protocol is described by a broadcast tree, since a protocol may send more than one message to some vertex. Nevertheless, given a k-step broadcast from v, there exists a k-step broadcast tree from v, consisting of those edges on which each vertex first receives the message. Hence, $b(v, G) \leqq k$ if and only if there is a k-step broadcast tree T rooted at v. Any such tree protocol may be controlled with at most $O(\lg n)$ bits overhead per message (the identity of the originator), although the local program length may be long. A parent in T might as well tell all its children the message as quickly as possible, so we may also require the additional rule:

5) The children of any $u \in T$ have consecutive labels $t(u) + 1, t(u) + 2, \cdots$.

Refer to those vertices v in T with $t(v) = s$ as *generation* s of T. If we do not require that T span G, then say T is a *partial* broadcast tree in G; i.e., it only broadcasts to those vertices that it spans.

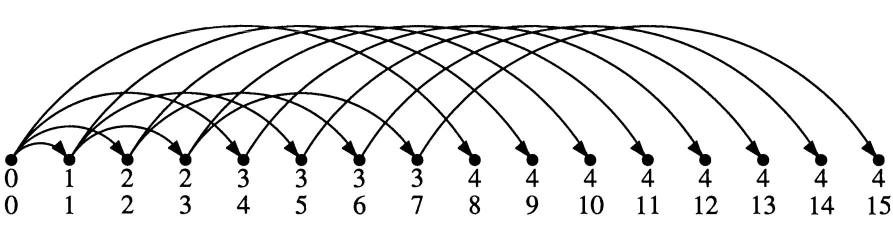


FIG. 1. *The Boolean broadcast tree T_4 describing a 4-step broadcast from 0 in the hypercube H_4. The first row of numbers is the time labeling $t(\cdot)$; the second row is the usual binary numbering.*

LEMMA 2.1. *In a (partial) broadcast tree T, the subtree rooted at a vertex u has size at most $2^{t(T)-t(u)}$.*

Proof. The subtree can at most double on each step after u gets the message. □

THEOREM 2.2. *Let G be an n-vertex broadcast graph. Then every vertex $v \in G$ has degree $d(v) \geqq L(n)$.*

Proof. Let $k = \lceil \lg n \rceil$, let T be a k-step broadcast tree from v, let $\delta \leqq d(v)$ be the degree of v in T, and let $v_1, \cdots, v_\delta$ be the children of v in T, labeled $t(v_s) = s$. Then the subtree rooted at v_s has size at most 2^{k-s}. Since these subtrees contain all vertices except v, $n-1 \leqq \sum_{i=1}^{\delta} 2^{k-i} = 2^k(1-2^{-\delta})$, so $\delta \geqq k - \lg(2^k-(n-1)) \geqq L(n)$. □

Note the last inequality is not tight; for example, when $n = 14$ we have $L(n) = 2$, but the proof really shows that the degree is at least three. For directed G the same argument shows $d_{\text{out}}(v) \geqq L(n)$ for all v; by averaging, there must also be a vertex v_0 with $d_{\text{in}}(v_0) \geqq L(n)$; hence, $d(v_0) \geqq 2 \cdot L(n)$.

COROLLARY 2.3. *For all $n \geqq 1$, we have $B(n) \geqq 1/2L(n) \cdot n$, $\vec{B}(n) \geqq L(n) \cdot n$, $D(n) \geqq L(n)$, and $\vec{D}(n) \geqq 2 \cdot L(n)$.*

Constructions in § 5 show the above bounds on $B(n)$ and $\vec{B}(n)$ are tight up to a constant factor. We need a further argument to get tight lower bounds for $D(n)$ and $\vec{D}(n)$.

For a given outdegree bound d and time bound t, we inductively construct $T_{d,t}$, the largest broadcast tree with $t(T) \leqq t$ and $\Delta_{\text{out}}(T) \leqq d$. The root v should have as many children as possible, so $d_{\text{out}}(v) = \min(d, t)$. Each child u of v must be the root of a maximum size subtree, so the tree rooted at u must be $T_{d,t-t(u)}$; this recursive construction uniquely defines $T_{d,t}$.

Let $b_d(t)$ be the number of vertices in $T_{d,t}$, and let $f_d(s)$ be the size of generation s in $T_{d,t}$, so $b_d(t) = \sum_{s=0}^{t} f_d(s)$. Since the parents of generation s are the vertices of the previous d generations, we have recurrences for $s, t > 0$:

$$b_d(t) = 1 + \sum_{1 \leqq i \leqq d} b_d(t-i), \tag{1}$$

$$f_d(s) = \sum_{1 \leqq i \leqq d} f_d(s-i), \tag{2}$$

where $b_d(0) = f_d(0) = 1$ for the originator and $b_d(s) = f_d(s) = 0$ for $s < 0$. The recurrence (2) defines the dth-order Fibonacci sequence [K, 5.4.2]. If $d \geqq t$ then the tree $T_{d,t}$ is simply the Boolean broadcast tree T_t.

The generating polynomial $x^d = x^{d-1} + \cdots + x + 1$ has one large real root λ near 2 dominating the growth rate of $f_d(t)$ and $b_d(t)$ (all other roots lie in the unit circle):

$$\lambda \sim 2 - 2^{-d} - \frac{d}{2} 2^{-2d} - O(d^2 2^{-3d}) \text{ as } d \rightarrow \infty.$$

For our purposes it suffices that $2 - 2^{1-d} < \lambda < 2 - 2^{-d}$ for all $d \geqq 2$.

LEMMA 2.4. *For $t \geqq 1$, $d \geqq 2$, let λ be defined as above. Then*

$$\lambda^{t-1} \leqq f_d(t) \leqq (2/\lambda)^{d-1} \lambda^{t-1},$$

$$\frac{\lambda^t - 1}{\lambda - 1} + 1 \leqq b_d(t) \leqq (2/\lambda)^{d-1} \frac{\lambda^t - 1}{\lambda - 1} + 1.$$

Proof. Fix d. For $1 \leqq t \leqq d$ we have $f_d(t) = 2^{t-1}$, which lies in the claimed range. Now for $t > d$ use induction on t and the fact that λ satisfies the generating polynomial. The bounds on $b_d(t)$ follow from summing the bounds on $f_d(t)$. □

THEOREM 2.5. *Let T be an n-vertex k-step broadcast tree with $d = \Delta_{\text{out}}(T)$. Then*

$$d > \lg\left(\frac{k \lg e}{k+1-\lg n}\right) - 1.$$

Proof. Since $T_{d,k}$ is the largest possible such tree, $n \leqq b_d(k)$. Estimated $(2/\lambda)^{d-1}/(\lambda - 1) \leqq 2$ (see Table 1), so $n \leqq b_d(k) \leqq 2(\lambda^k - 1) + 1 < 2\lambda^k$, and $\lg n < 1 + k \lg \lambda$. Now estimate $\lg \lambda < 1 - 2^{-(d+1)} \lg e$ and solve for d. □

In particular, we consider trees arising in broadcast graphs and relaxed broadcast graphs.

COROLLARY 2.6. *Let T be an n-vertex broadcast tree and $t(T) \leqq c + \lg n$. Then $\Delta_{\text{out}}(T) > \lg \lg n - 0.5 - \lg(c + 1)$. In particular if $t(T) \leqq \lceil \lg n \rceil$ then $\Delta_{\text{out}}(T) > \lg \lg n - 1.5$, and if $t(T) \leqq \lceil \lg n \rceil + 1$ then $\Delta_{\text{out}}(T) > \lg \lg n - 2.1$.*

COROLLARY 2.7. *$D(n)$ and $\vec{D}(n)$ are $\Omega(L(n) + \lg \lg n)$; $D'(n)$ and $\vec{D}'(n)$ are $\Omega(\lg \lg n)$.*

We have shown that any n-vertex $\lceil \lg n \rceil$-step broadcast tree T has $\Delta(T) \geqq \max(L(n), \lg \lg n - 1.5)$. We now show that this lower bound is tight up to a leading factor of 2 and a small additive constant.

LEMMA 2.8. *Given d, l, t with $d \geqq l + \lg t$, $b_d(t) > (1 - 2^{-l})2^t$.*

Proof. Estimate $b_d(t) \geqq \lambda^t > 2^t(1 - 2^{-d})^t > 2^t(1 - t2^{-d}) = 2^t - t2^{t-d} \geqq 2^t - 2^{t-l}$ by the condition on d. □

COROLLARY 2.9. *For $n \leqq 2^t$, let $d = L(n) + \lceil \lg \lg n \rceil + 1$. Then $n \leqq b_d(t)$.*

3. Boolean constructions. This section and the following one describe some initial constructions, which will be combined in § 5 to yield the desired results. Specifically, this section concerns constructions based on variations of the hypercube.

Given n and a $S \subset \mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$, the difference digraph $\mathbb{Z}_n[S]$ is defined with vertex set $\mathbb{Z}_n$ and edge set $\{i \rightarrow i + s : i \in \mathbb{Z}_n, s \in S\}$ (these are also known as directed star polygons). $\mathbb{Z}_n[S]$ has $|S|n$ edges. For example, define the *Boolean difference digraph* as $BD(n) = \mathbb{Z}_n[\{2^i : 0 \leqq i < \lceil \lg n \rceil\}]$. $BD(n)$ has broadcast properties similar to the hypercube, but it is defined even for n not a power of two.

THEOREM 3.1. *$BD(n)$ is a broadcast digraph, with a 1-bit overhead protocol.*

Proof. By translational symmetry we may assume that 0 is the originator. Let $k = \lceil \lg n \rceil$. On step s, $1 \leqq s \leqq k$, every vertex i that knows the message and knows $i + 2^{k-s} < n$ sends the message to $i + 2^{k-s}$ (note that we have reversed the bit order used in the H_k protocol described at the beginning of the previous section). This protocol will reach every processor exactly once.

Note that the processors know the time step by observing on which edge the message arrives (so this works asynchronously as well). To decide whether $i + 2^{k-s} < n$, the

TABLE 1
λ, $(2/\lambda)^{d-1}/(\lambda - 1)$, *and* $f_d(t)$ *for* $2 \leqq d \leqq 6$, $0 \leqq t \leqq 6$.

d	λ	$\frac{(2/\lambda)^{d-1}}{\lambda - 1}$	$f_d(0)$	$f_d(1)$	$f_d(2)$	$f_d(3)$	$f_d(4)$	$f_d(5)$	$f_d(6)$
2	1.6180	2.0000	1	1	2	3	5	8	13
3	1.8393	1.4088	1	1	2	4	7	13	24
4	1.9276	1.2043	1	1	2	4	8	15	29
5	1.9659	1.1089	1	1	2	4	8	16	31
6	1.9836	1.0595	1	1	2	4	8	16	32

processors pass along an extra bit. When processor i sends a message to $j = i + 2^{s-1}$, the extra yes/no bit tells j whether $j + 2^{s-1} \geqq n$. If "no," then j knows that its subtree will not be truncated anywhere, and so it sends "no" bits to all its children. Otherwise, j computes $n' = n \bmod 2^{s-1}$ and recursively originates the $BD(n')$ protocol (this recursion adds no overhead to the messages; we only require that each processor knows the value of n). An originator knows n and should send a "yes" message to the first child such that $2^{s-1} < n$, and a "no" message to every child after that.

We observe that the resulting broadcast is tree-shaped, and hence this protocol will also work in the asynchronous model. □

Let H_r denote the directed r-dimensional hypercube (i.e., H_r has a pair of directed edges wherever the undirected hypercube has an edge). From H_r we construct a related digraph $H_{r,t}$ with 2^{r+t} vertices: at each $v \in H_r$ root a copy of the Boolean broadcast tree T_t. Refer to the original H_r vertices as "root" vertices and the new $2^r(2^t - 1)$ vertices as "tree" vertices. Any root of $H_{r,t}$ may originate an $(r + t)$-step broadcast: for the first t steps broadcast across H_r to all the roots, and then for the remaining t steps broadcast up all the trees. This protocol requires no overhead bits, since the dimensions are always used in a fixed order. Now modify $H_{r,t}$ by adding a back-edge from every tree vertex back to the root of its tree; call the resulting digraph $H'_{r,t}$. Then $H'_{r,t}$ is a relaxed broadcast digraph. Root vertices originate a broadcast as before; tree vertices take one step to notify their root, and then let the root take care of the broadcast from there. In this case the protocol does not trace out a tree of messages, since the originator will receive a copy of the message; nevertheless, the protocol is still valid in the asynchronous model because the first message of the originator cannot collide with any future messages.

Just using $H'_{r,t}$ we may construct a sparse ($O(n)$-edge) relaxed broadcast digraph. Given n let $k = \lceil \lg n \rceil$, $t = \lceil \lg k \rceil$, and $r = k - t$. Then $H'_{r,t}$ has 2^k vertices; throw out $2^k - n$ leaves (this will not disrupt the protocol). The resulting digraph has n vertices, $(r-2)2^r + 2n < 3n$ directed edges and maxdegree $\Delta = 2r + t + 2^t - 1 < 4k = O(\lg n)$.

4. Fibonacci constructions.

4. Fibonacci constructions. In this section we construct partial broadcast digraphs FIB1, FIB2, FIB3; all rely on one idea, an "addressing" scheme based on the generalized Fibonacci numbers of § 2. Construction FIB1 is the simplest illustration of the idea. Construction FIB2 takes care of some wraparound problems, allowing any node to be an originator. Construction FIB3 allows the originator to send fewer messages; this graph will be the "backbone" for the final broadcast graph constructions in § 5. These constructions have parameters d, t, and l (corresponding roughly to maxdegree, broadcast time, and $L(n)$).

For string $\alpha = \alpha_1 \cdots \alpha_t \in \{0, 1\}^t$, let $\langle \alpha \rangle_d$ denote $\sum_{i=1}^{t} a_i \cdot f_d(i)$. Let $\mathscr{B}_{d,t} \subset \{0, 1\}^t$ denote the strings that do not have the substring 0^d1, and let $\mathscr{F}_{d,t} \subset \mathscr{B}_{d,t}$ be those with $\alpha_t = 1$ ($\mathscr{F}_{d,0}$ and $\mathscr{B}_{d,0}$ both contain the empty string). We have the following numbering theorem; it is a "dense" version of the dth-order Fibonacci number system [K, Exercise 5.4.2.10].

LEMMA 4.1. *For $d \geqq 2$, $|\mathscr{F}_{d,t}| = f_d(t)$, $|\mathscr{B}_{d,t}| = b_d(t)$, and the map $\alpha \mapsto \langle \alpha \rangle_d$ from $\mathscr{B}_{d,t}$ to $\{0, \cdots, b_d(t) - 1\}$ is bijective.*

Proof. Give each vertex u of $T_{d,t}$ a t-bit address α corresponding to the path from the root to u, where $\alpha_s = 1$ if and only if the path includes a vertex of generation $s \geqq 1$ (see Fig. 2). Inductively, the addresses in generation s are $\alpha = \beta 0^{t-s}$, where $\beta \in \mathscr{F}_{d,s}$, $b_d(s-1) \leqq \langle \alpha \rangle_d < b_d(s)$. Finally, note $\mathscr{B}_{d,t}$ is the disjoint union $\mathscr{B}_{d,t} = \bigcup_{s=0}^{t} \mathscr{F}_{d,s} 0^{t-s}$. □

Given $d \leqq t$, we construct a digraph $\mathrm{FIB1}_{d,t}$ (see Fig. 3) based on this addressing scheme. $\mathrm{FIB1}_{d,t}$ has vertex set $\mathbb{Z}_{b_d(t)} \times \mathbb{Z}_t$; we let $x \in \mathbb{Z}_{b_d(t)}$ index columns and $s \in \mathbb{Z}_t$ index

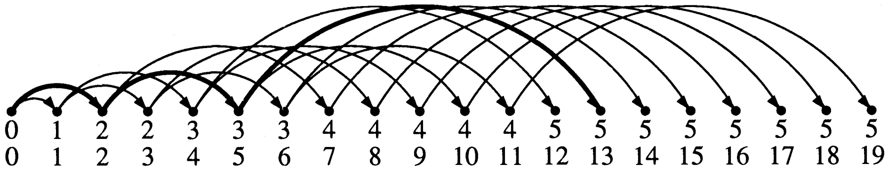


FIG. 2. *Broadcast tree* $T_{2,5}$ *with generation labels and the numbering of Lemma* 4.1. *Bold edges show* $13 = 2 + 3 + 8 = \langle 01101 \rangle_2$.

rows. It has the following single class of edges (we start a list of classes here, because we will add more soon).

Class 1. For all x, $1 \leqq s \leqq t$, and $1 \leqq i \leqq \min(d, s)$, connect $(x - f_d(s), s - i)$ to (x, s).

$\mathrm{FIB}1_{d,t}$ has $b_d(t) \cdot t$ vertices, less than $b_d(t) \cdot dt$ directed edges, and maxdegree $2d$.

THEOREM 4.2. *For every vertex* $(x, 0) \in \mathrm{FIB}1_{d,t}$ *there is a t-step partial broadcast tree* $T1_{d,t}(x)$ *rooted at* $(x, 0)$ *such that*:

(i) $T1_{d,t}(x)$ *contains exactly one vertex in each column* y.

(ii) *Generation* s *of* $T1_{d,t}(x)$ *lies entirely in row* s.

(iii) $T1_{d,t}(x)$ *is isomorphic to* $T_{d,t}$.

(iv) *The protocol for* $T1_{d,t}(x)$ *needs no bit overhead on messages*.

Proof. Let $T1_{d,t}(x)$ correspond to the following protocol: on step s, each vertex (y, r) that has received the message within the last d steps $(s - d \leqq r < s)$ sends it to $(y + f_d(s), s)$ (see Fig. 3). Then we are simply reconstructing $T_{d,t}$: vertex u in generation s of $T_{d,t}$ is reconstructed as $(s, x + \langle \alpha \rangle_d)$ in $T1_{d,t}(x)$ where α is the address of u. Hence, we generate all $\alpha \in \mathcal{B}_{d,t}$, and touch every column exactly once. Note the vertices do not

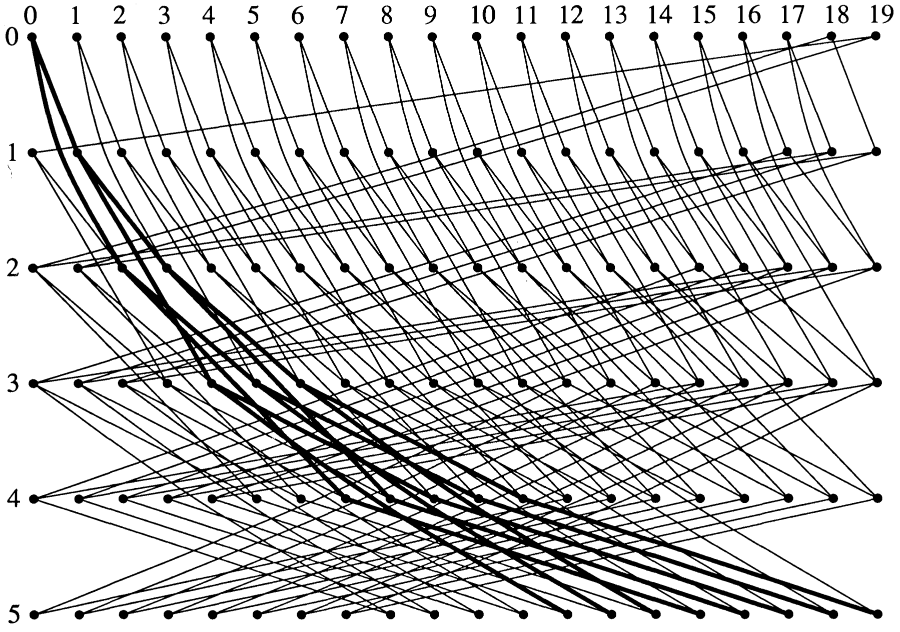


FIG. 3. *The digraph* $\mathrm{FIB}1_{2,5}$ *with subtree* $T1_{2,5}(0)$. *All edges are directed downward*; *the first and last rows are identified*.

need to know any dynamic information such as α or x to run this protocol (the vertices may even be oblivious to which edge delivers an incoming message). □

$\mathrm{FIB1}_{d,t}$ is still a long way from a broadcast digraph; our next construction allows *any* vertex to originate a partial broadcast, with only slightly higher costs. Construct a second digraph $\mathrm{FIB2}_{d,t}$ by augmenting $\mathrm{FIB1}_{d,t}$ with three further edge classes (Fig. 4 illustrates how the four classes connect the rows of $\mathrm{FIB2}_{d,t}$).

Class 2. For all x, $t - d \leqq r < t$, $1 \leqq s \leqq d$, connect $(x - f_d(s), r)$ to (x, s).

Class 3. For all x and $1 \leqq s \leqq t - 1$, add an edge from (x, s) to $(x, s + 1)$.

Class 4. For all x and $1 \leqq s < t - d$, add an edge from (x, s) to $(x, s + d + 1)$.
We refer to edges of Classes 1 and 2 as "Fibonacci" edges, because sending a message along such an edge always involves a jump of $f_d(s)$ columns, where $s \in \{1, \cdots, t\}$ is the row in which the jump ends. We refer to edges of Classes 3 and 4 as "Zero" edges since they begin and end in the same column.

THEOREM 4.3. *For every vertex* $(x, r) \in \mathrm{FIB2}_{d,t}$ *there is a t-step partial broadcast tree* $T2_{d,t}(x, r)$ *rooted at* (x, r) *such that*:

(i) $T2_{d,t}(x, r)$ *contains at least one vertex in each column* y.

(ii) *Generation* s *of* $T2_{d,t}(x, r)$ *lies entirely in row* $r + s$.

(iii) $\Delta_{\mathrm{out}}(T2_{d,t}(x, r)) \leqq 2d$.

(iv) *With* $\lg t + \lg d + O(1)$ *bits overhead per message we may implement this protocol and furthermore appoint a unique "leader" vertex of* $T2_{d,t}(x, r)$ *in each column*.

Proof. By translational symmetry among the columns we may assume that $x = 0$. If $r = 0$ we use the protocol from $\mathrm{FIB1}_{d,t}$; otherwise, $1 \leqq r < t$. Again we construct the tree by describing a protocol. In the previous protocol we constructed all addresses $\alpha \in \mathscr{B}_{d,t}$ by jumping along Fibonacci edges; in this protocol we will construct all addresses $\alpha = \gamma\beta$, where $\gamma \in \mathscr{B}_{d,r}$ and $\beta \in \mathscr{B}_{d,t-r}$. In particular, this includes all $\alpha \in \mathscr{B}_{d,t}$. The protocol proceeds in two phases: in phase I (the first $t - r$ steps) we construct β while leaving $\gamma = 0^r$, and in phase II (the last r steps) we wrap around and construct γ. In phase I we use Zero edges to pass completed $0^r\beta$ addresses down their columns to where they will eventually wrap around; phase II is essentially identical to the $\mathrm{FIB1}_{d,t}$ protocol. We maintain the invariant that all messages sent on step τ arrive in row $s = \tau + r \bmod t$, so each generation is confined to a single row.

In the following discussion let (y, s) refer to a vertex receiving the message on time step τ, $1 \leqq s, \tau \leqq t$. We call messages "Fibonacci" or "Zero" depending on the type of edge they traverse. Let messages carry the following additional information fields:

- The current time step τ and the row r of the originator. Since the receiver is in row $s = r + \tau \bmod t$, only one of these fields really needs to be sent.
- A routing address $\alpha \in \{0, 1\}^t$ telling the receiver (y, s) what path the message has followed so far on its way here from the originator (x, r). Bit α_i tells whether the message made a Fibonacci jump to row i; hence, $y = x + \langle \alpha \rangle_d$. We maintain $\alpha = \gamma\beta$ for some $\gamma \in \mathscr{B}_{d,r}$, $\beta \in \mathscr{B}_{d,t-r}$.

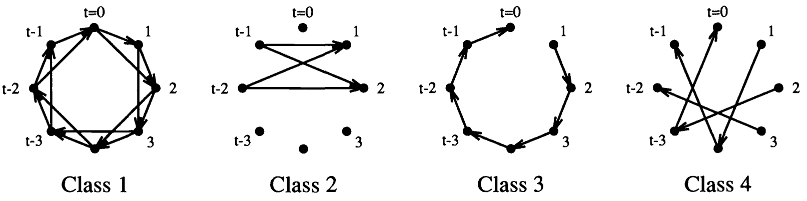


FIG. 4. *The classes of edges between rows of* $\mathrm{FIB2}_{2,8}$. *Edges here represent collections of edges between the corresponding rows* (*Fibonacci edges for Classes* 1 *and* 2, *Zero edges for Classes* 3 *and* 4).

We will show later that the protocol does not need to carry along all t bits of α, but only $\lg d + O(1)$ bits; so, the total message overhead will be $\lg t + \lg d + O(1)$ bits.

To start the protocol we pretend that on step $\tau = 0$ the originator (x, r) receives a Fibonacci message with address $\alpha = \gamma\beta = 0^t$. We describe the protocol by the actions of any vertex (y, s) after receiving a message on step τ. There are several cases.

I. If the received message is a phase I message (precisely, $\tau = 0$ or $r < s \leqq t$), the protocol is still working on β while $\gamma = 0^r$. The received message could be of either type.

Zero: If the received message is type Zero, then we are simply passing this $0^r\beta$ down the column without changing β further. There are two cases:

—If $s < t$, then just send the same Zero message on to the next row of this column, using an edge of Class 3.

—If $s = t$, then phase II will begin on the next step, so the receiver (y, t) must start sending Fibonacci messages to modify γ. It sends them consecutively to rows $1, \cdots, \min(d, r)$, using edges of Class 1.

Fibonacci: If the received message is type Fibonacci, we are actively building β, although γ is still 0^r. We have two goals: continue modifying β by sending Fibonacci messages to higher rows, and also let this current $\alpha = 0^r\beta$ wrap around the column to fill in its low-order bits. There are two cases:

—If $s < t - d$, then the next $d + 1$ steps (rows) are also phase I. Send Fibonacci messages to rows $s + 1, \cdots, s + d$, using edges of Class 1. Finally, send a Zero message to row $s + d + 1$ using an edge of Class 4.

—If $s \geqq t - d$, then (y, s) must both finish phase I and start phase II. For the remaining $t - s$ phase I steps, (y, s) sends Fibonacci messages to rows $s + 1, \cdots, t$ using edges of Class 1. On the next $\min(d, r)$ phase II steps (y, s) sends Fibonacci messages to rows $1, \cdots, \min(d, r)$ using edges of Class 2.

II. In phase II we fill in γ; only Fibonacci messages may be received in this phase. (y, s) reacts by sending Fibonacci messages to rows $s + 1, s + 2, \cdots$ for the next $\min(d, t - \tau = r - s)$ steps using edges of Class 1. This is essentially the protocol of $\mathrm{FIB1}_{d,t}$.

Recall that all messages sent on step τ arrive in row $s = r + \tau \bmod t$. The following claims characterizing the messages sent on each step may be proven inductively.

I. On step τ in phase I there is one Fibonacci message sent for each $\alpha = 0^r\beta 0^{t-s}$, where $\beta \in \mathscr{F}_{d,\tau}$. There is one Zero message sent for each $\alpha = 0^r\beta 0^{(d+1)+t-s}$, where $\beta \in \mathscr{B}_{d,\tau-(d+1)}$.

II. On step τ in phase II only Fibonacci messages are sent, one for each $\alpha = \gamma 0^{r-s}\beta$, where $\gamma \in \mathscr{F}_{d,s}$ and $\beta \in \mathscr{B}_{d,t-r}$.

Every $\alpha \in \mathscr{B}_{d,r}\mathscr{B}_{d,t-r}$ appears exactly once as the address of a Fibonacci message (an α may appear several times as a Zero message address), so in particular each $\alpha \in \mathscr{B}_{d,t}$ appears exactly once as the address of a Fibonacci message (we count the dummy message sent to the originator). Then there is a natural "leader" in each column: the vertex receiving an address $\alpha \in \mathscr{B}_{d,t}$ in a Fibonacci message. To show that the protocol defines a tree, we first check that there are no collisions on any step, i.e., all addresses α sent on a step map to distinct $\langle \alpha \rangle_d$ modulo $b_d(t)$.

I. In phase I, all addresses of messages of either type are of the form $\alpha = 0^r\beta$, where $\beta \in \mathscr{B}_{d,t-r}$. Since $1^r\beta 0^{t-s} \in \mathscr{B}_{d,t}$, the values $\langle 1^r\beta 0^{t-s} \rangle_d$ are all distinct modulo $b_d(t)$ by Lemma 4.1. Since $\langle \alpha \rangle_d = \langle 1^r\beta 0^{t-s} \rangle_d - \langle 1^r 0^{t-r} \rangle_d$, the $\langle \alpha \rangle_d$ must also be distinct.

II. In phase II, since all addresses of messages sent to row s are of the form $\gamma 0^{r-s}\beta$ for $\gamma \in \mathscr{F}_{d,s}$, $\beta \in \mathscr{B}_{d,t-r}$. The shifted addresses $\gamma 1^{r-s}\beta$ are all in $\mathscr{B}_{d,t}$; hence, the $\langle \alpha \rangle_d$ are distinct by a similar argument.

To finish showing that the protocol defines a tree, we check that there are no vertices appearing in two generations. This could only happen if generation t collides with generation 0 (the originator) in row r. But all the addresses received by generation t are in $\mathscr{F}_{d,r}\mathscr{B}_{d,t-r}$, and, hence, not congruent to 0.

Now we consider the bit overhead to implement this protocol. The entire protocol is oblivious to x, which just translates the tree vertices in $\mathbb{Z}_{b_d(t)}$. During the protocol no use is ever made of the bits of α; all the vertices need to know is the time τ (which must be given to them in the message overhead) and what kind of edge delivers the message (which we have assumed they know for free). At the end of the protocol each vertex $(y, s) \in T2_{d,t}(x, r)$ needs to know whether their $\alpha \in \mathscr{B}_{d,t}$ to decide if it is the leader of column y. Since $\gamma \in \mathscr{B}_{d,r}$ and $\beta \in \mathscr{B}_{d,t-r}$, the only way α could fail to be in $\mathscr{B}_{d,t}$ is if the substring $0^d 1$ appears on the boundary between γ and β. To decide, (y, s) needs to know:

- whether γ has a 1 in its last d bits and where it is,
- how many consecutive 0's begin β (either $< d$ or all of β).

The first may be determined just from the identity of the sender (local information); the second may be maintained with $\lg d + O(1)$ additional bits overhead per message. With $\lg t$ more bits to maintain the clock, we use the claimed number of bits. Since the resulting partial broadcasts are tree-shaped, this protocol works asynchronously as well. □

For our third construction FIB3 our goal is to reduce the number of messages sent by the originator. We introduce a new parameter $l \leqq d$ that corresponds to the $L(n)$ function of § 2. We modify FIB2 to work when the originator sends only l messages. Define

$$b_d(l, t) = 1 + \sum_{i=1}^{l} b_d(t-i).$$

In particular, $b_d(0, t) = 1$. Then $b_d(l, t)$ is the size of the maximum t-step broadcast tree $T_{d,l,t}$ whose root has degree l and all other vertices have outdegree at most d. We make the following simple estimate.

LEMMA 4.4. $b_d(l, t) \geqq (1 - 2^{-l})b_d(t) + 1$.

Proof. Since $b_d(t-i) \geqq 2^{-i}b_d(t)$,

$$b_d(l, t) - 1 \geqq \sum_{i=1}^{l} 2^{-i}b_d(t) = (1 - 2^{-l})b_d(t). \qquad \square$$

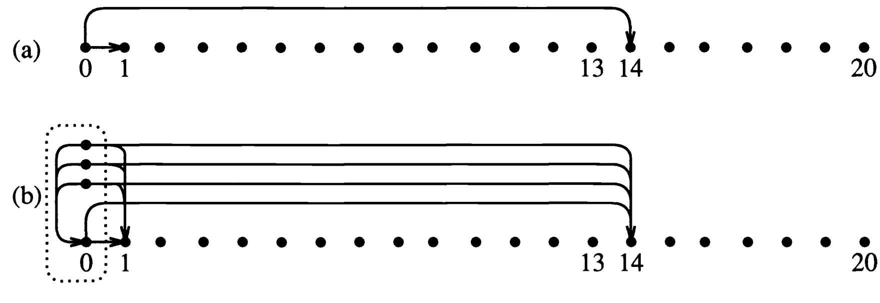


FIG. 5. (a) *Class* 5 *edges from node* 0 *in some row r of* FIB3 ($d = 3, l = 2, t = 5$). (b) *Corresponding class* 5′ *edges from* $S(0, r)$ *in* FIB4. *Node* 1 *is responsible for columns* 1–13, *node* 14 *for columns* 14–20.

We construct digraph $\mathrm{FIB3}_{d,l,t}$ on vertex set $\mathbb{Z}_{b_d(l,t)} \times \mathbb{Z}_t$. We use the same edge class definitions 1, 3, and 4 used for constructing $\mathrm{FIB2}_{d,t}$ (but now x in the definitions ranges over a different number of columns). We enlarge Class 2 and define a new Class 5 of "Root" edges going horizontally across rows (see Fig. 5(a)):

Class 2′. For all x, $t-l-d \leqq r < t$, $1 \leqq s \leqq d$, connect $(x - f_d(s), r)$ to (x, s).

Class 5. For all (x, r), $1 \leqq i \leqq l$, connect (x, r) to $(x + b_d(i-1, t), r)$.

THEOREM 4.5. *For every vertex* $(x, r) \in \mathrm{FIB3}_{d,l,t}$ *there is a t-step partial broadcast tree* $T3_{d,l,t}(x, r)$ *rooted at* (x, r) *such that:*

(i) $T3_{d,l,t}(x, r)$ *contains at least one vertex in each column* y.

(ii) *The root* (x, r) *has l children, all connected by Root edges. The subtree rooted at the ith child spans columns* $x + b_d(i-1, t)$ *through* $x + b_d(i, t) - 1$.

(iii) *All vertices besides the root have degree at most* $2d$.

(iv) *With* $\lg l + \lg d + \lg t + O(1)$ *bits overhead per message we may implement this protocol and furthermore appoint a unique "leader" vertex of* $T3_{d,l,t}(x, r)$ *in each column.*

Proof. We have arranged the edges so that for any originator (y, r) and any i such that $1 \leqq i \leqq l$, we may run the broadcast protocol for $\mathrm{FIB2}_{d,t-i}$ in rows $0, \cdots, t-i$. The only new trick is that if $t - i \leqq r < t$, then we use the $\mathrm{FIB1}_{d,t-i}$ protocol, i.e., (y, r) sends Fibonacci messages to rows $1, \cdots, d$, using Class 2′ edges. This generates a copy of $T2_{d,t-i}(y, r)$ spanning and confined to columns $y, \cdots, y + b_d(t-i) - 1$ of $\mathrm{FIB3}_{d,l,t}$. We call this the $\mathrm{FIB2}_{d,t-i}$ subprotocol started by (y, r).

To start the main $\mathrm{FIB3}_{d,l,t}$ protocol from the root (x, r), the root on step i $(1 \leqq i \leqq l)$ sends a message to its ith child $(x + b_d(i-1, t), r)$ telling it to start running the $\mathrm{FIB2}_{d,t-i}$ subprotocol. We have spaced the Root edges so that the child subtrees span disjoint consecutive blocks of columns; hence, the subprotocols never collide. Together with the root, they span every column of $\mathrm{FIB3}_{d,l,t}$. To control the ith subprotocol we need to send along the value of i with those messages, so that they know which rows to skip when they wrap around. This introduces $\lg l$ additional message overhead bits to those already needed to control the $\mathrm{FIB2}_{d,t-i}$ subprotocols. To select column leaders, we simply select the leaders from each subprotocol together with the root (x, r). □

We remark that this FIB3 protocol in fact never uses the Fibonacci edges to row t, and so there is a slightly better construction where FIB3 has only $t - 1$ rows. We have chosen to avoid this modification for simplicity of presentation.

5. Main constructions.

Using the constructions of the previous two sections, we are now ready to construct broadcast digraphs and relaxed broadcast digraphs within constant factors of the lower bounds of § 2. Note that for broadcast graphs, if $L(n)$ is within a constant factor of $\lg n$, say $L(n) \geqq \lceil \lg n \rceil / 3$, then we may use the Boolean difference digraph $BD(n)$, which has $n \lceil \lg n \rceil \leqq 3 \cdot \vec{B}(n)$ edges and maxdegree $2 \lceil \lg n \rceil \leqq 3 \cdot \vec{D}(n)$; i.e., both are within constant factors of the lower bounds.

Otherwise (when $L(n)$ is $O(\lg n)$, or when we want relaxed broadcast graphs) we rely on the constructions of § 4. We want to augment FIB3 with additional vertices so that most vertices have degree $O(l)$ but are still able to originate a broadcast. The general idea is to put each vertex (x, r) of FIB3 in charge of a small set of vertices $S(x, r)$, such that every vertex in the set may originate a broadcast in the same way as (x, r), and (x, r) can broadcast a received message back to all of $S(x, r)$. Thus, most of the vertices in $S(x, r)$ may have lower degree than their leader (x, r). This idea is detailed below.

We construct digraph $\mathrm{FIB4}_{d,l,t_1,t_2}$ (with parameters $1 \leqq l \leqq d \leqq t_1/2$, $\lg t_1 \leqq t_2$) by augmenting $\mathrm{FIB3}_{d,l,t_1}$. We also add new vertices in this construction; we refer to the original vertices of $\mathrm{FIB3}_{d,l,t_1}$ as FIB3-vertices.

Define $\tau = \lceil \lg t_1 \rceil$. For each column x of FIB3$_{d,l,t_1}$ construct a copy $H(x)$ of $H_{\tau,t_2-\tau}$ (from § 3). Identify the t_1 FIB3-vertices of column x with t_1 distinct roots of $H(x)$; the other $2^{t_2} - t_1$ vertices of $H(x)$ are new to the digraph FIB4. Repeating this for every column x defines all the new vertices of FIB4.

Partition the vertices of $H(x)$ into t_1 subsets $S(x, s)$ of size at most $\lceil 2^{t_2}/t_1 \rceil \leqq 2^{t_2-\tau+1}$, so that $(x, s) \in S(x, s)$. Thus we have partitioned all the vertices of FIB4 into approximately equal size sets represented by the FIB3-vertices. Now add the following class of Root edges, extending the previous definition of Class 5 (see Fig. 5(b)):

Class 5′. For all FIB3-vertices (x, r) and all $1 \leqq i \leqq l$, connect each $v \in S(x, r)$ to $(x + b_d(i-1, t), r)$. Also if $v \neq (x, r)$, connect v to (x, r).

We now bound the number of edges and the maxdegree of FIB4.

LEMMA 5.1. *Given* $1 \leqq l \leqq d \leqq t_1/2$, $\tau = \lceil \lg t_1 \rceil \leqq t_2$, *then* FIB4$_{l,d,t_1,t_2}$ *has exactly* $n = 2^{t_2} \cdot b_d(l, t_1)$ *vertices, less than* $n(l+2) + b_d(l, t_1) \cdot 2t_1(d + \tau + 1)$ *directed edges, and maxdegree less than* $3(d + l + t_2 + 2) + (l + 1)2^{t_2}/t_1$.

Proof. There are $b_d(l, t_1)$ columns y and 2^{t_2} vertices in each $H(y)$, hence the number of vertices. To count edges and maxdegree, there are six kinds of edges: those in classes 1, 2′, 3, 4, and 5′, and those in the $H(y)$ subgraphs. First we count edges. Every vertex has at most $l + 1$ outedges of Class 5′, and at most one incoming tree-edge from $H(y)$, hence the leading term of $n(l + 2)$. Now we count the remaining edges contributed per column, and then multiply by $b_d(l, t_1)$, the number of columns. In column y, Class 1 contributes $< dt_1$ edges, Class 2′ contributes $d(l + d) \leqq dt_1$ edges, Classes 3 and 4 contribute $< t_1$ edges each, and the digraph $H(y)$ has $\tau 2^\tau < 2t_1\tau$ root edges besides the tree edges already counted above. Altogether, this gives the claimed bound on edges.

To bound maxdegree we simply add the maxdegrees of the six cases. Class 1 has maxdegree $2d$ (note we must count both in and out degree), Class 2′ has maxdegree $l + d$, Classes 3 and 4 have maxdegree 2 each, class 5′ has maxdegree $\lceil 2^{t_2}/t_1 \rceil (l + 1) + l < (l + 1)(2 + 2^{t_2}/t_1)$ (attained at the FIB3-vertices), and each $H(y)$ subgraph has maxdegree $t_2 + \tau < 3t_2$. Adding these gives the claimed bound on maxdegree.

THEOREM 5.2. *For every vertex* $v \in$ FIB4$_{d,l,t_1,t_2}$ *there is a* $(t_1 + t_2)$*-step broadcast protocol starting from* v. *Furthermore,*

(i) *After* t_1 *steps of the protocol, for each column* y *there is a root vertex in* $H(y)$ *that knows the message.*

(ii) *The maximum number of messages sent by any vertex is* $2d + t_2$.

(iii) *We may implement this protocol with* $\lg l + \lg t + \lg d + O(1)$ *bits overhead per message in the synchronous model.*

Proof. Let (x, s) be the FIB3-vertex such that $v \in S(x, s)$. The protocol proceeds in two phases; the first phase of t_1 steps is essentially like the FIB3 protocol, with v taking the role of (x, s). For steps $1 \leqq i \leqq l$, v sends the message along a Root edge (Class 5′) to $(x + f_d(i - 1, t), s + i)$. Just as in the FIB3 protocol, these children of v initiate a local FIB2$_{d,t_1-i}$ protocol. Hence after t_1 steps, all of these subprotocols finish simultaneously and appoint a FIB3-vertex leader in every column except column x (since these leaders are FIB3-vertices they are also roots of their respective $H(y)$'s). Column x itself is a special case; if $v = (x, s)$, then v itself is the leader of column x. Otherwise, v sends the message to (x, s) on step $l + 1$; there is time to do this since we have assumed $l < t_1$.

At the start of the second phase, the FIB3 protocol has established a unique root leader in each FIB3-column y. This leader is a root in the corresponding $H(y)$, so simply follow the $H(y)$ t_2-step protocol to notify every vertex of FIB4. Hence, every vertex knows the message in $t_1 + t_2$ steps.

Note that the leader of column x receives the message on step $l + 1$ and may as well start broadcasting up $H(x)$ on step $l + 2$. Since v will then receive a copy of its own message, we cannot claim that the $H(x)$ protocol is tree shaped; nevertheless, it will work asynchronously, because the message that v receives is a descendant of v's last outgoing message. The message overhead is essentially the same as for the $\mathrm{FIB3}_{d,l,t_1}$ protocol, since the second phase introduces no new costs. □

This FIB4 protocol will not work in the asynchronous model. Phase 1 will appoint a unique leader in a given column y, but it may also make temporary use of other vertices from that column (as in the FIB2 protocol). In phase 2 the leader broadcasts to all of $H(y)$, including these "temporary" vertices used in the phase 1. Hence in the asynchronous model one of the temporary vertices may simultaneously receive two messages, one from phase 1 and the other from phase 2. Just from the general arguments of § 2 we know there is an asynchronous tree-shaped protocol, but with $O(\lg n)$ bits overhead.

Note that if t_2 is strictly greater than τ (i.e., $H(y)$ is not just a hypercube) then at least half the vertices in FIB4 are deletable; these are the leaves of the $T_{t_2 - \tau}$ subtrees used to build each $H(y)$. Removing some or all of these vertices will not disrupt the protocol, since they are always the last to receive the message in any broadcast. Deleting vertices preserves the leading term $n(l + 2)$ in the statement of Lemma 5.1, since each vertex contributed at most $l + 2$ edges to that term.

Given n, we now show that by setting the FIB4 parameters appropriately and possibly deleting some vertices, we get broadcast digraphs and relaxed broadcast digraphs of size n with degree and edge costs within constant factors of their lower bounds. The choices for l, d, and $t = t_1 + t_2$ are more or less fixed for us; our main freedom is in partitioning t into t_1 and t_2. As t_2 increases, the number of edges decreases and the maxdegree increases, but both are reasonably small for t_2 around $\lg dt/l$. The following two theorems make this precise.

THEOREM 5.3. *Given n such that $L(n) < (1/3)\lg n$, let $t = \lceil \lg n \rceil$. Choose $l = L(n) + 2$, $d = \lceil \lg\lg n \rceil + l$, $t_2 = \lceil \lg dt/l \rceil + 1$, $t_1 = t - t_2$. Then* $\mathrm{FIB4}_{d,l,t_1,t_2}$ *(possibly after deleting some leaves) is a t-step n-vertex broadcast digraph with $O(L(n)\cdot n)$ edges and $O(L(n) + \lg\lg n)$ maxdegree.*

Proof. First we show FIB4 has at least n vertices. By the definition of $L(n)$ we know $n \leqq 2^t(1 - 2^{-(L(n)+1)})$. Thus $n \leqq 2^t(1 - 2^{1-l}) < 2^t(1 - 2^{-l})^2$. By Lemma 4.4 we have $b_d(l, t) \geqq (1 - 2^{-l})b_d(t)$, and by Lemma 2.8 we have $b_d(t) \geqq (1 - 2^{-l})2^t$; hence, $b_d(l, t) \geqq n$. For any $k \leqq t$ we may estimate $2^k b_d(l, t - k) \geqq b_d(l, t)$, so in particular ($k = t_2$) FIB4 has at least n vertices.

We estimate $t_1 < t$, $\tau \leqq \lceil \lg\lg n \rceil$, $2dt/l \leqq 2^{t_2} < 4dt/l$, $t_2 < 2\lceil \lg\lg n \rceil$, and $b_d(l, t_1) \leqq 2^{t_1} = 2^t/2^{t_2} < 2^t(l/2dt) < nl/dt$. Now we apply Lemma 5.1. The number of edges is $<n(l+2) + b_d(l, t_1)\cdot 2t_1(d + \tau + 1) < n(l + 2) + (nl/dt)\cdot 2t\cdot 2d = 5nL(n) + 12n = O(nL(n))$. Similarly, the maxdegree is $<3(d + l + t_2 + 2) + (l + 1)2^{t_2}/t_1 < 3(3d) + (l + 1)(4dt/l)/t_1$. Since $t/t_1 \leqq \frac{3}{2}$ (for sufficiently large n) and $(l + 1)/l \leqq \frac{4}{3}$, the maxdegree is at most $17d = O(L(n) + \lg\lg n)$.

Finally, we need $2^{t_2} \geqq 2t_1$ so that there are enough deletable leaf vertices to get exactly n vertices. This is guaranteed by our choice of t_2. □

COROLLARY 5.4. *$B(n)$, $\vec{B}(n) = \Theta(L(n)\cdot n)$, and $D(n)$, $\vec{D}(n) = \Theta(L(n) + \lg\lg n)$.*

We comment that an alternative construction (presented in a previous version of this paper [Pe]) yields a better bound of $(L(n) + 2)n$ on the number of edges. However, that construction has linear indegree.

THEOREM 5.5. *Given n such that $L(n) < (1/3)\lg n$, let $t = \lceil \lg n \rceil + 1$. Choose $l = 2$, $d = \lceil \lg\lg n \rceil$, $t_2 = \lceil \lg dt \rceil$, $t_1 = t - t_2$. Then* $\mathrm{FIB4}_{d,l,t_1,t_2}$ *(possibly after deleting some* *leaves) is a t-step n-vertex relaxed broadcast digraph with $O(n)$ edges and $O(\lg \lg n)$ maxdegree.*

Proof. Again we start by showing FIB4 has at least n vertices. We know $n < 2^{t-1} < (1-2^{-l})^2 2^t$. Arguing as in the last proof we have $n < b_d(t)(1-2^{-l}) < b_d(l, t) < 2^{t_2} b_d(l, t_1)$; hence, the graph is large enough.

We estimate $t_1 < t$, $\tau \leqq \lceil \lg \lg n \rceil$, $dt \leqq 2^{t_2} < 2dt$, $t_2 < 2 \lceil \lg \lg n \rceil$, and $b_d(l, t_1) \leqq 2^{t_1} = 2^t / 2^{t_2} \leqq 2^t / dt < 4n/dt$. Again we apply Lemma 5.1. The number of edges is $< 21n = O(n)$ and the maxdegree is $< 17d + 12 = O(\lg \lg n)$.

We need $2^{t_2} \geqq 4t_1$ to insure there are enough deletable vertices; this is guaranteed by our choice of t_2 for n sufficiently large. □

COROLLARY 5.6. *$B'(n)$, $\vec{B}'(n) = \Theta(n)$, and $D'(n)$, $\vec{D}'(n) = \Theta(\lg \lg n)$.*

By Theorem 5.2 we know that the graphs of Theorems 5.3 and 5.5 have synchronous protocols with $O(\lg \lg n)$ bit overhead, while we need $O(\lg n)$ bits to get tree-shaped asynchronous protocols. We have no corresponding lower bounds on bit complexity.

6. c-Broadcast graphs.

A wide variety of models for broadcast (and communication networks in general) are considered in the literature. Examples of such models are radio broadcast networks (cf. [GVF]) and line broadcasts (cf. [F2]). One particular variant of the model considered here is a model that allows vertices to communicate simultaneously with all their neighbors in one time unit. This model, which is quite common in the field of synchronous distributed and parallel computing (cf. [A], [BD], [Fi] among others), is a natural one to consider when the system supplies hardware mechanisms enabling such an operation, or in cases where message transmission time is negligible compared to the time required for processing within the vertices between consecutive rounds. In such a model, broadcast can be achieved in time D in every network of diameter D; hence, in the complete network broadcast requires only one time unit.

The dichotomy between the above two extreme models suggests a natural intermediate model which provides for "conference calls" of limited size, i.e., in which a vertex is allowed to send a message simultaneously to up to c neighbors at a time, for some constant $c \geqq 1$. We refer to broadcast in this model as *c-broadcast*. In this section we extend the basic results known for $c = 1$ to every $c \geqq 1$. (A related generalization is studied in [RL]; there, the communication network is represented by a $(c+1)$-uniform hypergraph, and each conference call involves the vertices of some hyperedge.)

Similar to the definitions of the standard (1-broadcast) model, let $b_c(u, G)$ denote the minimum time required to broadcast from the vertex u in the graph G in the c-broadcast model, and let $b_c(G) = \max \{ b_c(u, G) \mid u \in V(G) \}$. The obvious lower bound on the time needed for c-broadcast is

LEMMA 6.1. *$b_c(G) \geqq \lceil \log_{c+1} n \rceil$ for every n-vertex network G.*

Consequently, a c-broadcast graph (respectively, relaxed c-broadcast graph) is an n-vertex communication network G such that $b_c(G) = \lceil \log_{c+1} n \rceil$ (respectively, $b_c(G) = \lceil \log_{c+1} n \rceil + 1$), and $B_c(n)$ (respectively, $B'_c(n)$) denotes the minimum number of edges of any n-vertex c-broadcast graph (respectively, relaxed c-broadcast graph).

We first extend the $n = 2^k$ result of [FHMP] to the c-broadcast model.

THEOREM 6.2. *$B_c(n) = cnk/2$ for every $n = (c+1)^k$, $k \geqq 1$.*

The results of the previous sections can be extended as well. Denote the exact number of consecutive leading c's in the $(c+1)$-ary representation of $n - 1$ by $L_c(n)$.

Extending the lower bound of Theorem 2.2, and using upper bound constructions from [Pe], we have the following.

THEOREM 6.3. *For all $n \geqq 1$, $c/2(L_c(n) - 1)n < B_c(n) < [c(L_c(n) + 1) + 1]n$,* i.e., $B_c(n) = \Theta(c \cdot L_c(n) \cdot n)$.

THEOREM 6.4. *$B'_c(n) < 2n$ for every $n \geqq 1$ and $c \geqq 1$.*

We may define the analogous maximal c-broadcast tree where each vertex sends messages at most d times (and hence has outdegree at most cd); the generation sizes are given by the sequence $f^c_d(s) = c \cdot \sum_{i=1}^{d} f^c_d(s-i)$ with growth rate $\lambda \sim (c+1) - c/(c+1)^d$. We may then extend Theorem 2.5.

THEOREM 6.5. *For every $n \geqq 1$, $D_c(n) = \Omega(L_c(n) + \log_{c+1} \log n)$ and $D'_c(n) = \Omega(\log_{c+1} \log n)$.*

The numbering scheme and first construction of § 4 carry through analogously. Further pursuit of the methods of § 5 may yield a construction proving the previous theorem tight as well; we have not pursued this further.

7. Open problems. A number of other interesting problems suggest themselves for further study. A significant area of problems concerns the design of broadcast schemes for given networks (as opposed to networks designed specifically for the purpose of broadcast). It is known that both determining the broadcast time $b(v, G)$ of an arbitrary vertex v in an arbitrary graph G [SCH] and recognizing a broadcast graph [FHMP] are NP-complete. Consequently, heuristic approaches for the problem of determining a near-optimal broadcast strategy in an arbitrary network were studied in [SW], and exact solutions were provided for special families of graphs, such as trees [P], [SCH] and grids [FH]. This line of research seems especially important, as in most cases the designer of a broadcast scheme faces an existing network with a fixed topology. Natural classes of graphs to be considered are families such as regular, planar, and bounded-degree graphs.

A related problem is that of distributing distinct pieces of information from *several* originators in the network simultaneously. In its ultimate form, this problem turns into the well-known *gossip problem* (cf. the bibliography of [HHL]). This problem involves n items of information, each initially held in one of the vertices, and the question is what resources (messages, time, edges, etc.) are required to let everyone know everything. This problem assumes a model in which a single message can carry an unlimited amount of information (or at least $O(n)$ bits). More realistic assumptions allow a message to carry no more than $O(\log n)$ bits of information, which makes the intermediate levels of the problem (i.e., with a limited number of originators) interesting in their own right.

Another interesting issue from a theoretical point of view is that of broadcasting on random graphs. The radius of random graphs has been well studied, but it may be worth looking at the broadcast radius $b(G)$ of random graphs. Pure random graphs will not make good broadcast graphs just out of degree constraints, but random graphs may still be useful components (e.g., use random edges instead of Fibonacci edges). More importantly, these random graphs may be fault tolerant. One may consider using a random protocol as well.

Acknowledgments. We thank Tom Leighton for his encouragement, Art Liestman for commenting on a previous version of the paper, and Alex Schäffer for helpful discussions.

REFERENCES

[A] B. AWERBUCH, *Complexity of network synchronization*, J. of the ACM, 32 (1985), pp. 804–823.

[BD] Ö. BABAOĞLU AND R. DRUMMOND, *Time-communication tradeoffs for reliable broadcast*, Report TR 85-687, Cornell University, Ithaca, NY, 1985.

[BHLP1] J.-C. BERMOND, P. HELL, A. L. LIESTMAN, AND J. G. PETERS, *Broadcasting in bounded degree graphs*, Technical Report TR 88-5, Simon Fraser University, Burnaby, Canada, 1988.

[BHLP2] ———, *New minimum broadcast graphs and sparse broadcast graphs*, Discrete Appl. Math., to appear.

[CL1] S. C. Chau and A. L. Liestman, *Constructing minimal broadcast networks*, J. Combin. Inform. System. Sci., 10 (1985), pp. 110–122.

[CL2] ———, *Constructing fault-tolerant minimal broadcast networks*, J. Combin. Inform. System. Sci., 11, (1986), pp. 1–18.

[F1] A. M. Farley, *Minimal broadcast networks*, Networks, 9 (1979), pp. 313–332.

[F2] ———, *Minimum-time line broadcast networks*, Networks, 10 (1980), pp. 59–70.

[FH] A. M. Farley and S. T. Hedetniemi, *Broadcasting in grid graphs*, Proc. 9th Southeastern Conference on Graph Theory and Computing, Baton Rouge, 1978, pp. 275–288.

[FP] A. M. Farley and A. Proskurowski, *Broadcasting in trees with multiple originators*, SIAM J. Algebraic Discrete Methods, 2 (1981), pp. 381–386.

[FHMP] A. M. Farley, S. T. Hedetniemi, S. Mitchell, and A. Proskurowski, *Minimum broadcast graphs*, Discrete Math., 25 (1979), pp. 189–193.

[Fi] M. J. Fischer, *The consensus problem in unreliable distributed systems* (*a brief survey*), Research Report #273, Department of Computer Science, Yale University, New Haven, CT, 1983.

[GVF] I. Gitman, R. M. Van Slyke, and H. Frank, *Routing in packet-switching broadcast radio networks*, IEEE Trans. Comm., 24 (1976), pp. 926–930.

[HHL] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman, *A survey of gossiping and broadcasting in communication networks*, Networks, 18 (1988), pp. 319–349.

[HL] P. Hell and A. L. Liestman, *Broadcasting in one dimension*, Discrete Appl. Math., 21 (1988), pp. 101–111.

[K] D. Knuth, *The Art of Computer Programming*, Addison-Wesley, Reading, MA, 1973.

[L] A. L. Liestman, *Fault-tolerant broadcast graphs*, Networks, 15 (1985), pp. 159–171.

[LP] A. L. Liestman and J. G. Peters, *Broadcast networks of bounded degree*, SIAM J. Discrete Math., 1 (1988), pp. 531–540.

[MH] S. Mitchell and S. T. Hedetniemi, *A census of minimum broadcast graphs*, J. Combin., Inform. System. Sci., 5 (1980), pp. 141–151.

[Pe] D. Peleg, *Tight bounds on minimum broadcast networks*, unpublished manuscript, July 1987.

[P] A. Proskurowski, *Minimum broadcast trees*, IEEE Trans. Comput., 30 (1981), pp. 363–366.

[RL] D. Richards and A. L. Liestman, *Generalizations of broadcasting and gossiping*, Networks, 18 (1988), pp. 125–138.

[SW] P. Scheuermann and G. Wu, *Heuristic algorithms for broadcasting in point-to-point computer networks*, IEEE Trans. Comput. 33 (1984), pp. 804–811.

[SCH] P. J. Slater, E. J. Cockayne, and S. T. Hedetniemi, *Information Dissemination in Trees*, SIAM J. Comput., 10 (1981), pp. 692–701.

[Wa] X. Wang, $B(18) = 23$, private communication, reported in [HHL], April 1986.

[We] D. B. West, *A class of solutions to the gossip problem*, *part* I, Discrete Math., 39 (1982), pp. 307–326.

SIAM J. DISC. MATH.
Vol. 4, No. 2, pp. 223-244, May 1991

NONCROSSING SUBGRAPHS IN TOPOLOGICAL LAYOUTS*

JAN KRATOCHVÍL†, ANNA LUBIW‡, AND JAROSLAV NEŠETŘIL§

Abstract. The computational complexity of the following type of problems is studied.

Given a topological layout (i.e., a drawing in the plane) of a graph, does it contain a noncrossing subgraph of a given type? It is conjectured that such problems are always NP-hard (provided planar subgraphs are looked for) regardless of the complexity of their nonplanar versions. This conjecture is verified for several cases in a very strong sense. In particular, it is shown that deciding the existence of a noncrossing path connecting two given vertices in a given topological layout of a 3-regular subgraph, as well as deciding the existence of a noncrossing cycle in such a layout, are NP-complete problems. It is also proved that deciding the existence of a noncrossing k-factor in a topological layout of a $(k + 1)$-regular graph is NP-complete for $k = 2, 3, 4, 5$. For $k = 1$, this question is NP-complete in layouts of 3-regular graphs, while it is polynomial solvable for layouts of graphs with maximum degree two.

Key words. topological graphs, planar layout, algorithmic complexity

AMS(MOS) subject classifications. 05C10, 05C99

1. Introduction. We consider finite undirected graphs without loops or multiple edges. (In a few particular cases when multiple edges are allowed, it will be specified explicitly that we are talking about multigraphs.) The vertex set and edge set of a graph G are denoted by $V(G)$ and $E(G)$, respectively, and we write $G = (V(G), E(G))$. An edge connecting vertices u and v is denoted by uv.

Let G be a (multi)graph considered together with a drawing (layout) in the plane (arcs corresponding to edges may cross but do not pass through vertices). Then we call G a *topological graph*. Let P be a graph property. Consider the following decision problem.

NONCROSSING P-SUBGRAPH.

(1) Instance: A topological graph G.
Question: Does there exist a subgraph G' of G with the property P such that the drawing of G' inherited from G is noncrossing?

We relate problem (1) to the following companion problem.

P-SUBGRAPH.

(2) Instance: A graph G.
Question: Does there exist a subgraph G' of G with the property P?

The problem (1) may sometimes be trivial and it can be solved in polynomial time for any property of bounded character (such as "to contain a fixed subgraph"). However, if the property P is not of bounded character, it has been conjectured in [N] that (1) is always an NP-hard problem, even if the companion problem (2) is polynomially solvable. We will say that P is a *topo-hard* property if (1) is an NP-hard problem.

In this paper we support this conjecture by several examples of properties for which problem (2) is easily solvable: connectivity, and maximum matching and k-factors. Perhaps the most striking example is the existence of a noncrossing path between two given vertices of a given layout. It is slightly surprising that all our properties remain topo-hard even if the input is reduced to very simple classes of graphs. It seems that it is difficult in general to find problems dealing with topological graphs that are polynomially solvable (cf. [KvL]).

* Received by the editors July 21, 1988; accepted for publication (in revised form) March 27, 1990.

† Department of Algebra, Charles University, Sokolovska 83, 186 00 Prague, Czechoslovakia.

‡ Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1.

§ Department of Applied Mathematics, Charles University, Malostranske nam. 25, 110 00 Prague, Czechoslovakia.

In most of the cases it suffices to describe a topological graph simply by stating which pairs of edges are crossing. This leads to a definition of abstract topological graphs in the next section. It is then natural to ask whether a given graph can be drawn in the plane respecting a given intersection pattern of edges. This is an abstract form of many problems in topological graph theory and VLSI circuits. In § 3, this problem is related to the recognition problem of string graphs (intersection graphs of curves in the plane, cf. [Si, KGK]).

In § 4, we prove NP-completeness of a modified *planar* 3-*sat* problem, which is a crucial starting point for the results on the *noncrossing path* and *noncrossing cycle* problems in § 5. Section 4 concludes the block of introductory sections where all auxiliary results are presented.

Sections 5–8 are devoted to examples of topo-hard properties. Besides *noncrossing path* and *noncrossing cycle*, another two problems of the same flavour (*noncrossing spanning tree* and *noncrossing connectedness*) are mentioned in § 5. An optimization problem of finding a maximum noncrossing matching is considered in § 6. Problems concerning k-factors are considered in §§ 7 and 8. Section 7 contains a polynomial algorithm for *noncrossing* 1-factor in layouts of graphs with maximum degree 2, while § 8 is devoted to NP-completeness results (which are proved in a unified form).

2. Topological and abstract topological graphs. A topological graph (T-graph) $G = (V, E)$ will sometimes be referred to as $G_T = (V_T, E_T)$ to indicate its drawing explicitly. An appropriate discretization of the drawing is a part of the data of a T-graph and it will be mostly irrelevant for our purpose.

An *abstract topological graph* (AT-graph) is a graph $G = (V, E)$ together with a set of $I \subset \binom{E}{2}$, i.e., and AT-graph is a triple $G = (V, E, I)$ ($\binom{E}{2}$ is the set of all two-element subsets of E).

For every T-graph $G_T = (V_T, E_T)$, we denote by $I(G_T)$ the set of all pairs of edges that are crossing in G_T, i.e., $I(G_T) = \{\{e_1, e_2\} | e_1^T \cap e_2^T \neq \emptyset\}$ (here e^T is the arc representing an edge e, and these arcs are considered open). To a T-graph G_T, we associate an AT-graph $G = (V, E, I(G_T))$ and we speak about the AT-graph corresponding to G_T.

We can expect that not every AT-graph corresponds to a T-graph. We say that an AT-graph $G = (V, E, I)$ is *realizable* if (V, E) has a drawing $G_T = (V_T, E_T)$ with $I(G_T) = I$, i.e., if G corresponds to a T-graph G_T. For example, the T-graph in Fig. 1 realizes the AT-graph $(\{a, b, c, d\}, \{ab, ad, bc, bd, cd\}, \{\{ab, cd\}, \{bc, bd\}\})$. On the other hand, an AT-graph $(V, E, \emptyset)$ is never realizable if the graph (V, E) is nonplanar. Of course, we can construct other examples of nonrealizable AT-graphs, e.g., $(\{a, b, c, d\}, \{ab, ac, ad, bc, bd, cd\}, \{\{ad, bc\}, \{bd, ac\}, \{cd, ab\}\})$ is a nonrealizable AT-graph on four vertices (cf. Fig. 2), while every AT-graph on three vertices is realizable (we leave a proof of this fact to the reader).

It is thus natural to consider the following decision problem.

Realizability of AT-Graphs.
(3) Instance: An AT-graph $G = (V, E, I)$.
Question: Is G realizable?

It has been shown recently by Kratochvíl that (3) is an NP-hard problem [K2].

Lubiw suggested the following approach to the question of realizability of AT-graphs. Considering realizability of an AT-graph $G = (V, E, I)$, I is the set of *prescribed* pairs of crossing edges. It seems more applicable to consider I to be the set of *allowed* pairs of crossing edges. This justifies the following definition. An AT-graph $G = (V, E, I)$ is *weak realizable* if there exists a T-graph $G_T = (V_T, E_T)$ such that $I(G_T) \subset I$. Note that if

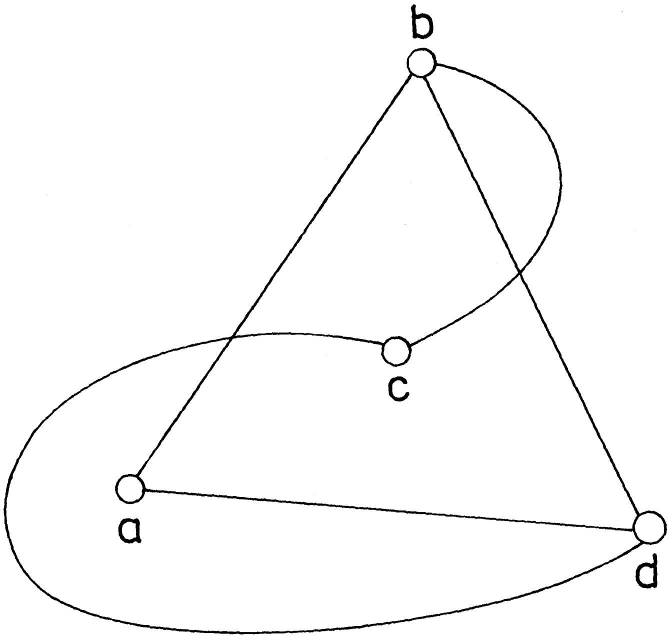


FIG. 1

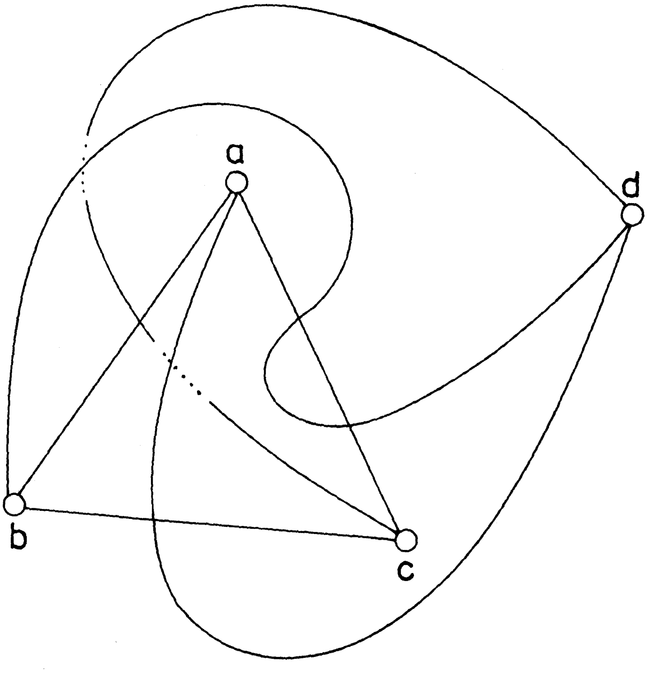


FIG. 2

(V, E) is planar then every AT-graph (V, E, I) is weak realizable, but may not be realizable (cf. Fig. 2). Consequently, we may consider the following decision problem.

WEAK REALIZABILITY OF AT-GRAPHS.

(4) Instance: An AT-graph $G = (V, E, I)$.
Question: Is G weak realizable?

Although problem (4) seems to be closer to planarity than (3), it is also NP-hard [K2]. Just in the opposite, we know that (3) is polynomially reducible to (4), but we do not know such a reduction from (4) to (3). Note also that we are calling these problems NP-hard because they are not yet proved to be in the class NP. There is even an evidence that they might not belong to NP at all: It is proved in [KM] that for every n, there exists a realizable AT-graph G_n with $O(n^2)$ vertices such that in every weak realization of G_n there are two edges that share 2^n common crossing points.

3. String graphs. A graph $G = (V, E)$ is called a *string graph* if it is isomorphic to the intersection graph of a system of curves in the plane, i.e., if there are curves $c(v)$, $v \in V$ such that $uv \in E$ if and only if $c(u) \cap c(v) \neq \emptyset$ for any two distinct $u, v \in V$. Such a system of curves is called a *string representation* of G (and the curves are called strings). See also [Si], [EET], [K1], [K2], [KGK].

Note that we may define string graphs in various ways, e.g., as intersection graphs of connected subgraphs in planar graphs, intersection graphs of arc-connected sets, or connected regions in the plane. All these definitions yield the same class of graphs [KGK], [Si].

String graphs present one of a few "natural" classes of graphs for which the NP-hardness of their recognition has been long open. Only recently it has been established by the first named author that the following problem is NP-hard [K2].

STRING GRAPHS RECOGNITION.

(5) Instance: A graph G.
Question: Is G a string graph?

String graphs are closed under edge contractions and under the taking of induced subgraphs, and hence the class of string graphs is closed in the induced minor order. The complexity of induced minor closed classes of graphs has been studied in [MNT]. The concept of induced minors differs drastically from minors studied by Robertson and Seymour [RS]. The class of all finite graphs fails to be well quasi-ordered by the induced minor order, and there are induced minor closed classes of graphs for which the recognition problem is even undecidable [MNT]. The class of string graphs is also the first "natural" induced minor closed class of graphs for which the recognition problem is known to be NP-hard.

Note also that *string graphs recognition* is actually a subproblem of *realizability of* AT-*graphs*. For, given a graph $G = (V, E)$, we may consider an AT-graph $H = (W, F, I)$ with $W = \{a_u, b_u | u \in V\}$, $F = \{a_u b_u | u \in V\}$, and $I = \{\{a_u b_u, a_v b_v\} | uv \in E\}$. Then every string representation of G is a realization of H (with vertices of W being the endpoints of the strings), and vice versa. Moreover, we have the following theorem.

THEOREM 3.1. *The problems* realizability of AT-graphs *and* string graphs recognition *are polynomially equivalent.*

We will use the following easy result.

LEMMA 3.2 [KGK]. *Let G be a string graph. Then G has a string representation with the following properties*:

(i) *Every point of the plane is contained in at most two strings*;

(ii) *Every pair of distinct strings shares a finite number of common intersecting points*;

(iii) *Every string that represents a vertex of degree* $\leqq 2$ *intersects any other string in at most one point.*

Proof of Theorem 3.1. Since (5) is a subproblem of (3), it suffices to show a reduction of (3) to (5). Let $H = (V, E, I)$ be an AT-graph. Define a graph $G = (W, F)$ by

$$W = V \cup E \cup \{(v, e) \mid v \in e \in E\},$$

$$F = \{\{v, (v, e)\}, \{(v, e), e\} \mid v \in e \in E\} \cup I.$$

If H corresponds to a topological graph H_T then we can easily transform H_T into a string representation of G. For every vertex $v \in V$, we choose a neighbourhood Ω_v of $v \in V_T$ so that Ω_v contains no crossing points of the edges from E_T. Then inside Ω_v, we replace the vertex v by a short segment and parts of the edges e^T by strings (v, e), as it is indicated in schematic Figs. 3(a), (b).

Conversely, if G is a string graph then it has a string representation satisfying (i)–(iii) of Lemma 3.2. In particular, the string representing a vertex $(v, e) \in W$ shares just one intersecting point with the string v, and one intersecting point with the string $e \in W$. Thus the situation looks locally like that depicted in Fig. 3(b). (Note that we may assume, without loss of generality, that every string e starts at its intersecting point with (v, e), since otherwise we can replace it as indicated in Fig. 3(c), (d).) Then contracting the strings v and (v, e) for $v \in e \in E$ into a point v, we obtain a T-graph H_T with intersection pattern $I(H_T) = I$. □

It follows that *string graphs recognition* also might not belong to NP. Again there are string graphs that require exponential numbers of intersecting points in any of their string representations. We conclude this section with several technical notions and results that will be used in the sequel.

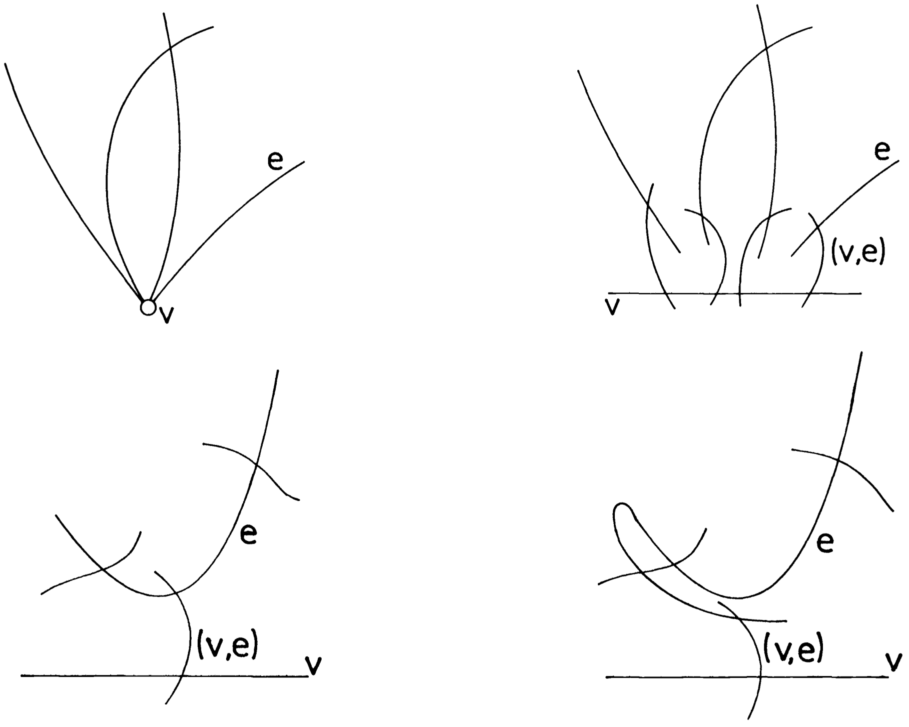


FIG. 3

DEFINITION 3.3. A graph G is called an *outerstring graph* if it has a string representation with all strings lying inside a halfplane and intersecting the borderline each in exactly one point (different strings in different points). Such a representation is called an *outerstring representation* of G.

DEFINITION 3.4. A graph G is called a *strong outerstring graph* if, for every ordering $v_1, v_2, \cdots, v_n$ of its vertices, there exists an outerstring representation of G with the strings intersecting the borderline in the ordering $v_1, v_2, \cdots, v_n$.

Note that instead of requiring the strings to lie inside a halfplane, we could require them to lie inside a disc and intersect the boundary of the disc each in one point. Obviously, this yields the same class of graphs (cf. [K1]).

LEMMA 3.5. *Let G_1, G_2, be strong outerstring graphs. Then the graph $G_1 + G_2$ is strong outerstring as well.* (*Here $G_1 + G_2$ is the Zykov sum of G_1 and G_2, i.e., the graph obtained from the disjoint union of G_1 and G_2 by adding all the edges between G_1 and G_2.*)

Proof. Denote $G = G_1 + G_2$, $n_1 = |V(G_1)|$, $n_2 = |V(G_2)|$, and $n = n_1 + n_2$. Let v: $\{1, 2, \cdots, n\} \twoheadrightarrow_{1-1} V(G)$ be an ordering of $V(G)$. Denote by $v^{(i)}$, $i = 1, 2$, the linear orderings $v^{(i)}: \{1, 2, \cdots, n_i\} \twoheadrightarrow V(G_i)$ inherited from v (i.e., for $u, u' \in V(G_i)$, $(v^{(i)})^{-1}(u) < (v^{(i)})^{-1}(u')$ if and only if $v^{-1}(u) < v^{-1}(u')$).

Fix a halfplane π with a borderline b, call b vertical. For $i = 1, 2$, fix outerstring representations R_i of G_i respecting the orderings $v^{(i)}$ and with strings lying in π. Let the representations be placed so that all the strings of R_1 lie above all the strings of R_2.

Consider a line b' parallel to b and lying outside π. Construct an outerstring representation of G that respects the ordering v as follows:

1. The strings of R_1 are extended by horizontal segments to reach the new borderline b';

2. The strings of R_2 are extended by "V-shaped" piecewise linear curves to reach the line b' so that all of them intersect all the horizontal extensions of the strings of R_1;

3. Steps 1 and 2 are done so that the endpoints of the extensions on b' respect the ordering v (cf. Fig. 4). □

Let us remark that Lemma 3.5 follows also from a characterization of strong outerstring graphs given in [K1].

COROLLARY 3.6. *Every complete multipartite graph* (*i.e., so called Turán graph*) *is strong outerstring.*

4. A modified planar 3-sat problem.

In the next section, we will use the following modification of the *planar satisfiability* problem:

PLANAR CYCLE 3-SAT.

(6) Instance: A formula ϕ with a set of clauses C over a set of variables X satisfying that

(i) every clause contains at most three literals;

(ii) there exists an ordering $c_1, c_2, \cdots, c_m$ of the clauses such that the graph

$G'_\phi = (X \cup C, \{xc \mid x \in c \in C \text{ or } \bar{x} \in c \in C\} \cup \{c_ic_{i+1} \mid i = 1, 2, \cdots, m\})$ is planar (here $c_{m+1} = c_1$).

Question: Is ϕ satisfiable?

Without the cycle $\{c_ic_{i+1} \mid i = 1, 2, \cdots, m\}$ in (ii), (6) would be the *planar* 3-*sat* problem that was proved to be NP-complete in [L] (cf. also [GJ]). Lichtenstein also proves that a cycle passing through all the vertices X of G_ϕ can be added without destroying planarity. We use a similar argument to prove the following proposition.

PROPOSITION 4.1. *The* planar cycle 3-sat *problem is* NP-*complete.*

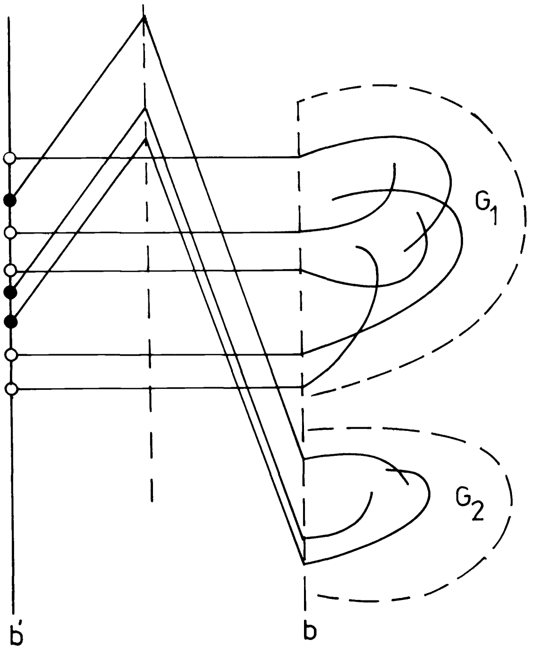


FIG. 4

Proof. We prove 3-*sat* $\propto$ *planar cycle* 3-*sat*. Consider a formula F with a set of 3-clauses C over a set of variables X. Draw the graph

$$G_F = (X \cup C, \{xc \mid x \in c \in C \text{ or } \bar{x} \in c \in C\})$$

as shown in Fig. 5(a) (the clause vertices are lined up at the left, and the variable vertices at the bottom). Every edge-crossing is replaced by a crossover box depicted in Fig. 5(b) and a graph (corresponding to a formula, say ϕ) depicted schematically in Fig. 5(c) is thus obtained. (In figures, variable vertices are depicted as circles, and clause vertices as squares. The signs + and − by an edge xc indicate that $x \in c$ or $\bar{x} \in c$, respectively). The crossover box is constructed so that in every satisfying truth assignment, $a_1 \Leftrightarrow a_2$ and $b_1 \Leftrightarrow b_2$ hold (and satisfying truth assignments exist for all values of $a_1 \Leftrightarrow a_2$ and $b_1 \Leftrightarrow b_2$). Hence F is satisfiable if and only if ϕ is. So far we have followed Lichtenstein [L], and the reader may check the details there.

It remains to show that we can add a cycle passing through all clause vertices of ϕ without destroying planarity. This is done as shown in Fig. 6(a) (note that we are adding fake crossover boxes to make pairs of neighbouring columns of boxes of the same height). Three ways the boxes are passed through by the cycle are depicted in Fig. 6(b). □

5. Noncrossing paths and cycles. In this section we prove the NP-completeness of *noncrossing path*, *noncrossing cycle*, and some related problems. Recall that in view of (1), we are considering the following problems.

NONCROSSING PATH.
Instance: A T-graph G_T and vertices $u, v \in V(G_T)$.
Question: Does G_T contain a noncrossing path connecting u and v?

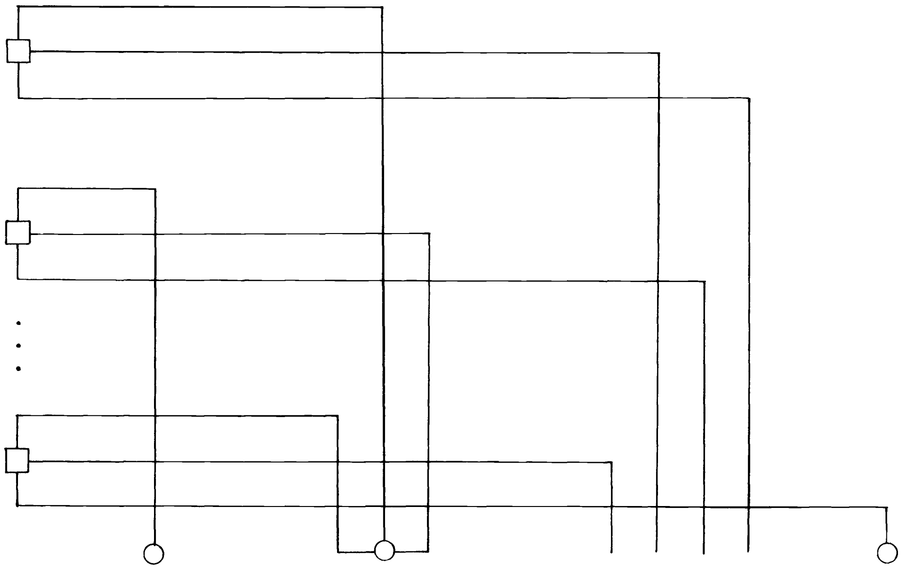


FIG. 5(a)

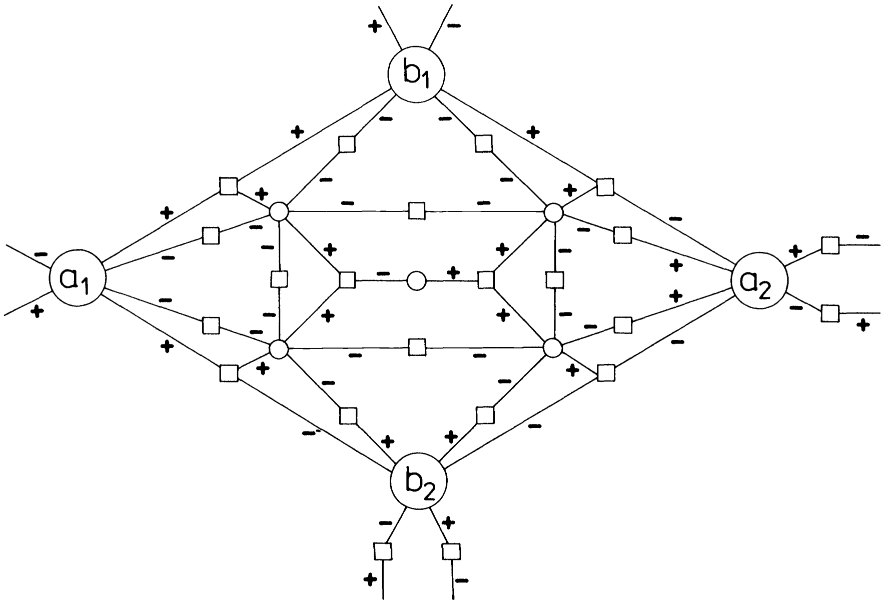


FIG. 5(b)

NONCROSSING CYCLE.
Instance: A T-graph G_T.
Question: Does G_T contain a noncrossing cycle?

Here we have Theorem 5.1.

THEOREM 5.1. *The* noncrossing path *and* noncrossing cycle *problems are* NP-*complete*.

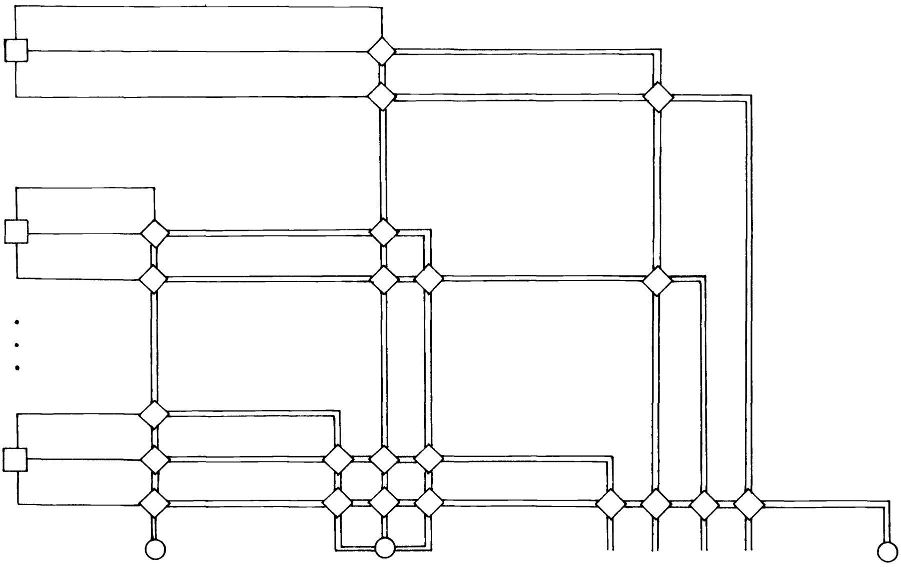


FIG. 5(c)

Proof of Theorem 5.1. 1. *Noncrossing path*. Let ϕ be a formula with an order $c_1, c_2, \cdots, c_m$ of the clauses such that the graph G'_ϕ is planar. We define an AT-graph $H = (V, E, I)$ as follows:

$$V = \{c_1, c_2, \cdots, c_m\} \cup \{c'_1, c'_2, \cdots, c'_m\},$$

$$E = \{c'_i c_{i+1} \mid i = 1, 2, \cdots, m-1\} \cup \{[c_i, c'_i]_x \mid x \in c_i \text{ or } \bar{x} \in c_i, 1 \leqq i \leqq m\}.$$

(Warning: this is considered as a multigraph; for every variable x with $x \in c_i$ or $\bar{x} \in c_i$ we have an edge $[c_i, c'_i]_x$.)

$$I = \{\{[c_i, c'_i]_x, [c_j, c'_j]_x\} \mid x \in c_i \text{ and } \bar{x} \in c_j\}$$
$$\cup \{\{[c_i, c'_i]_x, [c_i, c'_i]_y\} \mid (x \in c_i \text{ or } \bar{x} \in c_i) \text{ and } (y \in c_i \text{ or } \bar{y} \in c_i)\}.$$

Thus the underlying graph (V, E) has the shape depicted in Fig. 7. We claim that in H, there exists a path from c_1 to c'_m that does not contain two edges forming a pair from I if and only if ϕ is satisfiable. If P is such a path, then for every i there exists a unique x such that $[c_i, c'_i]_x \in P$. We put

$$x = \begin{cases} \text{TRUE if } [c_i, c'_i]_x \in P \text{ and } x \in c_i \text{ for some } i, \\ \text{FALSE if } [c_i, c'_i]_x \in P \text{ and } \bar{x} \in c_i \text{ for some } i; \end{cases}$$

truth values of other variables are chosen arbitrarily. It follows that every clause is satisfied and this definition is correct, since if $x \in c_i$ and $\bar{x} \in c_j$ then $[c_i, c'_i]_x$ and $[c_j, c'_j]_x$ cannot both be in P. Conversely, if ϕ is satisfiable then for each i, consider an edge $[c_i, c'_i]_x$ for which c_i contains a TRUE literal x. These edges, together with edges $c'_i c_{i+1}$, $i = 1, 2, \cdots, m-1$ form a noncrossing path from c_1 to c'_m.

It remains to show that the AT-graph H corresponds to a T-graph H_T. We obtain a planar representation of H_T by a (local) modification of the planar drawing of G'_ϕ:

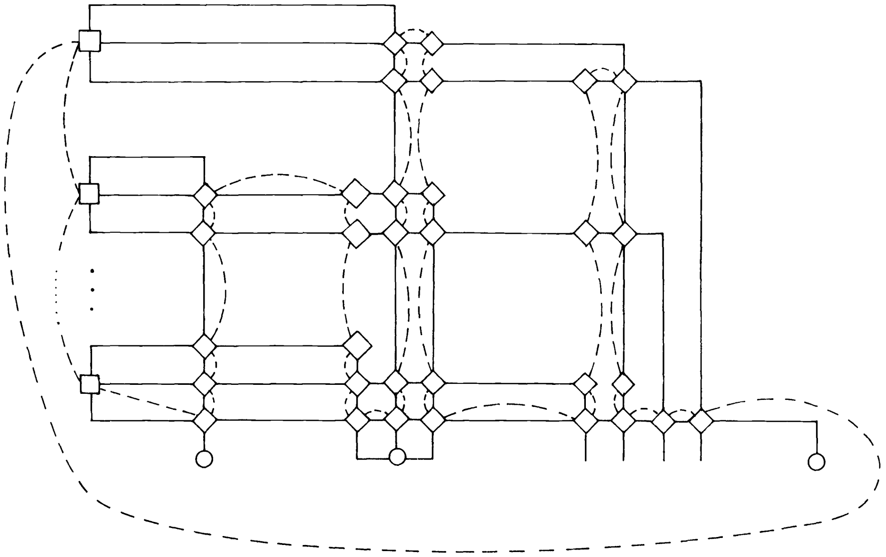


FIG. 6(a)

First we "double" vertices that correspond to clauses and "omit" vertices that correspond to variables, as indicated in Fig. 8. It is clear that we can achieve the intersections of edges within each clause; see Fig. 9.

Finally, it is necessary to realize intersections of edges within variables. This leads to the following problem: We consider the vertex x of G'_ϕ, which corresponds to a variable x and the edges incident with x. These edges are either positive or negative according to the occurrence of either x or $\bar{x}$ in the corresponding clauses. Now we want to replace every edge by an arc in such a way that any two arcs intersect if and only if they correspond to edges of different signs. However, this is possible to do locally as the desirable intersection graph is complete bipartite and, by Corollary 3.6, it is strong outerstring. See Fig. 10.

Putting an extra vertex in the middle of each multiple edge, we get rid of multiple edges. Denoting the resulting AT-graph by H', we see that H' contains a noncrossing path connecting c_1 and c'_m if and only if ϕ is satisfiable.

2. *Noncrossing cycle*. Follow the previous proof, starting with

$$E=\{c'_i c_{i+1} \mid i=1,2,\cdots,m\}\cup\{[c_i,c'_i]_x \mid x\in c_i \text{ or } \bar{x}\in c_i,\ 1\leqq i\leqq m\} \quad (c_{m+1}=c_1)$$

in the definition of the AT-graph H. □

Remark 5.2. With a simple modification of the above construction, we can show that *noncrossing path* and *noncrossing cycle* remain NP-complete for layouts of (simple) 3-regular graphs.

Spanning trees play a key role in many graph-theoretical and optimization questions. Hence it is natural to consider the problem *noncrossing spanning tree* (which is defined as problem (1) for P = "to be a spanning tree"). It follows directly from the proof of Theorem 5.1 that the following corollary holds.

COROLLARY 5.3. *The* noncrossing spanning tree *problem is* NP-*complete*.

It is straightforward to call a T-graph *noncrossing connected* if every two vertices are connected by a noncrossing path. Note that a noncrossing connected graph need not contain a noncrossing spanning tree (cf. the graph depicted in Fig. 11). Thus the following problem does not coincide with *noncrossing spanning tree*.

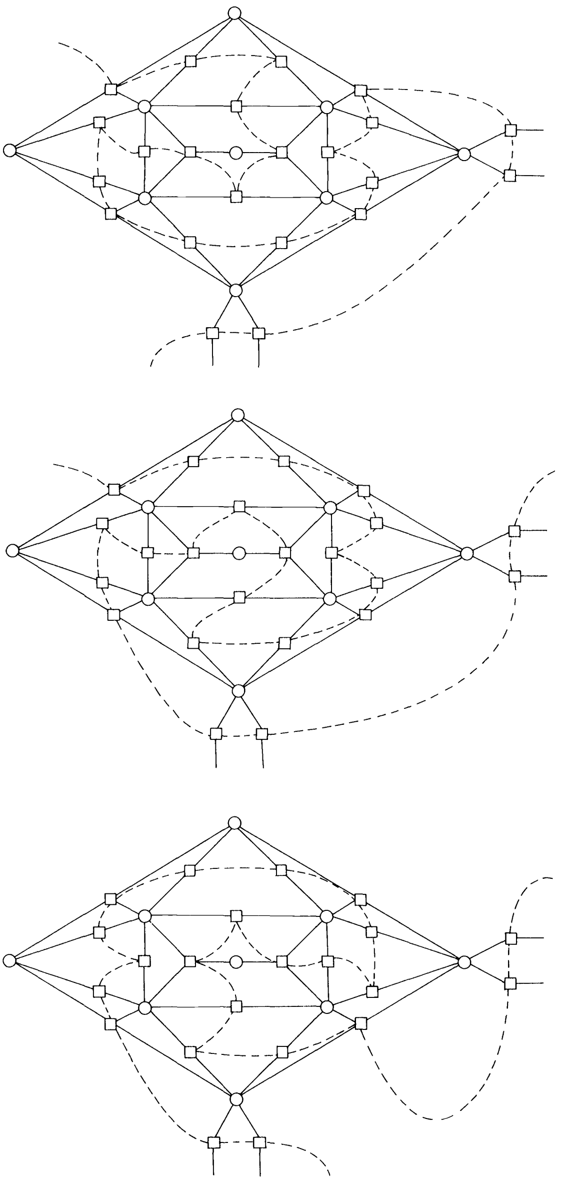


FIG. 6(b)

NONCROSSING CONNECTEDNESS.

Instance: A T-graph G_T.

Question: Is G_T noncrossing connected?

However, the AT-graph H' from the proof of part 1 of Theorem 5.1 is noncrossing connected if and only if it contains a noncrossing path from c_1 to c'_m (and then contains a noncrossing spanning tree). Hence the following corollary holds.

COROLLARY 5.4. *The* noncrossing connectedness *problem is* NP-*complete.*

Remark 5.5. We can again show easily that *noncrossing spanning tree* and *noncrossing connectedness* are NP-complete for layouts of 3-regular graphs.

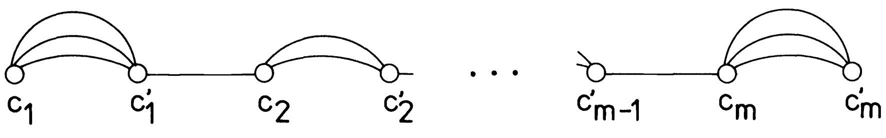


FIG. 7

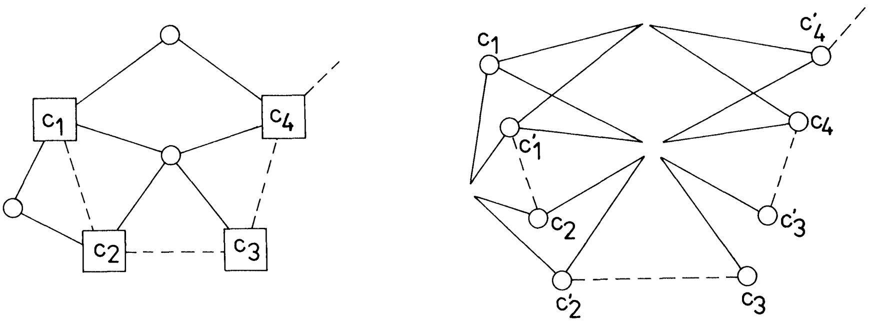


FIG. 8

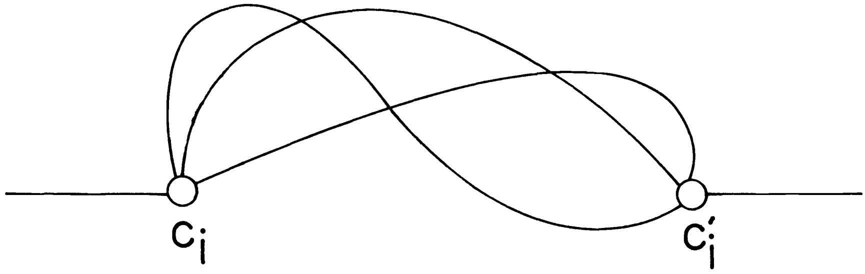


FIG. 9

6. Noncrossing matching. In this section we consider the following problem, which arises from one of the basic optimization problems in graph theory.

NONCROSSING MATCHING.

Instance: A T-graph G_T and a positive integer k.

Question: Does G_T contain a noncrossing matching of size $\geqq k$?

THEOREM 6.1. *The* noncrossing matching *problem is* NP-*complete.*

Proof. In view of the remark before Theorem 3.1, *noncrossing matching* in 1-regular T-graphs is exactly the *independent set* problem for string graphs. Every planar graph is a string graph ([EET, KGK, Si]) and *independent set* is NP-complete in planar graphs [GJ]. Hence *noncrossing matching* is NP-complete even in layouts of 1-regular graphs. □

7. Noncrossing 1-factor. A k-factor is a spanning k-regular subgraph. A 1-factor (also called a perfect matching) in a graph G is thus a matching of size $|V(G)|/2$. In the last two sections we are concerned with the following problem.

NONCROSSING k-FACTOR.

Instance: A T-graph G_T.

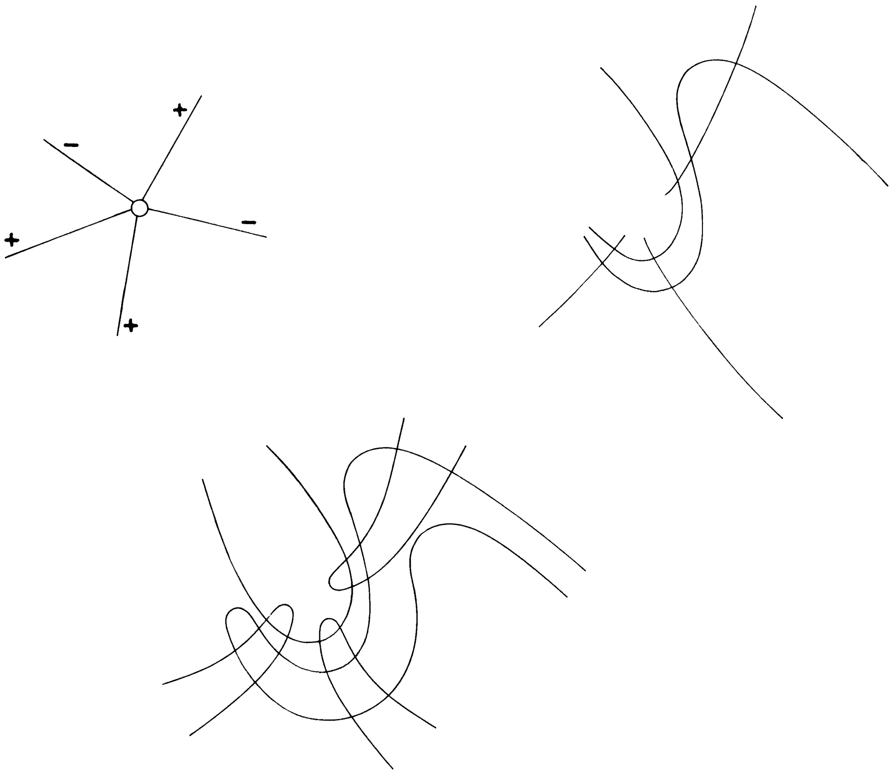


FIG. 10

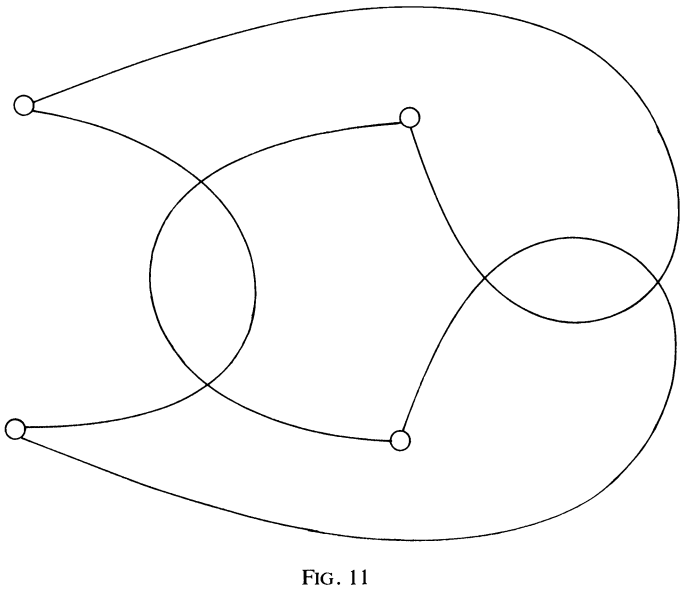


FIG. 11

Question: Does G_T contain a noncrossing k-factor?

We have the following results.

THEOREM 7.1. *The* noncrossing 1-factor *problem is polynomially solvable for layouts of graphs with maximum degree* $\leqq 2$, *and it is* NP-*complete for* 3-*regular* T-*graphs*.

THEOREM 7.2. *For* $k = 2, 3, 4, 5$, noncrossing k-factor *is* NP-*complete in* $(k+1)$-*regular* T-*graphs*.

THEOREM 7.3. *For* $k \geqq 6$, noncrossing k-factor *becomes trivial*, *but it is* NP-*complete if the input is a* (*nonrealizable*) AT-*graph*.

The first (polynomial) part of Theorem 7.1 will be proved in this section. The NP-complete results will be proved in a unified way in the next section.

Proof of Theorem 7.1, *part* 1. We prove that *noncrossing 1-factor* is polynomially solvable even for AT-graphs with maximum degree $\leqq 2$. (A subgraph (V, F) of an AT-graph (V, E, I) is noncrossing if $e \neq e'$, $e, e' \in F$ imply $\{e, e'\} \notin I$.) Let $H = (V, E, I)$ be an AT-graph such that the underlying graph $G = (V, E)$ has all degrees $\leqq 2$. Let $G_1, G_2, \cdots, G_k, G_{k+1}, \cdots, G_s$ be the connected components of G ordered so that $G_1, \cdots, G_k$ are cycles and $G_{k+1}, \cdots, G_s$ are paths. Explicitly, put

$$G_i = (\{v_1^i, \cdots, v_{n_i}^i\}, \{v_1^i v_2^i, \cdots, v_{n_i}^i v_1^i\}) \quad \text{for } i \leqq k,$$

$$G_i = (\{v_1^i, \cdots, v_{n_i}^i\}, \{v_1^i v_2^i, \cdots, v_{n_i-1}^i v_{n_i}^i\}) \quad \text{for } k+1 \leqq i \leqq s.$$

Without loss of generality we may suppose that all n_i's are even (otherwise H has no 1-factor), and we put

$$C_i = \{v_1^i v_2^i, v_3^i v_4^i, \cdots\}, C_i' = \{v_2^i v_3^i, v_4^i v_5^i, \cdots\} \quad \text{for } i = 1, 2, \cdots, s.$$

Suppose (V, F), $F \subset E$ is a *1*-factor in G and denote $F_i = F \cap \binom{V(G_i)}{2}$ for $i = 1, 2, \cdots, s$. Then $F_i = C_i$ or $F_i = C_i'$ for $i = 1, 2, \cdots, k$, and $F_i = C_i$ for $i = k+1, \cdots, s$. We encode this fact via a formula ϕ with variables $x_1, x_2, \cdots, x_s$ so that x_i is TRUE if and only if $F_i = C_i$, and ϕ is satisfied by a truth valuation if and only if the corresponding 1-factor is noncrossing. We let ϕ be the conjunction of the following clauses:

1. x_i for $i = k+1, \cdots, s$;

2. $\bar{x}_i$ if C_i is not noncrossing (i.e., the graph $(C_i, I \cap \binom{C_i}{2})$) fails to be discrete), $i = 1, 2, \cdots, s$;

3. x_i if C_i' is not noncrossing, $i = 1, 2, \cdots, k$;

4. $\bar{x}_i \vee \bar{x}_j$ if $C_i \cup C_j$ is not noncrossing, $i, j = 1, 2, \cdots, s$;

5. $\bar{x}_i \vee x_j$ if $C_i \cup C_j'$ is not noncrossing, $i \leqq s, j \leqq k$;

6. $x_i \vee x_j$ if $C_i' \cup C_j'$ is not noncrossing, $i, j = 1, 2, \cdots, s$.

Straightforwardly, ϕ is satisfiable if and only if H contains a noncrossing 1-factor. This completes the proof, since satisfiability of formulas with binary clauses can be decided in polynomial time [GJ]. □

8. Noncrossing *k*-factors. We will prove the NP-completeness of general factors by means of the "signal-sender" technique. It is more convenient to formulate and prove a more general statement.

Let P and R be graph properties that are connected (i.e., a graph G has the property P (respectively, R) if and only if each connected component of G has P (respectively, R)).

DEFINITION 8.1. A T-graph $G_T = (V_T, E_T)$ together with edges t, f is called a 2-*gadget* if it has the following properties.

1. The edges t and f intersect each other and both meet the outer face of the layout G_T;

2. There are subsets F_t and F_f of E_T such that
 (i) the graphs (V_T, F_t) and (V_T, F_f) are noncrossing subgraphs of G_T;
 (ii) (V, F_t) and (V, F_f) have property P;
 (iii) $F_t \cap \{t, f\} = \{t\}, F_f \cap \{f, t\} = \{f\}$;
3. If a subset F of E_T is noncrossing and has property P then $|F \cap \{t, f\}| = 1$;
4. The graph $G = (V, E)$ has property R.

DEFINITION 8.2. A T-graph $H_T = (V'_T, E'_T)$ together with edges e_1, e_2, e_3 is called a 3-*gadget* if it has the following properties.

1. The edges e_1, e_2, e_3 meet in the clockwise order the outer face of the layout H_T;
2. There are subsets $F_i \subset E'_T$, $i = 1, 2, 3$ such that
 (i) the graphs (V'_T, F_i), $i = 1, 2, 3$ are noncrossing subgraphs of H_T;
 (ii) the graphs (V'_T, F_i), $i = 1, 2, 3$ have property P;
 (iii) $F_i \cap \{e_1, e_2, e_3\} = \{e_i\}$ for $i = 1, 2, 3$;
3. If F is a noncrossing subset of E'_T and has property P then $F \cap \{e_1, e_2, e_3\} \neq \emptyset$;
4. The graph $H = (V', E')$ has property R.

We consider the following general problem.

NONCROSSING P-SUBGRAPH IN A TOPOLOGICAL R-SUBGRAPH (PTR).

Instance: A T-graph G_T with property R.

Question: Does G_T contain a noncrossing subgraph with property P?

THEOREM 8.3. *Let P and R be connected graph properties such that* 2- *and* 3-*gadgets exist. Then the* PTR *problem is* NP-*complete.*

Proof. We will prove *planar* 3-*sat* $\propto$ PTR. Let ϕ be a formula with a set C of 3-clauses over a set of variables X. We assume that the graph $H_\phi = (X \cup C, \{xc \mid x \in c \in C \text{ or } \overline{x} \in c \in C\})$ is planar.

Let us fix a noncrossing planar drawing of H_ϕ and, for every clause c, we list its variables clockwise $x_1(c)$, $x_2(c)$, $x_3(c)$ (according to the edges that leave c in direction to $x_i(c)$ in the drawing of H_ϕ).

For every $x \in X$, let $G_T(x)$ be a copy of the 2-gadget G_T (concerning gadgets we preserve the above notation). The copies of edges t and f are denoted by t^x and f^x, respectively. Similarly, for a clause $c \in C$, we denote by $H_T(c)$ a copy of the 3-gadget H_T; the copies of the edges e_1, e_2, e_3 are denoted by $e^c_{x_1(c)}$, $e^c_{x_2(c)}$, $e^c_{x_3(c)}$, respectively.

We define an AT-graph $H(\phi)$ as follows:

$$V(H(\phi)) = \bigcup_{x \in X} V(G_T(x)) \cup \bigcup_{c \in C} V(H_T(c)),$$

$$E(H(\phi)) = \bigcup_{x \in X} E(G_T(x)) \cup \bigcup_{c \in C} E(H_T(x)),$$

$$I(H(\phi)) = \bigcup_{x \in X} I(G_T(x)) \cup \bigcup_{c \in C} I(H_T(c)) \cup \{\{f^x, e^c_x\} \mid x \in c \in C\}$$

$$\cup \{\{t^x, e^c_x\} \mid \bar{x} \in c \in C\}.$$

Since $H(\phi)$ is a disjoint union of graphs $G_T(x)$ and $H_T(c)$ it has property R. It is also easy to see that the AT-graph $H(\phi)$ is realizable, i.e., that it corresponds to a T-graph $H_T(\phi)$: a realization of $H(\phi)$ is obtained from the planar drawing of H_ϕ by replacing every vertex x by a layout $G_T(x)$, and every vertex c by a layout $H_T(c)$. Intersections of edges t^x (respectively, f^x) with e^c_x are realized by extending the edges t^x (respectively, f^x) in the direction of the edge xc of H_ϕ.

We prove that $H(\phi)$ contains a noncrossing subgraph with property P if and only if ϕ is satisfiable:

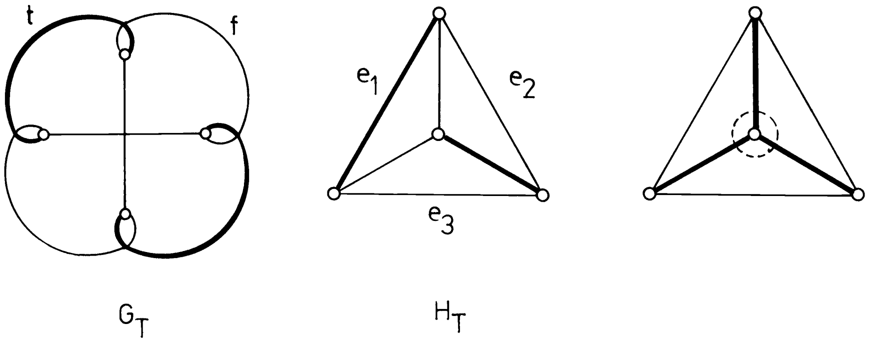


FIG. 12

1. Let $H(\phi)$ contain a noncrossing subgraph with property P, say $F \subset E(H(\phi))$ is such that $(V(H(\phi)), F)$ has property P and no two edges from F form a pair from $I(H(\phi))$. Since the property P is connected, we have

$$F = \bigcup_{x \in X} F(x) \cup \bigcup_{c \in C} F(c),$$

where $F(x) \subset E(G_T(x))$, $x \in X$, and $F(c) \subset E(H_T(c))$, $c \in C$ are noncrossing and have property P.

Define a truth assignment $\varphi : X \rightarrow \{\text{TRUE}, \text{FALSE}\}$ by

$$\varphi(x) = \begin{cases} \text{TRUE} & \text{if } t^x \in F(x), \\ \text{FALSE} & \text{if } f^x \in F(x). \end{cases}$$

By condition 3 of Definition 8.1, $|F(x) \cap \{t^x, f^x\}| = 1$, and $\varphi(x)$ is defined correctly.

Now consider a clause $c \in C$. By Definition 8.2.3, there is an $x \in X$ such that $e_x^c \in F(c)$. We have either $x \in c$ or $\bar{x} \in c$. If $x \in c$ then f^x and e_x^c intersect and (as F is noncrossing) we get $t^x \in F(x)$. Thus $\varphi(x) = \text{TRUE}$ and c is satisfied. On the other hand, if $\bar{x} \in c$ then t^x and e_x^c intersect and we get $f^x \in F(x)$. Thus $\varphi(x) = \text{FALSE}$ and c is satisfied. It follows that φ is a satisfying truth assignment for ϕ.

2. Let ϕ be satisfied by a truth valuation $\varphi : X \rightarrow \{\text{TRUE}, \text{FALSE}\}$. For every clause $c \in C$, denote by $g(c)$ a variable x that guarantees the validity of c (i.e., either $x \in c$ and $\varphi(x) = \text{TRUE}$ or $\bar{x} \in c$ and $\varphi(x) = \text{FALSE}$; $g(c)$ need not be unique).

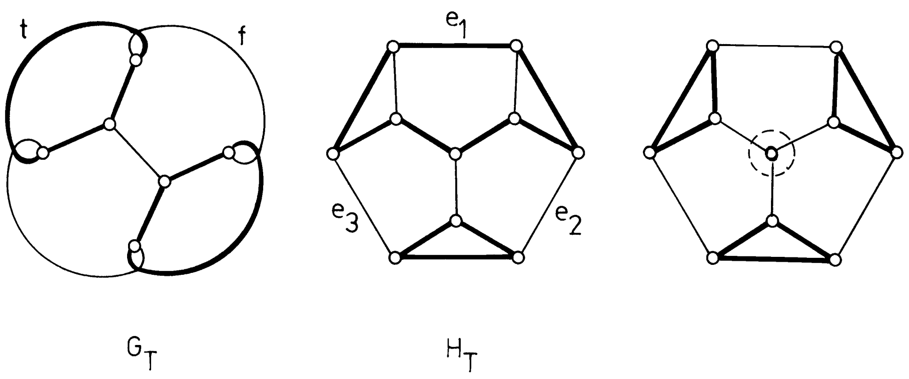


FIG. 13

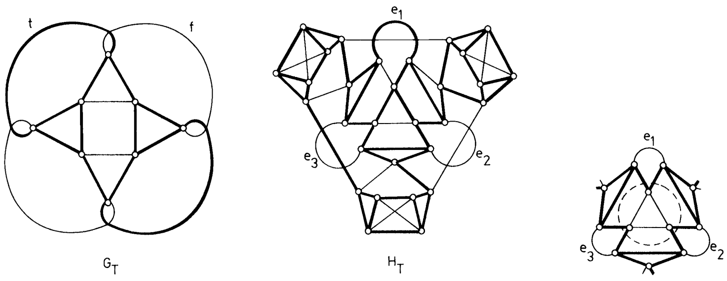


FIG. 14

We put

$$F(x)=\begin{cases} F_t(x)(\text{i.e., } t^x\in F_t(x)) & \text{if } \varphi(x)=\text{TRUE},\\ F_f(x)(\text{i.e.,} f^x\in F_f(x)) & \text{if } \varphi(x)=\text{FALSE}\end{cases}$$

for every $x\in X$ (cf. Definition 8.1.2).

$F(c) = F_i(c)$ (i.e., $e^c_{x_i(c)}\in F_i(c)$), where $g(c) = x_i(c)$ for every $c\in C$ (cf. Definition 8.2.2) and

$$F=\bigcup_{x\in X} F(x)\cup\bigcup_{c\in C} F(c).$$

Since P is a connected property, $(V(H(\phi)), F)$ has P. We can show easily that $(V(H(\phi)), F)$ is a noncrossing subgraph of $H(\phi)$. □

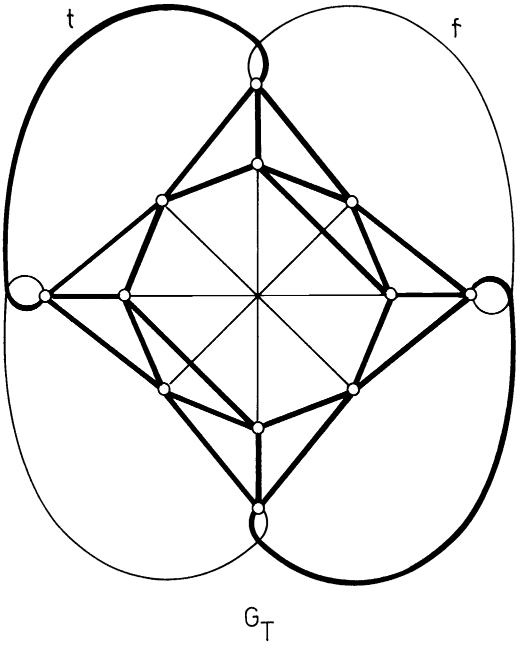


FIG. 15

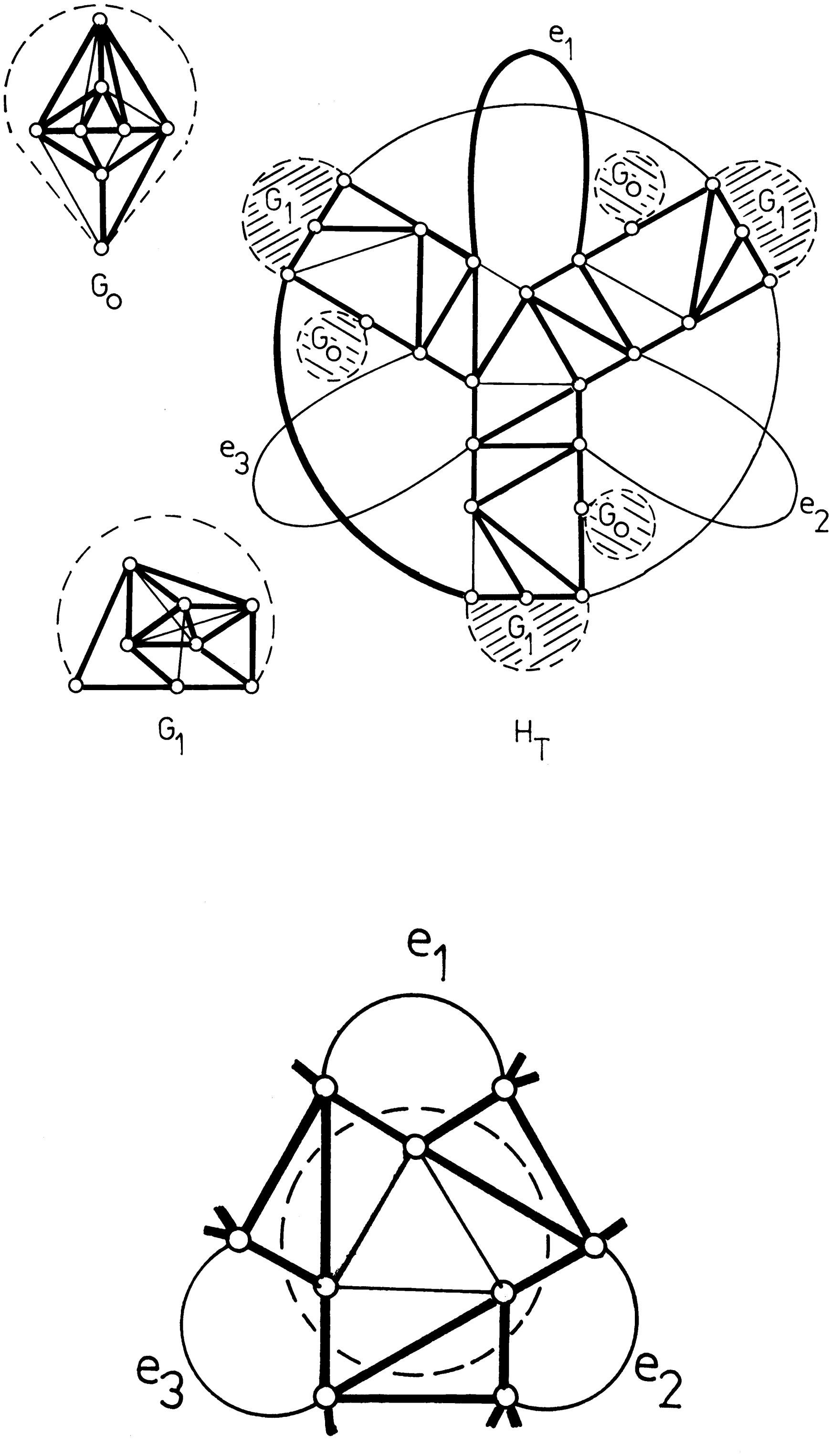


FIG. 15 (*continued*).

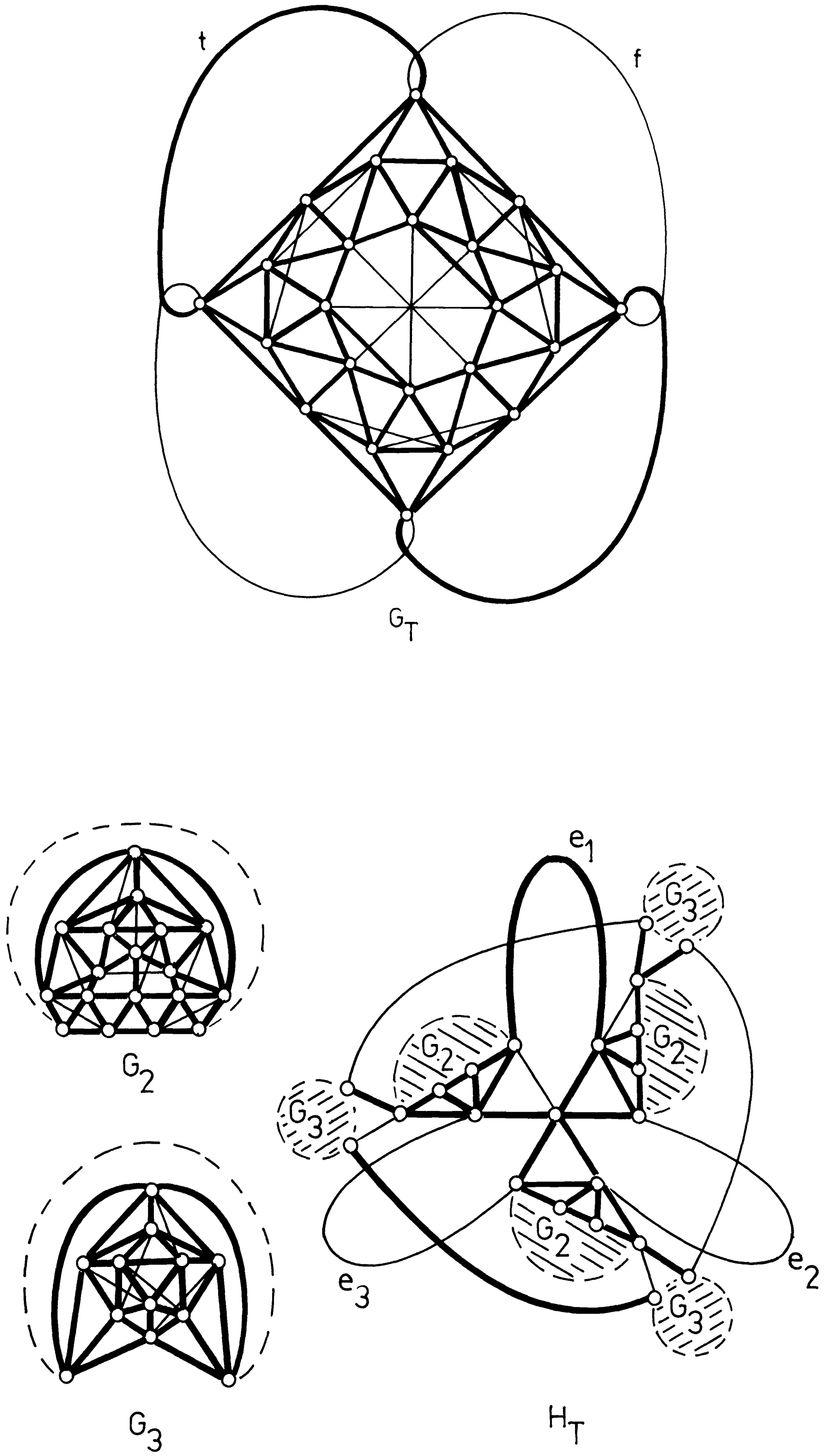


FIG. 16

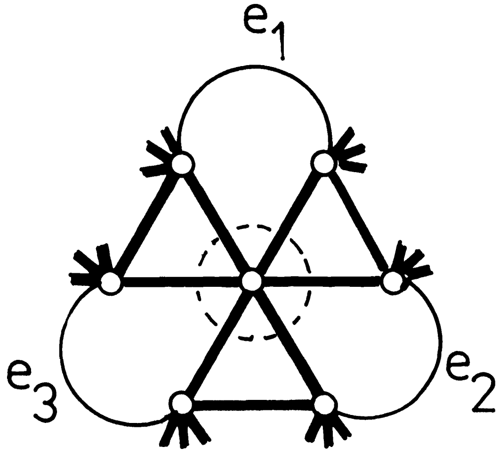


FIG. 16 (*continued*).

Proof of Theorem 7.1, *part* 2, *and proof of Theorem* 7.2. For the following pairs of properties P, R, the PTR problem is NP-complete:

	P	R
A	1-factor	3-regular
B	2-factor	3-regular
C	3-factor	4-regular
D	4-factor	5-regular
E	5-factor	6-regular

According to Theorem 8.3, it suffices to exhibit corresponding gadgets G_T and H_T. Gadgets for cases A, B, C, D, E are depicted in Figs. 12, 13, 14, 15, and 16, respectively. Note that only the factors F_t and F_1 are indicated; factors F_f and F_2, F_3 follow from the symmetry. The nonexistence of a factor $F_0 \subset E(H_T)$ that would be disjoint with $\{e_1, e_2, e_3\}$ is illustrated in part (c) of each figure by indicating those edges that would have to be contained in every such factor. That would lead to high (or low) degrees of some vertices (indicated by a circle). □

Proof of Theorem 7.3. It is well known that every planar graph contains a vertex of degree $\leqq 5$, and so no k-regular graph with $k \geqq 6$ admits a noncrossing drawing in the plane. Thus for $k \geqq 6$, the answer for any instance of *noncrossing k-factor* is negative.

The latter part of the theorem is proved via Theorem 8.3 again (more precisely, via a modified version of Theorem 8.3, since now we have AT-graphs at the input and not T-graphs. Hence the gadgets are AT-graphs as well and they are not necessarily realizable.). We show here an explicit construction of gadgets in the case of k being even.

Define an AT-graph $H = (V, E, I)$ by

$$V = \{a,b,c,d\} \cup \{x_i, y_i \mid i = 1,2,\cdots,k-1\},$$

$$E = \{ab, bc, cd, ad\} \cup \{ax_i, bx_i, cy_i, dy_i, x_iy_i \mid i = 1,2,\cdots,k-1\}$$

$$\cup \{x_ix_j, y_iy_j \mid 1 \leqq i < j \leqq k-1\},$$

$$I = \{\{ab, bc\}\},$$

$$t = ab, f = bc.$$

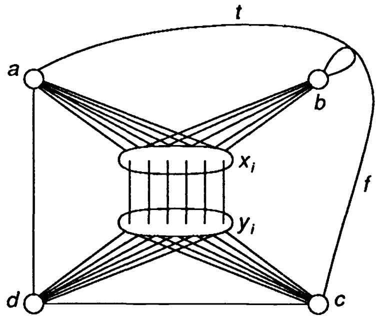


FIG. 17(a)

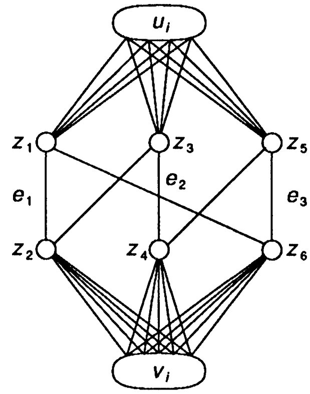


FIG. 17(b)

It is easily seen that H is $(k + 1)$-regular, every noncrossing k-factor in H contains either the edge t, or the edge f, and H contains noncrossing k-factors F_t and F_f (e.g., $F_t = E - (\{bc, ad\} \cup \{x_iy_i \mid i = 1, 2, \cdots, k - 1\}))$. Thus we take H for a 2-gadget. See Fig. 17(a).

We define an AT-graph $H' = (V', E', I')$ by

$$V' = \{z_1, z_2, \cdots, z_6\} \cup \{u_i, v_i \mid i = 1, 2, \cdots, k-1\},$$

$$E' = \{z_1z_2, z_2z_3, \cdots, z_6z_1\} \cup \{z_iu_j, z_{i+1}v_j \mid i = 1, 3, 5, j = 1, 2, \cdots, k-1\}$$
$$\cup \{u_iu_j, v_iv_j \mid 1 \leqq i < j \leqq k-1\},$$

$$I' = \emptyset,$$

$$e_1 = z_1z_2, \quad e_2 = z_3z_4, \quad e_3 = z_5z_6.$$

Again, it is easily seen that H' is $(k + 1)$-regular and that it contains (noncrossing) k-factors containing exactly one of the edges e_1, e_2, e_3 (e.g., $F_1 = E' - (\{e_2, e_3, z_1u_{k-1}, z_2v_{k-1}\} \cup \{u_iu_{i+1}, v_iv_{i+1} \mid i = 1, 3, 5, \cdots\}))$. Suppose $F \subset E'$ is a k-factor such that $F \cap \{e_1, e_2, e_3\} = \emptyset$. Then $\{z_iu_j \mid i = 1, 3, 5, j = 1, 2, \cdots, k - 1\} \subset F$ and the subgraph induced by F on $\{u_i \mid i = 1, 2, \cdots, k - 1\}$ should be $(k - 3)$-regular. This is impossible, since k is even and hence both $k - 1$ and $k - 3$ are odd. Thus we take H' for a 3-gadget and the proof is completed. See Fig. 17(b). The construction of a 3-gadget in the case of k being odd is a bit more technical. □

Note that the *noncrossing* k-factor problem for $k \geqq 6$ is thus the only case presented in this paper for which a topological embedding of the input AT-graph (or more precisely its realizability) actually matters.

REFERENCES

[B] F. Barahona, *On via minimization*, preprint OR 87491, Universität Bonn, 1987.

[EET] G. Ehrlich, S. Even, and R. E. Tarjan, *Intersection graphs of curves in the plane*, J. Combin. Theory Ser. B, 21 (1976), pp. 8–20.

[GJ] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of* NP-*Completeness*, W. H. Freeman, San Francisco, 1979.

[K1] J. Kratochvíl, *String graphs* I. *There are infinitely many critical nonstring graphs*, J. Combin. Theory Ser. B, to appear.

[K2] ———, *String graphs* II. *Recognizing string graphs is* NP-*hard*, J. Combin. Theory Ser. B, to appear.

[KGK] J. Kratochvíl, M. Goljan, and P. Kučera, *String Graphs*, Academia, Prague, 1986.

[KM] J. Kratochvíl and J. Matoušek, *String graphs requiring exponential representations*, J. Combin. Theory Ser. B, to appear.

[L] D. Lichtenstein, *Planar formulae and their uses*, SIAM J. Comput., 11 (1982), pp. 329–343.

[MNT] J. Matoušek, J. Nešetřil, and R. Thomas, *On polynomial time decidability of induced minor closed classes*, Comment. Math. Univ. Carolin., 29, 4 (1988), pp. 703–711.

[N] J. Nešetřil, *Problem*, in Selected problems from 15th Winter School on Abstract Analysis, KAM Series 87-33.

[RS] N. Robertson and P. Seymour, *Graph minors* I.–XVIII.

[Sch] A. Schrijver, *Decomposition of graphs on surfaces and homotopic circulation theorem*, preprint.

[Si] F. W. Sinden, *Topology of thin film* RC-*circuits*, Bell System Tech. J., (1966), pp. 1639–1662.

[KvL] M. R. Kramer and J. Van Leeuwen, *The complexity of wire routing and finding minimum area layouts for arbitrary* VLSI *circuits*, in Advances in Computing Research, F. P. Preparata, ed., JAI Press, 1984, pp. 115–128.

SIAM J. DISC. MATH.
Vol. 4, No. 2, pp. 245-252, May 1991

009

THREE-DIMENSIONAL STABLE MATCHING PROBLEMS*

CHENG NG† AND DANIEL S. HIRSCHBERG†

Abstract. The stable marriage problem is a matching problem that pairs members of two sets. The objective is to achieve a matching that satisfies all participants based on their preferences. The stable roommate problem is a variant involving only one set, which is partitioned into pairs with a similar objective. There exist asymptotically optimal algorithms that solve both problems.

In this paper, the complexity of three-dimensional extensions of these problems is investigated. This is one of twelve research directions suggested by Knuth in his book on the stable marriage problem. It is shown that these problems are NP-complete, and hence it is unlikely that there exist efficient algorithms for their solutions.

The approach developed in this paper provides an alternate NP-completeness proof for the hospitals/residents problem with couples—an important practical problem shown earlier to be NP-complete by Ronn.

Key words. stable marriage problem, stable roommate problem, NP-complete problems, three-dimensional matching

AMS(MOS) subject classifications. 68Q25, 90B99

Introduction. Consider the problem of assigning $3n$ students to n disjoint work groups of three students each. The students must guard against any three individuals abandoning their assignments and instead conspiring to form a new group that they consider more desirable.

The following procedure is followed: each student ranks all $\frac{1}{2}(3n-1)(3n-2)$ possible pairs of fellow students according to his/her preference for working with the pairs. A *destabilizing triple* for an assignment M consists of three students such that each ranks the remaining two (as a pair) more desirable than the pair that he/she is assigned to in M. The students' task, the *3-person stable assignment problem* (or 3PSA for short), is to find a *stable assignment*, one that is free of all destabilizing triples, if such an assignment exists.

Readers will recognize that 3PSA is a three-dimensional generalization of the *stable roommate problem*, which partitions $2n$ persons into n pairs of stable roommates. A better known variation is the *stable marriage problem*, which divides the participants into two disjoint sets, male and female. Each pair in a stable marriage must include a male and a female. The stable marriage problem has a similar generalization in three dimensions, which we name the *3-gender stable marriage problem* (or 3GSM for short) and define in the next section.

The stable roommate and stable marriage problems have been studied extensively [3]–[5], [9]. There exist efficient algorithms for both problems that run in $O(n^2)$ time [1], [6], [10]. Ng and Hirschberg [12] have obtained lower bound results proving that these algorithms are asymptotically optimal. Since no significant improvement is possible on the original problems, it is then natural to consider their three-dimensional generalizations, 3GSM and 3PSA. This is one of twelve research directions suggested by Knuth in his treatise on the stable marriage problem [9].

In this paper, we show that both 3GSM and 3PSA are NP-complete. Hence, it is unlikely that fast algorithms exist for these problems. The NP-completeness of 3GSM has been independently established by Subramanian [15]. In [11] we extend the approach developed in this paper to the study of two problems dealing with the task of matching married couples to jobs.

* Received by the editors March 8, 1989; accepted for publication (in revised form) May 8, 1990.

† Department of Information and Computer Science, University of California–Irvine, Irvine, California 92717.

Definitions. An instance of 3GSM involves three finite sets A, B, and D. These sets have equal cardinality k, which is the *size* of the problem instance. A *marriage* in 3GSM is a complete matching of the three sets, i.e., a subset of $A \times B \times D$ with cardinality k such that each element of A, B, and D appears exactly once.

For each element a of A, we define its *preference*, denoted by $\geqq_a$, to be a linear order on the elements of $B \times D$. The intuitive meaning of $(\beta_1, \delta_1) \geqq_a (\beta_2, \delta_2)$ is that a prefers (β_1, δ_1) to (β_2, δ_2) in a marriage. For $b \in B$ and $d \in D$, there are also analogous definitions $\geqq_b$ and $\geqq_d$ on the Cartesian products $A \times D$ and $A \times B$, respectively. When the subscript in the relation is evident from context, we omit it from the $\geqq$ notation.

A marriage is *unstable* if there exists a triple $t \in A \times B \times D$ such that t is not in the marriage and each component of t prefers the pair that it is matched with in t to the pair that it is matched with in the actual marriage. A *stable marriage* is a marriage where no such *destabilizing triple* can be found. Formally, a stable marriage is a marriage M, such that, for all $(a, b, d) \notin M$ and for the triples $(a, \beta_1, \delta_1), (\alpha_2, b, \delta_2), (\alpha_3, \beta_3, d) \in M$; either $(\beta_1, \delta_1) \geqq_a (b, d)$, $(\alpha_2, \delta_2) \geqq_b (a, d)$, or $(\alpha_3, \beta_3) \geqq_d (a, b)$.

A 3PSA instance of size n involves a set S of cardinality $n = 3k$, where k is an integer. The *preference* of $s \in S$, denoted $\geqq_s$, is a linear order on the set of unordered pairs $\{\{s_1, s_2\} | s_1 \neq s_2 \text{ and } s_1, s_2 \in S - \{s\}\}$. A *stable assignment* M in 3PSA is a partition of S into k disjoint three-element subsets, such that, for all $\{s_1, s_2, s_3\} \notin M$ and for the subsets $\{s_1, \sigma_{11}, \sigma_{12}\}$, $\{s_2, \sigma_{21}, \sigma_{22}\}$, $\{s_3, \sigma_{31}, \sigma_{32}\} \in M$; either $\{\sigma_{11}, \sigma_{12}\} \geqq_{s_1} \{s_2, s_3\}$, $\{\sigma_{21}, \sigma_{22}\} \geqq_{s_2} \{s_1, s_3\}$ or $\{\sigma_{31}, \sigma_{32}\} \geqq_{s_3} \{s_1, s_2\}$.

When referring to preferences, we adopt the convention that items are listed in decreasing order of favor. For example, the listing $p_1 p_2 \cdots p_k$, where each p_i denotes a pair, represents the preference $p_1 \geqq p_2 \geqq \cdots \geqq p_k$. We also use the simpler notation xyz to denote the ordered triple (x, y, z) or unordered $\{x, y, z\}$. Similarly, xy denotes (x, y) or $\{x, y\}$.

Although 3GSM is similar to its 2-gender counterpart in that an instance can have more than one stable marriage,[1] it differs from the 2-gender counterpart in that there exist instances that have no stable marriage. Figure 1 shows a 3GSM instance with $A = \{\alpha_1, \alpha_2\}$, $B = \{\beta_1, \beta_2\}$, and $D = \{\delta_1, \delta_2\}$. A complete list of all possible marriages, each shown with a corresponding destabilizing triple, confirms that no stable marriage exists for this instance of 3GSM.

NP-completeness of 3GSM. In the previous section, we noted that some instances of 3GSM do not have stable marriages. In this section, we will show that deciding whether a given instance of 3GSM has a stable marriage is an NP-complete problem. This is accomplished by giving a polynomial transformation from the three-dimensional matching problem (or 3DM for short) to 3GSM. A proof that 3DM is NP-complete is first given in Karp's [8] landmark paper.

An instance of 3DM involves three finite sets of equal cardinality—which we denote by A', B', and D', relating them to A, B, and D of 3GSM. Given a set of triples $T' \subseteq A' \times B' \times D'$, the 3DM problem is to decide if there exists an $M' \subseteq T'$ such that M' is a complete matching, i.e., each element of A', B', and D' appears exactly once in M'.

Given a 3DM instance I', we construct a corresponding 3GSM instance I. Although our construction can be adapted to work for any 3DM instance in general, we assume, in order to simplify the presentation, that no element of A', B', or D' appears in more

[1] In fact, the number of stable marriages in many instances is exponential in the instances' size. Irving and Leather [7] give a proof of this for the 2-gender case. Extending the proof to cover the 3-gender case is straightforward.

α_1	$\beta_1\delta_2$	$\beta_1\delta_1$	$\beta_2\delta_2$	$\beta_2\delta_1$
α_2	$\beta_2\delta_2$	$\beta_1\delta_1$	$\beta_2\delta_1$	$\beta_1\delta_2$

β_1	$\alpha_2\delta_1$	$\alpha_1\delta_2$	$\alpha_1\delta_1$	$\alpha_2\delta_2$
β_2	$\alpha_2\delta_1$	$\alpha_1\delta_1$	$\alpha_2\delta_2$	$\alpha_1\delta_2$

δ_1	$\alpha_1\beta_2$	$\alpha_1\beta_1$	$\alpha_2\beta_1$	$\alpha_2\beta_2$
δ_2	$\alpha_1\beta_1$	$\alpha_2\beta_2$	$\alpha_1\beta_2$	$\alpha_2\beta_1$

Possible Marriage	Destabilizing Triple
$\{\alpha_1\beta_1\delta_1, \alpha_2\beta_2\delta_2\}$	$\alpha_1\beta_1\delta_2$
$\{\alpha_1\beta_1\delta_2, \alpha_2\beta_2\delta_1\}$	$\alpha_2\beta_1\delta_1$
$\{\alpha_1\beta_2\delta_1, \alpha_2\beta_1\delta_2\}$	$\alpha_1\beta_1\delta_2$
$\{\alpha_1\beta_2\delta_2, \alpha_2\beta_1\delta_1\}$	$\alpha_2\beta_2\delta_2$

FIG. 1. *An instance of* 3GSM *that has no stable marriage.*

than three triples of T'. This assumption is made without loss of generality. In their reference work on NP-completeness, Garey and Johnson [2, p. 221] mention that 3DM remains NP-complete with this restriction.

We construct I by first building a "frame" consisting of the elements $\alpha_1, \alpha_2 \in A$, $\beta_1, \beta_2 \in B$, and $\delta_1, \delta_2 \in D$. The preferences of these elements do not depend on the structure of I' and are displayed in Fig. 2. In Fig. 2 and subsequent figures, we are only interested in the roles played by a few items in each preference list. Therefore, we use the notation Π_{Rem} to denote any fixed but arbitrary permutation of the remaining items.

We will prove in Lemma 2 that the triples $\alpha_1\beta_1\delta_1$ and $\alpha_2\beta_2\delta_2$ must be included in any stable marriage. Note that $\alpha_1\beta_1\delta_1$ is the weakest link in such a marriage because it represents the least preferred match for both β_1 and δ_1. Consequently, if any element $a \in A$ is matched in marriage with a pair that it prefers less than $\beta_1\delta_1$, then $a\beta_1\delta_1$ becomes a destabilizing triple.

The above observation gives us a strategy that uses the pair $\beta_1\delta_1$ as a "boundary" in the preferences of A's remaining elements. A necessary condition for a stable marriage in I is that all remaining elements of A must match with pairs located left of the boundary, i.e., $\geqq \beta_1\delta_1$. Using information from T' to construct the set of items to be positioned left of the boundary, we ensure that this condition for stable marriage can be met only if T'

α_1	$\beta_1\delta_1$	$\beta_2\delta_1$	$\beta_1\delta_2 \cdots \Pi_{\text{Rem}} \cdots$
α_2	$\beta_2\delta_2$		$\cdots \Pi_{\text{Rem}} \cdots$
$\vdots$			
β_1	$\alpha_1\delta_2$		$\cdots \Pi_{\text{Rem}} \cdots \alpha_1\delta_1$
β_2	$\alpha_2\delta_2$	$\alpha_1\delta_1$	$\cdots \Pi_{\text{Rem}} \cdots$
$\vdots$			
δ_1	$\alpha_1\beta_2$		$\cdots \Pi_{\text{Rem}} \cdots \alpha_1\beta_1$
δ_2	$\alpha_1\beta_1$	$\alpha_2\beta_2$	$\cdots \Pi_{\text{Rem}} \cdots$
$\vdots$			

FIG. 2. *Preferences of the elements* $\alpha_1, \alpha_2, \beta_1, \beta_2, \delta_1, \delta_2$.

contains a complete matching. The remaining difficulty is to ensure that matching all elements of A left of the boundary is sufficient to yield a stable marriage. Before giving details of the construction that provides the solution, we first prove the lemmas that establish the frame's properties.

LEMMA 1. *If a stable marriage M exists for I constructed by extending the frame in Fig. 2, then $\alpha_1\beta_2\delta_1 \notin M$.*

Proof. The proof is by contradiction. Suppose $\alpha_1\beta_2\delta_1 \in M$. Since $\alpha_1\beta_2\delta_1 \in M$, δ_2's match cannot be $\alpha_1\beta_1$ or $\alpha_2\beta_2$. From δ_2's preference, $\alpha_1\beta_1$ is the only pair $\geqq_{\delta_2} \alpha_2\beta_2$. Therefore, $\alpha_2\beta_2 \geqq_{\delta_2} \delta_2$'s match in M. Moreover, $\beta_2\delta_2$ and $\alpha_2\delta_2$ are the first preference choices of α_2 and β_2, respectively. Hence, $\alpha_2\beta_2\delta_2$ is a destabilizing triple for M, a contradiction. □

LEMMA 2. *If a stable marriage M exists for I constructed by extending the frame in Fig. 2, then $\alpha_1\beta_1\delta_1 \in M$ and $\alpha_2\beta_2\delta_2 \in M$.*

Proof. We first prove $\alpha_1\beta_1\delta_1 \in M$. Suppose β_1 is not matched with $\alpha_1\delta_1$ in M, we can then find a destabilizing triple for M. There are two cases.

Case 1. β_1 is matched with $\alpha_1\delta_2$. $\alpha_1\beta_1\delta_2 \in M$ implies that $\alpha_2\beta_2\delta_2$, $\alpha_1\beta_1\delta_1$, and $\alpha_1\beta_2\delta_1 \notin M$. By an argument similar to that of Lemma 1, $\alpha_1\beta_2\delta_1$ is a destabilizing triple.

Case 2. β_1 is not matched with $\alpha_1\delta_2$ nor $\alpha_1\delta_1$. $\alpha_1\beta_2\delta_1 \notin M$ by Lemma 1. Also, $\alpha_1\beta_1\delta_1 \notin M$, which implies that $\alpha_1\beta_1\delta_2$ is a destabilizing triple in this case.

Hence, we conclude that $\alpha_1\beta_1\delta_1 \in M$, which implies that $\alpha_1\beta_1\delta_2 \notin M$. It is now easy to verify that if $\alpha_2\beta_2\delta_2 \notin M$, then it is a destabilizing triple. □

If the sets of I' (A', B', and D') each has k elements, then the sets of I (A, B, and D) each has $3k + 2$ elements. The α's, β's, or δ's, which are in the frame, account for two elements. The remaining $3k$ elements are defined as follows.

Suppose $A' = \{a_1, a_2, \cdots, a_k\}$, $B' = \{b_1, b_2, \cdots, b_k\}$, and $D' = \{d_1, d_2, \cdots, d_k\}$. According to an earlier assumption, each element $a_i \in A'$ appears in no more than three triples of T'. We clone three copies of a_i and replace a_i with the clones $a_i[1]$, $a_i[2]$, and $a_i[3]$ in A. These clones' preferences are set up to make it possible for *exactly one* of their matches in a stable marriage to correspond to a triple in T'.

To prevent the two remaining clones from interfering with the above setup, we add elements w_{a_i}, y_{a_i} to B and x_{a_i}, z_{a_i} to D. In a stable marriage, the pairs $w_{a_i}x_{a_i}$ and $y_{a_i}z_{a_i}$ are required to match with two of a_i's clones, putting them out of action. We complete the sets B and D by adding to them the elements of B' and D', respectively. To summarize, $A = \{\alpha_1, \alpha_2\} \cup \bigcup_{a_i \in A'} \{a_i[1], a_i[2], a_i[3]\}$, $B = B' \cup \{\beta_1, \beta_2\} \cup \bigcup_{a_i \in A'} \{w_{a_i}, y_{a_i}\}$, and $D = D' \cup \{\delta_1, \delta_2\} \cup \bigcup_{a_i \in A'} \{x_{a_i}, z_{a_i}\}$.

Given that $a_ib_{j_1}d_{l_1}$, $a_ib_{j_2}d_{l_2}$, and $a_ib_{j_3}d_{l_3}$ are the triples containing a_i in T', the preferences in Fig. 3 accomplish the objectives outlined above. When there exist fewer than three triples containing a_i, we equate two or more of the j's and l's.

The following lemma establishes the roles of w_{a_i}, x_{a_i}, y_{a_i}, and z_{a_i}.

LEMMA 3. *If a stable marriage M exists for I constructed with the preferences shown in Fig. 3, then for every $a_i \in A'$, there exist $j_1, j_2 \in \{1, 2, 3\}$, $j_1 \neq j_2$ such that*

(a) $a_i[j_1]w_{a_i}x_{a_i} \in M$, and

(b) $a_i[j_2]y_{a_i}z_{a_i} \in M$.

Proof. Consider the triple $a_i[1]w_{a_i}x_{a_i}$, which represents the third preference choice of x_{a_i} and the first preference choices of $a_i[1]$ and w_{a_i}. It becomes a destabilizing triple unless x_{a_i} is matched with one of its first three preference choices, proving part (a) of the lemma.

Similarly, z_{a_i} must be matched with one of its first three preference choices. Otherwise, $y_{a_i}z_{a_i}$ forms a destabilizing triple with $a_i[1]$ or $a_i[2]$, depending on which a_i clone is matched in part (a). □

α_1				
α_2				
$\vdots$				
$a_i[1]$	$w_{a_i}x_{a_i}$	$y_{a_i}z_{a_i}$	$b_{j_1}d_{l_1}$	$\beta_1\delta_1\cdots\Pi_{\text{Rem}}$
$a_i[2]$	$w_{a_i}x_{a_i}$	$y_{a_i}z_{a_i}$	$b_{j_2}d_{l_2}$	$\beta_1\delta_1\cdots\Pi_{\text{Rem}}$
$a_i[3]$	$w_{a_i}x_{a_i}$	$y_{a_i}z_{a_i}$	$b_{j_3}d_{l_3}$	$\beta_1\delta_1\cdots\Pi_{\text{Rem}}$
$\vdots$				

β_1			
β_2			
$\vdots$			
w_{a_i}	$a_i[1]x_{a_i}$	$a_i[2]x_{a_i}$	$a_i[3]x_{a_i}\cdots\Pi_{\text{Rem}}$
y_{a_i}	$a_i[1]z_{a_i}$	$a_i[2]z_{a_i}$	$a_i[3]z_{a_i}\cdots\Pi_{\text{Rem}}$
$\vdots$			
b_i	$\cdots$	Π_{Rem}	
$\vdots$			

δ_1			
δ_2			
$\vdots$			
x_{a_i}	$a_i[3]w_{a_i}$	$a_i[2]w_{a_i}$	$a_i[1]w_{a_i}\cdots\Pi_{\text{Rem}}$
z_{a_i}	$a_i[3]y_{a_i}$	$a_i[2]y_{a_i}$	$a_i[1]y_{a_i}\cdots\Pi_{\text{Rem}}$
$\vdots$			
d_i	$\cdots$	Π_{Rem}	
$\vdots$			

FIG. 3. *Preferences in the* 3GSM *instance I. The column of $\beta_1\delta_1$'s represents the boundary. Preferences of α's, β's, and δ's are those shown in Fig.* 2.

We are now ready to prove the NP-completeness of 3GSM by showing that I has a stable marriage if and only if T' has a complete matching of I'.

THEOREM 1. *If T' contains a complete matching M' of the* 3DM *instance I', then the constructed* 3GSM *instance I has a stable marriage M.*

Proof. We show that it is possible to construct a stable marriage M. Begin by adding $\alpha_1\beta_1\delta_1$ and $\alpha_2\beta_2\delta_2$ to M.

For each element $a_i \in A'$, the only triples in T' containing a_i are $a_ib_{j_1}d_{l_1}$, $a_ib_{j_2}d_{l_2}$, and $a_ib_{j_3}d_{l_3}$ using the notations found in Fig. 3. One of these triples is in M'.

$$\text{Add to } M\begin{cases} a_i[1]b_{j_1}d_{l_1}, & a_i[2]w_{a_i}x_{a_i}, \quad \text{and} \quad a_i[3]y_{a_i}z_{a_i} \quad \text{if } a_ib_{j_1}d_{l_1}\in M'; \\ a_i[1]w_{a_i}x_{a_i}, & a_i[2]b_{j_2}d_{l_2}, \quad \text{and} \quad a_i[3]y_{a_i}z_{a_i} \quad \text{if } a_ib_{j_2}d_{l_2}\in M'; \\ a_i[1]w_{a_i}x_{a_i}, & a_i[2]y_{a_i}z_{a_i}, \quad \text{and} \quad a_i[3]b_{j_3}d_{l_3} \quad \text{if } a_ib_{j_3}d_{l_3}\in M'. \end{cases}$$

Since M' is a complete matching, the above construction guarantees that those elements of B and D that originate from B' and D' are used exactly once in M. It is easy to verify that all other elements of A, B, and D are also used exactly once. Hence, M is a marriage.

To show that M is stable, it is sufficient to show that no element of A is a component of a destabilizing triple. α_1 and α_2 satisfy this condition immediately because they are matched with their first preference choices.

Referring to Fig. 3, each of the remaining elements of A is matched with a pair located to the left of the boundary. Hence, the only pairs that can form destabilizing triples are $w_{a_i}x_{a_i}$ and $y_{a_i}z_{a_i}$. However, w_{a_i}'s (y_{a_i}'s) match is one of its first three preference choices. These three choices are in exact reverse order of x_{a_i}'s (z_{a_i}'s). This eliminates w_{a_i} and y_{a_i} from participating in any destabilizing triple. □

THEOREM 2. *If the* 3GSM *instance I has a stable marriage, then T′ contains a complete matching of the* 3DM *instance I′.*

Proof. Suppose I has a stable marriage M. Lemma 2 requires M to include $\alpha_1\beta_1\delta_1$ and $\alpha_2\beta_2\delta_2$. Lemma 3 requires that, for each $a_i \in A'$, two of the a_i clones match with $w_{a_i}x_{a_i}$ and $y_{a_i}z_{a_i}$. Let M' represent the matching that results when M is restricted to the remaining elements that are without predetermined matches.

For each $a_i \in A'$, only one a_i clone remains to be matched in M'. Therefore, we will drop the distinction between an a_i clone and the a_i it represents, without the risk of introducing any ambiguity in M'. The elements that participate in M' can then be characterized as exactly those elements of A', B', and D'. Since M' is a subset of a marriage, it represents a complete matching.

Due to the absence of destabilizing triples, every a_i in M' must match with a preference choice located to the left of the boundary. The construction of I, as illustrated in Fig. 3, restricts this choice to the third item in the preference list since the first two items are already matched. Moreover, the triple formed by a_i and this item is contained in T'. Hence, every triple in M' is also a triple in T', and M' is the desired complete matching contained in T'. □

THEOREM 3. 3GSM *is* NP-*complete.*

Proof. It is easy to verify that the construction of I from I' can be accomplished within a polynomial time bound. Therefore, Theorems 1 and 2 establish that 3GSM is NP-hard. It is also possible to check the stability of a given marriage in polynomial time, establishing 3GSM's membership in NP. □

NP-completeness of 3PSA. The NP-completeness of 3PSA follows from that of 3GSM because the former is a generalization of the latter. Given a 3GSM instance I where $A = \{a_1, a_2, \cdots, a_k\}$, $B = \{b_1, b_2, \cdots, b_k\}$, and $D = \{d_1, d_2, \cdots, d_k\}$, we can extend it into a 3PSA instance $\hat{I}$ by defining $S = A \cup B \cup D$. Each element of S retains its entire preference list from I as the first k^2 preference items in $\hat{I}$. We refer to these k^2 items as *inherited items*. All remaining items are inconsequential in $\hat{I}$ and are arranged in fixed but arbitrary permutations following the inherited items. The result is illustrated in Fig. 4.

THEOREM 4. 3PSA *is* NP-*complete.*

Proof. Any stable marriage M in I is an assignment in $\hat{I}$. Any destabilizing triple for M in $\hat{I}$ is simultaneously a destabilizing triple for M in I. Therefore, the stability of M in I implies its stability in $\hat{I}$.

We claim that any stable assignment $\hat{M}$ in $\hat{I}$ involves only inherited items and is therefore a marriage in I. This is equivalent to claiming that $\hat{M}$ is a complete matching of $A \times B \times D$. Otherwise, there exist elements $a_i \in A$, $b_j \in B$, $d_l \in D$ not matched to inherited items, which implies that $a_ib_jd_l$ is a destabilizing triple.

Since $\hat{M}$ involves only inherited items, any destabilizing triple for $\hat{M}$ in I is simultaneously a destabilizing triple for $\hat{M}$ in $\hat{I}$. Therefore, the stability of $\hat{M}$ in $\hat{I}$ implies its stability in I. □

Related results. In addition to the interest generated among computer scientists, the stable marriage problem has also received substantial attention from game theorists. It is used to model economic problems that require matching representatives from different market forces, such as matching labor to the job market. Since 1951, the National Resident Matching Program (NRMP) has based its success on an algorithm that solves the stable

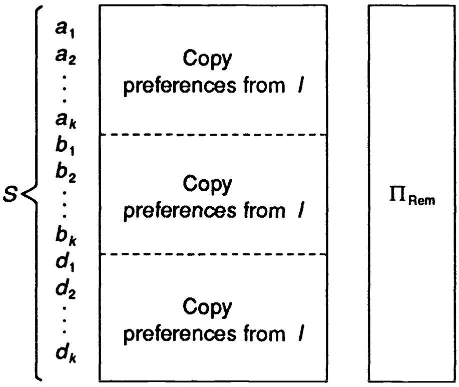


FIG. 4. *Preferences in the* 3PSA *instance* $\hat{I}$.

marriage problem [14]. The NRMP is the centralized national program in the United States that matches medical school graduates to hospital resident positions.

In recent years, NRMP administrators have recognized that an increasing proportion of medical school graduates comes from the set of married couples who are both medical students graduating in the same year. In 1983, NRMP instituted a "couples program" that allows a participating couple to increase the probability of being matched with two resident positions in close proximity. To participate in this special program, a couple submits a combined preference list that ranks pairs of resident positions.

In 1984, Roth [14, p. 1008] discovered a dilemma with NRMP's couples program. He showed that there are instances where no stable matching can exist. Recently, Ronn [13] proved that the problem of deciding whether a stable matching exists in an instance of the couples program is NP-complete.

As an extension of our work in this paper, we have obtained an alternate NP-completeness proof for NRMP's couples program [11]. We model the couples program as a job matching problem for dual-career couples where only a single job market is involved. Each couple has a preference list that ranks pairs of available positions. However, each employer ranks applicants individually without regard to marriage relations. A matching is stable if no couple can find an alternate pair of employers such that all four participants benefit from the new arrangement.

The NP-completeness proof for the problem in the above model is an adaptation of those developed in this paper. We refer interested readers to [11] for further details. In addition, we also examine the simpler problem that results when the employers are partitioned into two disjoint job markets, one for the male and female participants, respectively. We show that the problem remains NP-complete even with this simplification.

Conclusions and open problems. We have shown that three-dimensional generalizations of the stable marriage and stable roommate problems are NP-complete. Our result also applies to the problem of finding stable job assignments for dual-career couples, resulting in an alternate NP-completeness proof for NRMP's couples program. It may be interesting, as a topic for further research, to investigate the possibility of applying our result to other matching problems and their variants.

The proofs in this paper exploit the ability to assign a somewhat "inconsistent" preference list. For example, in Fig. 2, δ_2 does not rank β_1 consistently ahead of β_2 but instead depends on with whom the β's are matched. In the example, $\alpha_1\beta_1 \geqq_{\delta_2} \alpha_1\beta_2$ but $\alpha_2\beta_2 \geqq_{\delta_2} \alpha_2\beta_1$. An interesting question to consider is whether the matching problems remain NP-complete if all preference lists must obey a "consistency property," namely, $xy \geqq_a xz$ holds for either all x's or no x.

There are other ways to generalize the stable marriage problem in three dimensions besides those considered in this paper. One approach allows A to rank only elements of B, B ranks only elements of D, and D ranks only elements of A. A triple $abd \notin M$ is destabilizing if $ab_1d_1, a_2bd_2, a_3b_3d \in M$ and $b \geqq_a b_1$, $d \geqq_b d_2$, $a \geqq_d a_3$. One of the referees, who called our attention to this problem, attributes its origin to Knuth and dubbed it "circular" 3GSM. The complexity of this problem is currently an open problem.

Acknowledgments. We thank the referees for many helpful suggestions that have improved the presentation of this paper and for pointing out the two open problems in the previous section.

REFERENCES

[1] D. Gale and L. Shapley, *College admissions and the stability of marriage*, Amer. Math. Monthly, 69 (1962), pp. 9–15.

[2] M. R. Garey and D. S. Johnson, *Computers and Intractability; a Guide to the Theory of* NP-*Completeness*, W. H. Freeman, New York, 1979.

[3] D. Gusfield, *Three fast algorithms for four problems in stable marriage*, SIAM J. Comput., 16 (1987), pp. 111–128.

[4] ———, *The structure of the stable roommate problem: efficient representation and enumeration of all stable assignments*, SIAM J. Comput., 17 (1988), pp. 742–769.

[5] D. Gusfield and R. W. Irving, *The Stable Marriage Problem: Structure and Algorithms*, MIT Press, Cambridge, MA, 1989.

[6] R. W. Irving, *An efficient algorithm for the "stable roommates" problem*, J. Algorithms, 6 (1985), pp. 577–595.

[7] R. W. Irving and P. Leather, *The complexity of counting stable marriages*, SIAM J. Comput., 15 (1986), pp. 655–667.

[8] R. M. Karp, *Reducibility among combinatorial problems*, in Complexity of Computer Computations, R. E. Miller and J. W. Thatcher, eds., Plenum Press, New York, 1972, pp. 85–103.

[9] D. E. Knuth, *Mariages Stables*, Les Presses de L'Université de Montréal, Montréal, 1976.

[10] D. G. McVitie and L. B. Wilson, *The stable marriage problem*, Comm. ACM, 14 (1971), pp. 486–492.

[11] C. Ng and D. S. Hirschberg, *Complexity of the stable marriage and stable roommate problems in three dimensions*, Technical Report 88-28, Department of Information and Computer Science, University of California, Irvine, 1988.

[12] ———, *Lower bounds for the stable marriage problem and its variants*, SIAM J. Comput., 19 (1990), pp. 71–77.

[13] E. Ronn, NP-*complete stable matching problems*, J. Algorithms, 11 (1990), pp. 285–304.

[14] A. Roth, *The evolution of the labor market for medical interns and residents: a case study in game theory*, J. Political Economy, 92 (1984), pp. 991–1016.

[15] A. Subramanian, *A new approach to stable matching problems*, Technical Report STAN-CS-89-1275, Department of Computer Science, Stanford University, Stanford, CA, 1989.

SIAM J. DISC. MATH.
Vol. 4, No. 2, pp. 253-274, May 1991

SPECIAL RIM HOOK TABLOIDS AND SOME NEW MULTIPLICITY-FREE S-SERIES*

J. B. REMMEL† AND M. YANG‡

Abstract. This paper develops some general methods for expanding series of the form $\prod_i (1 + \sum_{k \geqq 1} f_k x_i^k)^{\pm 1}$ as a sum of Schur functions. That is, a combinatorial interpretation is given in terms of special rim hook tabloids and the coefficients f_k, of the coefficients d_λ where

$$\prod_i \left(1 + \sum_{k \geqq 1} f_k x_i^k\right)^{\pm 1} = \sum_\lambda d_\lambda S_\lambda$$

and S_λ denotes the Schur function associated with the partition λ. Such a series is called *multiplicity free* if d_λ is 1, -1, or 0 for all λ. The general methods are then applied to give explicit algorithms to find the coefficients c_λ where

$$\prod_i \left(\frac{1-x_i^{mn}}{1-x_i^n}\right)^{\pm 1} = \sum_\lambda c_\lambda S_\lambda.$$

In particular, it is shown that for all m, $n > 0$, c_λ is always 1, -1, or 0 and hence series of the form $\prod_i ((1 - x_i^{mn})/1 - x_i^n))^{\pm 1}$ are always multiplicity free. It is also shown that all series of the form $\prod_i ((1 - x_i^m)/(1 - x_i^2))^{\pm 1}$ are also multiplicity free for all m.

Key words. Schur functions, multiplicity-free S-series, special rim hooks tabloids, hook Schur functions

AMS(MOS) subject classifications. 05A19, 20C35

In some recent work, Yang and Wybourne [16], King, Yang, and Wybourne [8], and Lascoux and Pragacz [9] have produced a number of new expansions of series in terms of Schur functions that are called S-functions series. For example, there are 32 formal expressions

$$\prod_i (1 \pm x_i)^{\pm 1} \prod_{i \overset{<}{\leqq} j} (1 \pm x_i x_j)^{\pm 1}, \tag{0.1}$$

where the different symbols + or − and < or ≦ can be freely chosen. Note that each of the series of (0.1) is symmetric, i.e., invariant under permutations of the variables, and hence has an expansion in the form

$$\sum_\lambda d_\lambda S_\lambda(x_1, x_2, \cdots), \tag{0.2}$$

where $S_\lambda(x_1, x_2, \cdots)$ is the Schur function associated with the partition λ and d_λ is some coefficient. The problem then is to find the coefficients d_λ for any particular symmetric series. We say that a series is *multiplicity free* if its expansion in the form of (0.2) has the property that $d_\lambda \in \{0, 1, -1\}$ for all λ. Littlewood [10] was the first to give the S-series for many of the 32-series of the form (0.1). Littlewood's work was expanded on by Yang and Wybourne [16], who gave expressions for 30 of 32 series, and Lascoux and Pragacz [9] solved the two cases left unsolved by Yang and Wybourne.

* Received by the editors October 27, 1989; accepted for publication (in revised form) June 27, 1990. This research was partially supported by National Science Foundation grant DMS 87-02473.

† Department of Mathematics, University of California at San Diego, La Jolla, California 92093-0112.

‡ Department of Mathematics, Western Illinois University, Macomb, Illinois 61455.

In addition to the series of the form in (0.1), all three papers above also studied series of the form

$$(0.3) \qquad \prod_i (1 + a_1 x_i + a_2 x_i^2 + \cdots + a_n x_i^n)^{\pm 1}.$$

It was known (see [3], [5], [15]) that series of the form $\prod_i (1 \pm x_i^p)$ and $\prod_i (1 \pm x_i^p)^{-1}$ are multiplicity free. Yang and Wybourne [14] gave explicit expressions for the series

$$\prod_i (1 + x_i + x_i^2)^{\pm 1}, \prod_i (1 - x_i + x_i^2)^{\pm 1}, \prod_i (1 + x_i + x_i^2 + x_i^3)^{\pm 1}, \text{and} \prod_i (1 - x_i + x_i^2 - x_i^3)^{\pm 1},$$

which shows that these S-series are also multiplicity free. In general, King, Yang, and Wybourne [8] gave techniques to expand series of (0.3) in terms of nonstandard Schur functions. Similarly, Lascoux and Pragacz [9], using techniques from the theory of λ-rings, were able to give a closed expression for the series

$$(0.4) \qquad \prod_i (1 + x_i + \cdots + x_i^{p-1}) = \sum_{0 \leq r \leq p-1} (-1)^{p-1-r} \sum_H S_H(x_1, x_2, \cdots),$$

where the sum runs over all sequences in $\mathbb{N}^p$ of the form $H = (h_0, \cdots, h_{p-1})$, where $N = \{0, 1, 2, \cdots\}$ and $h_0, \cdots, h_{p-r-1} \equiv 0 \bmod p$ and $h_{p-r}, \cdots, h_{p-1} \equiv 1 \bmod p$. Now there are standard techniques for transforming a nonstandard Schur function into either plus or minus a standard Schur function. The problem is that even when we have an explicit formula like (0.4), it is difficult in general to be able to find the coefficients d_λ that appear in the expansion of the series in terms of standard Schur functions as in (0.2). Indeed it is not at all clear from (0.4) that the series $\prod_i (1 + x_i + \cdots + x_i^{p-1})$ is multiplicity free.

The main result of this paper is to show that for all integers $n, p > 0$, the S-series for series of the form

$$(0.5) \qquad \prod (1 + x_i^n + x_i^{2n} + \cdots + x_i^{(p-1)n})^{\pm 1}$$

are all multiplicity free. In fact, we show that there is a very simple algorithm to determine the coefficients d_λ. We use two basic tools to derive our results. First, we use the generalized Cauchy identity for the so-called hook Schur functions $HS_\lambda(x, y)$ (see [11], [12]), which up to sign factors are the functions $S_\lambda(X - Y)$ in λ-ring notation. Second, we use a combinatorial interpretation for the inverse of the Kostka matrix given by Egecioglu and Remmel [6] in terms of special rim hook tabloids.

S-function series have many applications to problems in physics involving Lie groups. Rowe, Wybourne, and Butler [14] and King and Wybourne [8] have shown how particular infinite series of S-functions arise in the determination of branching rules from certain noncompact Lie groups to compact Lie groups. Such branching rules are particularly relevant for calculations of nuclear models that exploit various properties of the noncompact sympletic groups $S_p(2n, R)$, where the series

$$D = \prod_{i \leq j} (1 - x_i x_j)^{-1}$$

and

$$M = \prod_i (1 - x_i)^{-1}$$

naturally occur. The construction of symmetrical wave functions necessitates the resolution of the Schur function expansion of plethysms of the D-series and M-series. This leads naturally to the need to evaluate the Schur function expansion of some of the series discussed in this paper.

The outline of this paper is as follows. In § 1 we establish our notation and state the necessary definitions and results from other papers that we need in order to prove our results. In § 2, we outline a general method for expanding series of the form

$$\prod_i \left(1+\sum_{k\geqq 1} f_k x^k\right)^{\pm 1} \tag{0.6}$$

for any formal power series $1+\sum_{k\geqq 1} f_k x^k$. In § 3, we then apply our general method to derive the S-series for the series in (0.5), as well as the hook Schur function analogue of these series.

1. Basic definitions and results. Let $\lambda = (0 < \lambda_1 \leqq \cdots \leqq \lambda_k)$ be a partition of n, i.e., $n = |\lambda| = \sum_{i=1}^{k} \lambda_i$. Each one of the integers λ_i is called a part of λ. We write $\lambda \vdash n$ to denote that λ is a partition of n. An alternative notation for λ is $\lambda = 1^{q_1} 2^{q_2} \cdots n^{q_n}$, where q_i is the number of parts of size i in λ. The length of λ, $l(\lambda)$, is the number of parts of λ.

The Ferrers diagram of shape λ, denoted by F_λ, is the set of left justified rows of squares or cells with λ_i cells in the ith row from the top for $i = 1, \cdots, k$. For example, see Fig. 1.1. In this context, the pair (i, j) denotes the cell in the ith row and jth column of F_λ, where we label the rows from bottom to top and the columns from left to right. We let λ' denote the conjugate of λ, i.e., $F_{\lambda'}$ is the Ferrers diagram that results from F_λ by transposing F_λ about the 45° line $\{(i, i) | i \geqq 0\}$.

Given two partitions $\lambda = (0 < \lambda_1 \leqq \cdots \leqq \lambda_k)$ and $\mu = (0 < \mu_1 \leqq \mu_2 \leqq \cdots \leqq \mu_l)$, we write $\mu \leqq \lambda$ if $l \leqq k$ and $\mu_{l-i} \leqq \lambda_{k-i}$ for $i = 0, \cdots, l-1$. The skew diagram $F_{\lambda/\mu}$ of shape λ/μ will consist of the cells of F_λ that remain after the cells of F_μ are removed. For example, see Fig. 1.2. We note that it is possible for a given skew diagram to represent $F_{\lambda/\mu}$ for many λ and μ. For example, the diagram pictured in Fig. 1.2 also equals $F_{(2,3,4,4)/(1,1,2,3)}$ and $F_{(1,2,3,3,3)/(1,2,3)}$. We write $\mu \subset \lambda$ if $\mu \leqq \lambda$, $l < k$, and $\mu_l < \lambda_k$. It is then easy to see that a given skew diagram is of the form $F_{\lambda/\mu}$ with $\mu \subset \lambda$ for unique μ and λ.

A *rim hook* H of a partition λ is a consecutive sequence of cells on its North-Eastern rim such that any two adjacent cells of H share a common edge and the removal of H from F_λ leaves a Ferrers diagram corresponding to a partition. H is *special rim hook*

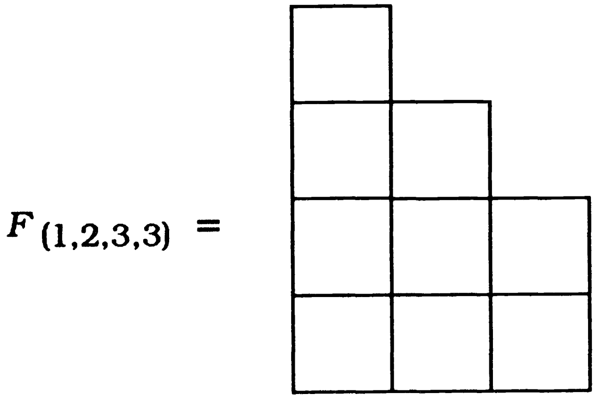


FIG. 1.1

$F_{(1,2,3,3)/(1,2)} =$

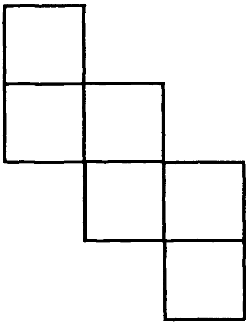


FIG. 1.2

(srh) if it starts in the cell $(l(\lambda),\ 1)$. H is a *transposed special rim hook* (t-srh) if it ends in cell $(1,\ \lambda_{l(\lambda)})$. For example, Fig. 1.3 pictures all the special rim hooks of $\lambda = (1, 2, 3, 3)$ and Fig. 1.4 pictures all the transposed special rim hooks of λ. Given any rim hook H of λ, we let $|H|$ = number of cells of H, $r(H)$ = number of rows of H, and $c(H)$ = number of columns of H. It is easy to see by induction on $|H|$ that for any rim hook H

$$|H| + 1 = r(H) + c(H). \tag{1.1}$$

We say H is a rim hook (srh, t-srh) of λ/μ if $F_{\lambda/\mu} = F_{\lambda^*/\mu^*}$, where $\mu^* \subset \lambda^*$ and H is a rim hook (srh, t-srh) of λ^* all of whose cells lie in $F_{\lambda/\mu} = F_{\lambda^*/\mu^*}$. For example, there are only two special rim hooks of $(1, 2, 3, 3)/(2, 2)$; see Fig. 1.5.

A *tabloid* T of shape λ/μ is a filling of $F_{\lambda/\mu}$ with positive integers. T is of *type* $\rho = 1^{q_1} \cdots i^{q_i} \cdots$, if i has frequency q_i in T. T_{ij} denotes the entry in the (i, j)th cell of T. A tabloid T of shape λ/μ is a *column-strict tableau* if the entries of T are weakly increasing in each row from left to right and strictly increasing in each column from bottom to top.

The Schur function S_λ of shape λ is defined by

$$S_\lambda(x) = \sum_T \prod_{(i,j)\in F_\lambda} x_{T_{ij}}, \tag{1.2}$$

where the summation is over all column-strict tableaux T of shape λ. Similarly, the skew Schur function $S_{\lambda/\mu}(x)$ of shape λ/μ is defined by

$$S_{\lambda/\mu}(x) = \sum_T \prod_{(i,j)\in F_{\lambda/\mu}} x_{T_{ij}}, \tag{1.3}$$

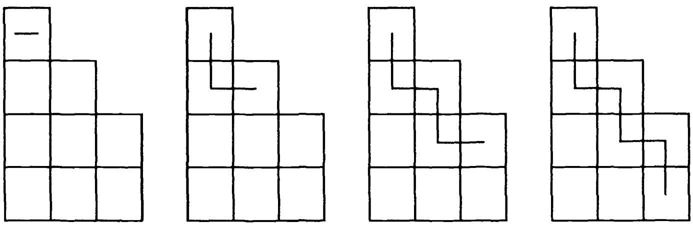


FIG. 1.3

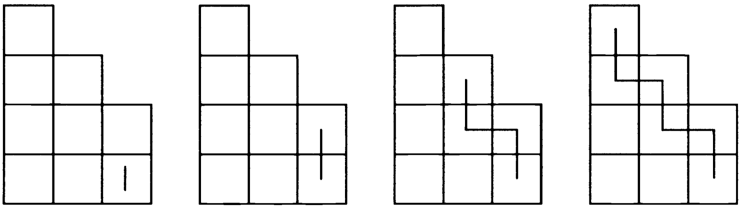


FIG. 1.4

where the summation is over column-strict tableaux T of shape λ/μ whenever $\mu \leqq \lambda$. The *homogeneous* symmetric function $h_\lambda(x)$ corresponding to a partition $\lambda \vdash n$ is given by

$$h_\lambda(x) = \prod_{i=1}^{k} h_{\lambda_i}(x), \tag{1.4}$$

where

$$h_r(x) = \sum_{0 < i_1 \leqq i_2 \leqq \cdots \leqq i_r} x_{i_1} x_{i_2} \cdots x_{i_r}. \tag{1.5}$$

The *elementary* symmetric function $e_\lambda(x)$ corresponding to a partition $\lambda \vdash n$ is given by

$$e_\lambda(x) = \prod_{i=1}^{k} e_{\lambda_i}(x), \tag{1.6}$$

where

$$e_r(x) = \sum_{0 < i_1 < i_2 < \cdots < i_r} x_{i_1} x_{i_2} \cdots x_{i_r}. \tag{1.7}$$

Next we give combinatorial interpretations due to Egecioglu and Remmel [6] for the coefficients that arise in the expansion of a skew Schur function $S_{\lambda/\mu}(x)$ in terms of either the elementary or homogeneous symmetric functions. That is, we want combinatorial interpretation for $H_{\nu,\lambda/\nu}$ and $E_{\nu,\lambda/\mu}$ where

$$S_{\lambda/\mu}(x) = \sum_{\nu \vdash |\lambda/\mu|} H_{\nu,\lambda/\mu} h_\nu(x) \tag{1.8}$$

and

$$S_{\lambda/\mu}(x) = \sum_{\nu \vdash |\lambda/\mu|} E_{\nu,\lambda/\mu} e_\nu(x). \tag{1.9}$$

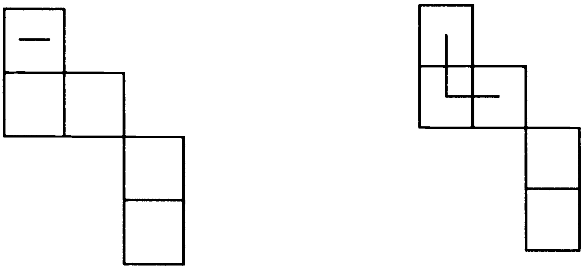


FIG. 1.5

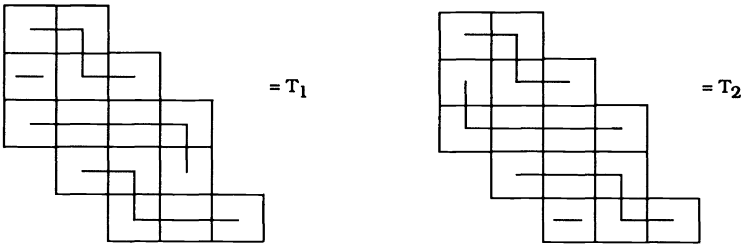


FIG. 1.6

To this end, we need to define *special rim hook tabloids* (SRHT) and *t-special rim hook tabloids* (t-SRHT). A special rim hook tabloid T of shape λ/μ is constructed recursively as follows. First, we pick a special rim hook H_1 in $F_{\lambda/\mu}$ and remove the cells of H_1 to produce a skew shape $\lambda^{(1)}/\mu^{(1)}$. Then the process is repeated for $F_{\lambda^{(1)}/\mu^{(1)}}$, i.e., we pick a special rim hook H_2 of $F_{\lambda^{(1)}/\mu^{(1)}}$, remove the cells of H_2 to produce a skew diagram of shape $\lambda^{(2)}/\mu^{(2)}$, etc. We continue this process until we produce a filling T of $F_{\lambda/\mu}$ with rim hooks $H_1, H_2, \cdots, H_r$ such that each rim hook H_i starts in the Western boundary of $F_{\lambda/\mu}$. The type of T is ν if $(|H_1|, \cdots, |H_r|)$ arranged in weakly increasing order produces the partition ν. A t-special rim hook tabloid T of shape λ/μ and type ν is defined recursively in the same way except that at each stage the rim hook H_i must be a t-special rim hook. For example, there are two SRHTs of shape $(2, 3, 4, 4, 5)/(1, 2)$ and type $(1, 4, 5, 5)$; see Fig. 1.6. The reader can check that there are no t-SRHTs of shape $(2, 3, 4, 4, 5)/(1, 2)$ and type $(1, 4, 5, 5)$. Figure 1.7 pictures the only t-SRHT of shape $(2, 3, 4, 4, 5)/(1, 2)$ and type $(1, 2, 2, 5, 5)$. Note that transposing a special rim hook tabloid T of shape λ/μ and type ν about the 45° line produces a t-special rim hook tabloid T' of shape λ'/μ' and type ν. We introduce two sign functions on rim hooks H, namely,

$$r\text{-sgn}(H) = (-1)^{r(H)-1} \tag{1.10}$$

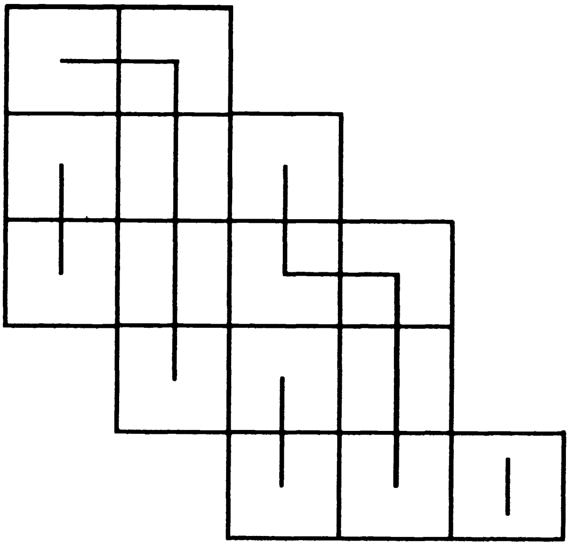


FIG. 1.7

and

$$c\text{-sgn}\,(H)=(-1)^{c(H)-1}. \tag{1.11}$$

These two sign functions induce two sign functions on SRHTs or t-SRHTs T by

$$r\text{-sgn}\,(T)=\prod_{i=1}^{r} r\text{-sgn}\,(H_i) \tag{1.12}$$

and

$$c\text{-sgn}\,(T)=\prod_{i=1}^{r} c\text{-sgn}\,(H_i). \tag{1.13}$$

For example, if T_1 is the special rim hook tabloid of Fig. 1.6, then

$$r\text{-sgn}\,(T_1)=(-1)^{2-1}(-1)^{1-1}(-1)^{2-1}(-1)^{2-1}=-1$$

and

$$c\text{-sgn}\,(T_1)=(-1)^{3-1}(-1)^{1-1}(-1)^{4-1}(-1)^{4-1}=1.$$

We let SRHT $(\nu, \lambda/\mu)$ (t-SRHT $(\nu, \lambda/\mu)$) denote the set of all SRHTs (t-SRHTs) of shape λ/μ and type ν. This given, it is proved in [6] that our desired combinatorial interpretation of $H_{\nu,\lambda/\mu}$ and $E_{\nu,\lambda/\mu}$ is given by

$$H_{\nu,\lambda/\mu}=\sum_{T\in \text{SRHT}\,(\nu,\lambda/\mu)} r\text{-sgn}\,(T) \tag{1.14}$$

and

$$E_{\nu,\lambda/\mu}=\sum_{T\in \text{t-SRHT}\,(\nu,\lambda/\mu)} c\text{-sgn}\,(T). \tag{1.15}$$

The hook Schur function $HS_\lambda(x;y)$ is defined as follows:

$$HS_\lambda(x;y)=\sum_{\mu\leqq\lambda} S_\mu(x)S_{\lambda'/\mu'}(y). \tag{1.16}$$

We note that $HS_\lambda(x;y)$ is clearly related to the function $S_\lambda(X-Y)$ in λ-ring notation. That is, if $X=x_1+x_2+\cdots$ and $Y=y_1+y_2+\cdots$, then

$$S_\lambda(X-Y)=\sum_{\mu\leqq\lambda} S_\mu(X)(-1)^{|\lambda'/\mu'|}S_{\lambda'/\mu'}(Y). \tag{1.17}$$

We let $HS_\lambda(0;y)$ and $(HS_\lambda(x;0))$ denote the result of setting all the variables $x_i(y_i)$ to be zero. Hook Schur functions have the following properties; see [1], [2], and [12]:

$$HS_\lambda(x;0)=S_\lambda(x), \tag{1.18}$$

$$HS_\lambda(0;y)=S_{\lambda'}(y), \tag{1.19}$$

$$HS_\lambda(x;y)=HS_{\lambda'}(y;x), \tag{1.20}$$

$$\sum_\lambda HS_\lambda(x;s)HS_\lambda(y;t)=\prod_{i,j}\frac{1}{1-x_iy_j}\prod_{i,j}\frac{1}{1-s_it_j}\prod_{i,j}(1+x_it_j)\prod_{i,j}(1+y_is_j). \tag{1.21}$$

We note that (1.21) is a generalization of the two Cauchy identities:

$$\sum_\lambda S_\lambda(x)S_\lambda(y)=\prod_{i,j}\frac{1}{1-x_iy_j} \tag{1.22}$$

and

$$(1.23) \qquad \sum_{\lambda} S_\lambda(x) S_{\lambda'}(t) = \prod_{i,j} (1 + x_i t_j).$$

That is, setting $s_i = t_j = 0$ for all i and j in (1.21) yields (1.22) and setting $s_i = y_j = 0$ for all i and j in (1.21) yields (1.23). Given an S-series, $S = \sum d_\lambda S_\lambda(x)$, we say the series $HS = \sum d_\lambda HS_\lambda(x, y)$ is the *hook Schur function analogue* of S. It is often the case that if the S-series S has a nice generating function, then the hook Schur function analogue also has nice generating function although there is no standard method to obtain such a generating function for HS from a generating function for S; see [11] and [12] for examples.

Finally, we should point out that there is another equivalent definition of the Schur function $S_\lambda(x)$. Namely, if $\lambda \vdash n$ and $\lambda = (0 \leqq \lambda_1 \leqq \lambda_2 \cdots \leqq \lambda_n)$, where we add a sequence of zero parts at the start of the nonzero parts of λ to give us n-parts, then we can express $S_\lambda(x)$ as the ratio of two $n \times n$ determinants:

$$(1.24) \qquad S_\lambda(x_1, \cdots, x_n) = \frac{\det |x_i^{\lambda_j + j - 1}|}{\det |x_i^{j-1}|}.$$

One advantage of (1.24) is that it makes sense for any sequence $\lambda = (\lambda_1, \cdots, \lambda_n)$ of nonnegative integers. If λ does not correspond to a partition, then we say $S_\lambda(x)$ is a *nonstandard Schur function*. By interchanging the columns of the determinant in the numerator of (1.24), we can transform any nonstandard Schur function into a standard Schur function up to a sign. In fact, analyzing the effect of column switches we are easily led to the following relations:

$$(1.25) \qquad S_\lambda(x) = 0 \quad \text{if } \lambda_i - 1 = \lambda_{i+1}$$

and

$$(1.26) \qquad S_\lambda(x) = -S_{(\lambda_1, \cdots, \lambda_{i+1}+1, \lambda_i - 1, \lambda_{i+2}, \cdots, \lambda_n)}(x).$$

2. Expansions of series of the form $\prod_i (1 + \sum_{k \geqq 1} f_k x_i^k)^{\pm 1}$. In this section we outline some general methods for expanding series of the form $\prod_i (1 + \sum_{k \geqq 1} f_k x_i^k)^{\pm 1}$ and closely related series.

First, suppose that we start with two polynomials $p(x) = 1 + p_1 x + \cdots + p_n x^n$ and $q(x) = 1 + q_1 x + \cdots + q_m x^m$. We can write

$$p(x) = (-1)^n p_n \prod_{i=1}^{n} (u_i - x) \quad \text{and} \quad q(x) = (-1)^m q_m \prod_{i=1}^{n} (v_i - x),$$

where $\{u_1, \cdots, u_m\} = R[p(x)]$ is the set of roots of $p(x)$ and $\{v_1, \cdots, v_m\} = R[q(x)]$ is the set of roots of $q(x)$. Note that

$$(-1)^n p_n \prod_{i=1}^{n} u_i = 1 = (-1)^m q_m \prod_{i=1}^{m} v_i$$

so that we can write

$$(2.1) \qquad p(x) = \prod_{i=1}^{n} (1 - x s_i)$$

and

$$q(x)=\prod_{i=1}^{m}(1-xt_i), \tag{2.2}$$

where $s_i=1/u_i$ and $t_j=1/v_j$ for all i and j. Thus $\{s_1,\cdots,s_n\}=IR[p(x)]$ is the set of inverses of the roots of $p(x)$ and $\{t_1,\cdots,t_m\}=IR[q(x)]$ is the set of inverses of the roots of $q(x)$. Now, specializing variables in the generalized Cauchy identity (1.21), we have

$$\begin{aligned}\prod_i\frac{q(-x_i)p(-y_i)}{p(x_i)q(y_i)}&=\prod_{i,j}\frac{1}{1-x_is_j}\prod_{i,j}\frac{1}{1-y_it_j}\prod_{i,j}(1+x_it_j)\prod_{i,j}(1+y_is_j)\\&=\sum_\lambda HS_\lambda(x;y)HS_\lambda(s_1,\cdots,s_n;t_1,\cdots,t_m)\\&=\sum_\lambda HS_\lambda(x;y)HS_\lambda(IR[p(x)];IR[q(x)]).\end{aligned}\tag{2.3}$$

Given a S-series $S=\sum_\lambda d_\lambda S_\lambda(x)$, the *conjugate* of S, S', is the series $S'=\sum_\lambda d_\lambda S_{\lambda'}(x)$ and the hook Schur function analogue of S, HS, is given by $HS=\sum_\lambda d_\lambda HS_\lambda(x;y)$. By further specializing variables in (2.3), we get two basic S-series and their conjugates and hook Schur function analogues.

PROPOSITION 2.1. *Given $p(x)$ and $q(x)$ as in* (2.1) *and* (2.2), *we have*

$$\text{(a)}\prod_i\frac{q(-x_i)}{p(x_i)}=\sum_\lambda S_\lambda(x)HS_\lambda(IR[p(x)];IR[q(x)]), \tag{2.4}$$

$$\text{(b)}\prod_i\frac{p(-x_i)}{q(x_i)}=\sum_\lambda S_{\lambda'}(x)HS_\lambda(IR[p(x)];IR[q(x)]), \tag{2.5}$$

$$\text{(c)}\prod_i\frac{q(-x_i)p(-y_i)}{p(x_i)q(y_i)}=\sum_\lambda HS_\lambda(x;y)HS_\lambda(IR[p(x)];IR[q(x)]). \tag{2.6}$$

Proof. The proof of (a) results from (2.3) by setting $y_i=0$ for all i; (b) results from (2.3) by setting $x_i=0$ for all i to obtain the series $\prod_i p(-y_i)/q(y_i)$ and then replacing y_i by x_i; and (c) is just (2.3). □

PROPOSITION 2.2.

$$\text{(a)}\prod_i\frac{1}{p(x_i)}=\sum_\lambda S_\lambda(x)S_\lambda(IR[p(x)]), \tag{2.7}$$

$$\text{(b)}\prod_i p(-x_i)=\sum_\lambda S_{\lambda'}(x)S_\lambda(IR[p(x)]), \tag{2.8}$$

$$\text{(c)}\prod_i\frac{p(-y_i)}{p(x_i)}=\sum_\lambda HS_\lambda(x;y)S_\lambda(IR[p(x)]), \tag{2.9}$$

$$\text{(d)}\prod_i\frac{p(-x_i)}{p(y_i)}=\sum_\lambda HS_{\lambda'}(x;y)S_\lambda(IR[p(x)]). \tag{2.10}$$

Proof. Equations (2.7), (2.8), and (2.9) result from (2.4), (2.5), and (2.6), respectively, by setting $t_i=0$ for all i, i.e., by setting $q(x)=1$. Equation (2.10) results from (2.9) by interchanging x's and y's and using the fact that $HS_\lambda(x;y)=HS_{\lambda'}(y;x)$. □

Thus the explicit computation of the series (2.4)–(2.10) reduces to the problem of computing $HS_\lambda(IR[p(x)]; IR[q(x)])$ or $S_\lambda(IR[p(x)])$. To this end, let us first consider the problem of computing $S_{\lambda/\mu}(IR[p(x)])$. Now by (1.9)

$$S_{\lambda/\mu}(IR[p(x)])=\sum_{\nu} E_{\nu,\lambda/\mu}e_\nu(IR[p(x)]). \tag{2.11}$$

But then

$$\begin{aligned} p(x)=1+p_1x+\cdots+p_nx^n&=\prod_{i=1}^{n}(1-s_ix)\\ &=\sum_{r=0}^{n}(-1)^r x^r e_r(s_1,\cdots,s_n)\\ &=\sum_{r=0}^{n}(-1)^r x^r e_r(IR[p(x)]). \end{aligned} \tag{2.12}$$

Thus

$$e_r(IR[p(x)])=(-1)^r p_r, \qquad r\geqq 0. \tag{2.13}$$

Now recall by (1.15),

$$E_{\nu,\lambda/\mu}=\sum_{T\in \text{t-SRHT}(\lambda/\mu,\nu)} c\text{-sgn}\,(T), \tag{2.14}$$

where c-sgn $(T)=\prod_{h\in T} c$-sgn (h) and for any rim hook h, c-sgn $(h)=(-1)^{c(h)-1}$. It follows from (2.13) and (2.14) that

$$E_{\nu,\lambda/\mu}e_\nu(IR[p(x)])=\sum_{T\in \text{t-SRHT}(\nu,\lambda/\mu)}\omega_p(T), \tag{2.15}$$

where

$$\omega_p(T)=\prod_{h\in T}\omega_p(h) \tag{2.16}$$

and for any rim hook h,

$$\begin{aligned} \omega_p(h)&=(-1)^{c(h)-1}(-1)^{|h|}p_{|h|}\\ &=(-1)^{c(h)-1}(-1)^{r(h)+c|h|-1}p_{|h|}\\ &=(-1)^{r(h)}p_{|h|}\\ &=-r\text{-sgn}\,(h)p_{|h|}. \end{aligned} \tag{2.17}$$

Combining (2.12)–(2.17), we have proved the following theorem.

THEOREM 2.3. *Suppose* $p(x)=1+p_1x+\cdots+p_nx^n=\prod_{i=1}^{n}(1-xs_i)$. *Then*

$$S_{\lambda/\mu}(s_1,\cdots,s_n)=S_{\lambda/\mu}(IR[p(x)])=\sum_{\nu\vdash|\lambda/\mu|}\omega_p(E_{\nu,\lambda/\mu}), \tag{2.18}$$

where

$$\omega_p(E_{\nu,\lambda/\mu})=\sum_{T\in \text{t-SRHT}(\nu,\lambda/\mu)}\omega_p(T) \tag{2.19}$$

and $\omega_p(T)$ *is given by* (2.16) *and* (2.17).

Of course, the special case of Theorem 2.3 when $\mu = \phi$ gives $S_\lambda(IR[p(x)])$. To compute $HS_{\lambda/\alpha}(IR[p(x)]; IR[q(x)])$, note that

(2.20)

$$\begin{aligned}HS_{\lambda/\alpha}(IR[p(x)];IR[q(x)]) &= \sum_{\alpha \leqq \mu \leqq \lambda} S_{\mu/\alpha}(IR[p(x)])S_{\lambda'/\mu'}(IR[q(x)]) \\ &= \sum_{\alpha \leqq \mu \leqq \lambda} \left(\sum_{\nu \vdash |\mu/\alpha|} \omega_p(E_{\nu,\mu/\alpha}) \right) \sum_{\gamma \vdash |\lambda'/\mu'|} \left(\sum_{\gamma} \omega_q(E_{\gamma,\lambda'/\mu'}) \right) \\ &= \sum_{\alpha \leqq \mu \leqq \lambda} \sum_{\nu \vdash |\mu/\alpha|} \sum_{\gamma \vdash |\lambda/\mu|} \sum_{\substack{T \in \text{t-SRHT}(\nu,\mu/\alpha) \\ R \in \text{t-SRHT}(\gamma,\lambda'/\mu')}} \omega_p(T)\omega_p(R).\end{aligned}$$

If we consider the pairs of t-special rim hook tabloids that occur on the right-hand side of (2.20), we are naturally led to the concept of a *bi-special rim hook tabloid* of shape λ. That is, if we transpose $R \in$ t-SRHT $(\gamma, \lambda'/\mu')$ about the 45° line, we get an element $R^t \in$ SRHT $(\gamma, \lambda/\mu)$. The pair (T, R^t) then gives us a filling of $F_{\lambda/\alpha}$. Thus we define a *bi-special rim hook tabloid* (bi-SRHT) B of shape λ/α and type $\langle \nu, \gamma \rangle$ to be a pair (T, S) where T is a t-special rim hook tabloid of shape μ/α, where $\alpha \leqq \mu \leqq \lambda$ and S is a special rim hook tabloid of shape λ/μ. For example, B in Fig. 2.1 is a bi-SRHT of shape $(2^2, 4, 6^3)$ and type $\langle (3^2, 4), (1, 2, 3, 5^2) \rangle$, where for emphasis we have separated the pair by a darkened line. We let B-SRHT $(\langle \nu, \gamma \rangle, \lambda/\alpha)$ denote the set of all bi-SRHTs of shape λ/α and type $\langle \nu, \gamma \rangle$. Also from (2.18), we see that we should define the $\omega_{p,q}$ weight of $B = (T, S) \in$ B-SRHT $(\langle \nu, \gamma \rangle, \langle \lambda/\alpha \rangle)$ by

(2.21)
$$\omega_{p,q}(B) = \prod_{t \in T} \omega_p(t) \prod_{h \in S} \bar{\omega}_q(h),$$

where

(2.22)
$$\omega_p(t) = -r\text{-sgn}\,(t)p_{|t|}$$

and

(2.23)
$$\bar{\omega}_q(h) = -c\text{-sgn}\,(h)q_{|t|}.$$

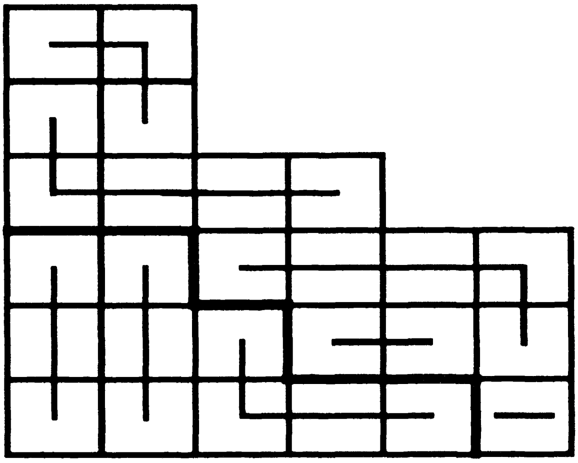


FIG. 2.1

Note that we need c-sgn (h) in (2.23) because when we transpose a t-special rim hook h^* of weight r-sgn $(h^*)q_{|h^*|}$ we get a special rim hook h of weight c-sgn $(h)q_{|h|}$. This given, we then have the following result.

THEOREM 2.4. *Let* $p(x) = 1 + p_1x + \cdots + p_nx^n = \prod_{i=1}^{n}(1 - xs_i)$ *and* $q(x) = 1 + q_1x + \cdots + q_mx^m = \prod_{i=1}^{m}(1 - xt_i)$, *then*

$$\begin{aligned} HS_{\lambda/\alpha}(s_1, \cdots, s_n; t_1, \cdots, t_m) &= HS_{\lambda/\alpha}(IR[p(x)]; IR[q(x)]) \\ &= \sum_{\langle\nu,\gamma\rangle} \omega_{p,q}(B_{\langle\nu,\gamma\rangle,\lambda/\alpha}), \end{aligned} \tag{2.24}$$

where

$$\omega_{p,q}(B_{\langle\nu,\gamma\rangle,\lambda/\alpha}) = \sum_{B\in\text{B-SRHT}(\langle\nu,\gamma\rangle,\lambda/\alpha)} \omega_{p,q}(B) \tag{2.25}$$

and $\omega_{p,q}(B)$ *is given by* (2.21).

Note that when we plug in the expressions for $S_\lambda(IR[p(x)])$ and $HS_\lambda(IR[p(x)]; IR[q(x)])$ into the series (2.4)–(2.10), we see that the coefficients that arise in the expansions of the series depend only on the coefficients of the polynomials p and q and not on the roots of p and q. Moreover, if $|\lambda| = k$, then the coefficients of $S_\lambda(x)$ or $HS_\lambda(x, y)$ in any of the series depends only on the coefficients $p_1, \cdots, p_k$ and $q_1, \cdots, q_k$ since any rim hook in t-SRHT or B-SRHT that contributes to the coefficient of $S_\lambda(x)$ or $HS_\lambda(x, y)$ can have length at most k. It thus follows that all our results apply to infinite series $p(x) = 1 + \sum_{k\geq 1} p_kx^k$ and $q(x) = 1 + \sum_{k\geq 1} q_kx^k$, as well as to polynomials. That is, suppose $p(x)$ and $q(x)$ are infinite series as above and we let $p_n(x) = 1 + p_1x + \cdots + p_nx^n$ and $q_n(x) = 1 + q_1x + \cdots + q_nx^n$. Then it is easy to see that the series

$$\prod_i \frac{q(-x_i)p(-y_i)}{p(x_i)q(y_i)} - \prod_i \frac{q_n(-x_i)p_n(-y_i)}{p_n(x_i)q_n(y_i)} \tag{2.26}$$

has no terms of degree $\leqq n$. Hence if $\lambda \vdash n$ then the coefficient of $HS_\lambda(x, y)$ in

$$\prod_i \frac{q(-x_i)p(-y_i)}{p(x_i)q(y_i)} \quad \text{and} \quad \prod_i \frac{q_n(-x_i)p_n(-y_i)}{p_n(x_i)q_n(y_i)}$$

are the same.

Thus we can summarize our results in the following theorems.

THEOREM 2.5. *Let* $p(x) = 1 + \sum_{k\geq 1} p_kx^k$ *and* $q(x) = 1 + \sum_{k\geq 1} q_kx^k$. *Then*

$$\text{(a)} \prod_i \frac{q(-x_i)}{p(x_i)} = \sum_\lambda S_\lambda(x)\left(\sum_{\langle\nu,\gamma\rangle} \omega_{p,q}(B_{\langle\nu,\gamma\rangle,\lambda})\right), \tag{2.27}$$

$$\text{(b)} \prod_i \frac{p(-x_i)}{q(x_i)} = \sum_\lambda S_{\lambda'}(x)\left(\sum_{\langle\nu,\gamma\rangle} \omega_{p,q}(B_{\langle\nu,\gamma\rangle,\lambda})\right), \tag{2.28}$$

$$\text{(c)} \prod_i \frac{q(-x_i)p(-y_i)}{p(x_i)q(y_i)} = \sum_\lambda HS_\lambda(x; y)\left(\sum_{\langle\nu,\gamma\rangle} \omega_{p,q}(B_{\langle\nu,\gamma\rangle,\lambda})\right), \tag{2.29}$$

where

$$\omega_{p,q}(B_{\langle\nu,\gamma\rangle,\lambda}) = \sum_{B\in\text{B-SRHT}(\langle\nu,\gamma\rangle,\lambda)} \omega_{p,q}(B)$$

and if $B = (T, S) \in$ B-SRHT $(\langle \nu, \gamma \rangle, \lambda) = \sum_{\mu \leqq \lambda}$ t-SRHT $(\nu, \mu) \times$ SRHT $(\gamma, \lambda/\mu)$, *then* $\omega_{p,q}(B) = \omega_{p,q}(T, S) = \prod_{t \in T} -r\text{-sgn}\,(t) p_{|t|} \prod_{h \in S} -c\text{-sgn}\,(h) q_{|h|}$.

THEOREM 2.6. *Let* $p(x) = 1 + \sum_{k \geqq 1} p_k x^k$. *Then*

$$\text{(2.30)} \qquad \text{(a)} \prod_i \frac{1}{p(x_i)} = \sum_\lambda S_\lambda(x) \left(\sum_\nu \omega_p(E_{\nu,\lambda}) \right),$$

$$\text{(2.31)} \qquad \text{(b)} \prod_i p(-x_i) = \sum_\lambda S_{\lambda'}(x) \left(\sum_\nu \omega_p(E_{\nu,\lambda}) \right),$$

$$\text{(2.32)} \qquad \text{(c)} \prod_i \frac{p(-y_i)}{p(x_i)} = \sum_\lambda HS_\lambda(x;y) \left(\sum_\nu \omega_p(E_{\nu,\lambda}) \right),$$

$$\text{(2.33)} \qquad \text{(d)} \prod_i \frac{p(-x_i)}{p(y_i)} = \sum_\lambda HS_{\lambda'}(x;y) \left(\sum_\nu \omega_p(E_{\nu,\lambda}) \right),$$

where

$$\omega_p(E_{\nu,\lambda}) = \sum_{T \in \text{t-SRHT}\,(\nu,\lambda)} \omega_p(T)$$

and if $T \in$ t-SRHT (ν, λ), *then*

$$\omega_p(T) = \prod_{h \in T} -r\text{-sgn}\,(h) p_{|h|}.$$

Note that the coefficients $\sum_{\langle \nu,\gamma \rangle} \omega_{p,q}(B_{\langle \nu,\gamma \rangle,\lambda})$ and $\sum_\nu \omega_p(E_{\nu,\lambda})$, which appear in Theorems 2.6 and 2.7, were derived from our expansion of $S_{\lambda/\mu}$ in terms of the elementary symmetric functions given by (1.9). We could go through a similar analysis based on the expansion of $S_{\lambda/\mu}$ in terms of the homogeneous symmetric functions given by (1.8). We will indicate briefly what happens if we use (1.8) in our analysis. So let $p(x) = 1 + \sum_{k \geqq 1} p_k x^k$ and $q(x) = 1 + \sum_{k \geqq 1} q_k x^k$. Suppose $\lambda \vdash n$, and let $p_n(x) = 1 + p_1 x + \cdots + p_n x^n = \prod_{i=1}^n (1 - x s_i)$ and $q_n(x) = 1 + q_1 x + \cdots + q_n x^n = \prod_{i=1}^n (1 - x t_i)$. Then we have shown that

$$\text{(2.34)} \qquad \sum_{\langle \nu,\gamma \rangle} \omega_{p,q}(B_{\langle \nu,\gamma \rangle,\lambda}) = HS_\lambda(s_1, \cdots, s_n; t_1, \cdots, t_n).$$

Then by (1.9)

(2.35)

$$HS_\lambda(s_1, \cdots, s_n; t_1, \cdots, t_n) = \sum_{\mu \leqq \lambda} S_\mu(s_1, \cdots, s_n) S_{\lambda'/\mu'}(t_1, \cdots, t_n)$$
$$= \sum_{\mu \leqq \lambda} \left(\sum_\nu H_{\nu,\mu} h_\nu(s_1, \cdots, s_n) \right) \left(\sum_\gamma H_{\gamma,\lambda'/\mu'} h_\gamma(t_1, \cdots, t_n) \right).$$

Next observe that

$$\text{(2.36)} \qquad \frac{1}{p_n(x)} = \prod_i \frac{1}{1 - x s_i} = 1 + \sum_{m \geqq 1} x^m h_m(s_1, \cdots, s_n).$$

Thus for $m \leqq n$

$$\text{(2.37)} \qquad h_m(s_1, \cdots, s_n) = u_m^*,$$

where

$$\frac{1}{p_n(x)} = 1 + \sum_{k \geqq 1} u_k^* x^k. \tag{2.38}$$

Moreover, it is easy to see that for $k \leqq n$

$$\left.\frac{1}{p_n(x)}\right|_{x^k} = \left.\frac{1}{p(x)}\right|_{x^k} = p_k^*. \tag{2.39}$$

Thus if we set

$$\frac{1}{p(x)} = 1 + \sum_{k \geqq 1} p_k^* x^k, \tag{2.40}$$

then for $m \leqq n$

$$h_m(s_1, \cdots, s_n) = p_m^*. \tag{2.41}$$

By a similar argument, if we set

$$\frac{1}{q(x)} = 1 + \sum_{k \geqq 1} q_k^* x^k, \tag{2.42}$$

then for $m \leqq n$

$$h_m(t_1, \cdots, t_n) = q_m^*. \tag{2.43}$$

Plugging (2.41) and (2.43) plus our interpretation of $H_{\nu,\lambda/\mu}$ from (1.14) into (2.35), we get for $\lambda \vdash n$

(2.44)

$$HS_\lambda(s_1, \cdots, s_n; t_1, \cdots, t_n) = \sum_{\mu \leqq \lambda} \left(\sum_{\nu} \sum_{T \in \mathrm{SRHT}\,(\nu,\mu)} v_p(T) \right) \left(\sum_{\gamma} \sum_{S \in \mathrm{SRHT}\,(\gamma,\lambda'/\mu')} v_q(S) \right),$$

where for $T \in$ SRHT (ν, μ),

$$v_p(T) = \prod_{h \in T} v_p(h) \tag{2.45}$$

and for a special rim hook $h \in T$

$$v_p(h) = r\text{-sgn}\,(h) p^*_{|h|}. \tag{2.46}$$

Similarly for $S \in$ SRHT $(\gamma, \lambda'/\mu')$,

$$v_q(S) = \prod_{h \in S} v_q(h), \tag{2.47}$$

where for a special rim hook $h \in S$

$$v_q(h) = r\text{-sgn}\,(h) q^*_{|h|}. \tag{2.48}$$

Now if we consider pairs (T, S) that occur on the right-hand side of (2.44), we see that if S^t denotes the transpose of S about the 45° line, then the pair (T, S^t) gives us a filling of F_λ. Thus we are led to the concept of a *transposed bi-special rim hook tabloid* B of shape λ/α and type $\langle \nu, \gamma \rangle$ as consisting of a pair (T, R), where for some $\alpha \leqq \mu \leqq \lambda$, $T \in$ SRHT $(\nu, \mu/\alpha)$ and $R \in$ t-SRHT $(\gamma, \lambda/\mu)$. We let tB-SRHT $(\langle \nu, \gamma \rangle,$

$\lambda/\alpha)$ denote the set of all transposed bi-special rim hook tabloids of shape λ/α and type $\langle\nu,\gamma\rangle$. Then combining (2.34), (2.44), and Theorem 2.5, we have the following theorem.

THEOREM 2.7. *Let*

$$p(x)=1+\sum_{k\geqq 1}p_kx^k,\frac{1}{p(x)}=1+\sum_{k\geqq 1}p_k^*x^k,q(x)=1+\sum_{k\geqq 1}q_kx^k$$

and

$$\frac{1}{q(x)}=1+\sum_{k\geqq 1}q_k^*x^k.$$

Then

$$\text{(a)}\prod_i\frac{q(-x_i)}{p(x_i)}=\sum_\lambda S_\lambda(x)\left(\sum_{\langle\nu,\gamma\rangle}v_{p,q}(t\text{-}B_{\langle\nu,\gamma\rangle,\lambda})\right),\tag{2.49}$$

$$\text{(b)}\prod_i\frac{p(-x_i)}{q(x_i)}=\sum_\lambda S_{\lambda'}(x)\left(\sum_{\langle\nu,\gamma\rangle}v_{p,q}(t\text{-}B_{\langle\nu,\gamma\rangle,\lambda})\right),\tag{2.50}$$

$$\text{(c)}\prod_i\frac{q(-x_i)p(-y_i)}{p(x_i)q(y_i)}=\sum_\lambda HS_\lambda(x;y)\left(\sum_{\langle\nu,\gamma\rangle}v_{p,q}(t\text{-}B_{\langle\nu,\gamma\rangle,\lambda})\right),\tag{2.51}$$

where

$$v_{p,q}(t\text{-}B_{\langle\nu,\gamma\rangle,\lambda})=\sum_{B\in\text{tB-SRHT}(\langle\nu,\gamma\rangle,\lambda)}v_{p,q}(B)$$

and if $B=(U,V)\in$ tB-SRHT $(\langle\nu,\gamma\rangle\lambda)=\sum_{\mu\leqq\lambda}$ SRHT $(\nu,\mu)\times$ t-SRHT $(\gamma,\lambda/\mu)$, *then* $v_{p,q}(B)=v_{p,q}(U,V)=\prod_{t\in U}r\text{-sgn}\,(t)p^*_{|t|}\prod_{h\in V}c\text{-sgn}\,(h)q^*_{|h|}$.

A similar argument will also allow us to derive the following analogue of Theorem 2.6.

THEOREM 2.8. *Let* $p(x)=1+\sum_{k\geqq 1}p_kx^k$ *and* $1/p(x)=\sum_{k\geqq 1}p_k^*x^k$. *Then*

$$\text{(a)}\prod_i\frac{1}{p(x_i)}=\sum_\lambda S_\lambda(x)\left(\sum_\gamma v_p(H_{\gamma,\lambda})\right),\tag{2.52}$$

$$\text{(b)}\prod_i p(-x_i)=\sum_\lambda S_{\lambda'}(x)\left(\sum_\gamma v_p(H_{\gamma,\lambda})\right),\tag{2.53}$$

$$\text{(c)}\prod_i\frac{p(-y_i)}{p(x_i)}=\sum_\lambda HS_\lambda(x;y)\left(\sum_\gamma v_p(H_{\gamma,\lambda})\right),\tag{2.54}$$

$$\text{(d)}\prod_i\frac{p(-x_i)}{p(y_i)}=\sum_\lambda HS_{\lambda'}(x;y)\left(\sum_\gamma v_p(H_{\gamma,\lambda})\right),\tag{2.55}$$

where

$$v_p(H_{\gamma,\lambda})=\sum_{T\in\text{SRHT}(\gamma,\lambda)}v_p(T)$$

and if $T\in$ SRHT (γ,λ), *then*

$$v_p(T)=\prod_{h\in T}r\text{-sgn}\,(h)p^*_{|h|}.$$

Finally, we should observe that we have left the series that appear in Theorems 2.5–2.8 in a form that derives directly from the generalized Cauchy identity (1.21). However, by replacing x_i by $-x_i$ or $p(x)$ by $1/r(x)$, we can get series of other forms. For example, replacing x_i by $-x_i$ in (2.31), we get

$$\prod_i p(x_i)=\sum_\lambda (-1)^{|\lambda|} S_{\lambda'}(x)\left(\sum_\nu \omega_p(E_{\nu,\lambda})\right) \tag{2.56}$$

or by replacing $p(-x)$ by $1/r(x)$ in (2.27) or (2.28) we get expansions of series of the form

$$\prod_i r(-x_i)q(-x_i) \quad \text{or} \quad \prod_i \frac{1}{r(x_i)q(x_i)}.$$

3. New multiplicity-free S-series. In this section, we use the results of § 2 to derive a number of new multiplicity-free S-series. We start by considering the series

$$\begin{aligned}\prod_i (1+x_i^p+x_i^{2p}+\cdots+x_i^{p(n-1)}) &= \prod_i \frac{1-x_i^{pn}}{1-x_i^p} \\ &= \sum_\lambda d_\lambda S_\lambda(x).\end{aligned} \tag{3.1}$$

Let ω be a primitive npth root of unity so that ω^n is a primitive pth root of unity. Then

$$\begin{aligned}p(x) &= (1+x^p+x^{2p}+\cdots+x^{p(n-1)}) = \frac{1-x^{pn}}{1-x^p} \\ &= \prod_{i=0}^{np-1}(1-\omega^i x)\Big/\prod_{k=0}^{p-1}(1-\omega^{kn}x) \\ &= \prod_{i=1}^{np-p}(1-s_i x),\end{aligned} \tag{3.2}$$

where $\{s_1, \cdots, s_{np-p}\} = \{\omega^j \mid 0 \leqq j \leqq np-1 \text{ and } j \neq 0 \bmod n\}$.

First we apply Proposition 2.2(b) with $-x_i$ replacing x_i and we get

$$\begin{aligned}\prod_i p(x_i) &= \sum_\lambda (-1)^{|\lambda'|} S_{\lambda'}(x) S_\lambda(s_1, \cdots, s_{np-p}) \\ &= \sum_\mu (-1)^{|\mu|} S_\mu(x) S_{\mu'}(s_1, \cdots, s_{np-p}).\end{aligned} \tag{3.3}$$

Note that $S_{\mu'}(s_1, \cdots, s_{np-p}) = 0$ if μ' has more than $np-p$ rows so that from (3.3) it follows that

$$d_\mu = 0 \text{ unless } F_\mu \text{ has } np-p \text{ or fewer columns.} \tag{3.4}$$

Next let $q(x) = 1-(-x)^{pn}$ and $r(x) = 1-x^p$. Then if we apply Theorem 2.5(a) to $q(x)$ and $r(x)$, we get

$$\begin{aligned}\prod_i p(x_i) &= \prod_i \frac{q(-x_i)}{r(x_i)} \\ &= \sum_\lambda S_\lambda(x)\left(\sum_{\langle\nu,\gamma\rangle} \omega_{r,q}(B_{\langle\nu,\gamma\rangle,\lambda})\right),\end{aligned} \tag{3.5}$$

where

$$\omega_{r,q}(B_{\langle\nu,\gamma\rangle,\lambda}) = \sum_{B \in \text{B-SRHT}(\langle\nu,\gamma,\rangle,\lambda)} \omega_{r,q}(B) \tag{3.6}$$

and if $B = (T, S) \in$ B-SRHT $(\langle\nu, \gamma\rangle, \lambda)$, then

$$\begin{aligned} \omega_{r,q}(B) &= \omega_{r,q}(T,S) \\ &= \prod_{t\in T} -r\text{-sgn}(t)r_{|t|} \prod_{h\in S} -c\text{-sgn}(h)q_{|h|} \\ &= \prod_{t\in T}(-1)^{r(t)}(-1)\chi(|t|=p)\prod_{h\in S}(-1)^{c(h)}(-1)^{pn+1}\chi(|h|=pn), \end{aligned} \tag{3.7}$$

where for any statement A, we set $\chi(A) = 1$ if A is true and $\chi(A) = 0$ if A is false. We can see from (3.6) and (3.7) that $\omega_{r,q}(B_{\langle\nu,\gamma\rangle,\lambda}) = 0$ unless ν is of the form (p^k) and γ is of the form (pn^l). We claim that if λ is a shape with at most $np - p$ columns, i.e., if $l(\lambda') \leqq np - p$, then there is at most one pair $\langle k, l\rangle$ for which there exists $B \in$ B-SRHT $(\langle(p^k), (pn^l)\rangle, \lambda)$ and if there is such a $\langle k, l\rangle$, then there is at most one such B. That is, suppose $B = (T, S)$ is a bi-special rim hook tabloid of shape λ and type $\langle(p^k), (pn^l)\rangle$. Thus for some $\mu \leqq \lambda$, $T \in$ T-SRHT $((p^k), \mu)$ and $S \in$ SRHT $((pn^l), \lambda/\mu)$. First, note that since each t-special rim hook t of T is of size p and must contain at least one cell in the first row of T, μ is a shape with at most p rows. Thus all cells in F_λ in a row higher than p must be covered by rim hooks from S. Moreover, we claim no rim hook from S can start in a row $i \leqq p$. For suppose h is a rim hook of size np that starts in cell $(i, 1)$ where $i \leqq p$, then$c(h) = |h| + 1 - r(h) = np + 1 - r(h) \geqq np + 1 - i \geqq np - p + 1$. That is, h covers at least $np - p + 1$ columns and hence cannot be contained in F_λ that has at most $np - p$ columns by assumption. But this means S is uniquely determined by λ since we must start in the top North-West square of F_λ and successively begin to fill F_λ with special rim hooks of size np until we reach a point where the next rim hook is to start in a row i where $i \leqq p$. The remaining cells that are not covered by the rim hooks of size np must be of shape μ. Then we start to fill F_μ with t-special rim hooks of size p starting in the South-East corner of F_μ. Since all t-special rim hooks of T are of the same size, we never have any choice in this filling so that T is uniquely determined by λ as well. Let us say that λ is (np, p)-viable if there is a bi-special rim hook tabloid of shape λ and type $\langle(p)^k, (np)^l\rangle$. Note that the following is an algorithm to determine if λ is (np, p)-viable if $l(\lambda') \leqq np - p$.

```
Algorithm to determine if $\lambda$ is $(np,p)$-viable for $l(\lambda') \leqq np - p$
Step 1. Set $\mu = \lambda$.
Step 2. If $l(\mu) > p$, go to Step 3, otherwise go to Step 4.
Step 3. If $\mu$ has a special rim hook $h$ of size $np$, let $\mu^* = F_\mu - h$ denote the shape
        that results by removing the cells of $h$ from $F_\mu$. Then set $\mu = \mu^*$ and go to
        Step 2. If $\mu$ has no such special rim hook $h$, stop; $\lambda$ is not $(np,p)$-viable.
Step 4. If $\mu$ has a $t$-special rim hook $t$ of size $p$, then set $\mu = F_\mu - t$ and go to Step
        4. If $\mu$ has no such special rim hook, stop; $\lambda$ is $(np,p)$-viable if and only if
        $\mu = \phi$.
```

As an example of the algorithm, suppose $n = 4$ and $p = 3$. Figure 3.1(a) shows that $\lambda = (2, 5^2, 6^2, 7, 8, 9)$ is not $(12, 3)$-viable because we get stopped at the second round of Step 3. Figure 3.1(b) shows $\lambda = (5, 7, 9^5)$ is not $(12, 3)$-viable because we get stopped the first time we reach stage 4. Figure 3.1(c) shows $(1^2, 3, 6, 8^2, 9^3)$ is $(12, 3)$-viable.

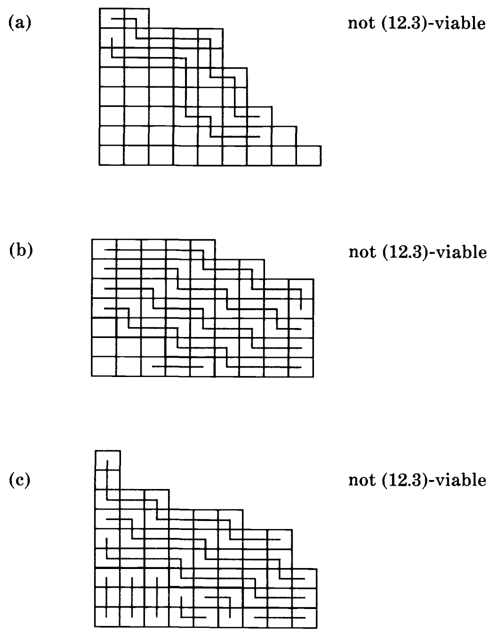


FIG. 3.1

Now if λ is (np,p)-viable and $B = (T, S)$ is the bi-special rim hook tabloid of shape λ and type $\langle (p^k), (np)^l \rangle$, then by (3.7) and the fact that $|h| + 1 = r(h) + c(h)$,

$$\omega_{p,q}(B) = \prod_{t \in T} r\text{-sgn}(t) \prod_{h \in S} -r\text{-sgn}(h). \tag{3.8}$$

Thus we have proved the following theorem.

THEOREM 3.1. *Suppose*

$$\prod_i (1 + x_i^p + x_i^{2p} + \cdots + x_i^{(n-1)p}) = \sum_\lambda d_\lambda S_\lambda(x). \tag{3.9}$$

Then

(a) $d_\lambda = 0$ *if* $|\lambda| \neq 0 \bmod p$ *or* $l(\lambda') > np - p$;

(b) *if* $|\lambda| = 0 \bmod p$ *and* $l(\lambda') \leqq np - p$, *then* $d_\lambda = 0$ *if* λ *is not* (np,p)*-viable and if* λ *is* (np,p)*-viable,* $d_\lambda = \prod_{t \in T} r\text{-sgn}(t) \prod_{h \in S} -r\text{-sgn}(h)$ *where* $B = (T, S)$ *is the unique bi-special rim hook tabloid of shape* λ *and type* $\langle (p^k), (np^l) \rangle$.

Note that we can now use Theorem 2.6 to derive the following series from Theorem 3.1.

COROLLARY 3.2. *Let* d_λ *be defined as in Theorem* 3.1. *Then*

$$\text{(a)} \prod_i (1 + x_i^p + \cdots + x_i^{p(n-1)})^{-1} = \sum_\lambda (-1)^{|\lambda|} d_{\lambda'} S_\lambda, \tag{3.10}$$

$$\text{(b)} \prod_i \frac{1 + (-y_i)^p + (-y_i)^{2p} + \cdots + (-y_i)^{(n-1)p}}{(1 + x_i^p + \cdots + x_i^{p(n-1)})} = \sum_\lambda (-1)^{|\lambda|} d_{\lambda'} HS_\lambda(x,y), \tag{3.11}$$

$$\text{(c)} \prod_i \frac{1 + x_i^p + \cdots + x_i^{p(n-1)}}{1 + (-y_i)^p + (-y_i)^{2p} + \cdots + (-y_i)^{(n-1)p}} = \sum_\lambda (-1)^{|\lambda|} d_\lambda HS_\lambda(x,y). \tag{3.12}$$

Next we consider series of the form $\prod_i (1-x_i^n)/(1-x_i^2)$. First, if $n=0$, it is known that the series $\prod_i 1/(1-x_i^2)$ is multiplicity free. Actually, for any $p>0$, the series $\prod_i 1/(1-x_i^p)$ is multiplicity free, see [3]–[5] and [15]. In fact, it is easy to see directly from Theorem 2.6(a) that

$$(3.13)\qquad \prod_i 1/(1-x_i^p)=\sum_\lambda c_\lambda S_\lambda(x),$$

where $c_\lambda = \prod_{h\in T} r\text{-sgn}(h)$ if there is a t-special rim hook tabloid T of type (p^k) for some k and $c_\lambda = 0$ if there is no such T. Here we use the fact that if $|\lambda| \neq 0 \bmod p$, then there is no t-SRHT T of shape λ and type (p^k) and if $|\lambda| = kp$, then there can be at most one t-SRHT T of shape λ and type (p^k). Thus $c_\lambda \in \{0, \pm 1\}$ for all λ and the series $\prod_i 1/(1-x_i^p)$ is multiplicity free in general. We note that our evaluation of c_λ in (3.13) is precisely the one given in [3].

Now if $n=2k$, then the series $\prod_i (1-x_i^{2k})/(1-x_i^2)$ is a special case of the series in Theorem 3.1 and hence is multiplicity free. Thus we need only consider series of the form

$$(3.14)\qquad \prod_i \frac{1-x_i^{2k+1}}{1-x_i^2}=\prod_i \frac{1+x_i+x_i^2+\cdots+x_i^{2k}}{1+x_i}=\sum_\lambda b_\lambda S_\lambda(x).$$

First we apply Proposition 2.1(a) with $p(x)=1+x$ and $q(x)=\prod_{i=1}^{2k}(1+\alpha^i x)$, where α is a primitive $(2k+1)$st root of unity to conclude that

$$(3.15)\qquad \begin{aligned}\prod_i \frac{q(-x_i)}{p(x_i)}&=\prod_i \frac{1+x_i+x_i^2+\cdots+x_i^{2k}}{1+x_i}\\ &=\sum_\lambda S_\lambda(x) HS_\lambda(-1;-\alpha,-\alpha^2,\cdots,-\alpha^{2k}).\end{aligned}$$

Now any hook Schur function $HS_\lambda(x_1;y_1,\cdots,y_{2k})=0$ if F_λ contains any cells in a row $i \geqq 2$ that lie in a column $j>2k$. That is, $HS_\lambda(x_1;y_1,\cdots,y_{2k})\neq 0$ implies F_λ must lie within the $(1, 2k)$-hook pictured in Fig. 3.2. Thus it follows immediately from (3.15) that

$$(3.16)\qquad b_\lambda = 0 \text{ if } F_\lambda \text{ does not lie in the } (1,2k)\text{-hook}.$$

Next we apply Theorem 2.5(a) with $q(x)=1+x^{2k+1}$ and $p(x)=1-x^2$ to conclude that

$$(3.17)\qquad \begin{aligned}\prod_i \frac{q(-x_i)}{p(x_i)}&=\prod_i \frac{1-x_i^{2k+1}}{1-x_i^2}\\ &=\sum_\lambda S_\lambda(x)\left(\sum_{\langle\nu,\gamma\rangle}\omega_{p,q}(B_{\langle\nu,\gamma\rangle,\lambda})\right),\end{aligned}$$

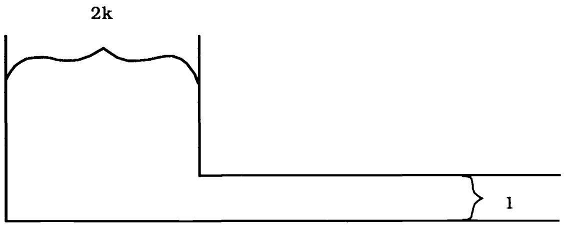


FIG. 3.2

where

$$\omega_{p,q}(B_{\langle\nu,\gamma\rangle,\lambda}) = \sum_{B\in \text{B-SRHT}(\langle\nu,\gamma\rangle,\lambda)} \omega_{p,q}(B) \tag{3.18}$$

and if $B = (T, S) \in$ B-SRHT $(\langle\nu, \gamma\rangle, \lambda)$, then

$$\begin{aligned} \omega_{p,q}(B) &= \omega_{p,q}(T,S) \\ &= \prod_{t\in T} -r\text{-sgn}\,(t)p_{|t|} \prod_{h\in S} -c\text{-sgn}\,(h)q_{|h|} \\ &= \prod_{t\in T}(-1)^{r(t)}(-1)\chi(|t|=2)\prod_{h\in S}(-1)^{c(h)}\chi(|h|=2k+1). \end{aligned} \tag{3.19}$$

We can see from (3.18) and (3.19) that $\omega_{p,q}(B_{(\langle\nu,\gamma\rangle\lambda)}) = 0$ unless ν is of the form (2^a) and γ is of the form $((2k+1)^b)$. We claim that if λ is a shape that fits inside the $(1, 2k)$-hook, then there is at most one pair $\langle a, b\rangle$ for which there exist $B \in$ B-SRHT $(\langle(2^a), ((2k+1)^b)\rangle, \lambda)$ and if there is such a pair $\langle a, b\rangle$, then there is at most one such B. That is, suppose $B = (T, S)$ is a bi-special rim hook tabloid of shape λ and type $\langle(2^a), ((2k+1)^b)\rangle$. Thus for some $\mu \leqq \lambda$, $T \in$ t-SRHT $((2^a), \mu)$ and $S \in$ SRHT $(2k+1)^a$, $\lambda/\mu)$. Since each t-special rim hook t of T is of size 2, it follows that μ is a shape with at most two rows and $|\mu| = 2a$. Thus all cells in F_λ in a row higher than 2 must be covered by rim hooks from S. Thus if $l(\lambda) > 2$, then we must start to fill F_λ, starting in the top North-West corner, with special rim hooks of size $2k+1$ until we reach a point where the next rim hook we place would start in a row i where $i \leqq 2$. Let $\nu \leqq \lambda$ be the shape of the cells that are not covered by rim hooks at this point. We know that $\mu \leqq \nu$. Note also that because λ is contained in the $(1, 2k)$-hook, it follows that any special rim hook of size $2k+1$ that starts in a row $i \leqq 2$ must reach row 1. This means that we can place at most one special rim hook h of size $2k+1$ in F_ν and h must end at the end of row 1 of F_ν. Thus there is at most one way in which we can place a special rim hook h of size $2k+1$ in F_ν. Now if $|\nu|$ is even, then it must be that S does not intersect ν because otherwise μ is the shape that results from ν by removing a rim hook of size $2k+1$ starting at the end of row 1. But then $|\mu| = |\nu| - (2k+1)$ is odd, contradicting our choice of $|\mu| = 2a$. Thus if $|\nu|$ is even, $\mu = \nu$. If $|\nu|$ is odd, then we cannot have $\mu = \nu$ so that S must contain the special rim hook of size $2k+1$ that ends at the end of row 1. In either case, we see that S is completely determined by λ. Once S has been determined, we see that T is completely determined as well because all special rim hooks of T are of the same length. Let us say that λ is $(2k+1, 2)$-viable if there is a bi-special rim hook tabloid of shape λ and type $\langle(2^a), ((2k+1)^b)\rangle$. The following algorithm determines if λ is $(2k+1, 2)$-viable is λ is contained in the $(1, 2k)$-hook.

Algorithm to determine if λ is $(2k+1,2)$-viable for λ contained in a $(1,2k)$-hook

Step 1. Set $\mu = \lambda$.

Step 2. If $l(\mu) > 2$, go to Step 3, otherwise go to Step 4.

Step 3. If μ has a special rim hook h of size $2k+1$, let $\mu^* = F_\mu - h$ denote the shape that results by removing the cells of h from F_μ. Set $\mu = \mu^*$ and go to Step 2. If μ has no such special rim hook h, stop; λ is not $(2k+1,2)$-viable.

Step 4. If $|\nu|$ is odd, go to Step 5; if $|\mu|$ is even, go to Step 6.

Step 5. If μ has a rim hook h of size $2k+1$ that ends in the first row of μ, then set $\mu = F_\mu - h$ and go to Step 6. If μ has no such special rim hook h, stop; λ is not $(2k+1,2)$-viable.

Step 6. If μ has a t-special rim hook t of size 2, then set $\mu = F_\mu - t$ and go to Step 6. If μ has no such special rim hook, stop; λ is $(2k+1,2)$-viable if and only if $\mu = \phi$.

Note that the algorithm simply tells us to fill in F_λ with special rim hooks of size $2k+1$ until our next rim hook is to start in a row $i \leqq 2$. At that point, we have a decision based on whether we have an odd number or even number of squares to fill. This situation is pictured in Fig. 3.3 for the case of $k = 2$. In both Figs. 3.3(a) and 3.3(b), on the left we have reached the point where the next special rim hook of size 5 should start in row 2. In Fig. 3.3(a), we have eight cells remaining so we must now start to fill in with t-special rim hooks of size 2. In Fig. 3.3(b), we have seven cells left, so we must add one more rim hook of size 5 ending at the end of row 1 and then fill in the rest of the cells with t-special rim hooks of size 5.

Now if λ is $(2k+1, 2)$-viable and $B = (T, S)$ is a bi-special rim hook tabloid of shape λ and type $\langle(2^a), ((2k+1)^b)\rangle$, then by (3.19), we have

$$\omega_{p,q}(B) = \prod_{t\in T} r\text{-sgn}\,(t) \prod_{h\in S} -c\text{-sgn}\,(h). \tag{3.20}$$

Thus we have proved the following theorem.

THEOREM 3.3. *Suppose*

$$\prod_i \frac{1-x_i^{2k+1}}{1-x_i^2} = \sum_\lambda b_\lambda S_\lambda(x). \tag{3.21}$$

Then

(a) $b_\lambda = 0$ *if* λ *is not contained in the* $(1, 2k)$*-hook*;

(b) *otherwise* $b_\lambda = 0$ *if* λ *is not* $(2k+1, 2)$*-viable and if* λ *is* $(2k+1, 2)$*-viable, then* $b_\lambda = \prod_{t\in T} r\text{-sgn}\,(t) \prod_{h\in S} -c\text{-sgn}\,(h)$, *where* $B = (T, S)$ *is the unique bi-special rim hook tabloid of shape* λ *and type* $\langle(2^a), ((2k+1)^b)\rangle$.

Note that we can now apply Theorem 2.5 to derive the following series from Theorem 3.3.

COROLLARY 3.4.

$$\text{(a)} \prod_i \frac{1-x_i^2}{1+x_i^{2k+1}} = \sum_\lambda S_\lambda(x) b_{\lambda'}, \tag{3.22}$$

$$\text{(b)} \prod_i \frac{(1-x^{2k+1})(1-y_i^2)}{(1-x_i^2)(1+y_i^{2k+1})} = \sum_\lambda HS_\lambda(x;y) b_\lambda. \tag{3.23}$$

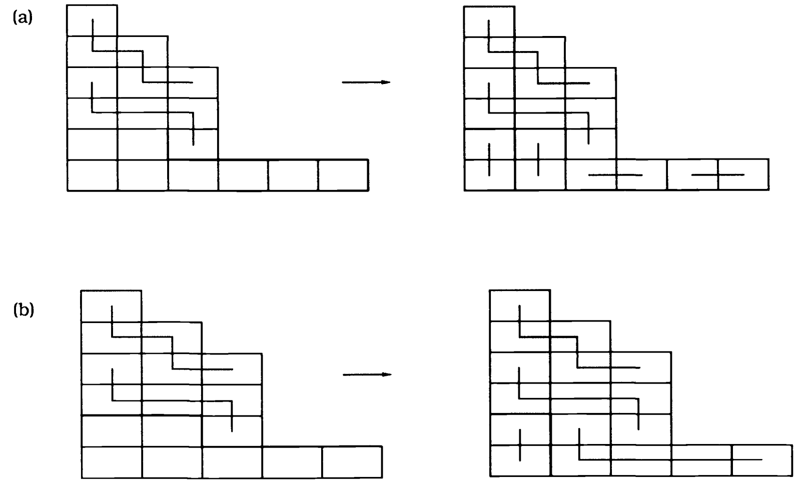


FIG. 3.3

We note that we can apply the same type of argument used to prove Theorem 3.3 to try to evaluate any series of the form $\prod_i (1 - x_i^m)/(1 - x_i^p) = \sum_\lambda a_\lambda S_\lambda(x)$. We can show that $a_\lambda = 0$ unless λ is contained in the $(p - 1, m - 1)$-hook. Also, $a_\lambda = 0$ unless there are bi-special rim hook tabloids B of shape λ and type $\langle (p^a), (m^b) \rangle$ for some a and b. However, for $p \geqq 3$, it is no longer the case that if λ is contained in the $(p - 1, m - 1)$-hook, then

$$|\bigcup_{a,b} \text{B-SRHT}(\langle (p^a),(m^b)\rangle,\lambda)| \leqq 1.$$

Thus we cannot immediately conclude that all such series are multiplicity free. We can, however, get a bound on a_λ for such series and in a number of special cases prove that the series is multiplicity free. For example, all series of the form $\prod_i (1 - x_i^m)/(1 - x_i^3)$ are multiplicity free. Such results will appear in [13].

Note added in proof. The authors can now show that all S-series of the form $\prod_i (1 - x_i^m)/(1 - x_i^n)$ are multiplicity free. These results will appear in [13].

REFERENCES

[1] A. Berele and A. Regev, *Hook Young diagrams with applications to combinatorics and to representations of Lie superalgebras*, Adv. in Math., 64 (1987), pp. 118–175.

[2] A. Berele and J. B. Remmel, *Hook flag characters and their combinatorics*, J. Pure Appl. Algebra, 35 (1985), pp. 222–245.

[3] Y. M. Chen, *Combinatorial Algorithms for Plethysm*, Ph.D. thesis, Univ. of California, San Diego, 1982.

[4] Y. M. Chen, A. M. Garsia, and J. B. Remmel, *Algorithms for Plethysm*, Contemp. Math., 34 (1984), pp. 109–153.

[5] D. G. Duncan, *On D. E. Littlewood's algebra of S-functions*, Canad. J. Math., 4 (1952), pp. 504–512.

[6] O. Egecioglu and J. B. Remmel, *A combinatorial interpretation of the inverse Kostka matrix*, Lin. and Multilin. Alg., 26 (1990), pp. 59–84.

[7] R. C. King and B. G. Wybourne, *Holomorphic discrete series and harmonic series unitary irreducible representations of non-compact Lie groups*: $S_p(2n, R)$, $U(p, q)$ *and* $SP^*(2n)$, J. Phys. A., 18 (1985), pp. 3113–3139.

[8] R. C. King, B. G. Wybourne, and M. Yang, *Slinkies and the S-function content of certain generating functions*, J. Phys. A: Math Gen. 22 (1989), pp. 4519–4535.

[9] A. Lascoux and P. Pragacz, *S-function series*, J. Phys. A, 21 (1988), p. 4105.

[10] D. E. Littlewood, *The Theory of Group Characters*, Second edition, Oxford University Press, London, 1950.

[11] J. B. Remmel, *Permutation statistics and* (k, l)*-hook Schur functions*, Discrete Math., 67 (1987), pp. 271–298.

[12] ———, *The combinatorics of* (k, l)*-hook Schur functions*, Contemp. Math., 34 (1984), p. 253.

[13] J. B. Remmel and M. Yang, *On the S-series for series of the form* $\prod_i (1 - x_i^n)/(1 - x_i^m)$, in preparation.

[14] D. J. Rowe, B. G. Wybourne, and P. H. Butler, *Unitary representations, branching rules and matrix elements for the non-compact symplectic groups*, J. Phys. A, 18 (1965), pp. 939–953.

[15] J. A. Todd, *A note on the algebra of S-functions*, Proc. Cambridge Philos. Soc., 45 (1949), pp. 328–334.

[16] M. Yang and B. G. Wybourne, *S-function series and non-compact Lie groups*, J. Phys. A, 19 (1986), pp. 3513–3525.

SIAM J. DISC. MATH.
Vol. 4, No. 2, pp. 275-280, May 1991

BOOTSTRAP PERCOLATION, THE SCHRÖDER NUMBERS, AND THE N-KINGS PROBLEM*

LOUIS SHAPIRO† AND A. B. STEPHENS‡

Abstract. A percolation process on $n \times n$ 0-1 matrices is defined. This process is defined so that a zero entry becomes one if two or more of its neighbors have the value one. Entries that have the value one never change. The process halts when no more entries can change. The initial matrices are taken to be all the $n \times n$ permutation matrices.

It is shown that the number of matrices that eventually become all ones is given by the Schröder numbers. Asymptotically, the proportion of such matrices approaches zero. Next, matrices that exhibit no growth at all are considered. The number of such matrices is given in terms of a generating function, and the proportion of such matrices approaches e^{-2} as n goes to infinity. The methods used involve bracketing, trees, and generating functions.

Key words. bootstrap percolation, Schröder numbers, trees, generating function, permutation matrix

AMS(MOS) subject classifications. 05A15, 82A43

In this paper we consider some combinatorial problems dealing with bootstrap percolation. In this percolation process we start with an $n \times n$ matrix where each entry is 0 or 1. The entries of this matrix change by iteration according to a certain rule. According to the rule, at each iteration we look for any 0 entry that has two or more of its nearest neighbors (above, below, right, left) equal to 1, and change all such zeros to ones. We repeat this process until no more changes are possible.

Following are some examples:

$$\text{(A)}\quad \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$\text{(B)}\quad \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

$$\text{(C)}\quad \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

We would say in example A that the matrix fills up. In example B, a 2×2 submatrix fills up, while in C there is no growth.

The process described above is an example of bootstrap percolation. Many variations are possible by changing the rule. For example, "two or more" could change, the matrix could be rectangular, cylindrical, or infinite, and so forth. The topics of percolation and, more generally, cellular automata have been very actively researched in recent years and the books of Grimett [9], Kesten [12], and Durrett [5] provide a good introduction, as does the survey article of Durrett [4]. See also [2] and [8].

* Received by the editors November 20, 1989; accepted for publication May 18, 1990.

† Department of Mathematics, Howard University, Washington, D.C. 20059.

‡ Department of Computer Science, University of Maryland, Baltimore County, Cantonsville, Maryland 21228.

This article, however, is essentially self-contained. We want to consider bootstrap percolation as discussed above in the special case where the initial matrix has exactly one 1 in each row and column; i.e., the matrix is a permutation matrix. For the $n \times n$ case there will be $n!$ initial matrices. Note that both examples A and C start with permutation matrices. If the matrix is considered to be a chessboard and a 1 becomes a rook then these permutation matrices become configurations of n nontaking rooks.

We will answer the following questions:

(1) How many of the $n!$ initial configurations will fill up?

(2) For large n what proportion fill up?

(3) How many initial configurations allow for no growth?

(4) For large n what proportion allow no growth?

The answers to (2) and (4) are 0 and e^{-2} while the answer to (1) is the sequence of Schröder numbers. A computer was used to find the first few terms: 1, 2, 6, 22, 90, 394, $\cdots$ of (1), which suggested the $(n-1)$st Schröder number (a surprise to us). The answer to (3) involves a simple generating function that has radius of convergence equal to 0.

1. Preliminaries. Since the Schröder numbers are central to this paper but not widely known, we give a brief discussion and some references. They are defined as follows. The Schröder numbers have as their generating function

$$R(x) = \sum_{n=0}^{\infty} r_n x^n, \quad \text{where } R(x) = 1 + x(R(x) + (R(x))^2)$$

or more briefly $R = 1 + x(R + R^2)$. Thus

$$R(x) = (1 - x - \sqrt{1 - 6x + x^2})/2x.$$

Another equivalent formulation is the recurrence

$$(n+1)r_n = 3(2n-1)r_{n-1} - (n-2)r_{n-2}, \qquad n \geqq 2$$

with $r_0 = 1$, $r_1 = 2$.

The first few Schröder numbers are $\{r_n\}_{n=0}^{\infty} = \{1, 2, 6, 22, 90, 394, 1806, \cdots\}$.

Two places where the Schröder numbers appear are the following:

(A) Count the number of random walks from (0, 0) to (N, N) that stay below the line $y = x + 1$, where the set of possible steps is (0, 1), (1, 0) and (1, 1).

(B) Count the permutations possible using a double-ended input-restricted queue (i.e., deque). See Knuth [13] for an excellent exposition complete with railroad tracks.

The Schröder numbers are closely related to the Catalan numbers [3], [7], [14], one connection being the equation

$$r_n = \sum_{i=0}^{n} \binom{2n-i}{i} C_{n-i},$$

where $C_m = (1/(m+1))\binom{2m}{m}$ is the mth Catalan number. In addition to the reference to Knuth, the Schröder numbers appear in [3], [15], and [16] while the original appearance [17] dates back to 1870. See also [6].

2. Matrices that fill up. We are now ready to prove our main result.

THEOREM 1. *The number of $n \times n$ permutation matrices that fill up is r_{n-1}, the $(n-1)$st Schröder number.*

Proof. Let $\pi = a_1 a_2 \cdots a_n$ be an arbitrary permutation of $\{1, 2, \cdots, n\}$. We first define inductively what we mean by a block.

(i) Each a_i is a block.

(ii) If two adjacent blocks B and B^* contain elements a and a^*, respectively, that are a consecutive increasing or decreasing pair, then form a new block (BB^*) or $[BB^*]$ using ( ) for the increasing case and [ ] for the decreasing case.

Note that blocks are disjoint and each block of cardinality m contains the integers $(i, i+1, \cdots, i+m)$ for some i, $1 \leqq i \leqq n$.

We repeat the following process until no new blocks can be formed.

The permutation π can be considered as a sequence of blocks. We read π left to right and use (ii) to form new blocks whenever possible.

This process is illustrated in the following example:

If $\pi = 1\ 9\ 8\ 3\ 4\ 5\ 2\ 7\ 6$ the first pass evolves as

$$1 \to 1[98] \to 1[98](34) \to 1[98]((34)5) \to 1[98][((34)5)2] \to 1[98][((34)5)2][76]$$

as we read from the left to right.

pass 1: 1[98][((34)5)2][76]

pass 2: 1[98]([((34)5)2][76])

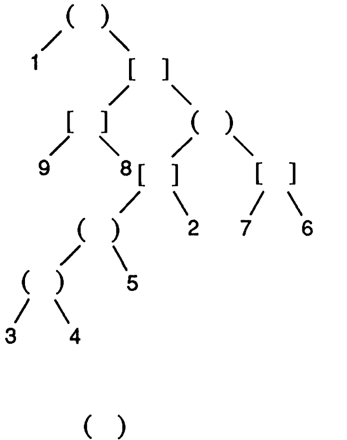
pass 3: 1[[98]([((34)5)2][76])]

pass 4: (1[[98]([((34)5)2][76])])

Those permutations ending up in a single block are in a direct correspondence to the permutation matrices that fill up.

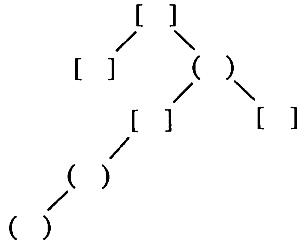
To see this, let P be a permutation matrix with corresponding permutation π. Blocks in π correspond to subsquares of P, which have already filled out. The case of two blocks being merged by rule (ii) corresponds to two subsquares of 1's within P, which meet at a corner and percolate to form a larger subsquare just large enough to contain both subsquares.

For such permutations we now form a binary tree as follows:



,

which we redraw as follows:



.

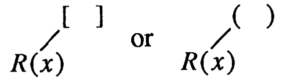
Note that we do not need the labels for the leaves since starting at the top we must have $\{1\}$ to the left and $\{2, 3, \cdots, 9\}$ to the right and so on.

Examining the structure of this tree in more detail, we note that the type, ( ) or [ ], of any node must differ from the type of its right child. To see this, note that if a node and its right child are both of type ( ) then this must result from two consecutive increases. But a string $j, j+1, j+2$ becomes $((j, j+1)j+2)$ which gives a left child, not a right child. Similarly, let B_j, B_{j+1}, B_{j+2} be blocks where the elements of B_j are less than those of B_{j+1}, which in turn are less than those of B_{j+2} and where consecutive blocks do contain consecutive elements. These blocks become $((B_j, B_{j+1})B_{j+2})$ which also gives a left child, not a right one. The reasoning for [[,]] is similar.

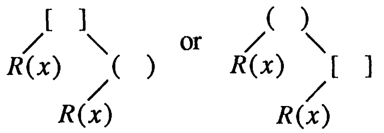
In the example above, the subsequence 3, 4, 5 is illustrative. If a node in the tree is a block consisting of more than one a_i then it must be of the form (BB^*) or $[BB^*]$. We say that B is its left child and B^* is its right child. A block consisting of a simple element is a leaf of the tree. If we now go over to generating functions, we obtain the equation

$$R(x) = 1 + 2xR(x) + 2x^2R^2(x) + 2x^3R^3(x) + \cdots,$$

where 1 counts the empty set, $2xR$ represents



while $2x^2R^2(x)$ represents



and so forth. Thus

$$R(x) = 1 + \frac{2xR(x)}{1 - xR(x)}$$

and

$$R(x) = 1 + x(R(x) + (R(x))^2).$$

This completes the proof.

Note that the generating function for $\{r_{n-1}\}_{n=1}^{\infty}$ is

$$xR(x) = r_0x + r_1x^2 + r_2x^3 + \cdots.$$

Having established that we have the Schröder numbers we can use the results of Knuth [13]. We obtain via the quadratic formula, partial fractions, and Stirling's formula that

$$r_n \sim \frac{C(3+\sqrt{8})^n}{n^{3/2}} \quad \text{with } C = \frac{1}{2}\sqrt{\frac{3\sqrt{2}-4}{\pi}} \approx 0.139,$$

which in turn answers question (2) as follows.

COROLLARY. $\lim_{n\to\infty}(r_{n-1}/n!) = 0$.

3. No growth configurations. We now turn to the question of counting matrices where no growth occurs. The first two interesting cases occur when $n = 4$, which yields the permutations 3 1 2 4 and 2 4 1 3. Here is a brief table for the first few no growth numbers.

n	0	1	2	3	4	5	6	7	8
a_n	1	1	0	0	2	14	90	646	5242

Let $A(x) = \sum_{n=0}^{\infty} a_n x^n$. It then follows that

$$A(xR(x)) = \sum_{n \geqq 0} n! x^n = \varepsilon(x),$$

$\varepsilon(x)$ being defined by the last equality. (Intuitively, we let each of $n!$ initial configurations fill out as far as possible. We end up with one, four, or more subsquares arranged in a no growth configuration. The growth in each subsquare is counted by the generating function $xR(x)$. For instance, if we were to end up with six subsquares the relevant generating function would be $a_6(xR(x))^6$. Summing yields the last result.) The $\varepsilon(x)$ function goes back to Euler and is highly nonconvergent but not completely unmanageable. Since the functional inverse of $xR(x)$ is $x((1-x)/(1+x))$ we have $A(x) = \varepsilon(x((1-x)/(1+x))$, which, at least in some sense, answers (3).

The no growth numbers (caffeine numbers?) actually go back to a problem in recreational mathematics: in how many ways can n kings be placed in an $n \times n$ board, one king on each row and column, so that no two kings can take each other? A result of Kaplansky [10], [11] says the following: Let $P(n, r)$ be the probability that on an $n \times n$ board there are r pairs of kings that can take each other. Then as n gets large the $P(n, r)$, $0 \leqq r \leqq n-1$ approach a Poisson distribution with $\lambda = 2$. In particular, $P(n, 0) = (a_n/n!) \rightarrow e^{-2}$, thus answering (4) while $A(x) = \varepsilon(x(1-x)/(1+x))$ is the answer to (3) in terms of a generating function. The coefficient of $w^k x^n$ in the expansion of $A(wxR(x))$ provides the number of $n \times n$ initial configurations that fill out to form k subsquares.

Further information on the n-kings problem is given in [1], while [18] is an invaluable reference for integer sequences.

REFERENCES

[1] M. Abramson and M. O. J. Moser, *Permutations without rising or falling w-sequences*, Ann. Math. Statist., 38 (1967), pp. 1245–1254.

[2] M. Aizenmann and J. Lebowitz, *Metastability effects in bootstrap percolation*, J. Physics (A), 21 (1988), pp. 3801–3813.

[3] L. Comtet, *Advanced Combinatorics*, D. Reidel, Boston, MA, 1974.

[4] R. Durrett, *Crabgrass, measles, and gypsy moths: An introduction to interacting particle systems*, Mathematical Intelligencer, 10 (1988), pp. 37–47.

[5] ———, *Lecture Notes on Particle Systems and Percolation*, Wadsworth and Brooks/Cole, Pacific Grove, CA, 1988.

[6] D. Gouyou-Beauchamp and B. Vauquelin, *Deux propriétés combinatoires des nombres de Schröder*, RAIRO Inform. Theor. Appl., 22 (1988), pp. 361–388.

[7] R. L. Graham, D. Knuth, and O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science*, Addison-Wesley, New York, 1988.

[8] D. Griffeath, *Cyclic random competition, A case history in experimental mathematics*, Notices Amer. Math. Soc., 36 (1988), pp. 1472–1480.

[9] G. Grimett, *Percolation*, Springer-Verlag, Berlin, New York, 1989.

[10] I. Kaplansky, *Symbolic solution of certain problems in permutations*, Bull. AMS, 50 (1944), pp. 906–914.

[11] ———, *The asymptotic distribution of runs of consecutive elements*, Ann. Math. Statist. 16 (1945), pp. 200–203.

[12] H. KESTEN, *Percolation Theory for Mathematicians*, Birkhausen, Basel, Switzerland, 1982.

[13] D. KNUTH, *The Art of Computer Programming*, *Fundamental Algorithms*, vol. 1, Addison-Wesley, New York, 1968. Second edition, 1973, Section 2.2.1.

[14] J. RIORDAN, *Combinatorial Identities*, John Wiley, New York, 1968.

[15] D. ROGERS AND L. SHAPIRO, *Some correspondence involving the Schröder numbers and relations*, in Combinatorial Mathematics, D. A. Holton, ed., Lecture Notes in Mathematics, vol. 686, Springer-Verlag, Berlin, New York, 1978.

[16] ———, *Deques, trees and lattice paths*, Combinatorial Mathematics, vol. VIII, K. L. McAvaney, ed., Lecture Notes in Mathematics, vol. 884, Springer-Verlag, Berlin, New York, 1981.

[17] E. SCHRÖDER, *Vier Combinatorische Probleme*, Zeitschrift fur Mathematik und Physik, 15 (1870), pp. 361–376.

[18] N. SLOANE, *A Handbook of Integer Sequences*, Academic Press, New York, 1973.

SIAM J. DISC. MATH.
Vol. 4, No. 2, pp. 281-292, May 1991

DISJOINT COVERS IN REPLICATED HETEROGENEOUS ARRAYS*

P. K. MCKINLEY†‡, N. HASAN†§, R. LIBESKIND-HADAS†, AND C. L. LIU†

Abstract. Reconfigurable chips are fabricated with redundant elements that can be used to replace the faulty elements. The fault cover problem consists of finding an assignment of redundant elements to the faulty elements such that all of the faults are repaired. In reconfigurable chips that consist of arrays of elements, redundant elements are configured as spare rows and spare columns.

This paper considers the problem in which a chip contains several replicates of a heterogeneous array, one or more sets of spare rows, and one or more sets of spare columns. Each set of spare rows is identical to the set of rows in the array, and each set of spare columns is identical to the set of columns in the array. Specifically, an ith spare row can only be used to replace an ith row of an array, and similarly with spare columns. Repairing the chip reduces to finding a cover for the faults in *each* of the arrays. These covers must be disjoint; that is, a particular spare row or spare column can be used in the cover of at most one array. Results are presented for three fault cover problems that arise under these conditions.

Key words. reconfigurable chips, fault covers

AMS(MOS) subject classification. 94C15

1. Introduction.

As chip density increases, the likelihood of fabrication defects on chips also increases. Maintaining an acceptable yield in chip production requires the capability to repair defective chips. To this end, reconfigurable chips are fabricated with redundant elements that can be used to replace faulty elements. The *fault cover problem* consists of finding an assignment of redundant elements to the faulty elements such that all of the faulty elements are replaced.

For reconfigurable chips that consist of arrays of elements, redundant elements are configured as spare rows and spare columns [15]. Examples of such *reconfigurable arrays* include not only arrays of memory elements [19], but also arrays of processors [11], [14]. A *line* refers to a row or column of an array. In a reconfigurable array, each spare line can be activated by programming selection circuitry after fabrication to effectively replace lines containing faulty elements. The fault cover problem seeks an assignment of the spare lines to the array such that all of the faulty elements are repaired. The set of replaced lines is referred to as a *cover.*

In the model studied previously [8], [12], a row that contains faulty elements can be replaced by *any* spare row, and a column that contains faulty elements can be replaced by *any* spare column. An example of this model is shown in Fig. 1, in which X's indicate faulty elements. Assigning spare rows to rows 1 and 4 and spare columns to columns 2 and 6, marked with arrows, represents one possible repair solution for this array. The fault cover problem for this type of reconfigurable array is NP-complete [12]. Several algorithms, including exhaustive approaches and heuristics, have been developed for this problem [3], [4], [7], [12], [18], [19].

The situation in which a particular row (column) can be replaced only by a member of a proper subset of the spare rows (columns) arises when the elements in the array are not all identical. For example, consider the array shown in Fig. 2, which contains four

* Received by the editors January 16, 1990; accepted for publication (in revised form) May 8, 1990. This research was supported in part by National Science Foundation grant MIP 87-03273.

† Department of Computer Science, University of Illinois, 1304 West Springfield Avenue, Urbana, Illinois 61801.

‡ Current address, Department of Computer Science, Michigan State University, East Lansing, Michigan 48824.

§ Current address, IBM Corporation, Post Office Box 950, Poughkeepsie, New York 12602.

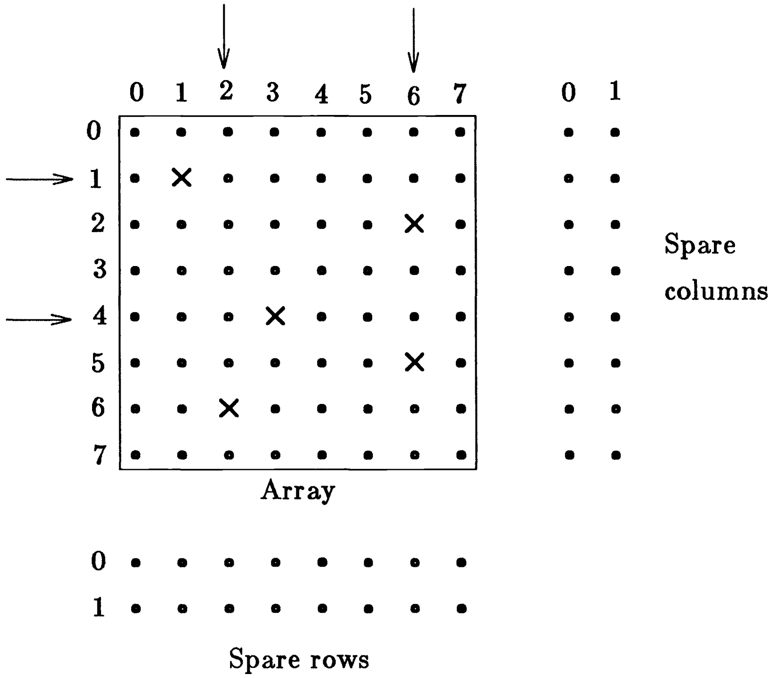


FIG. 1. *Reconfigurable* 8 × 8 *array with two spare rows and two spare columns.*

types of elements. In the configuration shown, the array comprises two types of rows and four types of columns. A spare row and column of each type is provided. Clearly, a line can be replaced only by a line of the same type.

We are concerned with problems in which such heterogeneity of array elements implies that the *i*th rows of all the arrays share one or more spares, and similarly for the

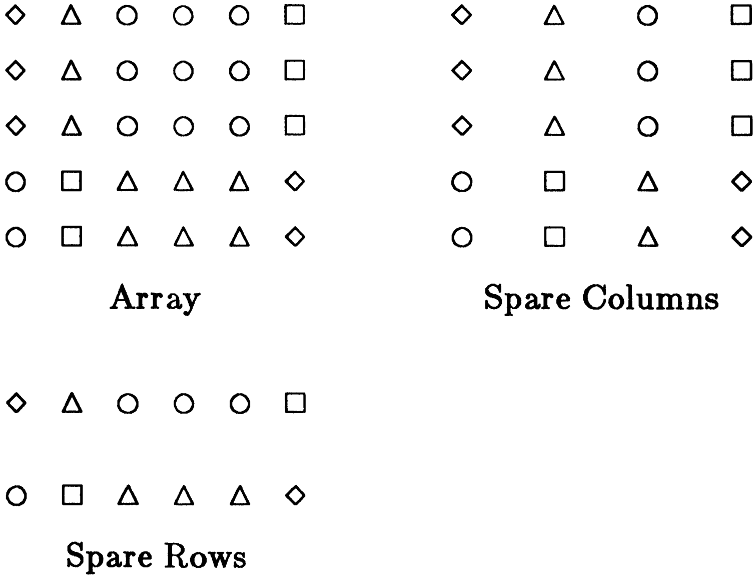


FIG. 2. *Heterogeneous reconfigurable array.*

*j*th columns of all the arrays. In other words, the set of spare rows is identical to the set of rows in the array, and the set of spare columns is identical to the set of columns in the array. When the chip contains a single array of elements, the problem of repairing faults is trivial. In fact, each fault can be covered in either of two ways, with a spare row or with a spare column. When the chip contains multiple copies of an array, however, repairing the chip reduces to finding a cover for the faults in *each* of the arrays. Figure 3 shows three copies of an array whose faults must be covered by lines from one set of spare rows and one set of spare columns. A spare line can be assigned to only one of the three arrays; that is, the three covers for the arrays must be *disjoint*.

To formulate the problem of finding disjoint covers for replicated heterogeneous arrays, we model each array as a (0, 1)-matrix, a 0 indicating a nonfaulty element and a 1 indicating a faulty element. Figure 4 shows an instance of the problem for the arrays depicted in Fig. 3. Each of the arrays contains two faulty elements. One solution to the cover problem, indicated with arrows, is the following: spare columns 1 and 2 are assigned to array 1; spare row 2 is assigned to array 2; spare column 4 is assigned to array 3. The covers are disjoint and all of the faults are covered.

In this paper, we present results for three fault cover problems for reconfigurable arrays in which the use of spare lines is constrained in the manner described above. These problems are the *feasibility problem*, the *disjoint minimum cover problem*, and the *multiple spare array problem*. In the first two problems, the chip is assumed to comprise *t* replicates of an array and one set each of spare rows and columns. The feasibility problem asks whether or not the chip can be repaired; the disjoint minimum cover problem seeks a feasible solution but with the stipulation that the individual cover of each array be minimum, that is, consisting of a minimum number of spare lines. In §§ 2 and 3, respectively, we show that these problems can be solved in polynomial time. The multiple spare array problem is a generalization of the feasibility problem in which more than two arrays of spares are available for use in covering faults. In § 4, we show that the multiple spare array problem is NP-complete. In § 5, we briefly discuss other potential applications for our fault cover model, and in § 6 we summarize our results.

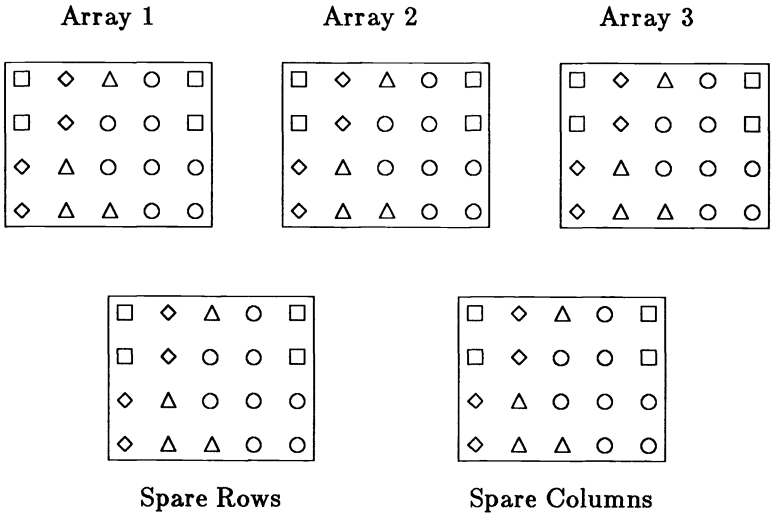
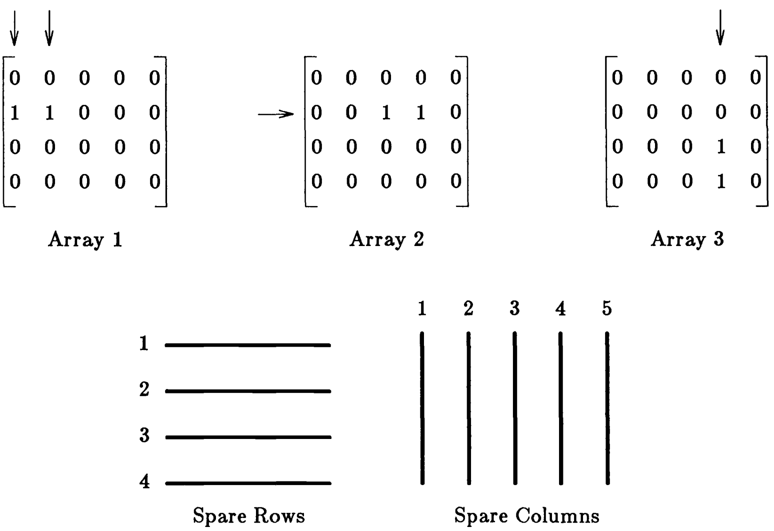


FIG. 3. *Replicated heterogeneous arrays and spares.*



$$\begin{bmatrix} 0&0&0&0&0\\ 1&1&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{bmatrix} \qquad \begin{bmatrix} 0&0&0&0&0\\ 0&0&1&1&0\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{bmatrix} \qquad \begin{bmatrix} 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&1&0\\ 0&0&0&1&0 \end{bmatrix}$$

FIG. 4. *Disjoint covers of arrays.*

2. Feasible disjoint covers. Given t copies of an $R \times C$ array, each containing zero or more faulty elements, an array of R spare rows and an array of C spare columns, the *feasibility problem* seeks an assignment of the spare lines to the arrays such that all faults are covered and no spare line is assigned to more than one array. The ith spare row may only be assigned to cover the ith row of an array, and the jth spare column may only be assigned to cover the jth column. To solve the feasibility problem, we show that the problem can be formulated as a multigraph coloring problem. This problem in turn can be reduced to the 2-satisfiability (2SAT) problem in which, given a set U of Boolean variables and a conjunction of 2-clauses over U, we seek an assignment of values to variables such that each of the clauses is *true*. The 2SAT problem is solvable in polynomial time using any one of several known algorithms [2], [5]. The multigraph is constructed as follows.

CONSTRUCTION 1. *Given a set of replicated arrays $A_1, A_2, \cdots, A_t$, we represent each fault i with a vertex v_i in a multigraph G. With each vertex v_i, we associate the label $(a_i{:}r_i, c_i)$, representing the array, row, and column, respectively, of the ith fault. For each pair of faults not in the same array that lie in the same rows of their respective arrays, we add a red edge to G between the vertices representing the faults. Similarly, we add a black edge for pairs of faults in the same columns of their respective arrays. Let V be the set of vertices and E the set of edges in G.*

The multigraph G may not be connected. In fact, if there exists a fault in row i and column j of an array, and if there are no faults in row i or column j of all other arrays, then this fault will be represented by an isolated vertex. Next, we consider the problem of assigning the colors red and black to the vertices of such a multigraph. We say that a coloring is *feasible* if every vertex is colored, no black edge has two black endpoints, and no red edge has two red endpoints.

THEOREM 1. *A feasible coloring for a multigraph resulting from Construction* 1 *exists if and only if there exist disjoint covers $K_1, K_2, \cdots, K_t$ for the arrays $A_1, A_2, \cdots, A_t$.*

Proof. In the following, $i, j, k, l \in \{1, 2, \cdots, |V|\}$. Assume that $K_1, K_2, \cdots, K_t$ are disjoint covers for $A_1, A_2, \cdots, A_t$. For each faulty element i, if spare row r_i is contained in K_{a_i}, then we color vertex v_i red; otherwise we color the vertex black. We claim that this coloring is feasible. If not, then there must exist a red edge whose endpoints, labeled $(a_i{:}r_i, c_i)$ and $(a_j{:}r_j, c_j)$, with $r_i = r_j$, are both red or a black edge whose endpoints, labeled $(a_k{:}r_k, c_k)$ and $(a_l{:}r_l, c_l)$, with $c_k = c_l$, are both black. The former case implies that row r_i is contained in two different covers, K_{a_i} and K_{a_j}. This contradicts our assumption that the covers are disjoint. The latter case implies that column c_k is contained in two different covers, K_{a_k} and K_{a_l}. Again, this is a contradiction.

Next, let C: $V \rightarrow \{red, black\}$ be a feasible coloring of the multigraph. We construct a set of covers $K_1, K_2, \cdots, K_t$ as follows. For each vertex v_i that is colored red, we include spare row r_i in cover K_{a_i}. For each vertex v_j that is colored black, we include spare column c_j in cover K_{a_j}. We claim that $K_1, K_2, \cdots, K_t$ are disjoint covers of $A_1, A_2, \cdots, A_t$. Since each vertex represents a faulty element, and for each vertex v_i at least one of row r_i and column c_i is included in cover K_{a_i}, it follows that $K_1, K_2, \cdots, K_t$ constitute covers for $A_1, A_2, \cdots, A_t$, respectively. Assume that $K_1, K_2, \cdots, K_t$ are not pairwise disjoint. If a row r_i is included in both K_{a_i} and K_{a_j}, then there must exist two vertices labeled $(a_i{:}r_i, c_i)$ and $(a_j{:}r_j, c_j)$, with $r_i = r_j$, both colored red and connected by a red edge, a contradiction. Similarly, if a column c_k is included in both K_{a_k} and K_{a_l}, then there must be two vertices labeled $(a_k{:}r_k, c_k)$ and $(a_l{:}r_l, c_l)$, with $c_k = c_l$, both colored black and connected by a black edge. Again, this is a contradiction. □

Figure 5 shows the multigraph corresponding to the arrays and their faults shown in Fig. 4. Red edges are depicted with solid lines, black edges with dashed lines. If vertices v_1, v_2, v_5, and v_6 are colored black and vertices v_3 and v_4 are colored red, then the coloring is feasible. From this solution, we generate disjoint covers for the arrays as follows: for each red vertex labeled $(a_i{:}r_i, c_i)$, we assign spare row r_i to cover row r_i in array a_i; for each black vertex labeled $(a_j{:}r_j, c_j)$, we assign spare column c_j to cover column c_j in array a_j. That is, spare columns 1 and 2 are assigned to array 1; spare row 2 is assigned

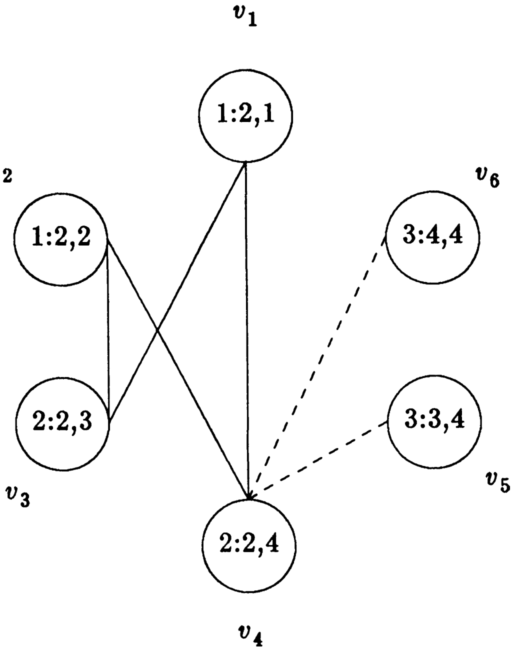


FIG. 5. *Multigraph for feasibility problem.*

to array 2; spare column 4 is assigned to array 3. This solution is indicated with arrows in Fig. 4.

To solve the feasibility problem, we require an algorithm to solve this multigraph coloring problem. Construction 2 shows that this problem can be formulated as an instance of 2SAT, solvable in polynomial time [2], [5].

CONSTRUCTION 2. *Given a multigraph $G = (V, E)$ with red edges and black edges, we construct a conjunction of clauses as follows: For each vertex $v_i \in V$, we introduce a Boolean variable s_i. For each red edge (v_i, v_j), we include the clause $(\bar{s}_i \vee \bar{s}_j)$; for each black edge (v_k, v_l), we include the clause $(s_k \vee s_l)$.*

Intuitively, setting s_i to *true* means that the fault represented by s_i is covered by a row. Similarly, setting s_i to *false* means that the fault represented by s_i is covered by a column.

THEOREM 2. *The conjunction of clauses resulting from Construction 2 is satisfiable if and only if a feasible coloring exists for the multigraph G.*

Proof. Let C: $V \rightarrow \{red, black\}$ be a feasible coloring of the multigraph G. We assign values to the Boolean variables as follows: For each vertex v_i that is colored red, we assign variable s_i the value *true*. For each vertex v_j that is colored black, we assign variable s_j the value *false*. Each red edge (v_i, v_j) has at least one black endpoint, so the clause $(\bar{s}_i \vee \bar{s}_j)$ is *true*. Each black edge (v_k, v_l) has at least one red endpoint, so the clause $(s_k \vee s_l)$ is *true*. Therefore, all the clauses in the conjunction are *true*.

Next, let TA: $\{s_i\} \rightarrow \{true, false\}$ be a truth assignment satisfying the conjunction of clauses. We color the multigraph as follows: For each *true* variable, we color its corresponding vertex red. For each *false* variable, we color its corresponding vertex black. Note that an isolated vertex will not be represented in the conjunction of clauses. For completeness, we color each isolated vertex red. Each clause of the form $(\bar{s}_i \vee \bar{s}_j)$ is *true*, so the red edge (v_i, v_j) it represents must have at least one black endpoint. Each clause of the form $(s_k \vee s_l)$ is *true*, so the black edge (v_k, v_l) it represents must have at least one red endpoint. Therefore, the coloring is feasible. □

As an example, we give the 2SAT formulation for the set of arrays shown in Fig. 4. Using the numbering of the vertices in Fig. 5, the conjunction of clauses is: $(\bar{s}_1 \vee \bar{s}_3) \wedge (\bar{s}_1 \vee \bar{s}_4) \wedge (\bar{s}_2 \vee \bar{s}_3) \wedge (\bar{s}_2 \vee \bar{s}_4) \wedge (s_4 \vee s_5) \wedge (s_4 \vee s_6)$. An example of a satisfying truth assignment is constructed by setting s_3 and s_4 to be *true* and setting s_1, s_2, s_5, and s_6 to be *false*.

3. Disjoint minimum covers. The disjoint minimum cover problem seeks a feasible solution to the fault cover problem, with the stipulation that the individual cover of each array be minimum. Finding minimum covers is one way to reduce the cost of repairing the chip [3]. To show that the disjoint minimum cover problem can be solved in polynomial time, we must first provide some background results. A minimum cover of a (0, 1)-matrix is a minimum set of lines that contain all the 1's. The problem may be represented by a graph. For a given (0, 1)-matrix, we construct a bipartite graph G, which consists of two sets of vertices, X and Y, and a set of edges E. For each row r_i of the matrix there is a vertex $x_{r_i} \in X$. For each column c_j of the matrix there is a vertex $y_{c_j} \in Y$. There is an edge between vertices x_{r_i} and y_{c_j} if there is a 1 in position (r_i, c_j) in the matrix. This construction is illustrated in Fig. 6. A cover of G is a set of vertices $K \subseteq X \cup Y$ such that every $e \in E$ is adjacent to some vertex $k \in K$.

A *matching* in a graph is a subset of the edges such that no two edges in the matching have a common endpoint. A *maximum matching* is a matching of maximum cardinality. The bold edges in Fig. 6 constitute a maximum matching. Given a graph G and a matching in G, a vertex is said to be *matched* if it is adjacent to an edge in the matching; otherwise,

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & & 1 & 1 & & \\ 2 & 1 & & 1 & & 1 \\ 3 & & & & & 1 \\ 4 & 1 & & & 1 & \end{array}$$

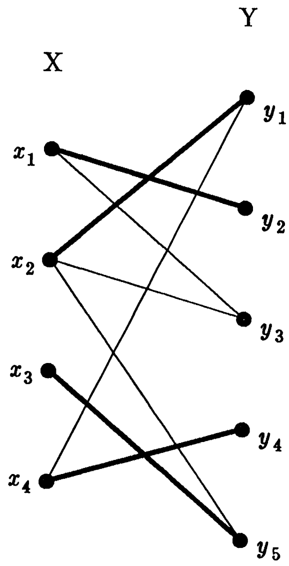


FIG. 6. *(0, 1)-matrix and corresponding bipartite graph.*

it is said to be *unmatched*. The König–Egerváry theorem [13] states that the size of a minimum cover in a (0, 1)-matrix is the same as the size of a maximum matching in the corresponding bipartite graph. Finding a maximum matching in a bipartite graph can be done in time $O(m\sqrt{n})$, where m is the number of edges and n is the number of vertices [10].

LEMMA 1. *Given a bipartite graph $G = (X \cup Y, E)$, a maximum matching M in G, and a minimum cover K of G, exactly one endpoint of each edge in M belongs to K.*

Proof. By definition, at least one endpoint of every edge must be included in K. Since edges in M have no endpoints in common, no vertex can cover two edges in M. So at least one endpoint of each edge in M must be in K. From the König–Egerváry theorem [13], we know that $|K| = |M|$. Therefore, at most one endpoint of each edge in M can be in K. □

LEMMA 2. *Given a bipartite graph $G = (X \cup Y, E)$ and a maximum matching M in G, an unmatched vertex does not belong to any minimum cover K of G.*

Proof. By Lemma 1, there must be at least $|M|$ matched vertices in the cover. Since $|K| = |M|$, the cover K can contain no other vertices. □

Using these two lemmas, we now show that the disjoint minimum cover problem, like the feasibility problem, can be reduced to 2SAT. The conjunction of clauses is formed using the following construction.

CONSTRUCTION 3. *Let R and C be the number of rows and columns, respectively, in each of t (0, 1)-matrices, $A_1, A_2, \cdots, A_t$. Let G_i be the bipartite graph corresponding to A_i. Let M_i be a maximum matching for A_i. For each row r_i, $1 \leqslant r_i \leqslant R$, we introduce t Boolean variables, $r_{i,1}, r_{i,2}, \cdots, r_{i,t}$. For each column c_i, $1 \leqslant c_i \leqslant C$, we introduce t Boolean variables, $c_{i,1}, c_{i,2}, \cdots, c_{i,t}$. The conjunction consists of four types of clauses*:

(1) *For each* 1, *we include the clause $(r_{i,k} \vee c_{j,k})$, where r_i, c_j, and A_k are the row, column, and array, respectively, that contain the* 1.

(2) *Next, for each row r_i that contains a* 1 *in one or more of the arrays, and for each unordered pair of matrices, A_k and A_l, we include the clause $(\bar{r}_{i,k} \vee \bar{r}_{i,l})$. For each column c_j that contains a* 1 *in one or more of the arrays, and for each unordered pair of*

matrices, A_k and A_l, we include the clause $(\overline{c}_{j,k} \vee \overline{c}_{j,l})$. Hence, for each line that contains a 1 in one or more of the arrays, $t(t-1)/2$ clauses are included in the conjunction.

(3) *For each 1 that is represented by an edge in M_k, we include the clause $(\bar{r}_{i,k} \vee \bar{c}_{j,k})$, where r_i and c_j are the row and column, respectively, that contain the 1.*

(4) *Finally, for each row r_i whose representative vertex in G_k is not matched, we include the clause $(\bar{r}_{i,k})$. For each column c_j whose representative vertex in G_k is not matched, we include the clause $(\bar{c}_{j,k})$.*

THEOREM 3. *The conjunction of clauses resulting from Construction 3 is satisfiable if and only if there exist disjoint minimum covers $K_1, K_2, \cdots, K_t$ for matrices $A_1, A_2, \cdots, A_t$.*

Proof. Assume that there exist disjoint minimum covers $K_1, K_2, \cdots, K_t$ for matrices $A_1, A_2, \cdots, A_t$. We assign truth values to variables as follows: For each spare row $r_i \in K_k$, we set $r_{i,k}$ to be *true*; for each spare column $c_j \in K_l$, we set $c_{j,l}$ to be *true*. Each 1 in A_k is covered by its row r_i or its column c_j, so its corresponding clause $(r_{i,k} \vee c_{i,k})$ must be *true*. Since each spare row r_i can be assigned to at most one cover, every clause $(\bar{r}_{i,k} \vee \bar{r}_{i,l})$ must be *true*. Since each spare column c_j can be assigned to at most one cover, every clause $(\bar{c}_{j,k} \vee \bar{c}_{j,l})$ must be *true*. By Lemma 1, we know that, for each edge in a matching M_k, exactly one of its endpoints must be included in a minimum cover of G_k. Therefore, every clause $(\bar{r}_{i,k} \vee \bar{c}_{j,k})$ must be *true*. Finally, by Lemma 2, an unmatched vertex in a bipartite graph G_k cannot be in a minimum cover of G_k, so all 1-clauses must be *true*. Hence, using the truth assignment above, the conjunction of clauses is *true*.

Conversely, assume the conjunction is satisfiable. Then there exists a truth assignment that forces every clause to be *true*. For each *true* variable $r_{i,k}$, include spare row r_i in cover K_k. For each *true* variable $c_{j,l}$, include spare column c_j in cover K_l. The clauses from step 1 imply that each 1 is covered. The clauses from step 2 imply that the covers are disjoint. The clauses from steps 3 and 4 imply that the covers are minimum. □

We omit the details of the conjunction for the example shown in Fig. 4. We note, however, that while there exists a solution to the feasibility problem for this example, there does not exist a set of disjoint minimum coverings for the three arrays shown. Such a set could involve no more than three spare lines, but the faults in the arrays cannot be covered with fewer than four.

4. Disjoint covers using multiple spare arrays. The multiple spare array problem is an extension of the feasibility problem discussed in § 2 to include the case in which the chip contains more than one set of spare rows, more than one set of spare columns, or both. Multiple sets of spares offer potential increases in chip yield because more defects can be successfully covered. Of course, the increase in yield must be balanced against the increase in fabrication and materials costs accompanying the use of additional spares.

The multiple spare array problem can be stated as follows: Given t (0, 1)-arrays of R rows and C columns each, SR $R \times C$ arrays of spare rows and SC $R \times C$ arrays of spare columns, the problem is to find an assignment of the spares to the arrays such that all the ones in the arrays are covered. An example of the multiple spare array problem, in which $SR = 2$ and $SC = 1$, is depicted in Fig. 7. Unfortunately, finding such disjoint covers is much more difficult than is the original problem, in which $SR = SC = 1$.

THEOREM 4. *The multiple spare array problem is* NP-*complete.*

Proof. The problem is in NP because we can guess an assignment of the spares to the arrays and check in polynomial time whether or not all the faulty elements are covered. Next, we want to use a reduction from a known NP-complete problem to the multiple spare array problem to show that the latter is NP-complete. Our reduction is

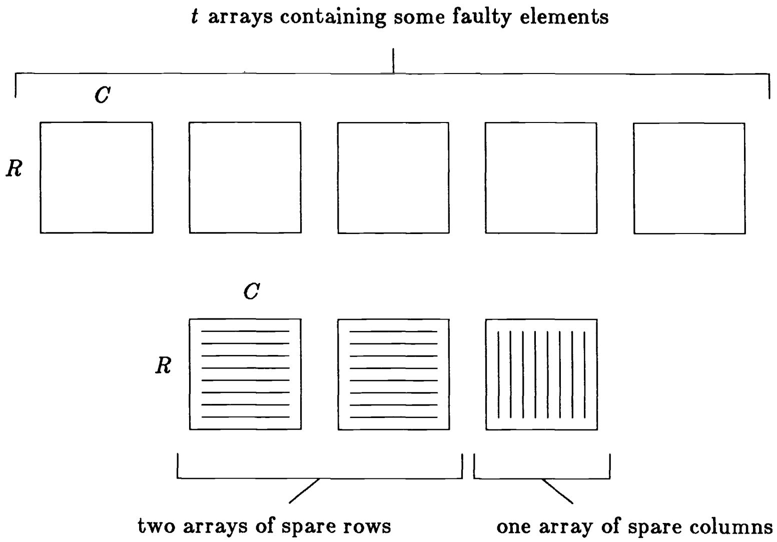

t arrays containing some faulty elements
C
R
C
R
two arrays of spare rows
one array of spare columns

FIG. 7. *Multiple spare array problem.*

from the *vertex cover* problem, which is as follows: Given a graph $G = (V, E)$, and an integer $K \leqq |V|$, does there exist a subset $V' \subseteq V$ with $|V'| \leqq K$ such that for each edge $(u, v) \in E$, at least one of u and v belongs to V'? The vertex cover problem is NP-complete [6].

Given an instance of the vertex cover problem, we want to construct an instance of the multiple spare array problem. Given $V = \{v_1, v_2, \cdots, v_n\}$, $E = \{e_1, e_2, \cdots, e_m\}$, and a positive integer $K \leqq n$, we construct n $1 \times m$ arrays, $A_1, A_2, \cdots, A_n$, K $1 \times m$ arrays of spare rows, and one $1 \times m$ array of spare columns. That is, $SR = K$ and $SC = 1$. An entry $(1, j)$ in array A_i is 1 if and only if vertex v_i is one of the endpoints of edge e_j. This means that the sum of the number of faulty elements in each column over all the arrays is exactly 2.

Now we want to show that there exists a solution to the instance of the vertex cover problem if and only if there exists a solution to the instance of the multiple spare array problem. If there is a solution to the instance of the vertex cover problem, then there is a subset $V' = \{v'_1, v'_2, \cdots, v'_l\}$ of V such that $l \leqq K$ and every edge in E has at least one endpoint in V'. For each vertex v'_i in V', we assign a spare row to the first (and only) row of array A_i. We use at most SR spare rows, because $SR = K$. Since every edge has at least one of its endpoints in V', the sum of the number of faulty elements that have not been covered by spare rows, in each column over all the arrays, is at most 1. This means that a spare column can be used to cover each of these 1's.

Suppose there is a solution to the instance of the multiple spare array problem. Let $A_{a_1}, A_{a_2}, \cdots, A_{a_l}$, where $l \leqq SR$, be the arrays to which spare rows are assigned. Let V' be $\{v_{a_1}, v_{a_2}, \cdots, v_{a_l}\}$. Since we have only one array of spare columns, this means that the sum of the number of faulty elements left uncovered by the spare rows in each column over all the arrays is at most 1. Recall that initially this number was 2. This means that the set V' contains at least one endpoint of each edge in E. □

Although chip yield may be increased with the use of multiple sets of spares, our NP-completeness result implies that heuristic algorithms are likely to be the only viable approach to the problem. The investigation of such heuristics is a potential area for future research.

5. Other fault cover applications. The results presented here have potential application in two other VLSI contexts. First, as the density of reconfigurable arrays continues to increase, with a corresponding increase in the number of elements in the arrays, repairing a chip will require a larger number of spare lines. Often, however, an individual row or column contains only a small number of faulty elements [17]. This implies that, for a reconfiguration method such as shown in Fig. 1, in which entire rows or columns are replaced, the spare elements will be used inefficiently because most of them will replace correctly functioning elements. The number of such wasted redundant elements increases as the size of the array increases and limits the number of chips that can be repaired.

One way to increase efficiency in the use of spares, and thus increase yield, is to replace the single large array with an array of smaller subarrays. The redundant elements are arranged such that rows and columns of individual subarrays may be replaced, independent of other subarrays, achieving the desired higher efficiency. Allocating spare lines for each subarray may be expensive. Alternatively, allowing a spare line to be used anywhere on the chip is not an attractive solution because the cost of wiring and the size of programmable decoders increases with the partitioning of the array. A compromise solution, used in [9], [16], is to limit the number of subarrays to which a particular spare line may be assigned. Figure 8 shows how our model may be used in this manner. The array has been partitioned into 16 subarrays. The spare elements have been arranged as one array of spare rows and one array of spare columns.

Another potential application of our model stems from recent interest in three-dimensional VLSI design [1]. Consider the situation depicted in Fig. 9, in which eight arrays are sandwiched between an array of spare rows and an array of spare columns. Arranging redundant elements in this manner, and requiring that a spare row be used to replace only one of the rows directly below it, and that a spare column be used to replace only a column directly above it, offers one way to reduce the circuit complexity in reconfigurable three-dimensional devices.

In both applications just described, the arrays may be homogeneous. Our model for heterogeneous arrays is applicable because it is assumed that the need to simplify wiring for reconfiguration imposes constraints on the use of spares.

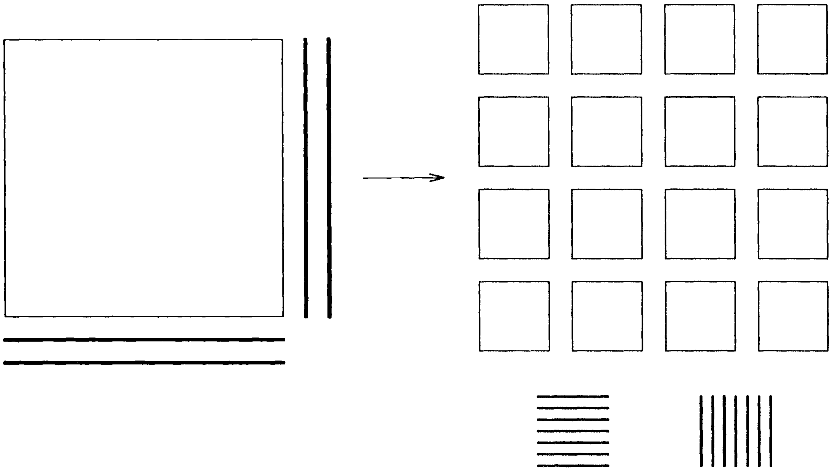


FIG. 8. *Reconfiguration of spares.*

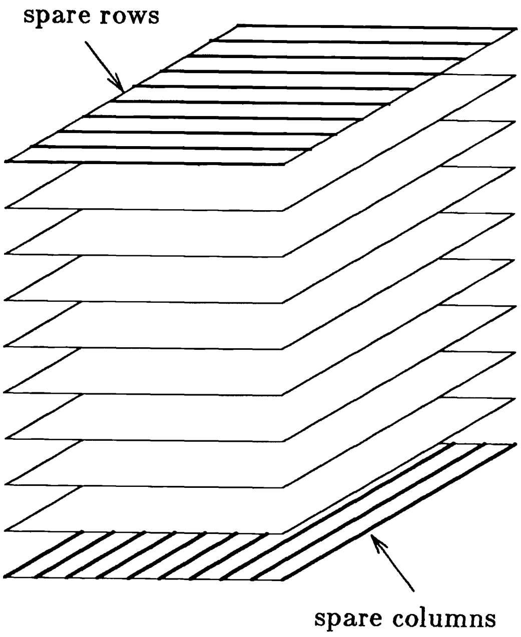


FIG. 9. *Three-dimensional arrays and spares.*

6. Summary. We have presented results for a set of fault cover problems in replicated, heterogeneous arrays of elements. First, a polynomial-time solution was given for the problem of finding a set of disjoint covers, if one exists, for the arrays using one set of spare rows and one set of spare columns. Second, a polynomial-time algorithm was given to find a feasible set of disjoint covers such that each is minimum. Finally, the problem of finding a feasible solution when multiple sets of spare lines are available was shown to be NP-complete. We briefly discussed two other potential applications of this work. We are currently studying extensions of the problems discussed here.

REFERENCES

[1] *Topical Meeting on Three Dimensional Integration*, Miyagi-Zao, Japan, May 30–June 1, 1988.

[2] M. DAVIS AND H. PUTNAM, *A computing procedure for quantification theory*, J. Assoc. Comput. Mach., 7 (1960), pp. 201–215.

[3] R. J. DAY, *A fault-driven comprehensive redundancy algorithm*, IEEE Design and Test, 2 (1985), pp. 35–44.

[4] R. C. EVANS, *Testing repairable* RAMS *and mostly good memories*, in Proc. IEEE Int. Test Conference, Philadelphia, PA, 1981, pp. 49–55.

[5] S. EVEN, A. ITAI, AND A. SHAMIR, *On the complexity of timetable and multicommodity flow problems*, SIAM J. Comput., 5 (1976), pp. 691–703.

[6] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of* NP-*Completeness*, W. H. Freeman, New York, 1979.

[7] R. W. HADDAD AND A. T. DAHBURA, *Increased throughput for the testing and repair of* RAMs *with redundancy*, in Proc. Intl. Conf. on Computer-Aided Design, Santa Clara, CA, 1987, pp. 230–233.

[8] N. HASAN AND C. L. LIU, *Minimum fault coverage in reconfigurable arrays*, in Proc. 18th Intl. Symp. on Fault-Tolerant Computing, Tokyo, Japan, June 27–30, 1988, pp. 348–353.

[9] Y. HAYASAKA, K. SHIMOTORI, AND K. OKADA, *Testing system for redundant memory*, in Proc. IEEE Int. Test Conference, Philadelphia, PA, 1982, pp. 240–244.

[10] J. HOPCROFT AND R. M. KARP, *An $n^{5/2}$ algorithm for maximum matching in bipartite graphs*, SIAM J. Comput., 2 (1973), pp. 225–231.

[11] I. KOREN AND D. K. PRADHAN, *Yield and performance enhancement through redundancy in* VLSI *and* WSI *multi-processor systems*, Proc. of IEEE, Vol. 74, No. 5 (1986), pp. 699–711.

[12] S. Y. KUO AND W. K. FUCHS, *Efficient spare allocation for reconfigurable arrays*, IEEE Design and Test, 4 (1987), pp. 24–31.

[13] D. KÖNIG, *Graphen und Matrizen*, Mat. Fiz. Lapok, 38 (1931), pp. 116–119.

[14] F. LOMBARDI, R. NEGRINI, M. G. SAMI, AND R. STEFANELLI, *Reconfiguration of* VLSI *arrays: A covering approach*, IEEE 17th Intl. Symp. on Fault-Tolerant Computing, Pittsburgh, PA, 1987, pp. 251–256.

[15] W. R. MOORE, *A review of fault-tolerant techniques for the enhancement of integrated chip yield*, Proc. of IEEE, Vol. 74, No. 5 (1986), pp. 684–697.

[16] Y. NISHIMURA, M. HAMADA, H. HIDAKA, H. OZAKI, AND K. FUJISHIMA, *A redundancy test-time reduction technique in* 1-Mbit DRAM *with a multibit test mode*, IEEE J. Solid-State Circuits, 24 (1989), pp. 43–49.

[17] S. E. SCHUSTER, *Multiple word/bit redundancy for semiconductor memories*, IEEE J. Solid-State Circuits, SC-13 (1978), pp. 698–703.

[18] M. TARR, D. BOUDREAU, AND R. MURPHY, *Defect analysis system speeds test and repair of redundant memories*, Electronics, January, 1984, pp. 175–179.

[19] C. L. WEY AND F. LOMBARDI, *On the repair of redundant* RAMS, IEEE Trans. Computer-Aided Design, Cad-6 (1987), pp. 222–231.

SIAM J. DISC. MATH.
Vol. 4, No. 2, pp. 293–303, May 1991

013

AN INTEGER PROGRAM FOR CODES *

MARTIN DOWD†

Abstract. For each value of the parameters A, n, d, a linear program exists whose integer solutions correspond to codes. The Plotkin bound gives a necessary and sufficient condition on n/d for feasibility. Some further simple remarks on the tableau of the linear program can be made; it can also be modified to consider only linear codes. For A divisible by 4, codes with optimum n/d with smallest possible value for n are Hadamard matrices. The question of a bound on n is an instance of the more general question of an upper bound on b for a block design with given v and k, which is conjectured to be polynomial in v. Some constructions related to these questions are given.

Key words. Plotkin bound, short block designs, Hadamard rectangles

AMS(MOS) subject classifications. 94B25, 05B05

1. The integer program.

A code may be considered as the rows of a 0-1 matrix, and conversely if the rows are distinct. Say that the minimum distance of the matrix is that of the code. Let $\mathcal{A} = \{1 \cdots A\}$.

THEOREM 1. *Up to column permutation the $A \times n$ matrices of minimum distance greater than or equal to d correspond to the integral solutions to the following system of inequalities:*

$$x_S \geq 0, \quad \textit{all } S \subseteq \mathcal{A} \tag{1}$$

$$\sum_{\substack{S \subseteq \mathcal{A}, \\ |S \cap T| = 1}} x_S \geq d, \quad \textit{all } T \subseteq \mathcal{A}, |T| = 2 \tag{2}$$

$$\sum_{S \subseteq \mathcal{A}} x_S = n. \tag{3}$$

Proof. Given a matrix, assign to x_S the number of columns which are 1 in exactly the rows of S. Conversely, a nonnegative integer solution determines how often each column appears.

THEOREM 2. *The system of Theorem* 1 *is feasible if and only if $n/d \geq 2(A-1)/A$ if A is even, and $n/d \geq 2A/(A+1)$ if A is odd.*

Proof. Divide the x_S by d and sum (2) over T, yielding

$$\sum_{S \subseteq \mathcal{A}} |S||\mathcal{A} - S| \frac{x_S}{d} \geq \binom{A}{2}.$$

Since $|S||\mathcal{A} - S|$ is at most $A^2/4$ if A is even, and $(A+1)(A-1)/4$ if A is odd, one direction follows. For the other, distribute the weight n evenly between the variables x_S with $|S| = A/2$ for A even, or $|S| = (A-1)/2$ for A odd.

The bound is the Plotkin bound, with A fixed rather than n and d. For some n and d with n/d meeting the bound integer solutions exist. The complete block design with $v = 2k + 1$ (where these are the block design parameters in the usual notation) yields an example, if a row of 0's is added.

*Received by the editors June 26, 1989; accepted for publication (in revised form) July 10, 1990.

†2532 Orange Avenue, Costa Mesa, California 92627.

In general, however, the Plotkin bound yields far too low a restriction on n for existence of an integer solution, for given A and d. For example, fix d and apply the sphere packing bound.

It is interesting to note that a "row" of an assignment can be defined by defining the matrix X with $X_{iS} = x_S$ if point i is in set S, or 0 otherwise. The distance between two rows is given by the Manhattan norm $\sum_S |X_{iS} - X_{jS}|$. However, the theorem shows that a packing argument in the $(2^k - 1)$-dimensional space over the reals will yield no improvement to the Plotkin bound.

2. Some properties of the tableau. It is clearly unlikely that any useful theorems concerning the tableau will be easy to derive. Some simple observations can be made, however. The system of equations (2) can be slightly simplified, by normalizing the solutions so that the last row is all 0's, i.e., by setting $x_S = 0$ if $A \in S$. With $v = A - 1$, let $V = \{1 \cdots v\}$, and let C^v, or simply C, denote the matrix whose rows are the coefficients of the resulting equations. If the rows of C are labelled with the sets $T \subseteq V, 1 \leq |T| \leq 2$, and the columns with the sets $S \subseteq V$, then $C_{TS} = 1$ if $|T \cap S|$ is odd (i.e., 1), and 0 otherwise. All solutions to the original system can be obtained by choosing a solution to the new system, and for each $S \subseteq V$ splitting x_S between S and $S \cup \{A\}$. More generally we may consider the family $C^{t,v}$ for $1 \leq t \leq v$, by considering T with $|T| \leq t$.

THEOREM 3. *The matrix $C^{t,v}$ has full rank.*

Proof. A basis for the solutions of $C^t x = 0$ may be obtained as follows. Let Q be the matrix whose rows and columns are labeled with the subsets of V, and where $Q_{TS} = (-1)^{|S \cap T|}$. It is well known that $Q^T Q = (1/2^{v-1})I$. Appending a row of -1's to C^t, adding twice this to every other row, and multiplying by -1 yields the rows of Q with $|T| \leq t$, so that transposing the remaining rows of Q yields all but one of the linearly independent solutions to the homogeneous system. The remaining solution is given by $x_S = 1$ if $S = \emptyset$, else $x_S = 0$.

Theorem 2 can be generalized to arbitrary t; let

$$\eta(u,s) = \sum_{r \text{ odd}} \binom{s}{r} \binom{v-s}{u-r}$$

be the number of u element sets T such that $|S \cap T|$ is odd for $|S| = s$. Then

$$\sum_{u=1}^{t} x_S \eta(u, |S|) \geq \sum_{u=1}^{t} \binom{a}{u}.$$

For $t = 1$, the bound $\sum x_S \geq 1$ is met by the solution $x_S = 1$ if $S = V$ and $x_S = 0$ otherwise. Furthermore, any set of v linearly independent columns of C that includes the column labeled V is a basis yielding this solution, since $B^{-1}j = e$ for some unit vector e, where j is the all 1's vector, if and only if some column of B is j. For $t = 3$ differentiating yields $|S| = (v^2 + 2)/2v$, which is nonintegral.

Let F^v denote the v-dimensional vector space over the two element field F; by identifying the members of V with a basis of F^v, each $S \subseteq V$ may be identified with a vector of F^v. The linear group on F^v then acts on the vectors $\langle x_S \rangle$ by permuting coordinates. For each $T \subseteq V$ let U_T be the $(v-1)$-dimensional subspace $\{S : |S \cap T| \text{ is even}\}$. We claim that an element of the linear group acts on the solutions to $C^{t,v}x = 0$ if and only if it preserves the U_T with $|T| \leq t$. If it preserves these subspaces, it maps the equations among themselves; and if it does not it maps some subspace corresponding to a solution to one corresponding to an equation. Let GS^v, or simply GS, denote this group, for $t = 2$. The following theorem gives a characterization of it; it can also be characterized as a group of operations on Hadamard rectangles [Do87].

THEOREM 4. *GS^v is the group generated by the permutation matrices and the matrix Δ, defined by $\Delta_{rc} = 1$ if $r = c$ or $c = v$ and 0 otherwise. It is isomorphic to the symmetric group on A points.*

Proof. Temporarily call the group generated by the matrices GS_1; it is easily verified that GS_1 acts as the symmetric group on the sets $\{i\}$ together with V (Δ acts by transposing V and $\{v\}$). Furthermore, considering these as the vertices of the complete graph on A vertices, the action on the U_S is the same as the action on the edges, where U_{ij} corresponds to the $\{i\}\{j\}$ edge; and U_i to the $\{i\}V$ edge. The first claim is by induction on v; for the basis, $v = 2$ and the group is the entire linear group. Using the induction hypothesis it is readily verified that a matrix M in GS may be brought by members of GS_1 to a form where $M_{rc} \neq 0$ only if $r = c$, $c = v - 1$ and $r \leq v - 1$, or $c = v$. Considering the action on U_{1v} shows that $M_{r,v-1} = 0$ for $r < v - 1$; and on U_{ij} for $1 \leq i \leq j < v$ that the M_{rv}, $r < v$, are equal.

THEOREM 5. *The minimum size circuits of C^v are isomorphic to C^3 (ignoring the 0 column in both).*

Proof. We may assume three of the columns are those labeled $\{1\}, \{2\}, \{3\}$. Inductively, the ith column, $4 \leq i < 7$, if it is a linear combination of the previous columns, would be disjoint from $\{4, \cdots, v\}$. Hence in rows $\{1\}, \{2\}, \{3\}$ it is distinct from the previous columns; hence linearly independent of them; and hence without loss of generality, disjoint from $\{4, \cdots, v\}$.

3. Remarks on applying pivots. The possibility exists that the tableau of the linear program may have some interesting properties involving sequences of pivots applied to it. A basic feasible solution (see [PS82] for terminology) has only $O(v^2)$ types of columns, a fact whose relevance will become apparent in §5. This leads us to ask whether anything further can be said about optimum basic feasible solutions, in particular if they are reached via the simplex method. For example, consider the "augmented" tableau $C|I$, with inhomogeneous column all 1's, say; we can ask which optimum basic feasible solutions can be reached, without repeating a pivot row. Call a sequence of pivots (where a pivot is a pair (r, c) consisting of a row number and column number) simple if no row is repeated.

A second question is whether the pivots of a simple sequence can be applied in any order. A characterization of when this is so can be given. Let T be a tableau, and $\sigma = (r_1, c_1), \cdots, (r_k, c_k)$ a simple sequence of pivot positions. Define the matrix M_σ to have (i, j) entry T_{r_i, c_j}. Define a simple sequence σ to be positive if, when the pivots are applied to T in this order, the pivot entry is always positive; and legal if every pivot is legal, that is, preserves feasibility. We will see that if σ is a simple sequence of pivots then σ may be arbitrarily reordered if and only if M_σ is a P-matrix, i.e., a matrix whose principle submatrices have positive determinant (see [BP79] for P-matrices). Computer experiments show that in the augmented tableau of the linear program for codes, such M_σ do not necessarily exist.

Let σ be a simple sequence; r some row not involved in σ; and c the inhomogeneous vector of the tableau T. Given a $k \times k$ principle submatrix M of M_σ define the matrices

M^{+r} and $M^{/r}$ as follows:

$$M_{ij}^{+r} = \begin{cases} M_{ij} & \text{if } i,j \leq k \\ T_{r_i,c} & \text{if } i \leq k, j = k+1 \\ T_{r,c_j} & \text{if } i = k+1, j \leq k \\ T_{r,c} & \text{if } i = j = k+1 \end{cases}$$

$$M_{ij}^{/r} = \begin{cases} M_{ij} & \text{if } i \neq r \\ T_{r_i,c} & \text{if } i = k. \end{cases}$$

THEOREM 6. *A simple sequence σ is positive if and only if every leading principle submatrix of M_σ has positive determinant. It is legal if and only if in addition, for each leading principle submatrix M of M_σ, whenever $M^{/r}$ has positive determinant M^{+r} has nonnegative determinant.*

Proof. The proof is by induction on the length k of σ. The basis for the first claim simply states that the pivot entry is positive. In one direction, it suffices to show that after applying the (r_1, c_1) pivot, the leading principle submatrices of the matrix of the remaining sequence in the new tableau are positive. Let α denote T_{r_1,c_1}; the determinant of each leading principle submatrix of M_σ is multiplied by $1/\alpha$ by applying the pivot. Furthermore, the submatrix obtained by then deleting the first row and column has the same determinant. For the converse, by induction if $\det(M_\sigma) \leq 0$, but $\det(M_\tau) > 0$ for all proper initial subsequences τ of σ, then the last pivot entry is ≤ 0. The second claim follows by a similar argument; the basis states that the pivot is legal.

With the aid of a computer, we have determined the following. For C^3, every six of the seven nonzero columns forms a basis. There are two basic feasible solutions. One has $x_S = \frac{1}{2}$ if $|S|$ is 1 or 3; it is of cost 2 and its bases are those omitting an S with $|S| = 2$. The other has $x_S = \frac{1}{2}$ if $|S|$ is 2, and is of cost $\frac{3}{2}$. The two solutions are stabilized by GS, and each set of bases is acted on transitively. Fixing a basis for the second solution, there are 24 sets of pivot points of legal simple sequences leading to the basis from the augmented tableau. The number of orderings that are legal ranges from 1 to 84.

For C^4, the bases may be classified according to the distribution of values. There are 10 types, namely,

$(-1/2)^4 1^1 (1/2)^5$, $(-1/2)^3 1^1 (1/2)^6$, $(-1/2)^4 (1/2)^6$, $0^5(-1/2)(1/2)^4$,
$0^4(-1/2)(1/2)^5$, $0^5(1/2)^5$, $0^2(1/4)^8$, $0^3(1/4)^7$, $(1/3)(1/6)^9$, and $(1/6)^{10}$.

The number of solutions of each type is, respectively,

30, 20, 5, 5, 10, 1, 5, 5, 10, 1;

and the number of bases

30, 20, 5, 810, 810, 162, 60, 160, 10, 1.

We have not determined which can be reached by simple sequences.

4. Linear codes. The linear program can be modified so that only linear codes are considered. Suppose $A = 2^k$, let F^k denote the k-dimensional vector space over the two element field F, and let $\mathcal{C}$ denote the family of set theoretic complements of the $(k-1)$-dimensional subspaces of F^k.

LEMMA 7. *An incidence matrix is the matrix of a linear code if and only if its rows can be labeled with the points of F^k so that the nonempty S for which $x_S > 0$ are the sets in $\mathcal{C}$.*

Proof. If the code is linear, choose k rows forming a basis and label them with a basis of F^k. This induces a labeling of the remaining rows in an obvious way. Given a column, the rows in which it is 0 are those sums of basis rows which involve an even number of basis rows which are 1 in the column; these form a subspace. Conversely, it suffices to consider the case $x_S = 1$ for all complements S of $(k-1)$-dimensional subspaces, and $x_S = 0$ for all other S. But it is easily seen that this is the dual Hamming code.

THEOREM 8. *If in the equations of Theorem* 1 *S is restricted to range over C, and T over pairs containing* 0, *the nonnegative integral solutions correspond to linear codes of minimum distance $\geq d$.*

Proof. The proof is immediate from the lemma, and the fact that only the weight need be bounded below for a linear code.

For any k there is a well-known Hadamard matrix corresponding to a linear code.

5. Codes with optimum n/d.

LEMMA 9. *If a code has constant distance d then d is even, if $A \geq 3$.*

Proof. Normalize the code to have a row of 0's; any nonzero row then has weight d. Given two distinct rows, suppose there are λ columns with 1's in both rows. Then the distance between the rows is $2(d-\lambda)$, so $d = 2\lambda$.

It is easily seen from the proof of the lemma that a code with constant distance (also called a simplex code) is essentially the same (up to a row of 0's when normalized) as a pairwise balanced design with constant row weight 2λ.

THEOREM 10. *A code achieving the bound on n/d has constant distance. For A even, such codes correspond to block designs with $v = A-1$ and $b/r = 2v/(v+1)$.*

Proof. By the proof of Theorem 2, in a code with minimum n/d the weight of any column is $A/2$ if A is even, or $(A-1)/2$ or $(A+1)/2$ if A is odd. To show that the code has constant distance, count pairs where one row is 0 and the other 1. The total is $nA^2/4$ for A even, and $n(A-1)(A+1)/4$ for A odd; if any distance is greater than d, summing over the pairs of rows will yield a larger value. The last claim follows either by noting that a PBD with constant column weight is a block design, or noting directly that the rows of the design have constant weight.

We will call block designs with v odd and $b/r = 2v/(v+1)$ H-designs. For v odd, $b/r = 2v/(v+1)$ is equivalent to $r = 2\lambda$, or $k = (v+1)/2$. For such designs, b is a multiple of v; indeed $b/v = 4\lambda/(v+1) = 2r - b$. Note that the code itself is never a block design, since for v even $k = v/2$ is equivalent to $b/r = 2$ rather than $b/r = 2(v-1)/v$.

The complete design, with all $(v+1)/2$ element subsets of the treatments as blocks, is an example of an H-design. The resulting code has length $\binom{A-1}{A/2}$. The smallest possible length (corresponding to the highest possible rate, since A is fixed), is easily seen to be $A-1$ if $A \equiv 0 \pmod 4$, (corresponding to a Hadamard matrix), or $2(A-1)$ if $A \equiv 2 \pmod 4$.

If in a code of minimum n/d we replace 0 by $+1$ and 1 by -1, and add $2d-n$ columns of $+1$'s, the rows of the resulting configuration are orthogonal. Such a configuration has been called a Hadamard rectangle, or a rectangular Hadamard matrix. Hadamard rectangles are very easy to construct; simply delete rows from a Hadamard matrix. To obtain H-designs, the number of columns must be divisible by the number of rows, and there must be sufficiently many columns which are all $+1$. Hadamard rectangles with v rows satisfying the first requirement exist, of length $O(v^3)$; this follows by Theorem 7.13 of [GS79].

If Hadamard matrices exist for all $A \equiv 0 \pmod 4$, then shortest possible codes achieving the bound on n/d exist for all A. For $A \equiv 2 \pmod 4$ the required Hadamard rectangle is obtained from a Hadamard matrix of order $2A$ by normalizing so that there is a column of +1's, and taking the rows with +1 in a second column. For A odd, delete a row from a code for $A+1$; indeed, the minimum n for A rows is at most that for $A+1$, but there may be A row codes not obtained by deleting a row from an $A+1$ row code, even of minimum length when the minimum lengths are equal.

If only codes with optimum n/d are considered, the linear program can be modified by making (2) an equality constraint. If also only S with $|S| = (v+1)/2$ are considered, the result is essentially a special case of VIII.(7.2.a) of [BJL]; Lemma VIII.7.3 of [BJL], due to Wilson, states that these have an integer solution if and only if λ is a multiple of λ_0 (see below for λ_0). A homogeneous system can be obtained by setting all sums other than the first equal to the first, rather than λ. It would be of interest to obtain a module basis for the integer solutions; one example of such, without the size restriction on S, is given in [Do87].

6. A more general question. The existence of Hadamard matrices is subsumed by the more general question of the smallest possible value of b when $v \equiv 3 \pmod 4$ and $k = (v+1)/2$. In general, for $2 \le k \le v$ let

$$d = \gcd(v(v-1), k(k-1), v(k-1)),$$
$$\lambda_0(v,k) = k(k-1)/d, \quad b_0 = v(v-1)/d;$$

it is easy to see that for any design with given v and k, $\lambda_0|\lambda$ and $b_0|b$.

Define $b_1(v,k)$ to be the smallest b for which a design actually exists, and similarly for λ_1. We conjecture that

1. b_1 is bounded by a polynomial in v.

Successively stronger versions of this conjecture can be made.

2. $b_1 = O(v^2)$, or equivalently $\lambda_1 = O(k^2)$.
3. $\lambda_1 \le K\lambda_0$ for some real number K.

It seems quite likely that version 2 is true, and there is no evidence that version 3 is not. Another question of interest is for what v, k does there exist a block design with $\lambda = k(k-1)$; for these v, k, version 2 holds.

If v is restricted to be a prime power q then there is a design with $\lambda = k(k-1)$, by the double transitivity of the affine group $AGL(1,q)$ on $GF(q)$. Whether version 3 holds when v is a prime power is question that should be investigated. If $v = q+1$ version 1 follows by the double transitivity of $PSL(2,q)$ on $PG(1,q)$; the bound on b_1 is $O(v^3)$. Higher-dimensional geometries give a polynomial bound for further v, but the degree increases with the dimension.

If k is fixed, λ_0 is periodic in v, with period $k(k-1)$ (and is at most $k(k-1)$). By Wilson's theorem, then, for sufficiently large v $b_1(v,k) = b_0(v,k)$. Thus, as v increases, there is an increasing bound k_v such that $b_1(v,k) = b_0(v,k)$ (and $b_1(v, v-k) = b_0(v, v-k)$) for $k \le k_v$.

It is also true ([BJL, 8.7.1]) that for fixed v and k, if λ is sufficiently large and $\lambda_0|\lambda$ then there is a block design with the given parameters. We may thus define $b_2(v,k)$ as the least b such that there exists a block design with $b + tb_0$ blocks for all $t \ge 0$, and similarly for λ_2. We conjecture that b_2 is $O(v^2)$. Note that if $b_1 = b_0$ then $b_2 = b_0$.

Let $B(K,\lambda)$ denote the values of v for which there exists a pairwise balanced design, with each pair occurring λ times and block sizes in K; if $\lambda = 1$ it may be omitted, and k written for $\{k\}$. Let $Q^{\ge k}$ denote the prime powers $q \ge k$. It is easily seen that $B(Q^{\ge k}) \subseteq B(k, k(k-1))$, since $B(k, k(k-1))$ is closed and contains $Q^{\ge k}$. It follows from the following that there is a design with $\lambda(v,k) = k(k-1)$ if $v \ge k^c$

where c is an appropriate constant. For fixed k c may be chosen smaller than the general value; for example for $k \leq 7$ c may be 1 [Ha75].

LEMMA 11. *There is a v_0 such that if $v \geq v_0$ and $v \geq k^{43}$ then $v \in B(Q^{\geq k})$.*

Proof. Let $N^{q_i \geq k}$ denote $\{q_1 \cdots q_r : q_i \in Q^{\geq k}\}$; $N^{q_i \geq k} \subseteq B(Q^{\geq k}) \cap TD(k)$, by Macneish's theorem and induction, where $TD(k)$ denotes those g for which there exists a transversal design with k groups and group size g. By [BJL, IX.2.1 and IX.2.6], it suffices to show that, if $v \geq v_0$ and $v \geq k^{43}$, there are nonnegative integers x, y, u such that

$$v = xy + u;\ y \in N^{q_i \geq k+1}; x, x+1 \in N^{q_i \geq b+1};\ y, u \in N^{q_i \geq b};\ u \leq y.$$

This follows by obvious modifications to the argument in [Ro64].

7. A group construction.

Group methods can be used to construct some short block designs. Let G be a group acting on a v element set S. Let O_i, $1 \leq i \leq \omega$, be the orbits of pairs, and let B_j, $1 \leq j \leq \beta$, be base blocks, i.e., k element subsets of S. For $B \subseteq S$ let B^G denote the multiset of images. More usually B^G denotes the set, but we will use $\{B^G\}$ for this. Also, let $h_{ij} = |O_i \cap B_j^{(2)}|$, where $B^{(2)}$ denotes the collection of 2-element subsets of B.

If multisets are used, the length b of the incidence matrix resulting from the action of G on the B_j equals $|G|\beta$, and if the matrix is pairwise balanced $\lambda = bk(k-1)/v(v-1)$. Thus, the matrix is a block design, provided

$$\sum_{j=1}^{\beta} h_{ij} = \frac{\lambda |O_i|}{|G|} = \frac{k(k-1)}{v(v-1)} |O_i| \beta \tag{4}$$

for each i. In particular the right sides must be integers. If the $\{B_j^G\}$ are used, the jth term of the sum must be divided by the order $|G_{B_j}|$ of the setwise stabilizer, and λ cannot be so easily determined. The set version is completely general, provided multiple copies of an orbit are allowed.

In the multiset version, if G yields a design, and H is a group containing G and acting on S, then H yields a design. If H is a permutation subgroup of G and H has the same orbits of pairs as G, then again H yields a design.

LEMMA 12. *If conditions* (4) *are met for all i but one, they are met for all i.*

Proof. This follows since

$$\sum_{i,j} \frac{h_{ij}}{k(k-1)} = \frac{\beta}{2} = \sum_i \frac{|O_i|}{v(v-1)} \beta.$$

Many constructions are known for G acting regularly; the family of base blocks is then called a difference family. It may be verified that for a difference family to exist, λ must be divisible by $\lambda_d = k(k-1)/d_d$ where $d_d = \gcd(k(k-1), v-1)$, and that $\lambda_d = \lambda_0 \gcd(v,k)$. The cyclic groop Z_v is one useful groop; another is $GF(q_1) \times \cdots \times GF(q_s)$ where $v = q_1 \cdots q_s$ and the q_i are powers of distinct primes.

If G_0 acts regularly on itself and A is a group of automorphisms of G_0, $G = A \times G_0$ acts on G_0 via $x \mapsto \alpha(x)t$ for $\alpha \in A$, $t \in G_0$. For example, if $G_0 = Z_v$ then the automorphisms are multiplication by elements of Units(Z_v), the units of the ring Z_v. The order of $A \times G_0$ when A is all the automorphisms is $v\phi(v)$ where ϕ is the Euler function. There is an orbit of pairs O_d for each proper divisor d of v, namely,

$$\{\{x,y\} : x - y \equiv d \,(\operatorname{mod} v)\} = \{\{x,y\} : x - y \in d^G\},$$

where

$$d^G = \{x \in Z_v : \gcd(x,v) = 1\} = d\,\mathrm{Units}(Z_{v/d}).$$

In particular, $|O_d| = (1/2)v\phi(v/d)$.

Although it turned out to be irrelevant, we noticed the following property of the d^G, which does not seem to be well known. d^G is closed under multiplication if and only if $\gcd(d,e) = 1$ where $e = v/d$; this follows because d^G is those x for which $\max(\mathrm{ord}_p(x), \mathrm{ord}_p(v))$ has a fixed value, for each p. Further, in this case d^G is a group; the identity is $d^{\phi(e)}$, since $d^{\phi(e)} \equiv 1 \pmod e$, so $d^{\phi(e)}d \equiv d \pmod{de}$. It is readily verified that the map $u \mapsto d^{\phi(e)}u$ from $\mathrm{Units}(Z_e)$ to d^G is an isomorphism.

If $v = 3q$ where q is a power of a prime other than 2,3, let $G_0 = GF(3) \times GF(q)$, and let A be multiplication by units, of which there are $2(q-1)$. The orbits of differences are as for q prime, and may be called O_1, O_3, and O_q; the sizes are $v/2$ times $2(q-1)$, $q-1$, and 2. If $\beta = 1$, for $3q-1$ to divide $k(k-1)(q-1)$ k must equal $(3q\pm1)/2$; we consider $k = (3q+1)/2$. Let a_i, $i = 0,1,2$, denote $|\{x \in B : x \equiv i \pmod 3\}|$; note that $h_3 = \sum_i a_i(a_i-1)/2$.

LEMMA 13. *Every nonnegative integer can be written in the form*

$$n = w_0^2 + 3w_1^2 + w_2^2 + 3w_3^2. \tag{5}$$

Proof. A proof is given in §9.

COROLLARY 14. *Every nonnegative integer can be written in the form*

$$n = 3x_1^2 + 2x_1 + y_1^2 + 3x_2^2 + 2x_2 + y_2^2.$$

Proof. By Lemma 13, $3n+2 = w_1^2+3y_1^2+w_2^2+3y_2^2$. This is impossible if either w_1 or w_2 is divisible by 3, so we may assume $w_1, w_2 \equiv 1 \pmod 3$. Letting $x_i = (w_i-1)/3$ yields the corollary.

THEOREM 15. *Suppose q is a power of a prime other than 2 or 3. Then*

$$b_1(3q, (3q+1)/2) \le 12q(q-1).$$

If $q = 3x^2 + 2x + y^2$ for some integers x, y then

$$b_1(3q, (3q+1)/2) \le 6q(q-1).$$

Proof. We apply the group $A \times G_0$ where $G_0 = GF(3) \times GF(q)$, A is multiplication by units, and $\beta = 2$. We have

$$2q = 3x_1^2 + 2x_1 + y_1^2 + 3x_2^2 + 2x_2 + y_2^2;$$

since $2q \equiv 2 \pmod 4$, exchanging y_1 and y_2 if necessary we may assume

$$x_j \not\equiv y_j \pmod 2, \quad j = 1,2.$$

Now, $3x^2 + 2x + y^2 \ge 0$ for integer x, y, so $3x_j^2 + 2x_j + y_j^2 \le 2q$; it follows that

$$x_j \le \frac{\sqrt{6q+1}-1}{3},\; x_j \pm y_j \ge -\frac{2\sqrt{6q+1}+1}{3}.$$

Suppose $q \equiv 1 \pmod 4$; then the quantities

$$a_{j01} = \frac{q+2x_j+2y_j+1}{4},\ a_{j02} = \frac{q+2x_j-2y_j+1}{4},\ a_{j12} = \frac{q-4x_j-1}{4}$$

are integers. From the above inequalities they are nonnegative if $q \geq 11$. Since $a_{j01}+a_{j02}+a_{j12} \leq q$ we may choose base blocks B_j, containing a_{jtu} pairs of the form $\{(t,v),(u,v)\}$. Let

$$a_{j0} = a_{j01}+a_{j02},\ a_{j1} = a_{j01}+a_{j12},\ a_{j2} = a_{j02}+a_{j12}.$$

Then $\sum_j a_{jt} = (3q+1)/4$ and $\sum_j a_{jt}^2 = (3q^2+4q+1)/2$. The latter implies the requirement on $\sum_j h_{3j}$, and the requirement on $\sum_j h_{qj}$ is clearly satisfied. The theorem in this case now follows by Lemma 12. If $q = 3x^2+2x+y^2$ then $q \equiv 1 \pmod 4$; letting $\beta = 1$ and suppressing j, the above argument goes through, for all q. Note that if $q \equiv 1 \pmod 4$ and $q < 11$ then $q = 5 = 3 \cdot 1^2 + 2 \cdot 1 + 0^2$. If $q \equiv 3 \pmod 4$ let

$$a_{101} = \frac{q+2x_1+2y_1-1}{4},\ a_{102} = \frac{q+2x_1-2y_1-1}{4},\ a_{112} = \frac{q-4x_1-3}{4},$$
$$a_{201} = \frac{q+2x_2+2y_2-5}{4},\ a_{202} = \frac{q+2x_2-2y_2-5}{4},\ a_{212} = \frac{q-4x_2+5}{4},$$
$$a_{10} = a_{101}+a_{102}+1,\ a_{11} = a_{101}+a_{112}+1,\ a_{12} = a_{102}+a_{112}+1,$$
$$a_{20} = a_{201}+a_{202}+3,\ a_{21} = a_{201}+a_{212},\ a_{22} = a_{202}+a_{212}.$$

The a_{jtu} are integers, and are nonnegative if $q \geq 21$. The remaining cases are as follows:

$$q = 7:\ x_1 = -2, y_1 = 1, x_2 = 1, y_2 = 0;$$
$$q = 11:\ x_1 = -3, y_1 = 0, x_2 = 0, y_2 = 1;$$
$$q = 19:\ x_1 = 3, y_1 = 0, x_2 = 1, y_2 = 0.$$

Remarks on the primes $q = 3x^2+2x+y^2$ may be found in §8. Note that $y = 0$ if and only if $q = 5$. If $y \neq 0$ one can verify that six distinct representations may be obtained by permuting the a_i (where $x = q - a_1 - a_2$, $y = a_1 - a_2$, $a_0 + a_1 + a_2 = (3q+1)/2$); this is so for q a prime power. There may be several sextuples of representations. More can doubtless be said about the number of representations.

8. Remarks on primes of the form $3x^2+2x+y^2$. It is undoubtedly the case that the asymptotic density of the primes of the form $3x^2+2x+y^2$ can be determined by straightforward methods, along the lines of those of [Pa73] and [Wi75]. Indeed, this can undoubtedly be done for any two variable polynomial.

It is well known that an odd prime p is representable as w^2+3u^2 if and only if $p \equiv 1 \bmod 6$; the representation is essentially unique. For an arbitrary integer a, writing $a = d^2e$ where e is square free, a is representable if and only if e is, and as is well known this is the case if and only if it is a product of primes congruent to 1 mod 6. This gives another characterization of the special q of Theorem 15, namely, those where $3q+1$ is of this form.

It follows from the test of Mann and Yamamoto that if there is a (v,k,λ) difference set and $3|v$ then any prime divisor of the square free part of $k-\lambda$ must be congruent to 0 or 1 mod 3 ([Ma65, Cor. 7.2.4]; [BJL, VI.5.6.a]). The restriction above, therefore, yields no new information for difference sets. Also, difference sets do not always exist; for example, there is no difference set when $q = 17$ ([BJL, table D]).

9. Proof of Lemma 13. To prove Lemma 13, we consider quaternions of the form

$$a = a_0 + a_1\sqrt{3}i + a_2 j + a_3\sqrt{3}k,$$

where the a_i are integers. We verify that these form a ring, and that $N(a) = a_0^2 + 3a_1^2 + a_2^2 + 3a_3^2$. Thus, the integers of the form (5) are closed under multiplication. They certainly contain 0 and 1. If n is of the form (5) and is even, and if a_0 and a_2 have the same parity mod 2, so do a_1 and a_3, and

$$\frac{n}{2} = \left(\frac{a_0 + a_2}{2}\right)^2 + 3\left(\frac{a_1 + a_3}{2}\right)^2 + \left(\frac{a_0 - a_2}{2}\right)^2 + 3\left(\frac{a_1 - a_3}{2}\right)^2.$$

Otherwise, by permuting the variables if necessary, we may assume that a_0 and a_1, and a_2 and a_3, have the same parity. If, say, a_0 and a_1 are odd, one of $a_0 \pm a_1$ is divisible by 4, and

$$\left(\frac{a_0 \mp 3a_1}{2}\right)^2 + 3\left(\frac{a_0 \pm a_1}{2}\right)^2 = a_0^2 + 3a_1^2,$$

so we may assume all the a_i are even and again $n/2$ is of the form (5).

It remains to prove the lemma for an odd prime p. For this we introduce the ring H of quaternions of the form

$$a_0 + a_1\zeta + a_2 j + a_3\zeta j,$$

where the a_i are integers and $\zeta = (-1 + \sqrt{3}i)/2$. This is clearly a ring, by the usual properties of ζ and $j\zeta = (-1 - \zeta)j$. Also, for $a \in H$

$$N(a) = a_0^2 - a_0 a_1 + a_1^2 + a_2^2 - a_2 a_3 + a_3^2,$$

and is an integer.

We claim that N is a "left" Euclidean norm on H, that is, given $a, d \in H$, $d \neq 0$, there are $q, r \in H$ with $a = qd + r$ and $N(r) < N(d)$. It suffices to prove this for d an integer n, since then $ad^* = qdd^* + rd^*$ where $N(rd^*) < N(dd^*)$ and it follows for all d. Now, for $a \in H$

$$N(a) = \tfrac{1}{4}(2a_0 - a_1)^2 + \tfrac{3}{4}a_1^2 + \tfrac{1}{4}(2a_2 - a_3)^2 + \tfrac{3}{4}a_3^2.$$

Writing δ_i for $a_i - nq_i$, is suffices to show that q can be chosen so that $|\delta_1| \leq n/2$ and $|2\delta_0 - \delta_1| \leq n$, and similarly for δ_3, δ_2. Certainly the required q_1 can be chosen, and q_0 then can also be since $a_0 - nq_0$ can be chosen in any desired interval of length n.

The remainder of the proof is as in [He64, Thm. 7.f], except $u = u_0 + u_1\zeta + u_2 j + u_3\zeta j$, from which

$$2u = (2u_0 - u_1) + u_1\sqrt{3}i + (2u_2 - u_3)j + u_3\sqrt{3}k$$

follows, and $4p$ is of the form (5).

10. Conclusion. For $v = 3q$, we have shown the existence of short H-designs. Questions that should be studied further include improvements to the bound for H-designs; bounds for any k; and the existence of designs with $\lambda = k(k-1)$. The method of difference families would doubtless yield some results.

The more general problems of bounds on $b_1(v,(v+1)/2)$ for v odd, and on $b_1(v,k)$ in general, seem to require more extensive developments. Additional results would doubtless follow by both group methods and recursive methods; in particular Lemma 11 can doubtless be improved, perhaps using additional recursions and direct sieving methods. Nonconstructive methods for these problems are more difficult than those used in the proof of Wilson's theorem, since v cannot be taken as large as necessary. Finally, methods based on the integer program might be further investigated.

REFERENCES

[BJL] T. Beth, D. Jungnickel, and H. Lenz, *Design Theory*, Bibliographisches Institut, 1985.

[BP79] A. Berman and R. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.

[Do87] M. Dowd, *Incidence matrices and systems of linear equations*, Ars Combinatoria, 24 (1987), pp. 45–49.

[GS79] A. Geramita and J. Seberry, *Orthogonal Designs: Quadratic Forms and Hadamard Matrices*, Marcel Dekker, New York, 1979.

[Ha75] H. Hanani, *Balanced incomplete block designs and related designs*, Discrete Math., 11 (1975), pp. 255–369.

[Her] I. N. Herstein, *Topics in Algebra*, Blaisdell Publishing Company, 1964.

[IR82] K. Ireland and M. Rosen, *A Classical Introduction to Modern Number Theory*, Springer-Verlag, New York, Berlin, 1982.

[Ma65] H. Mann, *Addition Theorems*, Wiley–Interscience, New York, 1965.

[Pa73] C. Parry, *Primes represented by binary quadratic forms*, J. Number Theory, 5 (1973), pp. 266–270.

[PS82] C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, 1982.

[Ro64] K. Rogers, *A note on orthogonal Latin squares*, Pacific J. Math., 14 (1964), pp. 1395–1397.

[Wi75] K. Williams, *Note on integers representable by binary quadratic forms*, Canad. Math. Bull., 13 (1975), pp. 123–125.

SIAM J. DISC. MATH.
Vol. 4, No. 3, pp. 305–322, August 1991

A FORMAL THEORY OF CONSENSUS*

J. P. BARTHÉLEMY† AND M. F. JANOWITZ‡

Abstract. A broad set-theoretic model for consensus methods is presented. Within the framework of this model, a very general characterization of the median rule as well as various types of quota rule and polynomial rule are obtained. The model encompasses those situations where a single consensus object is achieved, as well as those in which a multiple consensus is allowed.

Key words. consensus, voting rule, median, quota rule, semilattice

AMS(MOS) subject classifications. 90A08, 06A12

Introduction. Consensus techniques have developed in such diverse fields as voter preference, taxonomy, phylogenetics, mathematical economics, anthropology, and sociology. For example:

- Social welfare functions and voting theory are concerned with the necessity for any society to arrive at a collective decision based on information provided by the individual members of that society.
- Evolutionary biology deals with the reconstruction of evolutionary trees from various estimates that are obtained from fossils as well as currently existing specimens.
- Classical statistics summarize numerical data by means of the concept of central value (e.g., mean, median, mode, etc.). Modern data analysis is becoming more and more concerned with information obtained from more complex objects (rankings, partitions, hierarchical trees, undirected trees, etc.). Due to the lack of a natural numerical structure for these objects, there is a need to develop a mathematical theory of consensus.

Arrow (1962) proposed a number of seemingly reasonable conditions that voting schemes might enjoy and proceeded to show in his celebrated "impossibility theorem" that no such voting scheme could exist. Since then there has been a flurry of activity designed to prove analogues of this theorem in other contexts, and to establish contexts in which the rather dismaying consequences of Arrow's theorem are not necessarily valid. The resulting theory has developed somewhat independently in a number of disciplines, and we often see the same theorem proved differently in different contexts.

What is needed is a general mathematical model in which these matters may be disposed of in a common setting. That is to say, we forget about the exact nature of the objects and, using some abstract structure on various sets of objects under consideration, concern ourselves instead with ways in which the structure can be used to summarize a given family of objects. Proceeding in this manner, two approaches are already available: one uses linear algebra (Rubinstein and Fishburn (1986)); the other uses ordered structures (Barthélemy, Leclerc, and Monjardet (1986)). The present paper deals with ordinal models. Since Barbut (1961), several authors have considered such models for a theory of consensus (see Monjardet (1980), (1990); Bandelt and Barthélemy (1984); Neumann and Norton (1986); Leclerc (1990)). In papers like Monjardet (1990) the emphasis is on those lattice polynomial rules that produce a unique consensus object from a fixed

* Received by the editors August 1, 1988; accepted for publication (in revised form) September 4, 1990. This research was supported in part by Office of Naval Research grant N00014-90-1008.

† Département Informatique, Ecole Nationale Superieure des Telecommunications, 75634 Paris Cedex 13, France.

‡ Department of Mathematics and Statistics, University of Massachusetts, Amherst, Massachusetts 01003.

number of objects that are to be summarized. In the present paper the approach is somewhat different:

(i) The number of objects to be summarized is not fixed. For example, in connection with voting rules, it is important to know what happens when new voters are added or when one or more ballots are disqualified.

(ii) A given family may have associated with it one or several consensus objects. For example, assume that the simple majority rule is used to determine the outcome of a vote on a motion. If there are more PRO votes than CON votes, the motion passes, and it fails if there are more CON votes than PRO votes. In case of equality between PRO and CON the decision is in doubt and simple majority rule can only produce {passes, fails} as a consensus.

Here is the general organization of the paper. Some preliminary ideas are discussed in § 0, with § 1 being devoted to a discussion of set-theoretic models for consensus methods. Median semilattices are introduced in § 2 in order to study the "majority rule" consensus, and characterizations are provided therein for the so-called "median" consensus. Section 3 contains material that generalizes the results of § 2. First of all, consensus rules are introduced that are more general than the simple majority rule. Second, these new rules are extended from median semilattices to distributive semilattices, and finally the need for distributivity is discussed.

0. Preliminaries. We recall here some definitions of mathematical objects, occurring as models in cluster analysis and social choice theory, that we shall use throughout this paper.

Let U be a finite set. A *weak order* on U is a binary relation W on U which is asymmetric (i.e., uWv implies not (vWu)) and negatively transitive (i.e., not (uWv) and not (vWw) imply not (uWw)). An important point is that W is a weak order if and only if its "dual negation" $N(W)$, $uN(W)v$ if and only if not (vWu), is a complete preorder (i.e., a complete transitive relation. Note that terminology in social choice theory is not always uniform, so that what we have called a weak order is sometimes called a "strict weak order," and what we called a complete preorder is sometimes called a "weak order." At any rate, with every weak order W there is associated an equivalence relation $E(W)$ defined by $uE(W)v$ if and only if $uN(W)v$ and $vN(W)u$. An equivalence class of $E(W)$ is called a *class* of the weak order W, and W induces a linear order $L(W)$ (or $\leqq$) on the set of all classes of W. Let $C_1, C_2, \cdots, C_p$ be the distinct classes of W, with $C_1 \leqq C_2 \leqq \cdots \leqq C_p$. The *q-section* S_q of W is the union of the first q classes, with the convention that $S_0 = \varnothing$. So we get $S_p = U$, and the sections of W constitute a chain of subsets of U, from $\varnothing$ to U. Conversely, to each chain of subsets of U, $\varnothing = U_0 \subset U_1 \subset \cdots \subset U_p = U$ there is associated the weak order W defined by uWv if and only if $u \in U_i$, $v \in U_j \setminus U_i$, with $i < j$. So we have a bijective map from the set $\mathscr{W}$ of all weak orders on U to the set C of all chains of subsets of U, from $\varnothing$ to U. Moreover, the weak order W is included in the weak order W' (in the sense that uWv implies $uW'v$) if and only if each section of W is a section of W'.

A *hierarchical tree* (alias *n-tree*) on U is a set $\mathbf{H}$ of subsets of U, such that $U \in \mathbf{H}$; $\varnothing \notin \mathbf{H}$; for each $u \in U$, $\{u\} \in \mathbf{H}$; and for each pair $A, B \in \mathbf{H}$, $A \cap B \in \{A, B, \varnothing\}$. A set $A \in \mathbf{H}$ is called a cluster of $\mathbf{H}$. The set U and the singletons $\{u\}$ are called the *trivial clusters*. We shall let $\mathscr{H}$ denote the set of hierarchical trees on U.

A *phylogenetic tree* on the set U is a graph-theoretic tree $\mathbf{T}$ together with a map f from U to the vertex set V of $\mathbf{T}$ such that each vertex in $V \setminus f(U)$ has a degree greater than or equal to three. The deletion of any edge of T induces a bipartition on U that is called a *split* of $\mathbf{T}$. The splits of $\mathbf{T}$ fulfill the following *compatibility property*: if $\{A, A'\}$

and $\{B, B'\}$ are splits then at least one among the four intersections $A \cap B$, $A' \cap B$. $A \cap B'$, $A' \cap B'$ is empty. We know from Buneman (1971) that a phylogenetic tree may be recaptured from its set of splits $B(\mathbf{T})$; the datum of a phylogenetic tree on U is equivalent to the datum of pairwise compatible bipartitions of U. So introducing the graph $G(U)$ with all bipartitions of U as vertices and all pairs of compatible bipartitions as edges, the phylogenetic trees may be interpreted as the complete subgraphs of $G(U)$. We shall use $\mathscr{T}$ to denote the set of phylogenetic trees on U.[1]

1. A generalized setting for consensus functions.

1. A generalized setting for consensus functions. Let X be a finite set. We are concerned with the determination of a consensus between several elements of X. We shall not assume that such a consensus is unique.

1.1. Consensus rules.

DEFINITION 1. A *consensus rule* on X is a map $c: X^* \to \mathbf{P}(X) \setminus \{\emptyset\}$, where $\mathbf{P}(X)$ is the power set of X, and $X^* = \bigcup_{k>0} X^k$ with X^k as the k-fold Cartesian product of X with itself. A *strict consensus rule* is a mapping $c: X^* \to X$.

An element $x^* = (x_1, \cdots, x_k)$ of X^* is called a *profile*. We denote by $(x)_k$ a *constant profile*; i.e., a profile having each of its k components equal to some fixed x. An element $y \in c(x^*)$ is called a *consensus* of x^*. In the case of a strict consensus rule, the element $c(x^*)$ is called *the* consensus of x^*.

Let k be a positive integer. A map from X^k into $\mathbf{P}(X) \setminus \{\emptyset\}$ (respectively, to X) is called a *k-consensus rule* on X (respectively, a *strict k-consensus rule* on X). Finally, the symbol V_k will be used to denote the set $\{1, 2, \cdots, k\}$ of the first k positive integers.

1.2. Stability and stability families. The general idea here is that the elements of X are complex objects built from bricks (e.g., the hierarchical trees are built from the clusters). In the following, we shall look at properties involving essentially the bricks occurring in the several consensuses of a profile, forgetting the ways in which the bricks are used to construct the different types of consensus. A *stability family* (Barthélemy and Monjardet (1981)) on X is a pair (S, f) where S is a set (the set of "bricks") and f a map from X into $\mathbf{P}(S)$. To each consensus rule c and each profile $x^* \in X^*$, the (S, f)-*solution set* $S(c, x^*)$ is defined by the requirement that

$$S(c, x^*) = \bigcup_{x \in c(x^*)} f(x).$$

If T is a fixed subset of S and $f(x) = T$ for all $x \in X$, then $S(c, x^*) = T$ for every consensus rule c and every profile x^*. Since the notion of a stability family now provides no information about the nature of consensus rules, there seems little point in considering this degenerate case. *It will therefore be assumed that any stability family (S, f) has the property that there exist $a, b \in X$ such that $f(a) \neq f(b)$.* Indeed, in many concrete situations, the mapping f is in fact one-to-one.

Examples 1 and 2 below are often considered in the literature on consensus.

Example 1. X is either the set $\mathscr{W}$ of all weak orders on the finite set U, or the set $\mathscr{L}$ of all linear orders on U, or the set $\mathscr{O}$ of all partial orders on U. The set S is the cartesian product $U \times U$. For each $x \in X$, $f(x)$ is the set of all ordered pairs (u, v) such that $u < v$ for the given order x.

Example 2. X is the set $\mathscr{P}$ of all partitions on the set U; $S = U \times U$, and for each $x \in X$, $f(x)$ is the set of all pairs (u, v) such that u and v are in the same class of the partition x. Thus $f(x)$ is the equivalence relation associated with the partition x.

[1] A more detailed explanation of this can be found in the paper *From copair hypergraphs to median graphs with latent vertices*, J. P. Barthélemy, Discrete Mathematics, 76 (1989), pp. 9–28.

DEFINITION 2. Let c be a consensus rule, and let (S, f) be a stability family on X:

(i) c is *stable on solutions* whenever, for every positive integer k, for all $s \in S$ and for all profiles x^*, $y^* \in X^k$,

$$\{i: s \in f(x_i)\} = \{i: s \in f(y_i)\}$$

implies that

$$s \in S(c, x^*) \quad \text{if and only if } s \in S(c, y^*).$$

(ii) c is *neutral on solutions* whenever, for every positive integer k, for all s, $t \in S$ and for all profiles x^*, $y^* \in X^k$,

$$\{i: s \in f(x_i)\} = \{i: t \in f(y_i)\}$$

implies that

$$s \in S(c, x^*) \quad \text{if and only if } t \in S(c, y^*).$$

(iii) c is *monotone neutral on solutions* if and only if for every positive integer k, for all s, $t \in S$ and for all profiles x^*, $y^* \in X^*$,

$$\{i: s \in f(x_i)\} \subseteq \{i: t \in f(y_i)\}$$

implies that

$$t \in S(c, y^*) \quad \text{whenever } s \in S(c, x^*).$$

The above definitions can be applied to strict consensus rules either by identifying x with $\{x\}$ for each element of X, or by defining the (S, f)-solution set by the rule $S(c, x^*) = f(c(x^*))$. Obviously, they can also be adapted to the case of k-consensus rules. In that spirit we recall in Proposition 1 a fundamental result for strict consensus rules in social choice and in cluster analysis.

In Examples 1 and 2, stability on solutions is usually called the *decisiveness condition* (when it is fulfilled, the consensus rule c is said to be decisive; Ferejohn and Fishburn (1979)). Recall that a strict consensus rule on the set of binary relations is said to be *paretian* if and only if for each profile $x^* \in X^k$, $\bigcap_{1 \leqq i \leqq k} x_i \subseteq c(x^*)$. Proposition 1 summarizes some familiar results in social choice theory and mathematical taxonomy; part (i) is essentially the classical Arrow theorem (Arrow (1962)), under the decisive case; part (ii) is essentially a result by Mas-Collel and Sonnenschein (1972) and Brown (1975); part (iii) is the Mirkin theorem (Mirkin (1975)) in the improved formulation by Leclerc (1984).

PROPOSITION 1. (i) *For $X = \mathscr{W}$ or $X = \mathscr{L}$, the decisive and paretian strict consensus rules are exactly the dictatorships; that is, for each integer k there exists an integer $i(k) \leqq k$ such that for each $x^* = (x_1, \cdots, x_k) \in X^k$, $c(x^*) = x_{i(k)}$.*

(ii) *For $X = \mathscr{O}$, the decisive and paretian strict consensus rules are exactly the oligarchic rules; that is to say, for each integer k, there exists a nonempty subset W_k of V_k such that for each $x^* = (x_1, \cdots, x_k) \in \mathscr{O}^k$, $c(x^*) = \bigcap_{i \in W_k} x_i$.*

(iii) *For $X = \mathscr{P}$, the decisive and paretian strict consensus rules are exactly the oligarchic rules.*

Another classical property of consensus rules is the notion of *symmetry*. Translated in terms of stability families this becomes the following definition.

DEFINITION 3. A consensus rule c is *symmetric on solutions* if and only if for each profile $x^* = (x_1, \cdots, x_k)$ and each permutation σ of V_k,

$$S(c, x^*) = S(c, \sigma(x^*)) \quad \text{where } \sigma(x^*) = (x_{\sigma(1)}, \cdots, x_{\sigma(k)}).$$

In case of a strict consensus rule c, we shall just say that c is *symmetric*. To illustrate the notion of symmetry, we mention the result below, which follows immediately from Proposition 1.

COROLLARY 2. (i) *For $X = \mathscr{W}$ or $X = \mathscr{L}$, there is no strict consensus rule that is paretian, decisive, and symmetric.*

(ii) *For $X = \mathscr{O}$ or $X = \mathscr{P}$, the only strict consensus rule that is paretian, decisive, and symmetric is the unanimity rule, which is defined by $c(x^*) = \bigcap \{x_i\colon 1 \leqq i \leqq k\}$ for $x^* \in X^k$.*

Working in the general framework of stability families, we now describe the structure of a neutral consensus rule.

1.3. Neutral consensus rules on a finite set. Let X be a finite set with (S, f) a stability family on X. We are concerned with the characterization of consensus rules that are either neutral or monotone neutral on solutions. The general flavor of the results can be seen at the outset from the following two lemmas.

LEMMA 3. *The following conditions on the consensus rule c are equivalent*:

(i) *c is neutral on solutions.*

(ii) *For each positive integer k, there is a collection $\mathbf{D}_k$ of subsets of V_k such that for any profile $x^* \in X^k$ it is true that*

$$s \in S(c, x^*) \textit{ if and only if } \{i\colon s \in f(x_i)\} \in \mathbf{D}_k.$$

Proof. The assertion that (ii) implies (i) is clear, so we assume the validity of (i) and seek to prove (ii). Fix a positive integer k and consider profiles $x^* \in X^k$. Define $\mathbf{D}_k$ by the rule $D \in \mathbf{D}_k$ if and only if for some profile x^* and some $s \in S$,

$$D = \{i \in V_k\colon s \in f(x_i)\} \quad \text{and} \quad s \in S(c, x^*).$$

By neutrality, if $t \in S$ and if $y^* = (y_1, \cdots, y_k) \in X^k$, then

$$\{i \in V_k\colon t \in f(y_i)\} \in \mathbf{D}_k \quad \text{implies } t \in S(c, y^*).$$

Define $\Phi\colon X^k \to \mathbf{P}(S) \setminus \{\varnothing\}$ by

$$s \in \Phi(x^*) \quad \text{iff } \{i \in V_k\colon s \in f(x_i)\} \in \mathbf{D}_k.$$

We need to prove that $\Phi(x^*) = S(c, x^*)$. If $s \in S(c, x^*)$, then by definition of Φ, $s \in \Phi(x^*)$. On the other hand, if $s \in \Phi(x^*)$, then $\{i \in V_k\colon s \in f(x_i)\} \in \mathbf{D}_k$, so for some $t \in S$ and some profile $y^* \in X^k$, we have

$$\{i \in V_k\colon t \in f(y_i)\} = \{i \in V_k\colon s \in f(x_i)\}$$

and $t \in S(c, y^*)$. By neutrality, $s \in S(c, x^*)$. □

LEMMA 4. *Let c be neutral on solutions and define $\mathbf{D}_k$ as in Lemma 3. Then c is monotone neutral on solutions if and only if each $\mathbf{D}_k$ is an order filter of $\mathbf{P}(V_k)$ in that $\mathbf{D}_k$ is not empty and $D \in \mathbf{D}_k$, $D \subset D'$ together imply that $D' \in \mathbf{D}_k$.*

Proof. If each $\mathbf{D}_k$ is an order filter of $\mathbf{P}(V_k)$, it is clear that c is monotone neutral on solutions. To obtain the converse, we assume that c is monotone neutral on solutions and $D \in \mathbf{D}_k$ with $D \subset D'$. Choose elements $a, b \in X$ such that $f(a)$ is not contained in $f(b)$. Define $x^* = (x_1, \cdots, x_k)$ by $x_i = a$ if $i \in D'$ and $x_i = b$ otherwise. The fact that $D \in \mathbf{D}_k$ implies the existence of a profile $y^* \in X^k$ and an element $s \in S$ such that $s \in S(c, y^*)$ and $D = \{i \in V_k\colon s \in f(y_i)\}$. Now if $s' \in f(a) \setminus f(b)$, we see that $D' = \{i \in V_k\colon s' \in f(x_i)\}$. Since c is monotone neutral this puts $s' \in S(c, x^*)$, whence $D' \in \mathbf{D}_k$. □

This also leads to a formula for $S(c, x^*)$ when c is monotone neutral on solutions.

PROPOSITION 5. *The following conditions on the consensus rule c are equivalent*:

(i) *c is monotone neutral on solutions.*

(ii) *For each positive integer k, there exists a family* $J_{k,1}, \cdots, J_{k,w(k)}$ *of subsets of* V_k*, no pair of which is comparable, having the property that for any* $x^* = (x_1, \cdots, x_k) \in X^k$,

$$S(c,x^*) = \bigcup_{1 \leqq p \leqq w(k)} \left[\bigcap_{i \in J_{k,p}} f(x_i) \right].$$

Proof. Condition (ii) clearly implies (i), so assume (i). Define $\mathbf{D}_k$ as in Lemma 3, and note that by Lemma 4, each $\mathbf{D}_k$ is an order filter of $\mathbf{P}(V_k)$. Let $J_{k,1}, \cdots, J_{k,w(k)}$ be the minimal elements of $\mathbf{D}_k$. Then if $x^* \in X^k$, we know that $s \in S(c, x^*)$ implies that

$$\{i \in V_k : s \in f(x_i)\} \in \mathbf{D}_k,$$

so it contains some $J_{k,p}$. But then $s \in \bigcap_{i \in J_{k,p}} f(x_i)$. On the other hand, $s \in \bigcap_{i \in J_{k,p}} f(x_i)$ puts $s \in S(c, x^*)$ by c being monotone neutral on solutions. □

DEFINITION 4. The *index* of the element $s \in S$ in the profile $x^* \in X^k$ is defined by the formula

$$\gamma(s,x^*) = |\{i \in V_k : s \in f(x_i)\}|/k.$$

We can now state Corollary 6.

COROLLARY 6. *The following two assertions on the consensus rule c are equivalent*:

(i) *c is monotone neutral and symmetric on solutions.*

(ii) *Corresponding to each positive integer k, there is a rational number* t_k $(0 \leqq t_k \leqq 1)$ *such that for all profiles* $x^* \in X^k$,

$$S(c,x^*) = \{s \in S : \gamma(s,x^*) \geqq t_k\}.$$

In that $s \in S(c, x^*)$ if and only if the assertion $s \in f(x_i)$ is true for a certain quota of indices i, this constitutes a characterization of "quota rules" on solutions. For a fixed positive integer k, these are often called "counting rules." This notion of quota rule will be further studied in § 3.

1.4. Consistent consensus rules and median consensus of a finite set.

DEFINITION 5. A consensus rule c on the set X is *consistent* (Young (1974), Young and Levenglick (1978)) if and only if for all profiles x^*, y^* the condition $c(x^*) \cap c(y^*) \neq \varnothing$ implies that

$$c(x^* y^*) = c(x^*) \cap c(y^*),$$

where $x^* y^*$ is the concatenation of the two profiles x^* and y^*.

For instance, if c is consistent and if $c(x) = \{x\}$ for each $x \in X$, then $x \in c\{x^*)$ implies $c(x^* x) = \{x\}$.

The median consensus is an important example of a consistent consensus rule. In order to define it, we must consider metrics on the finite set X.

DEFINITION 6. The *median consensus* on the finite metric space (X, d) is the consensus rule m on X defined by

$$m(x^*) = \{x \in X : D(x,x^*) \text{ is a minimum}\},$$

where $x^* \in X^*$ and $D(x, x^*) = \sum_{1 \leq i \leq k} d(x, x_i)$.

LEMMA 7. *The median consensus on the finite metric space* (X, d) *is consistent.*

Proof. Let $x \in m(x^*y^*)$ and $y \in m(x^*) \cap m(y^*)$. Then $D(y, x^*) \leqq D(x, x^*)$, $D(y, y^*) \leqq D(x, y^*)$, and $D(x, x^*y^*) \leqq D(y, x^*y^*)$. It follows that

$$\begin{aligned} D(x,x^*)+D(x,y^*) &= D(x,x^*y^*) \leqq D(y,x^*y^*) \\ &= D(y,x^*)+D(y,y^*) \leqq D(x,x^*)+D(x,y^*). \end{aligned}$$

This forces $D(x, x^*y^*) = D(y, x^*y^*)$, so $y \in m(x^*y^*)$. But now

$$D(x,x^*)+D(x,y^*)=D(y,x^*)+D(y,y^*)$$

with $D(y, x^*) \leqq D(x, x^*)$ and $D(y, y^*) \leqq D(x, y^*)$ also forces

$$D(x,x^*)=D(y,x^*) \quad \text{and} \quad D(x,y^*)=D(y,y^*)$$

from which it follows that $x \in m(x^*) \cap m(y^*)$. □

2. The median consensus on median semilattices.

2.1. Median semilattices. Median semilattices constitute an immediate and important generalization of both distributive lattices and tree semilattices. They arise in a variety of situations and for that reason results true for all median semilattices often have far-reaching consequences. For classical results on median semilattices, median graphs and median algebras, the reader is referred to Bandelt and Hedlíková (1983).

We ask the reader to recall that a *semilattice* is a partially ordered set S having the property that every pair of elements has a meet. The dual notion is called a *join semilattice*, and to say that S is a lattice is to say that both S and its dual are semilattices. Finally, the lattice S is called *distributive* if for all $a, b, c \in S$, it is true that $(a \vee b) \wedge c = (a \wedge c) \vee (b \wedge c)$. M is an order ideal of the partially ordered set S if M is a nonempty subset of S having the property that $x \in M$ and $u \leqq x$ together imply that $u \in M$. An order ideal of the form $(x] = \{t: t \leqq x\}$ for some $x \in S$ is called a *principal ideal* of S.

DEFINITION 7. A *median semilattice* is a semilattice such that:

(i) Every principal ideal is a distributive lattice, and

(ii) Any three elements have an upper bound whenever each pair of them has an upper bound.

A useful tool for working with median semilattices is the so-called *Sholander embedding*.

PROPOSITION 8 (Sholander (1954)). *Every median semilattice (finite or infinite) $(M, \leqq)$ can be embedded in a distributive lattice $(L, \leqq)$ such that:*

(i) *M is an order ideal of $(L, \leqq)$ and*

(ii) *each element of L is the join of finitely many elements of M.*

An element s of a partially ordered set P is called *sup-irreducible* if it is not the smallest element of P and it cannot be expressed as the supremum of finitely many elements distinct from s. The next remark follows directly from Proposition 8.

Remark 9. The sup-irreducibles of a median semilattice are exactly the sup-irreducibles of the Sholander distributive lattice described in Proposition 8.

A partially ordered set P is said to have *finite length* if there is an upper bound on the lengths of its chains. In a poset P of finite length with smallest element $\mathbf{0}$, we can define the *height* $h(x)$ of the element x to be the least upper bound of the lengths of the chains connecting x with $\mathbf{0}$. The element x *covers* y when $x > y$ and there is no element z such that $x > z > y$. P is said to be *graded* by its height function provided that $h(x) = h(y) + 1$ whenever x covers y. In particular, a median semilattice M with $\mathbf{0}$ of finite length is graded by its height function h, and the *least move* (or *shortest path*) metric on the diagram of M is given as in Monjardet (1981) by

$$d(x,y)=h(x)+h(y)-2h(x \wedge y).$$

From here on, d will be referred to as the *lattice metric* on M. From Proposition 8, it follows that a height function on M extends naturally to a height function of its Sholander distributive lattice L. Hence we have Remark 10.

Remark 10. The embedding of a median semilattice having finite length with $\mathbf{0}$ into its Sholander lattice L is an isometry with respect to the lattice metric d.

Proposition 8 allows us to view a median semilattice as an order ideal of a distributive lattice. On the other hand, not every order ideal of a distributive lattice will produce a median semilattice. An easy example may be obtained by letting L denote the lattice of all subsets of the three element set $S = \{x, y, z\}$, and taking $M = L \setminus \{S\}$. Then $\{x\}$, $\{y\}$, and $\{z\}$ have pairwise upper bounds while these three elements of M have no common upper bound. For further details, see Janowitz (1991).

2.2. Examples of median semilattices. A first example of a median semilattice (Bandelt (1982)) is obtained from the frames of a poset. Let $(E, \leqq)$ be a finite poset, and call Max and Min the sets of, respectively, maximal and minimal elements of $(E, \leqq)$. In connection with the next definition we agree that a closed interval of a poset $(P, \leqq)$ will be a set of the form $[a, b] = \{c\colon a \leqq c \leqq b\}$, where $a \leqq b$. A *subposet* of $(P, \leqq)$ is a nonempty subset F of P equipped with the same partial order as that of P.

DEFINITION 8. A *frame* of the finite poset $(E, \leqq)$ is a subset F of E such that:

(i) $\mathrm{Min} \subseteq F$, $\mathrm{Max} \subseteq F$, and

(ii) Every closed interval of the induced subposet $(F, \leqq)$ is a chain.

On the other hand, a *tree-poset* is a poset $(P, \leqq)$ in which every principal ideal is a chain. By condition (ii), frames appear as generalized tree-posets. Moreover, for a finite set U, a subset H of $E = \mathbf{P}(U) \setminus \{\varnothing\}$ is a hierarchical tree if and only if it is a frame of $(E, \subseteq)$.

If $(E, \leqq)$ has a largest element $\mathbf{1}$ and a smallest element $\mathbf{0}$, then the frames of E are nothing but the chains from $\mathbf{0}$ to $\mathbf{1}$. Thus the weak orders on the finite set U may be interpreted as the frames of $\mathbf{P}(U)$ ordered by inclusion.

Let $\mathrm{Fr}\,(E, \leqq)$ denote the set of frames of $(E, \leqq)$, and let $\mathrm{Fr}\,(E, \leqq)$ be ordered by inclusion. Obviously, if $F \in \mathrm{Fr}\,(E, \leqq)$ and $\mathrm{Max} \cup \mathrm{Min} \subseteq G \subseteq F$, then $G \in \mathrm{Fr}\,(E, \leqq)$. Moreover, we can easily show that if A, B, C are frames of $(E, \leqq)$ such that $A \cup B$, $A \cup C$ and $B \cup C$ are each frames, then so is $A \cup B \cup C$. This establishes Proposition 11.

PROPOSITION 11. $\mathrm{Fr}\,(E, \leqq)$ *ordered by set inclusion is a median semilattice.*

Note that in $\mathrm{Fr}\,(E, \leqq)$, the sup-irreducible elements are precisely the atoms $\mathrm{Max} \cup \mathrm{Min} \cup \{x\}$ with x neither maximal nor minimal. Furthermore, the principal ideals of $\mathrm{Fr}\,(E, \leqq)$ are more than just distributive lattices since they are, in fact, Boolean algebras.

We have already noted that $\mathrm{Fr}\,(\mathbf{P}(U) \setminus \{\varnothing\}, \subseteq)$ is the set $\mathscr{H}$ of all hierarchical trees on U, while $\mathrm{Fr}\,(\mathbf{P}(U), \subseteq)$ is the set $\mathscr{W}$ of all weak orders on U. In $\mathscr{H}$, the smallest element is the bush (i.e., the hierarchical tree admitting only trivial clusters) and the sup-irreducibles are the hierarchical trees with only one nontrivial cluster. So the set of sup-irreducibles of $\mathscr{H}$ may be identified with the nonsingleton proper subsets of U. In $\mathscr{W}$ the set inclusion order may be stated as follows: $W \subseteq W'$ if and only if each class of W is the union of classes of W'. The smallest element of $\mathscr{W}$ is the trivial weak order (which admits only the class U) and the maximal elements of $\mathscr{W}$ are the linear orders on U. The sup-irreducibles of $\mathscr{W}$ are the two-class weak orders $A < U \setminus A$, so the set of all sup-irreducibles of $\mathscr{W}$ may be identified with $\mathbf{P}(U) \setminus \{\varnothing, U\}$. For further investigations into the structure of $\mathscr{H}$ and $\mathscr{W}$ see Leclerc (1985), Barthélemy, Leclerc, and Monjardet (1986), and Janowitz (1985), respectively.

It is easy to show that the set $\mathscr{C}$ of complete relations on U also forms a median semilattice. Here the sup-irreducibles are those relations of the form $(U \times U)\setminus\{a, b\}$ with $a \neq b$, and the maximal elements are those complete relations R having the property that aRb with $a \neq b$ implies the failure of bRa. Thus every linear order is a maximal element of $\mathscr{C}$ as well as $\mathscr{W}$.

Another example of a median semilattice is a tree semilattice. This is a semilattice $(T, \leqq)$ with a smallest element $\mathbf{0}$ such that every principal ideal is a chain (whence $T \in \mathrm{Fr}(T, \leqq)$). The diagram of $(T, \leqq)$ is a directed tree that is rooted at $\mathbf{0}$. Since principal ideals of tree semilattices are chains, while principal ideals of frames are Boolean algebras, it follows that no tree semilattice of height greater than two can be a semilattice of frames of a poset.

A final example of a median semilattice is obtained from the complete subgraphs (or cliques) of a graph. Let $\mathrm{Cl}(G)$ denote the set of cliques of G ordered by set inclusion. Obviously, if $C \in \mathrm{Cl}(G)$ and $C' \subseteq C$, then $C' \in \mathrm{Cl}(G)$. Moreover, we can easily show that if A, B, C are cliques of G such that $A \cup B$, $A \cup C$, and $B \cup C$ are each cliques, then so is $A \cup B \cup C$. We state this formally as Proposition 12.

PROPOSITION 12. *$\mathrm{Cl}(G)$ ordered by set inclusion is a median semilattice.*

Since the phylogenetic trees on U appear as cliques in some graph, the set $\mathscr{T}$ of all phylogenetic trees on U is a median semilattice for the inclusion order between split sets. The smallest element of $\mathscr{T}$ is the one-vertex tree and the sup-irreducibles are the bone-trees (i.e., the trees with just one edge). Bone-trees are in one-to-one correspondence with bipartitions of U. So the set of all sup-irreducibles of $\mathscr{T}$ may be identified with the set $\mathbf{B}(U)$ of all bipartitions of U. Further investigations of the structure of $\mathscr{T}$ may be found in Barthélemy and Guenoche (1988).

2.3. Some additional considerations for consensus rules on semilattices. Let X be a finite semilattice. Denote by $\mathbf{0}$ the smallest element of X. Take as a stability family (§ 1.2) the set S consisting of $\mathbf{0}$ and the sup-irreducibles of X, and the map f defined by

$$f(x)=\{s\in S: s\leqq x\}.$$

Note that each nonzero element of X may be expressed as the join of a family of sup-irreducibles, so the mapping f is one-to-one.

In order to avoid any possible confusion, we shall use terms like "stable on sup-irreducible solutions (SIS)" or "symmetric on SIS" in place of the corresponding terms "stable on solutions" or "symmetric on solutions."

For $x \in X$ and $s \in S$, let $x[s] = s$ if $s \leqq x$ with $x[s] = \mathbf{0}$ otherwise. For $x^* = (x_1, \cdots, x_k) \in X^*$, let $x^*[s]$ be the profile $(x_1[s], \cdots, x_k[s])$. Such profiles are called *skeleton profiles*.

DEFINITION 9. A consensus function c on X is said to be *efficient* if and only if it is true that:

(i) For each constant profile $(x)_k$, $c((x)_k) = \{x\}$.

(ii) For each $s \in S$ and $x^* \in X^*$, $c(x^*[s]) \subseteq [\mathbf{0}, s]$.

In the remainder of this paper, we shall look for characterizations of general consensus rules as defined in § 1.1. Median semilattices and especially the majority rule in median semilattices constitute a good clue as to what to expect and will be studied first.

2.4. Majority rule and median consensus. Consider again a finite median semilattice M and the stability family (S,f) defined as in § 2.3. Then the index of the sup-irreducible s in the profile $x^* = (x_1, \cdots, x_k)$ is the number

$$\gamma(s,x^*)= |\{i: s\leqq x_i\}|/k.$$

Before proceeding we need a pair of definitions.

Definition 10. The elements x, y of a semilattice are said to be *compatible* when they have a join.

Definition 11. For each integer k and for each $x^* = (x_1, \cdots, x_k) \in M^k$, the *majority rule object* $\alpha(x^*)$ is given by

$$\alpha(x^*) = \bigvee\left\{\left(\bigwedge_{i \in I} x_i\right): I \subseteq V_k,\ |I| = \left[1 + \frac{k}{2}\right]\right\},$$

where for each real number h, $[h]$ denotes the largest integer $n \leqq h$.

Definition 11 provides a generalization of the well-known Condorcet (1785) procedure. From Bandelt and Barthélemy (1984), we know that $\alpha(x^*)$ always exists for any profile x^*.

Lemma 13. *For each profile x^* and each $s \in S$ for which $\gamma(s, x^*) = \frac{1}{2}$, s is compatible with $\alpha(x^*)$.*

Proof. This result follows from the following observations:

(1) Because $\gamma(s, x^*) = \frac{1}{2}$, for each $I \subseteq V_k$ with $|I| = [1 + k/2]$, $s \leqq x_i$ for at least one $i \in I$, so s is compatible with x_i for at least one $i \in I$.

(2) If s is compatible with $x \in M$, it is compatible with y for each $y \leqq x$.

(3) If s is compatible with each of $y_1, \cdots, y_p$ and if $y = y_1 \vee \cdots \vee y_p$ exists, then s is compatible with y. □

Now consider the lattice metric d on M (cf. § 2.1) and the median consensus m on the metric space (M, d) (cf. § 1.4). A result from Bandelt and Barthélemy (1984), which extends a result of Barbut (1961), stipulates the next result.

Proposition 14. *Let M be a finite median semilattice. Then for each profile $x^* \in M^k$, $\alpha(x^*) \in m(x^*)$. Moreover, if k is odd, then $m(x^*) = \{\alpha(x^*)\}$.*

When M is a distributive lattice, we know from Monjardet (1980) and Leclerc (1990) that $m(x^*)$ is the interval $[\alpha(x^*), \beta(x^*)]$, where

$$\beta\{x^*) = \bigwedge\left\{\left(\bigvee_{i \in I} x_i\right): I \subseteq V_k,\ |I| = \left[1 + \frac{k}{2}\right]\right\}$$

$$= \bigvee\left\{\left(\bigwedge_{j \in J} x_j\right): J \subseteq V_k,\ |J| = \left[\frac{k+1}{2}\right]\right\}.$$

In any median semilattice $m(x^*)$ is the set $M \cap L[\alpha(x^*), \beta(x^*)]$, where $L[\alpha(x^*), \beta(x^*)]$ is the interval $\{z \in L: \alpha(x^*) \leqq z \leqq \beta(x^*)\}$ of the Sholander distributive lattice L of M (cf. § 2.1). Our first goal is to arrive at a more detailed definition of $m(x^*)$. To do that, we need some additional technical facts.

Lemma 15. *Assume that M is a distributive lattice. Let $x, y \in M$ and $s \in S$. Then $s \leqq x \vee y$ implies $s \leqq x$ or $s \leqq y$.*

Proof. The proof is well known and follows from the fact that

$$s = (s \wedge x) \vee (s \wedge y). \qquad \square$$

Lemma 16. *Assume that M is a distributive lattice and that $x^* \in M^k$. Then:*

(i) *The set of all sup-irreducibles below $\alpha(x^*)$ is $\{s \in S: \gamma(s, x^*) > \frac{1}{2}\}$, and*

(ii) *The set of all sup-irreducibles below $\beta(x^*)$ is $\{s \in S: \gamma(s, x^*) \geqq \frac{1}{2}\}$.*

Proof. To establish (i), we must show that for x^*,

(1) $\gamma(s, x^*) > \frac{1}{2}$ implies $s \leqq \alpha(x^*)$, and

(2) $t \in S$ and $t \leqq \alpha(x^*)$ implies $\gamma(t, x^*) > \frac{1}{2}$.

(1) Consider $s \in S$ with $\gamma(s, x^*) > \frac{1}{2}$. There exists a subset I of V_k with $|I| = [1 + k/2]$ with $s \leqq x_i$ for each $i \in I$. Hence $s \leqq \alpha(x^*)$.

(2) Let $t \in S$ with $t \leqq \alpha(x^*)$. It follows from Lemma 15 that there exists a subset I of V_k with cardinality $[1 + k/2]$ such that $t \leqq \bigwedge_{i \in I} x_i$. In other words, $\gamma(t, x^*) > \frac{1}{2}$.

The same type of argument will establish (ii). □

LEMMA 17. *Assume that M is a distributive lattice. For each profile x^*, $m(x^*)$ is the set of all elements of the form $\alpha(x^*) \vee y$ where y is a join of sup-irreducibles s such that $\gamma(s, x^*) = \frac{1}{2}$.*

Proof. If y is a join of sup-irreducibles s such that $\gamma(s, x^*) = \frac{1}{2}$, then $y \leqq \beta(x^*)$, whence $\alpha(x^*) \vee y \leqq \beta(x^*)$. Conversely, if $x \in m(x^*)$, and if s is a sup-irreducible below x, then $\gamma(s, x^*) \geqq \frac{1}{2}$. □

Now we return to the context of median semilattices.

PROPOSITION 18. *Let M be a median semilattice, and let $x^* \in M^*$. Then:*

(i) *$\alpha(x^*)$ is the supremum of all $s \in S$ such that $\gamma(s, x^*) > \frac{1}{2}$.*

(ii) *$m(x^*)$ is the set of all elements of the form*

$$\alpha(x^*) \vee s_1 \vee \cdots \vee s_p$$

such that the indicated supremum exists in M, and $\gamma(s_i, x^*) = \frac{1}{2}$ for each index i.

Proof. Use the Sholander embedding of M into a distributive lattice L. The proposition then follows from Lemma 16 and the following observations:

(1) The sup-irreducibles of L and M are exactly the same (see Remark 9).

(2) Medians in M are medians in L (see Remark 10).

(3) Joins in L are either joins in M or else elements of $L \backslash M$ (see Proposition 8). □

The freedom we have to either put, or not put, a sup-irreducible s such that $\gamma(s, x^*) = \frac{1}{2}$ into a median of x^* is called, following Young and Levenglick (1978), the quasi-Condorcet property.

DEFINITION 12. A consensus rule c on the semilattice X is *quasi-Condorcet* if and only if, for each $s \in S$ and each $x^* \in X^*$ such that $\gamma(s, x^*) = \frac{1}{2}$, s compatible with x implies that $x \in c(x^*)$ if and only if $x \vee s \in c(x^*)$.

2.5. A characterization of the median consensus in a median semilattice.

THEOREM 19. *Let c be a consensus rule on the median semilattice M. Then $c = m$ if and only if c satisfies the following five conditions:*

(i) *c is efficient.*

(ii) *c is stable on* SIS.

(iii) *c is symmetric on* SIS.

(iv) *c is consistent.*

(v) *c is quasi-Condorcet.*

Proof. m is obviously efficient and symmetric on SIS. Stability on SIS comes from the fact that for each $x \in M^*$, $S(m, x^*) = \{s \in S: \gamma(s, x^*) \geqq \frac{1}{2}\}$. From Lemma 7, m is consistent, and the quasi-Condorcet condition follows from Proposition 18.

Conversely assume that c satisfies (i)–(v). Because of (v) it suffices to establish that for each $s \in S$, each $x^* \in M^*$ and each $x \in c(x^*)$:

(1) If $\gamma(s, x^*) > \frac{1}{2}$, then $s \leqq x$.

(2) If $\gamma(s, x^*) < \frac{1}{2}$, then $s \leqq x$ fails.

(1) Let $x^* \in M^k$, $\gamma(s, x^*) > \frac{1}{2}$ and suppose that for some $x \in c(x^*)$, $s \leqq x$ fails. Then $s \leqq x_i$ for p components x_i of x^*, where $p > k - p$. Set $y^* = x^*(x)_{2p-k}$. By efficiency, $c((x)_{2p-k}) = \{x\}$. Hence by consistency, $c(y^*) = \{x\}$. Now consider the skeleton profile $y^*[s]$. Using efficiency and the quasi-Condorcet condition, we get $s \in S(c, y^*[s])$ and by stability, $s \in S(c, y^*)$. But this forces $s \leqq x$, a contradiction.

(2) Suppose now that $\gamma(s, x^*) < \frac{1}{2}$. Assume that $s \in S(c, x^*)$ and consider the induced skeleton profile $x^*[s]$. By stability on SIS, $s \in S(c, x^*[s])$. Suppose s occurs p times in $x^*[s]$ where $p < k - p$. Let $y^* = (s)_p(\mathbf{0})_p$ and note that by quasi-Condorcet and efficiency, $\mathbf{0} \in c(y^*)$. By efficiency, $c((\mathbf{0})_{2k-p}) = \{\mathbf{0}\}$. Then by symmetry,

$$S(c,x^*[s]) = S(c,y^*(\mathbf{0})_{k-2p}) = \{\mathbf{0}\} \quad \text{a contradiction.} \qquad \square$$

2.6. Applications. In particular, Theorem 19 may be applied to the median semilattices described in § 2.2. When M is the median semilattice $\mathscr{H}$ of all hierarchical trees on the finite set U, we re-obtain a result from Barthélemy and McMorris (1986). In this paper the fact is also established that each condition of Theorem 19 is also necessary. We leave the reader to state the result when M is the median semilattice $\mathscr{T}$ of all phylogenetic trees on U. In the case where M is the median semilattice $\mathscr{W}$ of all weak orders on U, we get a new possibility theorem for "social choice." Because this field is mainly characterized by impossibility theorems (like Arrow's theorem when nondictatorship is required; cf. Proposition 1), this theorem deserves to be fully stated.

According to the median semilattice structure of $\mathscr{W}$ described in § 2.2 we consider as a stability family the pair $(\mathbf{P}(U), f)$, where the map f assigns to each weak order its set of sections (cf. § 0). So for a profile $W^* = (W_1, \cdots, W_k)$ of weak orders and a consensus rule c on $\mathscr{W}$ a *solution section* is a subset V of U such that there exists some $W \in c(W^*)$ so that V is a section of W. The notions of stability on solution sections, symmetry on solution sections, efficiency, and the quasi-Condorcet condition follow immediately.

Any weak order W on a finite set U may be transformed to any other weak order on U by applying the operations of either merging two consecutive classes or dividing a class into two consecutive classes. The lattice metric on $\mathscr{W}$ is simply the count of the minimal number of such operations needed to perform the required transformation. Proposition 18 provides the description of the median consensus on $\mathscr{W}$ and it follows from Theorem 19 that: *The median consensus on $\mathscr{W}$ is the only consensus rule that is efficient, stable on solution sections, symmetric on solution sections, consistent, and quasi-Condorcet.*

3. The *t*-consensus rules in a semilattice.

3.1. From median consensus to *t*-consensus. In this section we shall be dealing exclusively with finite semilattices. A median semilattice structure is needed to obtain the fact that for any profile x^*, $m(x^*)$ is the set of all elements of the form $\alpha(x^*) \vee s_1 \vee \cdots \vee s_p$ such that the indicated supremum exists in X and $\gamma(s_i, x^*) = \frac{1}{2}$ for $i = 1, 2, \cdots, p$ (see Proposition 18).

Suppose now that X is a semilattice having the property that for each $x^* \in X^*$, $\alpha(x^*) = \vee\{s: \gamma(s, x^* > \frac{1}{2})\}$ exists and define the consensus rule $m_{1/2}$ by

$$m_{1/2}(x^*) = \{\alpha(x^*) \vee y: y \text{ is the supremum of sup-irreducibles } s$$
$$\text{with } \gamma(s,x^*) = \tfrac{1}{2} \text{ and } y \text{ compatible with } \alpha(x^*)\}.$$

Does Theorem 19 characterize this $m_{1/2}$ consensus rule? The next proposition shows that the answer is no.

PROPOSITION 20. *For a semilattice X on which $m_{1/2}$ can be defined, the following conditions are equivalent:*

(i) *$m_{1/2}$ is consistent.*

(ii) *Each principal ideal of X is a distributive lattice.*

Proof. Since X is a finite semilattice, every principal ideal of X is a lattice. To show that (i) implies (ii), it is sufficient to verify that the consistency of $m_{1/2}$ prevents either

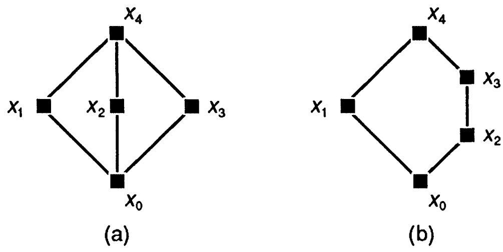


FIG. 1

of the lattices depicted in Fig. 1 from being sublattices of X. By routine but tedious arguments we can verify that in both sublattices of Figs. 1(a) and 1(b),

$$m_{1/2}(x_4,x_1,x_2)=\{x_4), \qquad x_4\in m_{1/2}(x_1,x_3),$$

while

$$x_4\notin m_{1/2}(x_4,x_1,x_1,x_2,x_3).$$

This shows that $m_{1/2}$ is not consistent. That implication (ii) implies (i) will be established in Lemma 25. □

This $m_{1/2}$ consensus rule appears as a special case of a more general type of consensus rule on a semilattice X. To see this, let t be a real number such that $0 \leqq t \leqq 1$. If for each profile $x^* \in X^*$, $\alpha_t(x^*) = \vee\{s \in S: \gamma(s, x^*) > t\}$ exists, then the m_t consensus rule is defined by taking $m_t(x^*)$ to be all elements that can be expressed in the form $\alpha_t(x^*) \vee s_1 \vee \cdots \vee s_p$ with $\gamma(s_i, x^*) = t$. Note that $m_0(x^*)$ is the principal filter generated by $\vee_i x_i$, while $m_1(x^*)$ is the principal ideal generated by $\wedge_i x_i$.

In a median semilattice M, for $t > \frac{1}{2}$, $\alpha_t(x^*)$ must necessarily exist because $\gamma(s, x^*) > \frac{1}{2}$ and $\gamma(s', x^*) > \frac{1}{2}$ together force s, s' to have a common upper bound. In view of this, $m_t(x^*) = [\alpha_t(x^*), \beta_t(x^*)]$, where $\beta_t(x^*) = \vee\{s \in S: \gamma(s, x^*) \geqq t\}$.

So the m_t consensus rules appear as a special case of the quota rules introduced in § 1.3. Our goal is to characterize the m_t rules in a manner analogous to that of the median rule. Proposition 20 tells us that this goal cannot be achieved unless every principal ideal of X is a distributive lattice.

DEFINITION 13. A semilattice X is called *distributive* if every principal ideal of X is a distributive lattice.

It is well known (see Lemma 15) that X is a distributive semilattice if and only if for compatible $x, y \in X$ and $s \in S$, $s \leqq x \vee y$ implies $s \leqq x$ or $s \leqq y$. Lemma 21 is an immediate consequence of this observation. From now on, we shall use it as a basic fact without specifically mentioning it.

LEMMA 21. *Let X be a distributive semilattice, and let $t \in [0, 1]$ be a real number for which α_t is defined. Then for each $x^* \in X^*$ we have $s \leqq \alpha_t(x^*)$ if and only if $\gamma(s, x^*) > t$.*

3.2. The quota number of a semilattice.

DEFINITION 14. The *quota number* $q(X)$ of the semilattice X is the infimum of the real numbers $t \in [0, 1]$ such that $\alpha_t(x^*)$ exists in X for each $x^* \in X^*$.

It is easy to show that $q(X)$ always exists for any finite semilattice X and that $\alpha_t(x^*)$ exists for each $x^* \in X^*$ if and only if $q(X) \leqq t \leqq 1$. Furthermore, to say that X is a lattice is equivalent to the assertion that $q(X) = 0$. More precisely, we have Proposition 22.

PROPOSITION 22. *Let X be a semilattice. Then either $q(X) = 0$ and X is a lattice, or else $q(X) \geqq \frac{1}{2}$.*

Proof. Assume $q(X) < \frac{1}{2}$. Let $a, b \in X$ and take x^* to be the profile (a, b). Then if $s \in f(a) \cup f(b)$, clearly $\gamma(s, x^*) \geqq \frac{1}{2}$. The existence of $a \vee b$ now follows from the fact that $\vee\{s: s \in f(a) \cup f(b)\}$ exists, and this establishes that $q(X) = 0$. □

As an obvious consequence of Proposition 22, we have Corollary 23.

COROLLARY 23. *Let M be a median semilattice. Then either $q(X) = 0$ and X is a distributive lattice, or else $q(M) = \frac{1}{2}$ and M is not a lattice.*

Remark 24. Quota numbers have been further investigated in Bandelt and Meletiou (1990) as well as Bandelt, Janowitz, and Meletiou (1990). It is shown there that if X is a semilattice in which every principal ideal is a lattice, then $q(X)$ exists and belongs to the set $\{0, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \cdots, 1\}$. Hence $q(X)$ is necessarily a rational number.

3.3. Consistency and t-Condorcet property for the m_t rule.

LEMMA 25. *Let X be a distributive semilattice, and let t be a real number in $[q(X), 1]$. Then the consensus rule m_t is consistent.*

Proof. This would follow if we could just show that if $m_t(x^*) \cap m_t(y^*) \neq \varnothing$, then:

(1) $\alpha_t(x^*) \vee \alpha_t(y^*) = \alpha_t(x^*y^*)$, and

(2) $\gamma(s, x^*y^*) \geqq t$ if and only if $\gamma(s, x^*) \geqq t$ and $\gamma(s, y^*) \geqq t$.

Assume then that $m_t(x^*) \cap m_t(y^*) \neq \varnothing$ and let s be such that $\gamma(s, x^*) > t$. Then $s \leqq w$ for each $w \in m_t(x^*)$ and consequently $s \in S(m_t, y^*)$. That is to say, we have:

(i) $\gamma(s, x^*) > t$ implies $\gamma(s, y^*) \geqq t$, and

(ii) $\gamma(s, y^*) > t$ implies $\gamma(s, x^*) \geqq t$.

In what follows, assume that $x^* \in M^{n_1}$ and $y^* \in M^{n_2}$.

(1) Let $s \in S$ with $s \leqq \alpha_t(x^*)$. Then by (i), $\gamma(s, y^*) \geqq t$. Assume $s \leqq x_i$ for k_1 indices and $s \leqq y_j$ for k_2 indices. This says that $k_1 > tn_1$ and $k_2 \geqq tn_2$, so $k_1 + k_2 > t(n_1 + n_2)$. But this forces $\gamma(s, x^*y^*) > t$, whence $s \leqq \alpha_t(x^*y^*)$. Since this is true for all sup-irreducibles below $\alpha_t(x^*)$, we see that $\alpha_t(x^*) \leqq \alpha_t(x^*y^*)$. A similar argument shows that $\alpha_t(y^*) \leqq \alpha_t(x^*y^*)$. Assume now that $s \in S$ and $s \leqq \alpha_t(x^*y^*)$. Then $\gamma(s, x^*y^*) > t$. Suppose $s \leqq x_i$ for k_1 indices and $s \leqq y_j$ for k_2 indices. It follows that $k_1 + k_2 > t(n_1 + n_2)$. In view of this we cannot have both $k_1 \leqq tn_1$ and $k_2 \leqq tn_2$, so at least one of the inequalities $\gamma(s, x^*) > t$ or $\gamma(s, y^*) > t$ must be true. But this says that $s \leqq \alpha_t(x^*)$ or $s \leqq \alpha_t(y^*)$ and we have established (1).

(2) Let $s \in S$ with $\gamma(s, x^*) \geqq t$ and $\gamma(s, y^*) \geqq t$. Then with k_1, k_2 defined as above, $k_1 \geqq tn_1$ and $k_2 \geqq tn_2$, so $k_1 + k_2 \geqq t(n_1 + n_2)$. Consequently, $\gamma(s, x^*y^*) \geqq t$. On the other hand, if $\gamma(s, x^*y^*) \geqq t$, and with k_1, k_2 as above, it must be true that $k_1 + k_2 \geqq t(n_1 + n_2)$. Using (i), we see that

$$k_1 < tn_1 \quad \text{implies } k_2 \leqq tn_2, \quad \text{and}$$

$$k_2 < tn_2 \quad \text{implies } k_1 \leqq tn_1.$$

In either case, the contradiction $(k_1 + k_2) < t(n_1 + n_2)$ is forced. □

DEFINITION 15. Let X be a semilattice, and let $t \in [0, 1]$ be a real number. A consensus rule c on X is *t-Condorcet* if and only if for each $s \in S$ and each $x^* \in X^*$ such that $\gamma(s, x^*) = t$, s compatible with x, implies that $x \in c(x^*)$ if and only if $x \vee s \in c(x^*)$.

So, the quasi-Condorcet condition, as defined in § 2.4 is just the $\frac{1}{2}$-Condorcet condition in a median semilattice.

3.4. A characterization of the m_t-rule.

THEOREM 26. *Let X be a distributive semilattice, let $t \in [q(X), 1]$ be a real number, and let c be a consensus rule on X. Then $c = m_t$ if and only if c satisfies the following five* *conditions*:

(i) *c is efficient.*

(ii) *c is stable on* SIS.

(iii) *c is symmetric on* SIS.

(iv) *c is consistent.*

(v) *c is t-Condorcet.*

Proof. For any choice of t, m_t is obviously efficient and symmetric on SIS. Stability on SIS comes from the fact that for each $x^* \in X^*$, $S(m_t, x^*) = \{s \in S: \gamma(s, x^*) \geqq t\}$. From Lemma 25, m_t is consistent, and the t-Condorcet conditions follow directly from distributivity and the definition of m_t.

Conversely, suppose that c satisfies (i)–(v). Because of (i) it suffices to show that for each $s \in S$, $x^* \in X^*$, and each $x \in c(x^*)$:

(1) If $\gamma(s, x^*) > t$, then $s \leqq x$.

(2) If $\gamma(s, x^*) < t$, then $s \leqq x$ fails.

We have already noted in Remark 24 that t is rational, so we write $t = m/n$ as a quotient in lowest terms of two positive integers, and let $x^* \in X^k$.

(1) Suppose $\gamma(s, x^*) > t$ and that for some $x \in c(x^*)$, $s \leqq x$ fails. Suppose that $s \leqq x_i$ for p components x_i of x^*, where $p/k > m/n$, so $np > mk$. Let y^* be $x^*x^* \cdots x^*$ (m factors) and $z^* = y^*(x)_{np-mk}$. By consistency, $c(y^*) = c(x^*)$ and by efficiency $c((x)_{np-mk}) = \{x\}$. A second application of efficiency now produces $c(z^*) = \{x\}$. Now consider the skeleton profile $z^*[s]$. By the t-Condorcet condition, $s \in S(c, z^*[s])$ and by stability, $s \in S(c, z^*)$. But this forces $s \leqq x$, a contradiction.

(2) Suppose $\gamma(s, x^*) < t$. Assume that $s \in S(c, x^*)$ and consider the induced skeleton profile $x^*[s]$. By stability on SIS, $s \in S(c, x^*[s])$. Suppose s occurs p times in $x^*[s]$ where $np < mk$. Let $y^* = (s)_{mp}(\mathbf{0})_{(n-m)p}$, so $\gamma(s, y^*) = t$. But by t-Condorcet and efficiency, $\mathbf{0} \in c(y^*)$. By efficiency, $c((\mathbf{0})_{mk-np}) = \{\mathbf{0}\}$. Hence by consistency, $c(y^*(\mathbf{0})_{mk-np}) = \{\mathbf{0}\}$, a contradiction. The point is that $x^*[s]$ concatenated with itself n times is a permutation of $y^*(\mathbf{0})_{mk-np}$. □

3.5. Discussion on distributivity. We see from Proposition 20 that the distributivity of the semilattice X is needed to obtain a characterization of $m_{1/2}$ using consistency. More generally, when the t-consensus rule m_t exists, it is, in fact, monotone neutral on SIS. In view of this, it seems reasonable to attempt a characterization of the m_t rules in terms of this concept. The first question we shall ask is when there exists a single real number t ($0 \leqq t \leqq 1$) such that one of the following conditions hold:

(a) For each profile $x^* \in X^*$, $S(c, x^*) = \{s \in S: \gamma(s, x^*) \geqq t)$;

(b) For each profile $x^* \in X^*$, $S(c, x^*) = \{s \in S: \gamma(s, x^*) > t)$.

To see the difference between the two conditions, take $t > \frac{1}{2} \geqq q(X)$ to be rational, and consider the strict consensus methods defined by taking $c(x^*) = \alpha_t(x^*)$ and $c'(x^*) = \beta_t(x^*)$. The rule c satisfies (b), while c' satisfies (a). As long as X has more than one nonzero sup-irreducible element, Proposition 5 is valid, so it is also informative to compare this with Corollary 6. The idea is that we wish to somehow replace the family $\{t_k\}$ with a single number t. For there to be any hope of doing this, the various order ideals $\mathbf{D}_k$ must be related to each other. Consistency would of course do this, but it turns out that we can get by with a much weaker condition.

DEFINITION 16. The consensus rule c is called *weakly consistent* if $c(x^*) = c((x^*)^k)$ for all $x^* \in X^*$ and all positive integers k. Here $(x^*)^k = x^*x^* \cdots x^*$ (k times).

DEFINITION 17. The *least index t* of the consensus rule c is defined to be the infimum of all numbers t such that for each profile x^*, $\gamma(s, x^*) > t$ implies $s \in S(c, x^*)$.

PROPOSITION 27. *Let c be a consensus rule on the semilattice X, and let t be the least index of c. Assume that $q(X) \leqq t \leqq 1$. Conditions* (iii), (iv), *and* (v) *all hold if and only if either* (i) *or* (ii) *is true*:

(i) *For each profile $x^* \in X^*$, $S(c, x^*) = \{s \in S: \gamma(s, x^*) > t\}$.*

(ii) *For each profile $x^* \in X^*$, $S(c, x^*) = \{s \in S: \gamma(s, x^*) \geqq t\}$.*

(iii) *c is weakly consistent.*

(iv) *c is monotone neutral on* SIS.

(v) *c is symmetric on* SIS.

Proof. Evidently, either (i) or (ii) will imply the validity of (iii), (iv), and (v). So let us assume that (iii), (iv), and (v) all hold. Consider $x^* \in X^*$. By construction of t, $s \in S(c, x^*)$ implies that $\gamma(s, x^*) \geqq t$. On the other hand, if $\gamma(s, x^*) \geqq t$, there must exist $s' \in S$ and $y^* \in X^{k'}$ such that $\gamma(s, x^*) \geqq \gamma(s', y^*)$ and $s' \in S(c, y^*)$. Let v^* be x^* concatenated with itself k' times, and w^* the result of concatenating y^* with itself k times. Then v^* and w^* are both members of $X^{kk'}$ and

$$\gamma(s, v^*) = \gamma(s, x^*) \geqq \gamma(s', y^*) = \gamma(s', w^*).$$

By c being monotone neutral and symmetric on SIS, this puts $s \in S(c, x^*)$ as desired.

If t is less than all $\gamma(s', y^*)$ for which $s' \in S(c, y^*)$, this establishes (i). If $t \in \gamma(s', y^*)$ for some $s' \in S(c, y^*)$, then the argument given above will show that whenever $\gamma(s, x^*) \geqq t$, it must follow that $s \in S(c, x^*)$, so that (ii) will hold. □

With our generalization of consistency, there will always be a "trivial" consensus rule fulfilling (iii), (iv), and (v) (and, equivalently, (i) or (ii)): namely, the rule s_t defined by $s_t(x^*) = \{s \in S: \gamma(s, x^*) > t\}$. Note that s_t is weakly consistent, but not consistent, and intuitively, is "far" from m_t. In fact we shall see that without consistency, there is no way to do better than Proposition 27 to get a characterization of consensus rules similar to m_t. To see this consider a number t such that $q(X) \leqq t \leqq 1$. For a profile x^*, recall that $\alpha_t(x^*) = \vee \{s \in S: \gamma(s, x^*) > t\}$.

Let X be a finite semilattice, and let t be a number such that $q(X) \leqq t \leqq 1$. Define a *t-secure consensus rule* on X as a consensus rule c with t as its least index such that: for each profile x^*, there exists $x \in c(x^*)$ such that $\alpha_t'(x^*) \leqq x$. We then have Proposition 28.

PROPOSITION 28. *Let X be a semilattice, and let t be a real number such that $q(X) \leqq t < 1$. The following two conditions are then equivalent*:

(i) *X is distributive.*

(ii) *There exists a t-secure consensus rule on X.*

Proof. Assume first that X is distributive. From the proof of Theorem 26, we have that for each $s \in S$ and each $x^* \in X^*$:

If $\gamma(s, x^*) > t$, then $s \leqq x$ for all $x \in m_t(x^*)$.
If $\gamma(s, x^*) < t$, then $s \leqq x$ must fail for all $x \in m_t(x^*)$.

Hence the result with $c = m_t$.

Assume now that X is not distributive. Then there exist compatible elements x, $y \in X$ and $s \in S$ such that $s \leqq x \vee y$ with s not bounded above by either x or y. For any real number t such that $q(X) \leqq t < 1$, it is always possible to consider positive integers p, q and r such that

$$\frac{p}{q} < t < \frac{p+r}{q} < \frac{p+2r}{q} < 1.$$

Set $x^* = (x)_p(y)_p(x \vee y)_r(\mathbf{0})_{q-2p-r}$. Assume for the moment that we can find a t-secure consensus rule c on X. Each $u \in S$ with $u \leqq x$ has the property that $u \leqq x \vee y$, so $\gamma(u, x^*) > t$ and $u \leqq \alpha_t(x^*)$. So there must exist $z \in c(x^*)$ with $x \leqq z$; similarly, we may establish that $y \leqq z$, whence $x \vee y \leqq z$; in other words, $s \in S(c, x^*)$. But $\gamma(s, x^*) = p/q < t$, contrary to the fact that t is the least index of c. □

Remark 29. In case X is distributive and c is consistent, we can easily prove that if $q(X) \leqq t \leqq 1$, then $c = m_t$, if and only if the following conditions are all true:

(i) t is the least index of c.
(ii) c is monotone neutral on SIS.
(iii) c is symmetric on SIS.
(iv) c satisfies the t-Condorcet condition.

Note added in proof. A minor modification of the proof of Propositions 20 and 28 will show that three additional conditions can be added to the list of equivalent conditions in Proposition 28:

(iii) *t is the least index of m_t.*
(iv) *t is the least index of α_t.*
(v) *m_t is consistent.*

REFERENCES

K. P. Arrow (1962), *Social Choice and Individual Values*, 2nd ed., John Wiley, New York (1st ed., 1951).

H. J. Bandelt (1982), private communication.

H. J. Bandelt and J. P. Barthélemy, *Medians in median graphs*, Discrete Appl. Math., 8, pp. 131–142.

H. J. Bandelt and J. Hedlíková (1983), *Median algebras*, Discrete Math., 45, pp. 1–30.

H. J. Bandelt and G. C. Meletiou (1990), *An algebraic setting for near-unanimity consensus*, Order, to appear.

H. J. Bandelt, M. F. Janowitz, and G. C. Meletiou (1990), *n-median semilattices*, preprint.

M. Barbut (1961), *Médiane, distributivité, éloignment*, repr. (1981) Math. Sci. Humaines, 70, pp. 5–31.

J. P. Barthélemy and A. S. Guenoche (1988), *Les arbres et le representation des proximités*, Masson, Paris.

J. P. Barthélemy, B. Leclerc, and B. Monjardet (1986), *On the use of ordered sets in problems of comparison and consensus of classifications*, J. Classification, 3, pp. 187–224.

J. P. Barthélemy and F. R. McMorris (1986), *The median procedure for n-trees*, J. Classification, 3, pp. 329–334.

J. P. Barthélemy and B. Monjardet (1981), *The median procedure in cluster analysis and social choice theory*, Math. Social Sci., 1, pp. 235–268.

D. J. Brown (1975), *Aggregation of Preferences*, Quart. J. Economics, 89, pp. 456–469.

P. Buneman (1971), *The recovery of trees from measures of dissimilarity*, in Mathematics in Archaeological and Historical Sciences, F. R. Hodgson, D. G. Kendall, and P. Tautu, eds., Edinburgh University Press, Edinburgh, pp. 387–395.

M. J. A. Condorcet (1785), *Essai sur l'application de l'analyse à la probabiltié des decision redues à la pluralité des voix*, Paris.

J. A. Ferejohn and P. C. Fishburn (1979), *Representations of Binary Rules by Generalized Decisiveness Structures*, J. Economic Theory, 21, pp. 28–45.

M. F. Janowitz (1984), *On the semilattice of weak orders of a set*, Math. Social Sci., 8, pp. 229–239.

——— (1991), *A converse to the Sholander embedding*, Discrete Appl. Math., to appear.

B. Leclerc (1984), *Efficient and binary consensus functions on transitively valued relations*, Math. Social Sci., 8, pp. 45–61.

——— (1985), *Les hierarchies de partiers et leur demi-trellis*, Math. Sci. Humaines, 89, pp. 5–34.

——— (1990), *Medians and majorities in semimodular lattices*, SIAM J. Discrete Math., 3, pp. 266–276.

A. Mas-Collel and H. Sonnenschein (1972), *General possibility theorems for group decision*, Rev. Economic Stud., 39, pp. 185–192.

B. G. Mirkin (1975), *On the problem of reconciling partitions*, in Quantitative Sociology. I, International Perspectives on Mathematical and Statistical Modeling, H. M. Bullock, A. Aganbegian, H. M. Borodkinm, R. Boudon, and V. Capecchi, eds., Academic Press, New York, pp. 441–449.

B. MONJARDET (1980), *Théorie et applications de la médiane dans les treillis distributifs finis*, Ann. Discrete Math., 9, pp. 87–91.

——— (1981), *Metrics on partially ordered sets, a survey*, Discrete Math., 35, pp. 173–184.

——— (1982), private communication.

——— (1990), *Arrowian characterizations of latticial federation consensus functions*, Math. Social Sci., 20, pp. 51–71.

D. A. NEUMANN AND V. T. NORTON (1986), *On lattice consensus methods*, J. Classification, 3, pp. 225–255.

A. RUBINSTEIN AND P. C. FISHBURN (1986), *Algebraic aggregation theory*, J. Economic Theory, 38, pp. 63–77.

M. SHOLANDER (1954), *Medians, lattices and trees*, Proc. Amer. Math. Soc., 5, pp. 808–812.

H. P. YOUNG (1974), *An axiomatization of Borda's rule*, J. Economic Theory, 9, pp. 43–59.

H. P. YOUNG AND A. LEVENGLICK (1978), *A consistent extension of Condorcet's election principle*, SIAM J. Appl. Math., 35, pp. 285–300.

SIAM J. DISC. MATH.
Vol. 4, No. 3, pp. 323-328, August 1991

A NEW APPROACH TO THE SERVER PROBLEM*

MAREK CHROBAK† AND LAWRENCE L. LARMORE†

Abstract. A new method for dealing with the server problem is proposed. The technique consists of embedding the given metric space M into a bigger metric space $\mathrm{cl}(M)$ called the *closure* of M, and allowing our servers to move in $\mathrm{cl}(M)$. How this technique can be applied to give a new optimal algorithm for two servers is shown.

Key words. algorithms, optimization problems, the server problem, competitive analysis

AMS(MOS) subject classification. 68Q25

1. Introduction. The k server problem can be formulated as follows: Let M be a metric space, in which we have k mobile servers that can occupy points of M. Initially, all servers are on some k specified points of M (called the *initial configuration*). At each time step we are given a request, specified by a location $r \in M$, and we have to choose which server to move to r to "serve" the request. Our measure of cost is the distance traveled by our servers, and the task is to design algorithms that minimize that cost.

The problem is that the requests have to be served on-line; that is, the choice of the server at the current step cannot depend on the future requests. It is known (see [2]) that if we were given the whole sequence of requests off-line, in advance, then an optimal schedule can be constructed efficiently in polynomial time. However, as it was shown in [6], no on-line algorithm can guarantee to yield a schedule that is better than k times the optimal one. Therefore, the on-line restriction is essential.

Recently, the research on on-line algorithms concentrates on so-called *competitive* algorithms. Let $\mathrm{cost}_{\mathrm{opt}}(K, \sigma)$ be the optimal cost of servicing the sequence of requests σ when the servers start from configuration K. By $\mathrm{cost}_{\mathcal{A}}(K, \sigma)$ we denote the respective cost of an on-line algorithm $\mathcal{A}$. An algorithm $\mathcal{A}$ is called *c-competitive* if for every initial configuration K there is a constant $b(K)$ such that for arbitrary sequence of requests σ we have

$$\mathrm{cost}_{\mathcal{A}}(K, \sigma) \leqq c \cdot \mathrm{cost}_{\mathrm{opt}}(K, \sigma) + b(K).$$

(We will often omit the parameters K and σ when they are understood.) In other words, for each initial configuration, the ratio $\mathrm{cost}_{\mathcal{A}}/\mathrm{cost}_{\mathrm{opt}}$ approaches c if $\mathrm{cost}_{\mathrm{opt}}$ is large.

It is not known whether there is an on-line algorithm that achieves $c = k$ for each k. The famous "k-server conjecture" of Manasse, McGeoch, and Sleator [6] states that this is indeed true. Up to now, it has been proven only for $k = 2$ in [6]. Irani and Rubinfeld [5] proved that a version of a balancing algorithm is 10-competitive for two servers. Some work has also been done on randomized algorithms. Raghavan and Snir [7] presented a randomized memoryless algorithm for two servers whose competitiveness constant is between 3 and 6. Berman, Karloff, and Tardos [1] proved that a similar algorithm is competitive for three servers, but the competitiveness constant is unknown.

In the general case, the solution is known only for some specific metric spaces. In [3] a k-competitive algorithm is given for trees, and it can be applied to all metric spaces that can be isometrically embedded in a tree (for example the weighted cache problem, see [2], [7]). Coppersmith et al. [4] gave a randomized k-competitive algorithm for a broad class of metric spaces that also includes trees.

* Received by the editors October 12, 1989; accepted for publication October 26, 1990.

† Department of Mathematics and Computer Science, University of California, Riverside, California 92521.

In this note we give a new on-line algorithm for two servers. Our algorithm is 2-competitive, and thus optimal. We employ some new techniques that we believe can be extended to three or more servers. Our method allows our servers to move in a bigger metric space $\mathrm{cl}(M)$, called the *closure* of M. All information about the past that the algorithm uses is recorded by the current positions of our servers in $\mathrm{cl}(M)$. This allows us to use a simple and intuitive potential argument in the proof of competitiveness.

2. The 2-server algorithm. Let M be the given metric space. For simplicity we assume that M is finite. At the end of this section, we describe how to implement the algorithm on infinite metric spaces. By $\|xy\|$ we denote the distance between points x, $y \in M$.

In the proof, we look at the computation as a game between our servers s_1, s_2, and the adversary's servers a_1 and a_2, and we compare our cost $\mathrm{cost}_{\mathscr{A}}$ to the adversary's cost $\mathrm{cost}_{\mathrm{adv}}$. Our goal is to show that, independently of the adversary's strategy, the inequality $\mathrm{cost}_{\mathscr{A}} \leqq c \cdot \mathrm{cost}_{\mathrm{adv}} + b$ holds. Each round is thought of as consisting of two steps: first the adversary moves a server and puts a request on its new position, and then our algorithm satisfies the request. Note that it is sufficient to show that $\mathrm{cost}_{\mathscr{A}} \leqq c \cdot \mathrm{cost}_{\mathrm{adv}} + b$ holds for arbitrary sequence of the adversary's moves, since one of the adversary schedules will correspond to the optimal schedule.

First we give an intuitive description of our method. In the algorithm, we visualize the computation as taking place in a bigger metric space $\mathrm{cl}(M)$ called the *closure* of M. Each point $u \in \mathrm{cl}(M)$ is defined by a set of distances between u and the points in M in such a way that the triangle inequality is preserved. We allow our servers to move through $\mathrm{cl}(M)$, while the requests and the adversary's servers are always in M. For simplicity, we also use notation $\|uv\|$ to denote the distance between u, $v \in \mathrm{cl}(M)$.

Our algorithm can be informally described as follows. Given a request on a point $r \in M$, we look at the points s_1, s_2 and r (for simplicity, s_i and a_i are also used to denote the current position of the corresponding server). If $\|rs_i\| + \|s_is_j\| = \|rs_j\|$, for $i \neq j$, then we move s_i to r. Otherwise, we do the following: Both servers move at the same speed toward the request, and simultaneously each of them moves at the same speed toward the other. Thus, if d is the speed of our servers (distance traveled in a unit of time), then after a unit of time both servers will get closer to the request by d, and closer to one another by $2d$. Eventually, one of our servers will be ahead of the other one, in the sense that the first case considered above will apply.

Formal description of the algorithm. By R^+ we denote the set of nonnegative reals. For a, b, $c \in R^+$, we define the predicate $\Delta(a, b, c)$ to mean that the numbers a, b, c satisfy the triangle inequalities: $a + b \geqq c$, $a + c \geqq b$, and $b + c \geqq a$. The *closure* of M is the set of all functions $u : M \rightarrow R^+$ such that for any x, $y \in M$, $\Delta(\|xy\|, u(x), u(y))$ holds. We set $\|uv\| = \max_{x \in M} |u(x) - v(x)|$, making $\mathrm{cl}(M)$ a metric space. M can be isometrically embedded into $\mathrm{cl}(M)$ by mapping each $x \in M$ to u_x, where $u_x(y) = \|xy\|$ for each $y \in M$. We will abuse notation by identifying x with u_x whenever convenient, allowing us to think of M as a subspace of $\mathrm{cl}(M)$.

Now we have to formalize the notion of moving a server, or both servers through $\mathrm{cl}(M)$. First we give some intuitive information. Suppose that the current location of s_i is u, and we want to move it towards v by some distance d. If $\|uv\| = e$, then for each $x \in M$ we have $|u(x) - v(x)| \leqq e$, and there is some $y \in M$ such that $|u(y) - v(y)| = e$. Suppose that d is very small. Then, in order to move s_i as needed, for each such y we can update $s_i(y)$ as follows. If $u(y) - v(y) = e$ then $s_i(y) \leftarrow s_i(y) - d$, and if $u(y) - v(y) = -e$ then $s_i(y) \leftarrow s_i(y) + d$. Since M is assumed to be finite, and since d is sufficiently small, all triangle inequalities will be preserved; that is, the new position of

s_i is a valid point in $\mathrm{cl}(M)$. And now the distance of s_i to v decreased by d and its distance to u is equal to d. If d is larger, then this movement can be divided into a sequence of steps as described above. At each consecutive step, the set of y's for which $s_i(y)$ has to be updated may vary. In a similar way, it is possible to define the movement of two servers in the first phase of the algorithm. In both cases, the whole process may be combined into one step. Since when one server moves, it always moves to a request point in M, this case is easy to formalize as one step. However, the first phase of our algorithm, when two servers move, ends when our servers may be in points that are not in M. Below we describe how those points can be determined.

We define now the function $\mathrm{step}_d(u, v, w) \in \mathrm{cl}(M)$, that determines the new position of a server whose current position is u and which is to be moved by d simultaneously towards v and w. For any $x \in M$, let $p(x)$ be the closest real number to $u(x)$ such that $|p(x) - v(x)| \leqq \|uv\| - d$ and $|p(x) - w(x)| \leqq \|uw\| - d$. Then we set $\mathrm{step}_d(u, v, w) = p$. In Lemma 1 below we show that the function step is well defined and satisfies the needed properties.

LEMMA 1. *Let* $u, v, w \in \mathrm{cl}(M)$, *and* $0 \leqq d \leqq \frac{1}{2}(\|uv\| + \|uw\| - \|vw\|)$. *Also let* $p = \mathrm{step}_d(u, v, w)$ *be defined as above. Then*:

(a) p *is a well-defined element of* $\mathrm{cl}(M)$,

(b) $\|pu\| = d$,

(c) $\|pv\| = \|uv\| - d$, *and* $\|pw\| = \|uw\| - d$.

(d) *Suppose that* $d = \frac{1}{2}(\|uv\| + \|uw\| - \|vw\|)$. *Then* $\|wp\| + \|pv\| = \|wv\|$.

Proof. (a) First we show that $p(x)$ is defined for each $x \in M$. Let I_x be the intersection of the closed interval of radius $\|uv\| - d$ centered at $v(x)$ and the closed interval of radius $\|uw\| - d$ centered at $w(x)$. I_x cannot be empty since the distance between those centers does not exceed the sum of those radii; i.e., $|v(x) - w(x)| \leqq \|vw\| \leqq \|uv\| + \|uw\| - 2d$, and in fact, I_x must contain a nonnegative real. Thus $p(x)$ exists. Write $I_x = [a_x, b_x]$. Next we show that $p \in \mathrm{cl}(M)$, i.e., $|p(x) - p(y)| \leqq \|xy\| \leqq p(x) + p(y)$ for all $x, y \in M$. Fix some $x, y \in M$.

CLAIM A: $|a_x - a_y| \leqq \|xy\|$ *and* $|b_x - b_y| \leqq \|xy\|$.

Without loss of generality, $a_x \leqq a_y$ and $a_x = v(x) - \|uv\| + d \geqq w(x) - \|uw\| + d$. If $a_y = v(y) - \|uv\| + d$, then $|a_x - a_y| = a_y - a_x = v(y) - v(x) \leqq \|xy\|$. Otherwise $a_y = w(y) - \|uw\| + d$, whence $|a_x - a_y| = a_y - a_x \leqq a_y - (w(x) + \|uw\| - d) \leqq w(y) - w(x) \leqq \|xy\|$. The second inequality can be proven in a similar way.

CLAIM B: $b_x + a_y \geqq \|xy\|$.

Without loss of generality, $a_y = v(y) - \|uv\| + d$. If $b_x = v(x) + \|uv\| - d$, then $b_x + a_y = v(x) + v(y) \geqq \|xy\|$. Otherwise, $b_x = w(x) + \|uw\| - d$, and then $b_x + a_y \geqq w(x) + \|uw\| + v(y) \geqq w(x) + w(y) \geqq \|xy\|$, using the definition of $\|uw\|$.

We now return to the proof of (a). We have established that p is well defined, but we need to show that $p \in \mathrm{cl}(M)$. Let $x, y \in M$. To show that $|p(x) - p(y)| \leqq \|xy\|$, it suffices (by symmetry) to show that $p(y) - p(x) \leqq \|xy\|$. If $p(x) \geqq u(x)$ and $p(y) \leqq u(y)$ we are done, since $p(y) - p(x) \leqq u(y) - u(x) \leqq \|xy\|$. If $p(x) < u(x)$, then $p(x) = b_x$ and $p(y) \leqq b_y$. By Claim A, $p(y) - p(x) \leqq b_y - b_x \leqq \|xy\|$. If $p(y) > u(y)$, then $p(y) = a_y$ and $p(x) \geqq a_x$. By Claim A, $p(y) - p(x) \leqq a_y - a_x \leqq \|xy\|$.

We now show that $p(x) + p(y) \geqq \|xy\|$. If $p(x) \geqq u(x)$ and $p(y) \geqq u(y)$, then we are done, since $p(x) + p(y) \geqq u(x) + u(y) \geqq \|xy\|$. If $p(x) < u(x)$, then $p(x) = b_x$ and $p(y) \geqq a_y$; so by Claim B, $p(x) + p(y) \geqq b_x + a_y \geqq \|xy\|$.

(b) We show first that $\|pu\| \leqq d$. Fix some $x \in M$. Without loss of generality assume that $p(x) \leqq u(x)$. Then $p(x) = b_x$. We can also assume that $p(x) = v(x) + \|uv\| - d$. Then $|p(x) - u(x)| = u(x) - p(x) = u(x) - v(x) - \|uv\| + d \leqq d$.

We now show that $\|pu\| \geqq d$. Pick $x \in M$ such that $|v(x) - u(x)| = \|uv\|$. Suppose $u(x) \leqq v(x)$. Then $u(x) = v(x) - \|uv\| \leqq a_x - d$; so $p(x) = a_x$ whence $p(x) - u(x) \geqq d$, hence $\|pu\| \geqq d$. The case $u(x) \geqq p(x)$ is proved similarly. Thus $\|pu\| = d$.

(c) By symmetry it is sufficient to prove only the first inequality. From the triangle inequality, $\|pv\| \geqq \|uv\| - \|pv\| = \|uv\| - d$, so we need only show that $\|pv\| \leqq \|uv\| - d$ by showing that $|p(x) - v(x)| \leqq \|uv\| - d$ for all $x \in M$. This holds since $p(x) \in I_x$, which is centered at $v(x)$ and has radius $\|uv\| - d$.

(d) The proof is by simple calculation, using (c) and the definition of d: $\|wp\| + \|pv\| = \|uw\| + \|uv\| - 2d = \|vw\|$. □

Algorithm $\mathscr{A}$. Let the request be on r. Let $d = \frac{1}{2} \min \{\|s_i s_j\| + \|s_i r\| - \|s_j r\|\}$ where the minimum is over the two choices of assignment of $\{i, j\}$ to $\{1, 2\}$. For the sake of exposition, it is convenient to present the move as if consisting of two phases, each possibly empty.

Phase 1:

(a) Move the first of our servers from s_1 to $s'_1 = \text{step}_d(s_1, s_2, r)$.

(b) Move the second of our servers from s_2 to $s'_2 = \text{step}_d(s_2, s'_1, r)$.

Phase 2:

If $\|s'_1 r\| \leqq \|s'_2 r\|$, move the first of our servers from s'_1 to r. Otherwise, move the other server from s'_2 to r.

Important remark. Recall that we visualize the algorithm in such a way that our servers move through $\text{cl}(M)$, and we can charge our servers their cost in $\text{cl}(M)$. In reality, however, our servers remain in M but only remember their virtual positions in $\text{cl}(M)$, and move to the request point only when they actually serve the request. Thus the cost we charge to our servers in $\text{cl}(M)$ may be different than the real cost. The adversary's servers are always on points of M (although we do not really need it for the proof, we could as well allow him to request points of $\text{cl}(M)$).

Let $\text{cost}'_{\mathscr{A}}$ be the cost of our servers in $\text{cl}(M)$. By the triangle inequality in $\text{cl}(M)$, and because $M \subseteq \text{cl}(M)$, we have the following fact.

FACT 1. $\text{cost}_{\mathscr{A}} \leqq \text{cost}'_{\mathscr{A}}$.

In order to prove that Algorithm $\mathscr{A}$ is 2-competitive, we define a potential function,

$$\Phi = 2\|M_{\min}\| + \|s_1 s_2\|,$$

where $M_{\min}$ is a minimum weight matching in the bipartite graph whose components are $\{s_1, s_2\}$, $\{a_1, a_2\}$, where the weights are equal to the distances between the servers. (Note that Φ is defined in $\text{cl}(M)$.) The same potential function was used in [4]. Recall now that we view the computation as a game, where each round consists of two steps: The adversary moves one of his servers to the request point, and then our algorithm serves the request. Consider a single round. Let $\Delta\text{cost}_{\text{adv}}$, $\Delta\text{cost}'_{\mathscr{A}}$ denote the change of cost_{adv} and $\text{cost}'_{\mathscr{A}}$ in this round. Let also $\Delta_{\text{adv}}\Phi$ and $\Delta_{\mathscr{A}}\Phi$ denote the changes of the potential, respectively, during the adversary's and our move.

LEMMA 2. $\Delta_{\text{adv}}\Phi \leqq 2 \cdot \Delta\text{cost}_{\text{adv}}$.

Proof. When the adversary moves, only one edge in $M_{\min}$ may change. If a_1 moves by d, then the distance from a_1 to its mate in $M_{\min}$ changes by at most d, so the change of the potential is at most $2d$. Therefore $\|M_{\min}\|$ cannot increase by more than $2d$. Note that the other matching may become minimum during the move, but if so, its weight cannot be greater than the one of $M_{\min}$. This implies the lemma. □

LEMMA 3. $\Delta_{\mathscr{A}}\Phi + \text{cost}'_{\mathscr{A}} \leqq 0$.

Proof. Consider the first phase, when both of our servers move by d. Suppose that the request is on a_1. By Lemma 1(c) one edge in the minimum matching gets shorter by d. But, by Lemma 1(b) and the triangle inequality, the length of the other cannot increase by more than d. Thus $\|M_{\min}\|$ cannot increase. But $\|s_1s_2\|$ drops by $2d$ by applying twice Lemma 1(c), and our cost is exactly $2d$.

In the second phase, assume that s_1 moves, that is, $\|a_1s_1\| + \|s_1s_2\| = \|a_1s_2\|$. Then $\|a_1s_1\| + \|a_2s_2\| \leqq \|a_1s_1\| + \|a_2s_1\| + \|s_1s_2\| = \|a_1s_2\| + \|a_2s_1\|$. This means that a_1 must be matched to s_1 in some minimum matching $M_{\min}$. Now, if s_1 moves by e, then $\|a_1s_1\|$ drops by e and $\|s_1s_2\|$ increases by e. Therefore, the potential must decrease by e, equal to our cost. □

From Lemmas 2 and 3, by a simple summation, and by application of Fact 1 we obtain the following theorem.

THEOREM 1. $\mathrm{cost}_{\mathscr{A}} \leqq 2\cdot\mathrm{cost}_{\mathrm{opt}} + \Phi_0$, *where* Φ_0 *is the initial potential. Therefore, Algorithm* $\mathscr{A}$ *is* 2-*competitive*.

Implementation. If M is finite and small, then the implementation is easy: After each round the server that is outside M remembers its position in $\mathrm{cl}(M)$. If $|M| = m$, this method uses $O(m)$ space and $O(m)$ time per request.

In cases when M is large, or even infinite, a more practical approach is to work only on the portion of M consisting of all the previous request points. If we are given n requests, this leads to a method that takes $O(n)$ space and $O(n)$ time per request.

We describe now the details of this implementation. Let x_0 be the initial point where the servers are located, and let x_t be the request point during the tth round, $t = 1, \cdots, n$. Denote by M_t the subspace of M induced by $x_0, \cdots, x_t$. Without loss of generality we can assume that $M = M_n$. We modify Algorithm 1 in such a way that at round t it works in $\mathrm{cl}(M_t)$ instead of $\mathrm{cl}(M)$, and when the new request x_{t+1} appears, $\mathrm{cl}(M_t)$ is embedded isometrically into $\mathrm{cl}(M_{t+1})$. Therefore, the whole computation can be in fact "embedded" into $\mathrm{cl}(M)$. This will assure that the cost we charge our servers is at least the real cost of the algorithm.

The embedding $\tau : \mathrm{cl}(M_t) \rightarrow \mathrm{cl}(M_{t+1})$ is defined as follows:

$$(\tau u)(x_i) = u(x_i), \qquad i = 0, \cdots, t,$$

$$(\tau u)(x_{t+1}) = \max_{i=0,\cdots,t} |u(x_i) - \|x_ix_{t+1}\||.$$

By a slight abuse of notation, we also refer to τu as u.

We need to show that τu is actually a member of $\mathrm{cl}(M_{t+1})$, i.e., that $\Delta(\|xy\|, u(x), u(y))$ holds for all $x, y \in M_{t+1}$. The only case we need to check is that $y = x_{t+1}$ itself and $x \in M_t$; that is, $|u(x) - u(x_{t+1})| \leqq \|xx_{t+1}\| \leqq u(x) + u(x_{t+1})$ for any $x \in M_t$. Now $u(x_{t+1}) \geqq |u(x) - \|xx_{t+1}\||$, which implies $\|xx_{t+1}\| \leqq u(x) + u(x_{t+1})$ and $u(x) - u(x_{t+1}) \leqq \|xx_{t+1}\|$. It remains only to show that $u(x_{t+1}) - u(x) \leqq \|xx_{t+1}\|$. Choose x_g such that $u(x_{t+1}) = |u(x_g) - \|x_gx_{t+1}\||$. If $u(x_{t+1}) = u(x_g) - \|x_gx_{t+1}\|$, then $u(x_{t+1}) = u(x_g) - \|x_gx_{t+1}\| \leqq u(x) + \|xx_g\| - \|x_gx_{t+1}\| \leqq u(x) + \|xx_{t+1}\|$, by two applications of the triangle inequality. On the other hand if $u(x_{t+1}) = \|x_gx_{t+1}\| - u(x_g)$, then $u(x_{t+1}) = \|x_gx_{t+1}\| - u(x_g) \leqq \|xx_{t+1}\| + \|xx_g\| - u(x_g) \leqq \|xx_{t+1}\| + u(x)$, again by two applications of the triangle inequality.

In order to prove that τ is indeed an isometry, it is sufficient to show that for any $u, v \in \mathrm{cl}(M_t)$ we have $|u(x_{t+1}) - v(x_{t+1})| \leqq \|uv\|$. Choose x_g such that $u(x_{t+1}) = |u(x_g) - \|x_gx_{t+1}\||$. Then $|u(x_{t+1}) - v(x_{t+1})| \leqq |u(x_g) - \|x_gx_{t+1}\|| - |v(y) - \|x_gx_{t+1}\|| \leqq |u(x_g) - v(x_g)| \leqq \|uv\|$ (because $|a - c| - |b - c| \leqq |a - b|$ for $a, b, c \in R$).

Acknowledgments. We would like to thank Allan Borodin, who noticed that our potential function can be expressed by the formula that was also used in [4]. This observation led to a simplification of the proof of Theorem 1. We also thank anonymous referees for insightful suggestions that helped us to improve the presentation of the paper.

REFERENCES

[1] P. Berman, H. Karloff, and G. Tardos, *A competitive algorithm for three servers*, in Proc. of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA, 1990, pp. 280–290.

[2] M. Chrobak, H. Karloff, T. Payne, and S. Vishwanathan, *New results on server problems*, in Proc. of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA, 1990, pp. 291–300.

[3] M. Chrobak and L. Larmore, *An optimal online algorithm for k servers on trees*, SIAM J. Comp., 20 (1991), pp. 144–148.

[4] D. Coppersmith, P. G. Doyle, P. Raghavan, and M. Snir, *Random walks on weighted graphs and applications to online algorithms*, in Proc. 22nd Annual ACM Symposium on Theory of Computing, Baltimore, MD, 1990, pp. 369–378.

[5] S. Irani and R. Rubinfeld, *A competitive 2-server algorithm*, manuscript.

[6] M. Manasse, L. A. McGeoch, and D. Sleator, *Competitive algorithms for server problems*, in Proc. 20th Annual ACM Symposium on Theory of Computing, Chicago, IL, 1988, pp. 322–333.

[7] P. Raghavan and M. Snir, *Memory versus randomization in online algorithms*, in 16th International Colloquium on Automata, Languages, and Programming, Stresa, Italy, Lecture Notes in Computer Science, Vol. 372, Springer-Verlag, Berlin, New York, 1989, pp. 687–703.

SIAM J. DISC. MATH.
Vol. 4, No. 3, pp. 329-341, August 1991

MIDPOINTS OF DIAGONALS OF CONVEX *n*-GONS*

PAUL ERDÖS†, PETER FISHBURN‡, AND ZOLTAN FÜREDI†

Abstract. Let $f(n)$ be the minimum over all convex planar n-gons of the number of different midpoints of the $\binom{n}{2}$ line segments, or diagonals, between distinct vertices. It is proved that $f(n)$ is between approximately $0.8\binom{n}{2}$ and $0.9\binom{n}{2}$. The upper bound uses the fact that the number of multiple midpoints, shared by two or more diagonals, can be as great as about $\binom{n}{2}/10$. Cases for which the number of midpoints is at least $\lceil n(n-2)/2 \rceil + 1$, the number for a regular n-gon when n is even, are noted.

Key words. convex n-gons, diagonal midpoints, multiple midpoints

AMS(MOS) subject classifications. 52A10, 52A25, 51M20

1. Introduction. Let M denote the set of midpoints of the $\binom{n}{2}$ line segments between distinct vertices of a convex n-gon in the plane. Let $f(n) = \min |M|$, taken over all convex n-gons. We prove that $f(n)$ is between about $0.40n^2$ and $0.45n^2$.

THEOREM 1. *For all* $n \geqq 3$,

$$\binom{n}{2} - \left\lfloor \frac{n(n+1)(1-e^{-1/2})}{4} \right\rfloor \leqq f(n) \leqq \binom{n}{2} - \left\lfloor \frac{n^2-2n+12}{20} \right\rfloor.$$

The lower bound proof, in § 2, is based in part on the following lemma.

PARALLELOGRAM LEMMA (Euclid). *Two finite crossing line segments in the plane have the same midpoint if and only if the ends of the segments are the vertices of a parallelogram.*

Section 2 also uses the notion of a multiple midpoint. Call a point in M *multiple* if it is the midpoint of two or more of the $\binom{n}{2}$ line segments between vertices of the convex n-gon. We let $\mathbf{M}$ denote the set of multiple midpoints.

Let $g(n) = \max |\mathbf{M}|$, taken over all convex n-gons. Clearly $f(n) + g(n) \leqq \binom{n}{2}$. The upper bound on $f(n)$ in Theorem 1 is a corollary of the following theorem.

THEOREM 2. *For all* $n \geqq 3$,

$$g(n) \geqq \left\lfloor \frac{n^2-2n+12}{20} \right\rfloor.$$

This quadratic lower bound on $g(n)$ is the largest lower bound presently known for $n \geqq 18$, but for most $n \leqq 17$ it is exceeded as follows:

n	5	6	7	8	9	10	11	12	13	14	15	16	17
$\left\lfloor \frac{n^2-2n+12}{20} \right\rfloor$	1	1	2	3	3	4	5	6	7	9	10	11	13
$g(n) \geqq$	1	2	3	3	4	5	6	8	9	10	11	13	14.

The construction for the improved lower bound on $g(n)$ is described in § 4.

* Received by the editors April 16, 1990; accepted for publication August 10, 1990. This research was supported by DIMACS (Center for Discrete Mathematics and Theoretical Computer Science), a National Science Foundation Science and Technology Center grant NSF-STC88-09648, and by Hungarian National Science Foundation grant 1817.

† Mathematical Institute, Hungarian Academy of Sciences, Budapest, Hungary.
‡ AT&T Bell Laboratories, Murray Hill, New Jersey 07974.

Section 5 concludes our study of M with remarks on $|M|$ when the number of multiple midpoints is small. Its main result, which includes all regular n-gons for even n, is the following theorem.

THEOREM 3. *If the number of multiple midpoints is less than* 3, *or if one vertex of the convex n-gon is an endpoint of diagonals whose midpoints include all multiple midpoints, then*

$$|M| \geqq \left\lceil \frac{n(n-2)}{2} \right\rceil + 1.$$

This inequality can fail when $|\mathbf{M}| = 3$.

We are not aware of previous contributions to the problems investigated here. Some time ago, Behrend [1] looked at sets of integers that contain no element midway between two others. More recently, Freiman [4], [5] obtained many results involving midpoints in additive number theory. One of these says that if $2 \leqq \lambda < 2^m$, $m \geqq 2$, then there is a constant $c_\lambda > 0$ such that every sufficiently large finite $X \subseteq \mathbb{R}^m$ whose points determine no more than $\lambda|X|$ midpoints has at least $c_\lambda|X|$ of its points in some hyperplane in $\mathbb{R}^m$. Fishburn [2], [3] gives an elementary proof of the planar version of Freiman's result and finds nearly best values of c_λ for $2 \leqq \lambda < 4$. The latter work uses results in the present paper.

2. Lower bounds on $f(n)$. Let $f(V) = |M|$ for a convex n-gon with vertex set V and nonempty multiple midpoint set $\mathbf{M}$. For each $\mu \in \mathbf{M}$ let

$$V(\mu) = \{x \in V : \mu = (x+y)/2 \text{ for some } y \in V\},$$

$$E(\mu) = \{\text{all diagonals with midpoint } \mu\},$$

$$D(\mu) = \{\text{all diagonals for } V(\mu) \text{ except those in } E(\mu)\}.$$

Thus $V(\mu)$ is the vertex set of $E(\mu)$, $|V(\mu)| = 2|E(\mu)|$, and $|D(\mu)| + |E(\mu)| = \binom{|V(\mu)|}{2}$. Let $\mu^* = |E(\mu)|$. Then

$$|D(\mu)| = 2\mu^*(\mu^* - 1).$$

Clearly, $E(\mu) \cap E(\lambda) = \varnothing$ when $\mu \neq \lambda$, μ, $\lambda \in \mathbf{M}$, and the same hypotheses and the Parallelogram Lemma are easily seen to imply $D(\mu) \cap D(\lambda) = \varnothing$. Obviously,

$$f(V) = \binom{n}{2} - \sum_{\mu \in \mathbf{M}} (\mu^* - 1).$$

We observe in passing that for $n \geqq 3$

$$f(V) \geqq \binom{n}{2} - \left\lfloor \frac{n^2 - 2n}{8} \right\rfloor$$

so that $f(n)$ is at least as great as about $(\frac{3}{8})n^2$. Observe that $\sum |D(\mu)| \leqq \binom{n}{2} - \lceil n/2 \rceil$ since for every $x \in V$ there is a $y \in V \setminus \{x\}$ across the n-gon from x such that $[x, y]$ is not a side of a parallelogram on four vertices of V. Therefore

$$\sum_{\mu \in \mathbf{M}} (\mu^* - 1) = \sum \frac{|D(\mu)|}{2\mu^*} \leqq \frac{1}{4} \sum |D(\mu)| \leqq \frac{1}{4}\left\{\binom{n}{2} - \lceil n/2 \rceil\right\},$$

and the given inequality for $f(V)$ follows from this and the concluding equality of the preceding paragraph.

The rest of this section is devoted to the better lower bound specified in the following lemma.

LEMMA 1.

$$f(V) \geqq \binom{n}{2} - \left\lfloor \frac{n(n+1)(1-e^{-1/2})}{4} \right\rfloor.$$

This shows that $f(n) > 0.4016n^2$ for all large n.

A new definition is needed. Let

$$C(x) = \{[y,z] \in E(\mu) : x \in V(\mu) \quad \text{and} \quad x \notin \{y,z\}\}$$

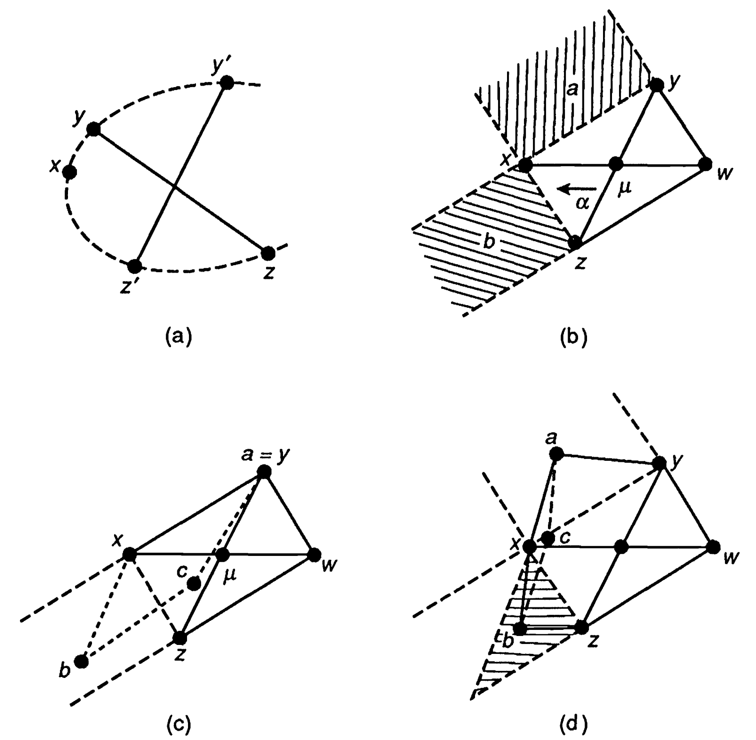
for each $x \in V$: see Fig. 1(a). The following result is central.

LEMMA 2. *Every two diagonals in $C(x)$ intersect in the interior of the n-gon.*

Proof. Suppose otherwise for $[y, z]$, $[a, b] \in C(x)$. Let μ be the midpoint of $[y, z]$ and of $[x, w]$, let α be the midpoint of $[a, b]$ and $[x, c]$, and suppose with no loss of generality that α lies in the x direction from $[y, z]$. Then a and b must lie in the three-sided dashed regions shown in Fig. 1(b), one in each region, or else convexity will be violated.

Assume that a is in the upper dashed region and b is in the lower dashed region. Suppose $a = y$: see Fig. 1(c). Then $b \neq z$ by our initial supposition, and since $[x, y]$, $[b, c]$, and $[z, w]$ are mutually parallel by the Parallelogram Lemma, we violate convexity. Therefore $a \neq y$. Similarly, $b \neq z$.

It follows that a and b are interior to their regions. Position a accordingly, anywhere in its region: see Fig. 1(d). Then convexity forces b to be interior to the shaded triangular



(a) (b)

(c) (d)

FIG. 1

region. But c, the fourth vertex of the parallelogram for α, will then lie in the interior of the hexagon with vertices $axbzwy$, which gives another violation of convexity. □

For each $v \in V$ let

$$c_v = |C(v)| = \sum_{\{\mu : v \in V(\mu)\}} (\mu^* - 1),$$

and for each diagonal $[x, y]$ of the n-gon define its "length" by

$$l(x,y) = 1 + \min \{\text{number of } V \text{ points on one side of } xy \text{ line},$$
$$\text{number of } V \text{ points on the other side of } xy \text{ line}\},$$

so that $1 \leqq l(x, y) \leqq \lfloor n/2 \rfloor$. If n is odd, there are n diagonals for each $l \in \{1, \cdots, (n-1)/2\}$; if n is even, there are n diagonals for each $l \in \{1, \cdots, (n-2)/2\}$ and $n/2$ diagonals with $l = n/2$. The following connection between c and l is immediate from Lemma 2.

COROLLARY 1. *$c_v \leqq l(x, y)$ for all $v \in V$ and all $[x, y] \in C(v)$.*

We now construct an $\binom{n}{2} \times n$ 0-1 matrix $A(V)$ that will be manipulated to yield the conclusion of Lemma 1. The $\binom{n}{2}$ rows of $A(V)$ are labeled by the diagonals in nonincreasing order of their l values: the final n rows have $l = 1$. The n columns of $A = A(V)$ are labeled by the vertices in nonincreasing order of their c_v. Write $i \twoheadrightarrow [x, y]$ when row i has label $[x, y]$, and $j \twoheadrightarrow v$ when column j has label v. We define A's entries by the following: when $i \twoheadrightarrow [x, y]$ and $j \twoheadrightarrow v$,

$$A_{ij} = \begin{cases} 1 & \text{if } [x,y] \in C(v), \\ 0 & \text{otherwise.} \end{cases}$$

When $j \twoheadrightarrow v$, $c_v = \sum_i A_{ij}$ and $\sum c_v = \sum_{\mathbf{M}} 2\mu^*(\mu^* - 1)$. Let $r_i = \sum_j A_{ij}$ for row i. When $i \twoheadrightarrow [x, y]$, $r_i = 0$ if $[x, y] \notin \cup_{\mathbf{M}} E(\mu)$, but if $[x, y] \in E(\mu)$ then

$$r_i = |\{v : [x,y] \in C(v)\}| = 2(\mu^* - 1).$$

Since μ^* rows have labels in $E(\mu)$,

$$\sum_{i=1}^{\binom{n}{2}} \frac{r_i}{r_i + 2} = \sum_{\mathbf{M}} \left[\frac{2(\mu^* - 1)}{2\mu^*}\right] \mu^* = \sum_{\mathbf{M}} (\mu^* - 1).$$

Therefore

$$f(V) = \binom{n}{2} - \sum_i \frac{r_i}{r_i + 2}.$$

Our lower bound on $f(V)$ is obtained from an upper bound on $\sum r_i/(r_i + 2)$.

Assume until later that n is odd. Then, by Corollary 1 and the nonincreasing order of rows by l,

$$\text{if } A_{ij} = 1 \quad \text{then } c_j \leqq \frac{n-1}{2} - \left\lfloor \frac{i-1}{n} \right\rfloor,$$

where c_j is c_v when $j \twoheadrightarrow v$.

Let $\mathscr{A}$ be the set of all $\binom{n}{2} \times n$ nonnegative integer matrices with column sums $c_1 \geqq c_2 \geqq \cdots \geqq c_n$, row sums $r_1, r_2, \cdots$, and

$$c_j \leqq \frac{n-1}{2} - \left\lfloor \frac{i-1}{n} \right\rfloor \quad \text{whenever entry } (i,j) \neq 0.$$

Clearly $A(V) \in \mathscr{A}$. Suppose $B \in \mathscr{A}$, $i < a, j < b$, and $B_{ib}B_{aj} > 0$. Let B' equal B except on $\{i, a\} \times \{j, b\}$, where

$$B'_{ij} = B_{ij} + 1,$$
$$B'_{ib} = B_{ib} - 1,$$
$$B'_{aj} = B_{aj} - 1,$$
$$B'_{ab} = B_{ab} + 1.$$

Then $B' \in \mathscr{A}$ since we have changed neither the column nor row sums, and in going to B' we need

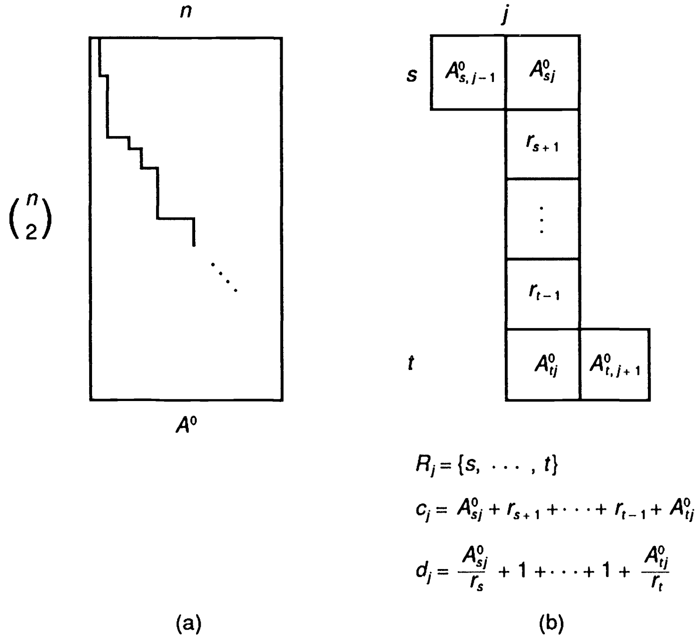
$$c_j \leqq \frac{n-1}{2} - \left\lfloor \frac{i-1}{n} \right\rfloor \quad \text{and} \quad c_b \leqq \frac{n-1}{2} - \left\lfloor \frac{a-1}{n} \right\rfloor.$$

The first of these is true since $i < a$ and for B we had $c_j \leqq (n-1)/2 - \lfloor (a-1)/n \rfloor$. The second is true since $j < b \Rightarrow c_b \leqq c_j$.

It follows from a finite sequence of switches as just described that $A(V)$ can be transformed into $A^0 \in \mathscr{A}$ so that no positive entry of A^0 is northeast or southwest of another positive entry. This implies that all positive entries of A^0 lie on a rectilinear staircase path as shown in Fig. 2(a). We suppose for convenience that all entries of A^0 on the path are positive: this is not needed for the desired conclusion, but it simplifies calculations by avoiding special notation that would continually refer to the set of all rows for which $r_i > 0$.

Let $W = \sum r_i/(r_i + 2)$ and let R_j be the set of rows i for which $A^0_{ij} > 0$. The staircase pattern gives $R_1 \leqq R_2 \leqq \cdots \leqq R_n$. Also let

$$W_j = \sum_{i \in R_j} \frac{A^0_{ij}}{r_i + 2} \qquad j = 1, \cdots, n.$$


n
$\binom{n}{2}$
A^0
j
s
$A^0_{s,j-1}$
A^0_{sj}
r_{s+1}
r_{t-1}
t
A^0_{tj}
$A^0_{t,j+1}$
$R_j = \{s, \cdots, t\}$
$c_j = A^0_{sj} + r_{s+1} + \cdots + r_{t-1} + A^0_{tj}$
$d_j = \frac{A^0_{sj}}{r_s} + 1 + \cdots + 1 + \frac{A^0_{tj}}{r_t}$

(a) (b)

FIG. 2

Then

$$W=\sum_i \frac{r_i}{r_i+2}=\sum_i\left(\sum_{j=1}^{n}\frac{A^0_{ij}}{r_i}\right)\frac{r_i}{r_i+2}$$

$$=\sum_{j=1}^{n}\sum_{i\in R_j}\frac{A^0_{ij}}{r_i+2}=\sum_j W_j.$$

We assign a fractional number of rows, d_j, to column j in which $A^0_{ij} > 0$, as follows:

$$d_j=\sum_{i\in R_j} A^0_{ij}/r_i.$$

If i is the first or last member of R_j, $0 < A^0_{ij}/r_i \leqq 1$, and if i is between the first and last members of R_j then $A^0_{ij}/r_i = 1$ since the r_i total for row i is all in column j: see Fig. 2(b).

Suppose $t = \max R_j$. Then

$$d_1+\cdots+d_j=t-1+\sum_{k\leqq j} A^0_{tk}/r_i\leqq t$$

and, by our earlier bound on c_j,

$$c_j\leqq\frac{n-1}{2}-\left\lfloor\frac{d_1+\cdots+d_j-1}{n}\right\rfloor.$$

LEMMA 3. *For all* j, $c_j/d_j\geqq 2$ *and*

$$W_j\leqq\frac{c_jd_j}{c_j+2d_j}.$$

Proof. For each paired piece of c_j and d_j as shown in Fig. 2(b), $A^0_{sj} \geqq (A^0_{sj}/r_s)2$, $r_{s+1} \geqq 1(2), \cdots$, so summation gives $c_j \geqq 2d_j$. When $|R_j| = m$, the inequality

$$W_j\leqq d_j\frac{c_j/d_j}{c_j/d_j+2}$$

can be put in the form

$$\sum_{k=1}^{m} p_k(\bar r-r_k)/(r_k+2)\geqq 0,$$

where $p_k > 0$, $\sum p_k = 1$ and $\bar r = \sum p_k r_k$. When multiplied by the product of the $(r_k + 2)$, this inequality becomes

$$\sum_{i<j} p_ip_j\left[\prod_{k\notin\{i,j\}}(r_k+2)\right](r_i-r_j)^2\geqq 0,$$

which is true. □

By Lemma 3, $\Sigma_{\mathrm{M}}(\mu^* - 1) = W \leqq \Sigma_j\, c_jd_j/(c_j + 2d_j)$. Denote the latter sum by $F(c, d)$, $c = (c_1, \cdots, c_n)$ and $d = (d_1, \cdots, d_n)$, and consider the following problem:

$$\text{maximize } F(c,d)=\sum_{j=1}^{n}\frac{c_jd_j}{c_j+2d_j}$$

subject to $c_1 \geqq c_2 \geqq \cdots \geqq c_n$ and, for $j = 1, \cdots, n$,

$$d_j > 0, \quad c_j/d_j \geqq 2, \quad c_j \leqq \frac{n-1}{2} - \left\lfloor \frac{d_1 + \cdots + d_j - 1}{n} \right\rfloor .$$

We replace the final constraint by the weaker but smooth $c_j \leqq (n+1)/2 - (d_1 + \cdots + d_j)/n$, observe that F increases in each c_j, and therefore take c_j as large as possible:

$$c_j^* = \frac{n+1}{2} - \frac{d_1 + \cdots + d_j}{n}.$$

Thus max $F(c, d) \leqq$ max $F(c^*, d)$ subject to $d_j > 0$ and $c_j^* \geqq 2d_j$.

LEMMA 4.

$$\max\ F(c^*, d) \leqq \frac{n(n+1)}{4}\left(1 - \frac{1}{\sqrt{e}}\right).$$

Proof. Let $c_0 = (n+1)/2$ and omit $*$ on c_j. Also let $x_j = c_j/d_j \geqq 2$. By the definition of c_j^* we have $c_j = [nx_j/(1 + nx_j)]c_{j-1}$. Therefore

$$c_j = c_0 \prod_{i=1}^{j} \frac{nx_i}{1 + nx_i},$$

$$d_j = c_0 \left(\prod_{i=1}^{j-1} \frac{nx_i}{1 + nx_i} \right) \frac{n}{1 + nx_j},$$

and

$$F(c, d) = c_0 \sum_{j=1}^{n} \left(\prod_{i=1}^{j} \frac{nx_i}{1 + nx_i} \right) \frac{1}{2 + x_j}$$

with each term in the sum $\leqq \frac{1}{4}$ since $x_j \geqq 2$. For $a, b > 0$

$$\left(\frac{na}{1+na}\right)\frac{1}{2+a} + \left(\frac{na}{1+na}\cdot\frac{nb}{1+nb}\right)\frac{1}{2+b} \geqq \left(\frac{nb}{1+nb}\right)\frac{1}{2+b} + \left(\frac{nb}{1+nb}\cdot\frac{na}{1+na}\right)\frac{1}{2+a}$$

if and only if $a \geqq b$. It follows that F is maximized when $x_1 \geqq x_2 \geqq \cdots \geqq x_n \geqq 2$, so assume the following.

Fix x_2 through x_n. Let $x = x_1$. Then

$$\frac{1}{c_0} F(c, d) = \left(\frac{nx}{1+nx}\right)\frac{1}{2+x} + \left(\frac{nx}{1+nx}\right)S,$$

where $S \leqq (n-1)/4$. Differentiation shows that the right-hand side decreases when

$$2 - nx^2 + S(2+x)^2 < 0,$$

which is true when $x \geqq 2 + 1/n$. We may therefore suppose that $x < 2 + 1/n$. But then S is much smaller than $n/4$, and the preceding inequality holds for all $x \geqq 2$. This implies that F is maximized at $x_1 = 2$, hence at $x_j = 2$ for all j, where

$$F = \frac{c_0}{4}\left[\frac{2n}{2n+1} + \left(\frac{2n}{2n+1}\right)^2 + \cdots + \left(\frac{2n}{2n+1}\right)^n\right]$$

$$= \frac{n(n+1)}{4}\left[1 - \left(\frac{2n}{2n+1}\right)^n\right] < \frac{n(n+1)}{4}\left(1 - \frac{1}{\sqrt{e}}\right). \qquad \square$$

Lemma 4 completes our proof of Lemma 1 when n is odd. When n is even, the preceding analysis is modified by replacing the bound on c_j obtained from Corollary 1 by

$$c_j \leqq \frac{n}{2} - \left\lfloor \frac{i+n/2-1}{n} \right\rfloor,$$

which corresponds to the remark on l for n even that precedes Corollary 1. Then c_j^* preceding Lemma 4 can be replaced by

$$c_j^* = \frac{n+1}{2} + \frac{1}{n} - \frac{d_1 + \cdots + d_j}{n}.$$

The only effect this has on the proof of Lemma 4 is to change c_0 there to $c_0 = (n+1)/2 + 1/n$. This changes the final equation in that proof to

$$F = \left[\frac{n(n+1)}{4} + \frac{1}{2}\right]\left[1 - \left(\frac{2n}{2n+1}\right)^n\right].$$

It is easily checked that this is less than $n(n+1)(1-e^{-1/2})/4$ when $n \geqq 10$. Therefore Lemma 4 holds for all $n \geqq 3$ except for $n \in \{4, 6, 8\}$. Lemma 1 claims for these three that $f(n=4) \geqq 5$, $f(n=6) \geqq 11$, and $f(n=8) \geqq 21$. Since $f(4) = 5$, $f(6) = 13$, and $f(8) \in \{24, 25\}$, Lemma 1 holds for all $n \geqq 3$.

3. Lower bound on $g(n)$.

THEOREM 2.

$$g(n) \geqq \left\lfloor \frac{(n^2-2n+12)}{20} \right\rfloor \quad \textit{for } n \geqq 3.$$

Proof. Let $a_k = k^2$ for $k = 4, 5, \cdots, 3m-1$ and $b_k = 3k^2$ for $k = 1, 2, \cdots, m$ with $m \geqq 2$. Take $N > 12m^2$ and construct the convex $(4m-4)$-gon that has m lower-left vertices $(-k, b_k)$ for $k = 1, \cdots, m$, and $3m-4$ upper-right vertices $(k, N-a_k)$ for $k = 4, \cdots, 3m-1$. For every $1 \leqq i < j \leqq m$ it is easily checked that

$$(*) \qquad b_j - b_i = a_{i+2j} - a_{2i+j}.$$

Since $j - i = (i+2j) - (2i+j)$, it follows that $[(-j, b_j), (i+2j, N-a_{i+2j})]$ and $[(-i, b_i), (2i+j, N-a_{2i+j})]$ have the common midpoint

$$c_{ij} = ((i+j)/2, (N-i^2-j^2-4ij)/2).$$

Moreover, if $i \neq k$, $i < j$, $k < l$, and $i + j = k + l$, then the vertical components of c_{ij} and c_{kl} are distinct. Therefore every multiple midpoint c_{ij} is distinct, so

$$g(4m-4) \geqq \binom{m}{2}.$$

The lower bound ratio of $g(n)/n^2$ in this case is approximately $(m^2/2)/(4m)^2 = 1/32$.

We get a larger ratio by deleting vertices at both ends of the upper-right part of the construction since the a_k pairs that match the b_k pairs as in $(*)$ are denser in the middle of the a_k sequence. A crude calculation for the quadratic terms shows that if we delete K vertices at each end of the a_k sequence then we lose about $K^2/3$ of the c_{ij}. This also removes $2K$ points from n, so the new ratio for $g(n)/n^2$ is approximately

$$\frac{m^2/2 - K^2/3}{(4m-2K)^2},$$

which is maximized through differentiation with respect to K at $K = 3m/4$, where the ratio is $1/20$.

To be more precise, suppose T vertices are removed from the a_k sequence, $\lfloor T/2 \rfloor$ at one end and $\lceil T/2 \rceil$ at the other end. Then, with details omitted, we get

$$g(4m-4-T) \geqq \binom{m}{2} - \left\lceil \frac{T}{6} \right\rceil \left(T+3-3\left\lceil \frac{T}{6} \right\rceil \right).$$

Given n, we then consider the (m, T) pairs that satisfy $n = 4m - 4 - T$ to determine the pair that maximizes the right-hand side of the preceding inequality. Further calculations show that the maximum is $\lfloor (n^2 - 2n + 12)/20 \rfloor$, as claimed in Theorem 2. □

4. Another construction for $g(n)$. The a_k and b_k of the preceding proof were chosen in an attempt to minimize n, given that each of the $\binom{m}{2}$ pairs from the lower left is matched by a pair from the upper right to yield a different multiple midpoint. We examined variations to this construction, but their lower bounds on $g(n)/n^2$ were smaller than $1/20$.

However, as mentioned earlier, a different construction gives larger lower bounds on $g(n)$ for most $n \leqq 17$. This other construction yields a lower bound on $g(n)/n$ of approximately $7/6$ for large n, as compared to $n/20$ for the quadratic construction used to prove Theorem 2, and is therefore much less powerful than the quadratic bound for large n.

Figure 3 illustrates the other construction that yields the largest lower bounds on $g(n)$ for small n that are presently known. Its 18 vertices are numbered in the order in which they enter the construction. We begin with the tall narrow rectangle for vertices

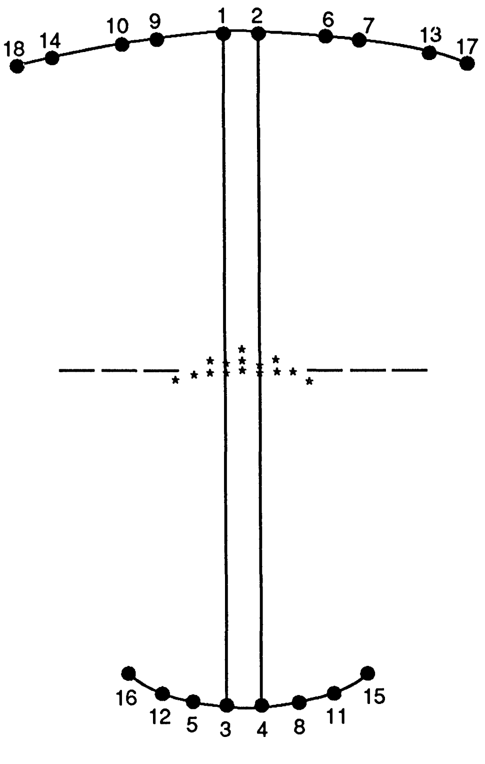


FIG. 3. *'s denote multiple midpoints.

1 through 4, then position 5 to the left of 3 so that the horizontal distances from 5 to 3 and from 3 to 4 are equal, with 5 slightly above the line through 3 and 4.

The other points are then positioned by midpoint restrictions and symmetry. Let $\mu(i,j)$ denote the midpoint between i and j. A complete account of multiple midpoints is shown in the following construction routine:

$\mu(1,4)=\mu(2,3)$
5: position as described above
6: $\mu(5,6)=\mu(2,4)$
7: $\mu(5,7)=\mu(3,6)$
8: position horizontally symmetric to 5
9: $\mu(8,9)=\mu(1,3)$
10: $\mu(8,10)=\mu(4,9)$
11: $\mu(10,11)=\mu(2,5)$
12: $\mu(7,12)=\mu(1,8)$ and $\mu(6,12)=\mu(9,11)$
13: $\mu(12,13)=\mu(3,7)$
14: $\mu(11,14)=\mu(4,10)$
15: $\mu(14,15)=\mu(2,12)$
16: $\mu(13,16)=\mu(1,11)$ and $\mu(7,16)=\mu(10,15)$
17: $\mu(12,17)=\mu(5,13)$
18: $\mu(11,18)=\mu(8,14)$.

Each μ equation here identifies a different multiple midpoint. The lower bound on $g(n)$ is the number of μ equations in place after point n is added.

The preceding construction shows that every vertex can be at the end of two or more diagonals whose midpoints are in $\mathbf{M}$. This occurs for the first time in the construction at $n=12$. Moreover, by Theorem 2, the smallest n presently known for which $g(n)>n$ is $n=23$. We do not know whether a smaller n suffices in either case.

5. Small numbers of multiple midpoints. This section focuses on situations that force $|M|$ to be much larger than the upper bound on $f(n)$ in Theorem 1. For each $\mu\in M$ let $V(\mu)$ be the set of all vertices at ends of diagonals that have midpoint μ. If $\mu\in\mathbf{M}$ then $|V(\mu)|\in\{4,6,8,\cdots\}$.

The following lemma is an easy consequence of the Parallelogram Lemma.

LEMMA 5. *If $\mu\in\mathbf{M}$ then all midpoints of line segments between points in $V(\mu)$ that differ from μ are different from each other. If $\lambda,\mu\in\mathbf{M}$, $\lambda\neq\mu$, and L is the line through λ and μ, then $L\cap V(\lambda)\cap V(\mu)=\varnothing$, $|V(\lambda)\cap V(\mu)|\leqq 2$ and, if $|V(\lambda)\cap V(\mu)|=2$, then λ or μ is the midpoint between the points in $V(\lambda)\cap V(\mu)$.*

Let R_n be a regular n-gon for even n. It follows immediately from the first part of Lemma 5 that

$$|M(R_n)|=\binom{n}{2}-\left(\frac{n}{2}-1\right)=\frac{n(n-2)}{2}+1.$$

An easy proof also shows that the maximum number of parallelograms that can be formed from the vertices of a convex n-gon for even n occurs at R_n and equals $n(n-2)/4$. The two diagonals of each parallelogram cross at the one multiple midpoint of R_n.

The initial observation in the preceding paragraph generalizes to the following theorem.

THEOREM 3. *If either* $|\mathbf{M}| \leqq 2$ *or* $\bigcap_{\mathbf{M}} V(\mu) \neq \varnothing$ *then*

$$|M| \geqq \left\lceil \frac{n(n-2)}{2} \right\rceil + 1.$$

The conclusion can fail if $|\mathbf{M}| = 3$.

Figure 4 verifies the final statement of the theorem. With $n = 11$ and $\mathbf{M} = \{\mu_1, \mu_2, \mu_3\}$, we have

$$V(\mu_1) = \{1,2,3,6,8,9\},$$
$$V(\mu_2) = \{2,4,6,11\},$$
$$V(\mu_3) = \{3,4,5,7,10,11\}$$

so that $V(\mu_1) \cap V(\mu_2) = \{2, 6\}$, $V(\mu_2) \cap V(\mu_3) = \{4, 11\}$, and $V(\mu_1) \cap V(\mu_3) = \{3\}$. Starting with all diagonals in place, we must remove five (two for μ_1, one for μ_2, two for μ_3) to have no multiple midpoint, so $|M| = \binom{11}{2} - 5 = 50$. On the other hand, $\lceil (11)(9)/2 \rceil + 1 = 51$.

The conclusion of Theorem 3 for $|\mathbf{M}| \leqq 2$ follows from Lemma 5. For example, if $\mathbf{M} = \{\lambda, \mu\}$ and $|V(\lambda) \cap V(\mu)| = 2$, then with $2c_\lambda = |V(\lambda)|$ and $2c_\mu = |V(\mu)|$ we have $2(c_\lambda + c_\mu) \leqq n + 2$. The removal of $(c_\lambda - 1) + (c_\mu - 1)$ diagonals gives a configuration in which no two remaining diagonals have the same midpoint, so

$$|M| \geqq \binom{n}{2} - (c_\lambda + c_\mu - 2) \geqq \binom{n}{2} + 2 - \left\lfloor \frac{n+2}{2} \right\rfloor$$
$$= \binom{n}{2} + 1 - \left\lfloor \frac{n}{2} \right\rfloor = \left\lceil \frac{n(n-2)}{2} \right\rceil + 1.$$

The following lemma is needed for the $\bigcap V(\mu) \neq \varnothing$ part of Theorem 3.

LEMMA 6. *Suppose* $|\mathbf{M}| = t \geqq 3$ *and* $\bigcap_{\mathbf{M}} V(\mu) \neq \varnothing$. *Let* α_k *be the number of vertices in exactly* k *of the* $V(\mu)$ *for* $\mu \in \mathbf{M}$. *Then* $\alpha_t = 1$, $\alpha_k = 0$ *for* $2 < k < t$, *and* $\alpha_2 \leqq t - 1$.

Proof. Take $x \in \bigcap_{\mathbf{M}} V(\mu)$ and $\mathbf{M} = \{\mu_1, \mu_2, \cdots, \mu_t\}$ as shown in Fig. 5(a). If vertex $y \neq x$ is in at least two $V(\mu)$, say $V(\mu_i)$ and $V(\mu_j)$, then Lemma 5 requires $y \in \{v_i, v_j\}$. It follows that $\alpha_t = 1$ and $\alpha_k = 0$ for all $2 < k < t$.

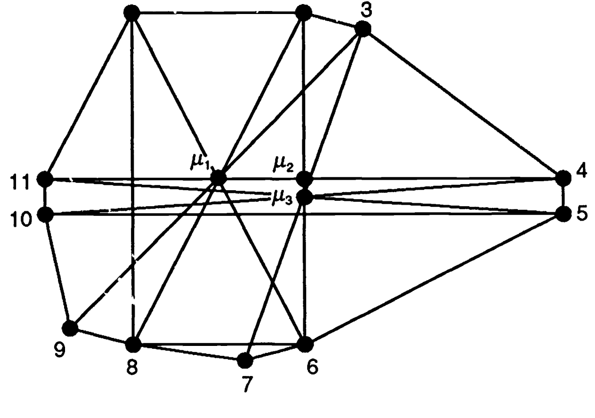
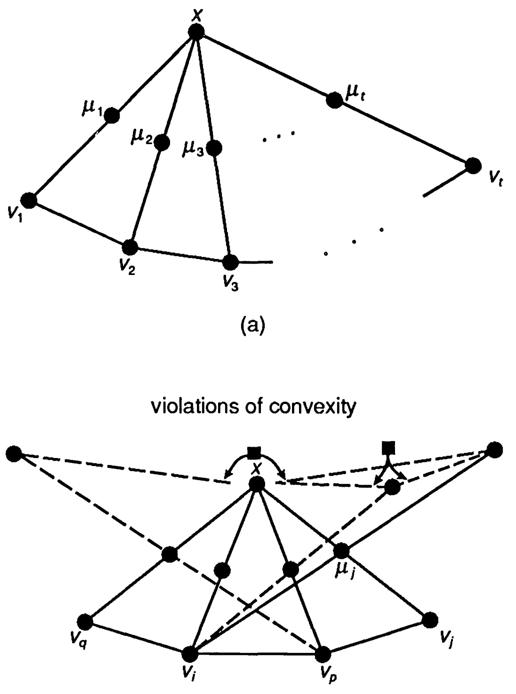


FIG. 4



(a)

(b)

FIG. 5

Lemma 5 and convexity imply the following: see Fig. 5(b).

Rule 1. $v_i \in V(\mu_j)$ for at most one $j \neq i$.

Rule 2. $[v_i \in V(\mu_j),\, i < j] \Rightarrow v_p \notin V(\mu_q)$ if $q \leqq i < p \leqq j$.

Rule 1 implies $\alpha_2 \leqq t$, and it follows easily from Rule 2 and its dual for $j < i$ that $\alpha_2 \leqq 2$ when $t = 3$. We use induction on t in what follows.

Suppose $\alpha_2 \leqq t - 1$ for $t = 3, \cdots, r - 1$ with $r \geqq 4$. Contrary to the lemma, suppose $\alpha_2 = r$ when $|\mathbf{M}| = r$. Suppose then that $v_i \in V(\mu_r)$ for some $i < r$. By Rule 2, $v_p \notin V(\mu_q)$ when $q \leqq i < p \leqq r$. Since $\alpha_2 = r$ and Rule 1 require every v_p to be in a $V(\mu_q)$, $q \neq p$, it follows that each of v_{i+1} through v_r is in one of $V(\mu_{i+1})$ through $V(\mu_r)$ that has a different index. But this is impossible by the induction hypothesis if $i < r - 2$, by Lemma 2 if $i = r - 2$, and by definition if $i = r - 1$. Hence no v_i for $i < r$ is in $V(\mu_r)$, a contradiction. Therefore $\alpha_2 \leqq r - 1$. □

Proof Completion (Theorem 3). Let $|\mathbf{M}| = t \geqq 3$ with $\mathbf{M} = \{\mu_1, \cdots, \mu_t\}$ and $|V(\mu_k)| = 2c_k$. Suppose $\cap_{\mathbf{M}} V(\mu) \neq \emptyset$. Since $(\alpha_2, \cdots, \alpha_t) \leqq (t - 1, 0, \cdots, 0, 1)$ by Lemma 6, $\sum (2c_k) \leqq n + 2(t - 1)$. Excision of $\sum (c_k - 1)$ diagonals gives a configuration in which no remaining diagonals have the same midpoint. A calculation similar to that preceding Lemma 6 yields $|M| \geqq \lceil n(n-2)/2 \rceil + 1$. □

6. Discussion. We have shown that $f(n)$ lies between about $0.8\binom{n}{2}$ and $0.9\binom{n}{2}$, that $|\mathbf{M}|$ can be as large as about $0.1\binom{n}{2}$, and that if either $|\mathbf{M}| \leqq 2$ or if some vertex lies at the ends of diagonals whose midpoints cover $\mathbf{M}$, then the corresponding n-gon has $|M| \geqq \lceil n(n-2)/2 \rceil + 1$.

Several open problems, in addition to exact values of $f(n)$ and $g(n)$, are suggested by our study. Does $\lim f(n)/n^2$ exist and, if so, what is its value? We ask a similar question for $g(n)/n^2$. Let $\mathbf{M}_3$ denote the set of all midpoints shared in common by at least three diagonals, and let $h(n) = \max |\mathbf{M}_3|$ over all convex n-gons. Does there exist $c > 0$ such that $h(n) > cn^2$ for all large n? If so, does a similar conclusion hold for midpoints with multiplicities that exceed 3?

REFERENCES

[1] F. A. BEHREND, *On sets of integers which contain no three terms in arithmetical progression*, Proc. Nat. Acad. Sci. USA, 32 (1946), pp. 331–333.

[2] P. C. FISHBURN, *On a contribution of Freiman to additive number theory*, J. Number Theory, 35 (1990), pp. 325–334.

[3] ———, *Sum set cardinalities of line restricted planar sets*, AT&T Bell Laboratories, Murray Hill, NJ, 1989.

[4] G. A. FREIMAN, *Foundations of a Structural Theory of Set Addition*, Vol. 37, Transl. Math. Monographs, American Mathematical Society, Providence, RI, 1973.

[5] ———, *What is the structure of K if K + K is small*? in Number Theory, New York, 1984–1985, Lecture Notes in Math., 1240, Springer-Verlag, New York, 1987, pp. 109–134.

SIAM J. DISC. MATH.
Vol. 4, No. 3, pp. 342–354, August 1991

004

NC ALGORITHMS FOR RECOGNIZING PARTIAL 2-TREES AND 3-TREES*

DANIEL GRANOT† AND DARKO SKORIN-KAPOV‡

Abstract. The existence of a k-separator in a partial k-tree graph is proved and a linear time algorithm is constructed that finds such a separator in k-trees. This algorithm can be used to obtain a balanced binary decomposition of a k-tree in $O(n \log n)$ time. Some other separation properties of partial k-trees are derived and used to construct a balanced decomposition of an embedding of a k-connected partial k-tree when $k = 2$, 3. Finally, NC algorithms are constructed for the recognition of a partial k-tree for $k = 2$, 3. For $k = 2$ and $k = 3$ these algorithms run in $O(\log^2 n)$ time using, respectively, $O(n^3)$ and $O(n^4)$ processors. Thus, the algorithms for $k = 2$, 3 improve considerably the processor bound of Chandrasekharan and Hedetniemi [*Proceedings of the* 26*th Annual Allerton Conference on Communication, Control and Computing*, 1989, pp. 283–292] general algorithm for the parallel recognition of partial k-trees that would require $O(\log n)$ time and, respectively, $O(n^{10})$ and $O(n^{12})$ processors in these cases.

Key words. k-tree, partial k-tree, k-separator, parallel algorithm

AMS(MOS) subject classifications. 05C75, 68Q10, 05C05

1. Introduction.

The study of the class of k-trees and their partial graphs was motivated by some practical questions concerning the reliability of communication networks in the presence of constrained line-and-site failures (Farley [10], Farley and Proskurowski [11], Neufeld and Colbourn [15], Wald and Colbourn [20]), and in view of their relevance in modeling the complexity of queries in data base systems (Arnborg [1]). Also, the class of k-trees is special in the sense that many problems that are NP-complete for arbitrary graphs were shown to be solvable in polynomial time when restricted to this class of graphs; see, for example, Arnborg and Proskurowski [5], Bodlaender [6], and Granot and Skorin-Kapov [12].

Arnborg, Corneil, and Proskurowski [2] have shown that the problem of finding the smallest number k, such that a given graph is a partial k-tree, is NP-complete. However, they also presented therein an $O(n^{k+2})$ time algorithm for the recognition of a partial k-tree when k is fixed. Robertson and Seymour [16] proved that there exists an $O(n^2)$ algorithm to recognize partial k-trees, and recently Bodlaender [6] developed an algorithm for recognizing and embedding a partial k-tree, for a fixed k, in $O(n^2)$ time. Wald and Colbourn [20] and Matousek and Thomas [14] have, respectively, constructed linear time sequential algorithms for the recognition of partial 2-trees and partial 3-trees. Finally, Bodlaender [6] and Chandrasekharan and Hedetniemi [9] have constructed NC-algorithms for the recognition of graphs that have a tree width less than k (or, equivalently, partial k-trees), which run in $O(\log n)$ time and use, respectively, $O(n^{3k+4})$ and $O(n^{2k+6})$ processors.

In this paper we introduce the notion of a *k-separator* ($k \geqq 2$) of an n-vertex graph $G = (V, E)$. Formally, a k-separator is a set of vertices S, $|S| = k$, which induces a partition of $V \backslash S$ into sets V_1 and V_2 satisfying (i) $|V_i| \leqq (k/k+1)n$, $i = 1, 2$, and (ii) no edge in E connects a vertex in V_1 with a vertex in V_2. We prove the existence of a k-

* Received by the editors April 27, 1989; accepted for publication (in revised form) July 13, 1990. This research was supported by Natural Sciences and Engineering Research Council of Canada grant A4181.

† Faculty of Commerce and Business Administration, The University of British Columbia, Vancouver, British Columbia, Canada.

‡ W. A. Harriman School of Management and Policy, State University of New York at Stony Brook, Stony Brook, New York.

separator for partial k-trees and construct a linear time algorithm for generating such a separator in k-trees. This algorithm can be used to construct a balanced binary decomposition tree of a k-tree in $O(n \log n)$ time. We further derive other separation properties of partial k-trees that are used to construct a balanced decomposition of an embedding of a k-connected partial k-tree into a k-tree when $k = 2, 3$. Finally, we develop parallel algorithms for the recognition of partial k-trees when $k = 2, 3$. These algorithms require $O(\log^2 n)$ time and use, respectively, $O(n^3)$ and $O(n^4)$ processors. For $k = 2, 3$ our algorithms improve considerably the processor bound of Chandrasekharan and Hedetniemi's [9] general algorithm for the recognition of partial k-trees, which would require, respectively, $O(n^{10})$ and $O(n^{12})$ processors for these cases.

The paper is organized as follows. In § 2 we present basic definitions and preliminary results. In § 3 we study k-separators of partial k-trees and describe an $O(n \log n)$ algorithm for constructing a balanced binary decomposition tree for k-trees. In § 4 we obtain additional separation properties of partial k-trees and show that they can be used to construct a binary balanced decomposition of an embedding of a k-connected partial k-tree when $k = 2, 3$. In § 5 we develop NC algorithms for the recognition and embedding of 2-connected (respectively, 3-connected) partial 2-trees (respectively, 3-trees), which could be extended for recognition of partial 2-trees that are not necessarily 2-connected. In § 6 we develop an algorithm for recognizing partial 3-trees that are not necessarily 3-connected.

2. Definitions and preliminaries. A *graph* G with vertex set V and edge set E will be denoted $G = (V, E)$. A *subgraph* of $G = (V, E)$ is a graph $G' = (V', E')$ such that $V' \subseteq V$ and $E' \subseteq E$. A *partial graph* of G is a subgraph of G containing all the vertices of G and a subset of its edges. We denote by $G(V')$, $V' \subseteq V$, a subgraph of G induced by V'. A *k-clique* is defined to be a complete graph on k vertices (it is not a clique in the standard terminology, i.e., a maximal completely connected subgraph). A set of vertices $S \subseteq V$ is a *separator* of $G = (V, E)$ if the subgraph $G(V \backslash S)$ induced by $V \backslash S$ has two or more connected components, and such a separator S is said to be minimal if no subset thereof is a separator of G. A graph $G = (V, E)$ is called *k-connected* if the cardinality of any minimal separator of G is at least k.

k-trees can be defined as follows. The k-clique is a k-tree, and a k-tree of more than k-vertices can be constructed by adding a new vertex and new edges connecting it to all vertices of some k-clique of a smaller k-tree. *Partial k-trees* are partial graphs of k-trees. A *contraction* of an edge $e = (u, v)$ in $G = (V, E)$ is obtained by removing $e = (u, v)$ from E and identifying the vertices u and v. A graph G is *contractible* to a graph G' if G' can be obtained from G by a sequence of edge contractions in G. Edge *extraction* of e in G results in a graph $G \backslash e$ with the same vertex set as G and the edge set $E \backslash \{e\}$. A graph H is a *minor* of a graph G if it can be obtained from G by a finite number of contractions and edge extraction operations. It follows from Arnborg, Proskurowski, and Corneil [3] that every minor of a partial k-tree is a partial k-tree. Thus, a subgraph of a partial k-tree is a partial k-tree.

For any vertex $v \in V$, the *neighborhood* of v is defined as the set of all vertices adjacent to v. The degree of a vertex v is the cardinality of its neighborhood. A *k-leaf* of a k-tree is a vertex v whose degree is equal to k. Every k-tree has at least two k-leaves, and we denote by $\mathrm{Adj}(v)$ the neighborhood of a k-leaf v. Note that the subgraph of G induced by $\mathrm{Adj}(v)$ is a k-clique. The graph obtained by removing a k-leaf v and its incident edges from a k-tree is a k-tree itself. This defines a *reduction process* for k-trees, and such a reduction process is complete when it ends up with some k-clique, R, the *root* of the reduction process.

A reduction sequence can be thought of as giving on orientation to a k-tree (Arnborg and Proskurowski [4]), where vertices are made descendants of k-cliques. Namely, if K is a k-clique, then v is a descendant of K in a given reduction sequence if and only if, when v was removed, each vertex of Adj(v) was either a member of K or a descendant of it.

3. *k*-separators. Let $G = (V, E)$ be a partial k-tree ($k \geqq 2$) with $|V| = n$. A *k-separator* of G is a set of vertices S, of cardinality $|S| = k$, which induces a partition $\{V_1, V_2\}$ of $V \backslash S$ satisfying

$$(1) \qquad |V_i| \leqq \frac{k}{k+1} n, \qquad i = 1, 2,$$

and

(2) no edge in E connects a vertex in V_1 with a vertex in V_2.

Observe that a k-separator extends the notion of a 2-separator defined for series parallel graphs (i.e., partial 2-trees), see, e.g., Hassin and Tamir [13].

THEOREM 3.1. *Every partial k-tree contains a k-separator. Moreover, for a k-tree, a k-separator can be constructed in linear time.*

Proof. Since any k-separator of a k-tree $G = (V, E)$ is also a k-separator of every partial graph of G, it is sufficient to prove the existence of a k-separator for k-trees. For that purpose, we construct below a linear time algorithm for finding a k-separator of a k-tree G. The algorithm follows a reduction process until reaching, for the first time, a k-leaf vertex v for which the corresponding set of descendant vertices, $D(K_v)$, of the k-clique K_v induced by Adj(v), contains at least $\frac{2}{3}n$ distinct vertices.

Formally, the algorithm has two steps.

Step 1. We initially set $D(A) = \phi$ for all k-cliques A of G. In general, let v be a k-leaf that is currently being considered for elimination in the reduction process. Let $V(K_v) = \text{Adj}(v) = \{u_1, \cdots, u_k\}$, and $V(\bar{K}_j) = \text{Adj}(v) \cup \{v\} \backslash \{u_j\}$, $j = 1, \cdots, k$, where $V(A)$ denotes the vertex set of a graph A. Then update $D(K_v)$, the set of descendant nodes of K_v, and $|D(K_v)|$ as follows:

$$(3) \qquad D(K_v) = \left(\bigcup_{j=1}^{k} D(\bar{K}_j) \right) \cup D(K_v) \cup \{v\},$$

$$(4) \qquad |D(K_v)| = \sum_{j=1}^{k} |D(\bar{K}_j)| + |D(K_v)| + 1.$$

(Observe that the sets $D(\bar{K}_j)$ in (3) and (4) are mutually disjoint since all k-cliques $\bar{K}_j$, $j = 1, \cdots, k$, are separators and $u_j \notin D(\bar{K}_j)$, $j = 1, \cdots, k$.) If $|D(K_v)| < \frac{2}{3}n$, we continue with the reduction process; otherwise, we go to Step 2, where the k-separator is produced. Observe that Step 1 will terminate whenever $n \geqq 3k$. If $n < 3k$ then every minimal separator of G is a k-separator.

Step 2. We need to distinguish between two cases.

Case 1. $\sum_{j=1}^{k} |D(\bar{K}_j)| + 1 \geqq \frac{2}{3}n$.

Let $V_j = D(\bar{K}_j)$, $j = 1, \cdots, k$, and $V_{k+1} = V \backslash \{(\cup_{j=1}^{k} V_j) \cup \text{Adj}(v) \cup \{v\}\}$, and let V_l be such that $|V_l| = \max_{j=1,\cdots,k+1} |V_j|$. We will prove that the set of vertices S,

$$S = \begin{cases} \text{Adj}(v) & \text{if } l = k+1 \\ V(\bar{K}_l) & \text{otherwise,} \end{cases}$$

is the required k-separator. Furthermore, $\{\tilde{V}_1, \tilde{V}_2\}$, where $\tilde{V}_1 = V_l$ and $\tilde{V}_2 = V\backslash(V_l \cup S)$, is the required partition of $V\backslash S$ satisfying (1)–(2).

Indeed, by assumption, $\sum_{j=1}^{k} |V_j| = \sum_{j=1}^{k} |D(\bar{K}_j)| \geqq \frac{2}{3}n - 1$, which, coupled with $\sum_{j=1}^{k+1} |V_j| = n - (k+1)$, implies that

$$(5) \qquad |V_{k+1}| \leqq n-(k+1)-(\tfrac{2}{3}n-1) \leqq \tfrac{2}{3}n.$$

Moreover, since v is the first node encountered by the reduction process for which $|D(K_v)| \geqq \frac{2}{3}n$, we have that

$$(6) \qquad |V_j| \leqq \tfrac{2}{3}n, \qquad j=1, \cdots, k.$$

Thus, by (5) and (6)

$$(7) \qquad |\tilde{V}_1| = |V_l| \leqq \frac{2}{3}n \leqq \frac{k}{k+1}n, \quad \text{for } k \geqq 2.$$

Furthermore, $|\tilde{V}_1| = |V_l| \geqq (n-(k+1))/(k+1)$, which implies that

$$(8) \qquad |\tilde{V}_2| \leqq n-k-\frac{n-(k+1)}{k+1} \leqq \frac{k}{k+1}n.$$

By the definition of $\tilde{V}_1$, $\tilde{V}_2$, and S, we also have that $\tilde{V}_1 \cap \tilde{V}_2 = \phi$, $\tilde{V}_1 \cup \tilde{V}_2 = V\backslash S$, and there is no edge in E with endpoints in $\tilde{V}_1$ and $\tilde{V}_2$. Thus, S is the required k-separator.

Case 2. $\sum_{j=1}^{k} |D(\bar{K}_j)| + 1 < \frac{2}{3}n$.

We will prove that if Case 2 holds then $S = \mathrm{Adj}(v)$ is the required k-separator. Let $\tilde{D}(K_v)$ denote the set of descendant notes of K_v just before it was updated by (3), which resulted in $|D(K_v)| \geqq \frac{2}{3}n$, and construct the new sets $V_1 = \bigcup_{j=1}^{k} D(\bar{K}_j) \cup \{v\}$ and $V_2 = \tilde{D}(K_v)$. By assumption, $|V_1| < \frac{2}{3}n$. Furthermore, $|V_2| < \frac{2}{3}n$, since otherwise Step 1 would have terminated earlier. Now, let $\tilde{V}_1 = V_1$ if $|V_1| \geqq |V_2|$ and $\tilde{V}_1 = V_2$ otherwise, and let $\tilde{V}_2 = V\backslash(\tilde{V}_1 \cup S)$, where $S = \mathrm{Adj}\,(v)$. Since $|V_i| \leqq \frac{2}{3}n$, $i = 1, 2$, we have that

$$|\tilde{V}_1| < \frac{2}{3}n \leqq \frac{k}{k+1}n, \qquad k \geqq 2.$$

Moreover, by (4) $|V_1| + |V_2| = |D(K_v)| \geqq \frac{2}{3}n$, which implies that $|\tilde{V}_1| = \max(|V_1|, |V_2|) \geqq \frac{1}{3}n$. Since $|\tilde{V}_1| + |\tilde{V}_2| = n - k$, we derive

$$|\tilde{V}_2| \leqq n-k-\frac{1}{3}n \leqq \frac{k}{k+1}n, \qquad k \geqq 2.$$

Again, by the definition of $\tilde{V}_1$, $\tilde{V}_2$, and S, we also have that $\tilde{V}_1 \cup \tilde{V}_2 = V\backslash S$, $\tilde{V}_1 \cap \tilde{V}_2 = \phi$, and there is no edge with endpoints in $\tilde{V}_1$ and $\tilde{V}_2$.

The above algorithm will produce a k-separator for a k-tree in linear time since Step 1 will be executed at most $n - k$ times, (3) and (4) require constant time for each iteration, and Step 2 requires constant time.

Let $G = (V, E)$ be a k-tree and S a k-separator of G that induces a partition of $V\backslash S$ into $\{V_1, V_2\}$ that satisfies (1) and (2). Then, the subgraphs $G(V_1 \cup S)$ and $G(V_2 \cup S)$ induced, respectively, by $V_1 \cup S$ and $V_2 \cup S$ are also k-trees that can be similarly decomposed. We can proceed in this manner to decompose G recursively until we end up with k-cliques. That entire decomposition can be represented by a *balanced* *binary decomposition tree* T, which is a rooted binary tree in which the root v represents the graph G, and if some vertex q is a parent of q_1 and q_2 in T, then q_1 and q_2 are sub-k-trees of q obtained by the above decomposition. Observe that (1) guarantees that T is at most $O(\log n)$ deep, i.e., the path from the root of T to any of its leafs contains at most $O(\log n)$ edges. Therefore, in view of Theorem 3.1 we have the following corollary.

COROLLARY 3.1. *Let $G = (V, E)$ be a k-tree. Then, a balanced binary decomposition tree T of G can be constructed in $O(n \log n)$ time.*

4. Some separation properties of partial k-trees. We derive in this section some separation properties of partial k-trees that will be used later to develop NC algorithms for the recognition and embedding of partial k-trees into k-trees when $k = 2, 3$.

LEMMA 4.1.[1] *Let $G = (V, E)$ be a graph and S a separator of G that induces a partition of $V \backslash S$ into $\{V_1, V_2\}$ such that* $0 \leqq |S| = l \leqq k$, $|V_i \cup S| \geqq k$, $i = 1, 2$, *and the subgraph of G induced by S, $G(S)$, is an l-clique. Then G is a partial k-tree if and only if the subgraphs $G_1 = G(V_1 \cup S)$* and *$G_2 = G(V_2 \cup S)$ induced, respectively, by $V_1 \cup S$ and $V_2 \cup S$ are partial k-trees.*

Proof. If G is a partial k-tree then since G_1 and G_2 are subgraphs of G they are also partial k-trees. On the other hand, assume that G_1 and G_2 are partial k-trees and let $\tilde{G}_1 = (V_1 \cup S, \tilde{E}_1)$ and $\tilde{G}_2 = (V_2 \cup S, \tilde{E}_2)$, respectively, be their embeddings into k-trees. Clearly, $G(S)$ is a subgraph of $\tilde{G}_1$ and $\tilde{G}_2$. Therefore, there exist k-cliques K_1 and K_2 contained, respectively, in $\tilde{G}_1$ and $\tilde{G}_2$ and such that both K_1 and K_2 contain the l-clique $G(S)$. Let $v_{l+1}, \cdots, v_k$ and $u_{l+1}, \cdots, u_k$ denote the nodes in $V(K_1)$ and $V(K_2)$, respectively, that are not in S, and consider the graph $\tilde{\tilde{G}}$ obtained from $\tilde{G} = (V, \tilde{E}_1 \cup \tilde{E}_2)$ after the addition of the edges (v_i, u_j), $i = l + 1, \cdots, k, j = l + 1, \cdots, k$, and $j \geqq i$. Then, it is easy to see that $\tilde{\tilde{G}}$ is a k-tree, and since G is a partial graph of $\tilde{\tilde{G}}$, it follows that G is a partial k-tree. (Note that in the degenerate case, when $l = 0$, we simply choose for K_1 and K_2 in the above proof any two k-cliques that are contained in $\tilde{G}_1$ and $\tilde{G}_2$.) □

We will use the following notation. For a graph $G = (V, E)$ and subsets S_1, S_2 such that $S_1 \subset S_2 \subseteq V$, we will denote by $G(S_2; K(S_1))$ the subgraph of G induced by S_2 that is augmented with all arcs between pairs of nodes of S_1 if they are missing in G.

LEMMA 4.2. *Let $G = (V, E)$ be a 2-connected graph and S, $S = \{s_1, s_2\}$, a separator of G that induces the partition of $V \backslash S$ into $\{V_1, V_2\}$ such that $V_i \neq \phi$ and $|V_i \cup S| \geqq k$, $i = 1, 2$. Then, G is a partial k-tree if and only if the subgraphs $G_1 = G(V_1 \cup S; K(\{s_1, s_2\}))$ and $G_2 = G(V_2 \cup S; K(\{s_1, s_2\}))$ are partial k-trees.*

Proof. If $G(V_i \cup S; K(\{s_1, s_2\}))$, $i = 1, 2$, are partial k-trees then by using Lemma 4.1 we can conclude that G is a partial k-tree. Thus, suppose that G is a partial k-tree. Since $V_1 \neq \phi$ and G is 2-connected, there exists a vertex $v_1 \in V_1$ and two disjoint paths, $p(v_1, s_1)$ and $p(v_1, s_2)$, joining v_1 with s_1 and s_2, respectively, in the subgraph $G(V_1 \cup S)$. Therefore, $p(v_1, s_1)$ and $p(v_1, s_2)$ form a simple path, $p(s_1, s_2)$, between s_1 and s_2 in $G(V_1 \cup S)$, and contracting the edges along $p(s_1, s_2)$ would yield a 2-clique with a vertex set $S = \{s_1, s_2\}$. Thus, G is contractible to $G(V_2 \cup S; K(\{s_1, s_2\}))$, which implies that $G(V_2 \cup S; K(\{s_1, s_2\}))$ is a partial k-tree. Analogously, we obtain that $G(V_1 \cup S; K(\{s_1, s_2\}))$ is a partial k-tree as well. □

LEMMA 4.3. *Let $G = (V, E)$ be a triconnected simple graph and S, $S = \{u_1, u_2, u_3\}$, a separator of G inducing a partition of $V \backslash S$ into $\{V_1, V_2\}$ such that*

[1] Similar to Theorem 2.7 in Arnborg and Proskurowski (1986).

$|V_i| \geqq 2$, $i = 1, 2$, and $|V_i \cup S| \geqq k$, $i = 1, 2$. Then, G is a partial k-tree if and only if the subgraphs $G(V_i \cup S; K(S))$, $i = 1, 2$, are partial k-trees.

Proof. If $G(V_i \cup S; K(S))$ are partial k-trees then by using Lemma 4.1 we can conclude that so is G. So, assume that G is a partial k-tree and let v_1, v_2 be in V_1. By the triconnectivity of G, there exist at least three vertex disjoint paths between v_1 and v_2 in G, and since S is a separator and $|S| = 3$, at least two of these paths are contained in $G(V_1 \cup S)$. Therefore, since G is assumed to be a simple graph, these two paths form a cycle $\mathscr{C}$ in $G(V_1 \cup S)$ with at least three vertices. By the triconnectivity of G there exist three vertices, say $\bar{s}_1$, $\bar{s}_2$, and $\bar{s}_3$, in $\mathscr{C}$ and vertex disjoint paths $p_1(s_1, \bar{s}_1)$, $p_2(s_2, \bar{s}_2)$, $p_3(s_3, \bar{s}_3)$ (some possibly degenerated to a single vertex) in $G(V_1 \cup S)$, such that $p_i(s_i, \bar{s}_i)$ $i = 1, 2, 3$, intersect with $\mathscr{C}$ only in nodes $\bar{s}_i$, $i = 1, 2, 3$, respectively. Clearly, if we contract edges in the paths $p_i(s_i, \bar{s}_i)$, $i = 1, 2, 3$, and identify s_i with $\bar{s}_i$, for $i = 1, 2, 3$, respectively, we create a cycle $\mathscr{C}'$ that contains s_1, s_2, s_3. Again, appropriate contractions of edges in $\mathscr{C}'$ will create a 3-clique $K(S)$. Thus, G is contractible to $G(V_2 \cup S; K(S))$ and we conclude that $G(V_2 \cup S; K(S))$ is a partial k-tree. Analogously, we can show that $G(V_1 \cup S; K(S))$ is a partial k-tree. □

We note that the above separation properties of partial k-trees imply the existence of a balanced decomposition tree T of an embedding of a biconnected (respectively, triconnected) partial 2-tree (respectively, 3-tree) G into a 2-tree (respectively, 3-tree). The root r of T is the graph G, and if q_1 and q_2 are sons of r in T then q_1 (respectively, q_2) is the graph $G(V_1 \cup S; K(S))$ (respectively, $G(V_2 \cup S; K(S)))$, where S is a 2-separator (respectively, 3-separator) of G satisfying (1) and (2), and whose existence follows from Theorem 3.1. The biconnectivity (respectively, triconnectivity) of G imply that $G(V_1 \cup S; K(S))$ and $G(V_2 \cup S; K(S))$ are biconnected (respectively, triconnected). Thus, if they have a sufficient number of nodes, they can be decomposed in a similar manner. In general, the leaves of the decomposition tree T are biconnected (respectively, triconnected) 2-trees (respectively, 3-trees) that are not necessarily decomposable in the same manner if they have less than 9 (respectively, 20) nodes.

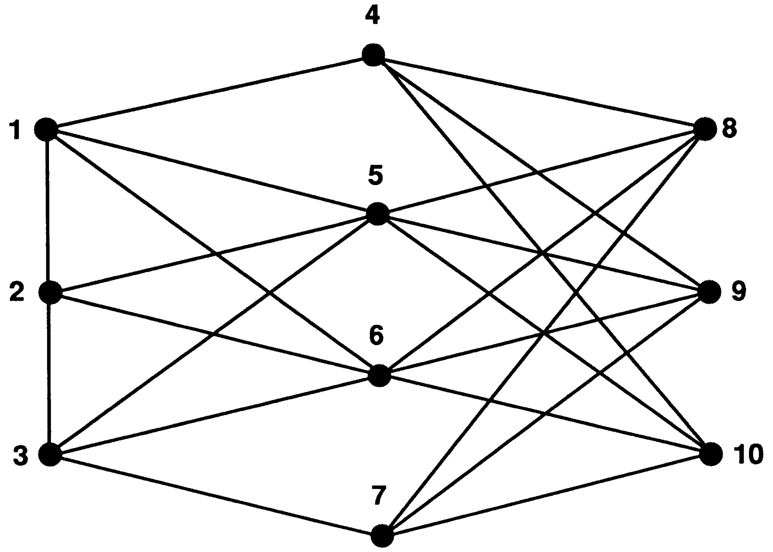


FIG. 4.1. $G = (V, E)$ *is* 4-*connected partial* 4-*tree*.

Apparently, the proofs of Lemmas 4.2 and 4.3 do not carry over for separators of cardinality 4 or more. Indeed, consider the graph $G = (V, E)$ depicted in Fig. 4.1.

Now, G can be easily seen to be a 4-connected partial 4-tree and $S = \{4, 5, 6, 7\}$ is a separator of G, inducing the partition of $V \backslash S$ to $V_1 = \{1, 2, 3\}$ and $V_2 = \{8, 9, 10\}$. Furthermore, there exists an embedding of G into a 4-tree in which S is a 4-clique. However, we can verify by inspection that edges in $G(V_1 \cup S)$ cannot be contracted so as to make S a 4-clique in the contracted graph.

5. NC algorithms for recognition of *k*-connected partial *k*-trees for *k* = 2, 3. In this section we present NC algorithms for recognizing a partial 2-tree (respectively, 3-connected partial 3-tree) graph G and finding its embedding into a 2-tree (respectively, 3-tree). Without loss of generality, we assume here and in § 6 that G is a simple graph.

```
Algorithm 5.1
Procedure  BBD(G, α, L)
Input  A 2-connected (respectively, 3-connected) simple graph $G = (V, E)$.
Output  $\alpha = 1$ if $G$ is a partial 2-tree (respectively, 3-tree), and $\alpha = 0$ otherwise. Furthermore, if $\alpha = 1$ then the output contains an embedding of $G$ into a 2-tree (respectively, 3-tree).
Step 0.  Set $\alpha = 1$ and $L = \phi$.
Step 1.
    (1.1)  If $|E| > 2n - 3$ (respectively, $3n - 6$) set $\alpha = 0$ and exit.
    (1.2)  Else, if $|V| \leqq 9$ (respectively, 20) check in constant time whether $G$ is a partial 2-tree (respectively, 3-tree). If yes, find an embedding, $\bar{G}$, of $G$ into a 2-tree (respectively, 3-tree), let $L = L \cup \{\bar{G}\}$. Else, set $\alpha = 0$. Exit.
Step 2.  Else, for each pair (respectively, triple) of distinct vertices $S = \{s_1, s_2\}$ (respectively, $S = \{s_1, s_2, s_3\}$) of $V$, in parallel, find all the connected components of $G(V \backslash S)$ and denote their vertex sets by $V_j^S : j = 1, \cdots, r_S$, for some $r_S \geqq 1$.
    (2.1)  If $|V_j^S| > \frac{2}{3}|V|$ (respectively, $|V_j^S| > \frac{3}{4}|V|$) for some $j = 1, \cdots, r_S$ reject $S$. If all the pairs (respectively, triples) $S$ were rejected set $\alpha = 0$ and exit.
    (2.2)  Else, if $\frac{1}{3}|V| \leqq |V_j^S| \leqq \frac{2}{3}|V|$ (respectively, $\frac{1}{4}|V| \leqq |V_j^S| \leqq \frac{3}{4}|V|$) for some $S$ and for some $j$, $1 \leqq j \leqq r_S$, set $M_1 = V_j^S \cup S$ and $M_2 = (V \backslash V_j^S)$.
    (2.3)  Else, for an unrejected $S$ compute $q_l = \sum_{j \leqq l} |V_j^S|, l = 1, \cdots, r_S$, and apply a binary search to find $\eta$ for which $\frac{1}{3}|V| \leqq q_\eta \leqq \frac{2}{3}|V|$ (respectively, $\frac{1}{4}|V| \leqq q_\eta \leqq \frac{3}{4}|V|$). Set $M_1 = \cup_{j=1}^{\eta} V_j^S \cup S$ and $M_2 = (V \backslash M_1) \cup S$.
    (2.4)  Call in parallel BBD(G(M_1; K(S)), α_1, L_1) and BBD(G(M_2; K(S)), α_2, L_2). If either $\alpha_1 = 0$ or $\alpha_2 = 0$, set $\alpha = 0$, else, set $L = L_1 \cup L_2$. Exit.
```

THEOREM 5.1. *Algorithm* 5.1 *recognizes whether a* 2-*connected* (*respectively*, 3-*connected*) *simple graph* $G = (V, E)$ *with* $|V| = n$ *is a partial* 2-*tree* (*respectively*, 3-*tree*) *and produces an embedding of* G *into a* 2-*tree* (*respectively*, 3-*tree*) *in* $O(\log^2 n)$ *time using* $O(n^3)$ (*respectively*, $O(n^4)$) *processors*.

Proof. We first show that the steps of Algorithm 5.1 are valid. If (1.1) in Step 1 holds then, since every 2-tree (respectively, 3-tree) has exactly $2n - 3$ (respectively, $2n - 6$) edges, it follows from Lemma 4.2 (respectively, Lemma 4.3) that G is not a partial 2-tree (respectively, 3-tree). If all pairs (respectively, triples) in Step 2 were rejected, G does not contain a 2-separator (respectively, 3-separator) and then, by Theorem 3.1, G is not a partial 2-tree (respectively, 3-tree). The biconnectivity (respectively, triconnectivity) of G implies the biconnectivity (respectively, triconnectivity) of the minors $G(M_i; K(S))$, $i = 1, 2$, created in Steps (2.2) and (2.3). Therefore, by Lemma

4.2 (respectively, Lemma 4.3), G is a partial 2-tree (respectively, 3-tree) if and only if all the minors $G(M_i; K(S))$, $i = 1, 2$, created in Step 2 are partial 2-trees (respectively, 3-trees).

Next, we will show that Algorithm 5.1 requires $O(\log^2 n)$ time and $O(n^3)$ (respectively, $O(n^4)$) processors. In Step 1, (1.2) can be performed in constant time using, for example, the sequential algorithm for the recognition and embedding of partial 2-trees (respectively, 3-trees) into 2-trees (respectively, 3-trees) of Wald and Colbourn [20] (respectively, Matousek and Thomas [14]). All connected components of $G(V\backslash S)$ can be found, by the parallel algorithm of Shiloach and Vishkin [18], in $O(\log n)$ time and using $O(n + m)$ processors, where $m = |E|$. Note that in our case $m = O(n)$. Since there are $O(n^2)$ (respectively, $O(n^3)$) pairs (respectively, triples) to be considered simultaneously, Step 2 requires $O(n^3)$ (respectively, $O(n^4)$) processors. Furthermore, in Step 2, (2.3) takes $O(\log n)$ time using $O(n^2)$ processors to compute partial sums, q_l, and perform a binary search on them. Since in (2.2) and (2.3) $|M_i| \leqq \frac{2}{3}|V|$ (respectively, $|M_i| \leqq \frac{3}{4}|V|$), $i = 1, 2$, Algorithm 5.1 will terminate after performing at most $O(\log n)$ nested calls of Procedure $BBD(G, \alpha, L)$. Thus, our algorithm recognizes whether a 2-connected (respectively, 3-connected) graph G is a partial 2-tree (respectively 3-tree) and delivers in L the leaves of a binary decomposition tree of an embedding of G into a 2-tree (respectively, 3-tree) in $O(\log^2 n)$ time using $O(n^3)$ (respectively, $O(n^4)$) processors. □

We note that in Algorithm 5.1 we have restricted G to be biconnected (respectively, triconnected). However, the recognition of partial 2-trees that are not necessarily biconnected can be easily carried out by a slight modification of Algorithm 5.1. Indeed, we can find all biconnected components of G using the parallel algorithm of Tarjan and Vishkin [19] in $O(\log n)$ time with $O(n)$ processors. If G does not have biconnected components, then it can be decomposed by zero or one separators into subgraphs of cardinality less than or equal to 3. Then, since every graph on three vertices is a partial 2-tree, Lemma 4.1 implies that G is a partial 2-tree. Otherwise, we perform Algorithm 5.1 on all biconnected components of G. By Lemma 4.1, G is a partial 2-tree if and only if all biconnected components of G are partial 2-trees. Clearly, the modified algorithm requires $O(\log^2 n)$ time and $O(n^3)$ processors.

The recognition of partial 3-trees that are not necessarily 3-connected requires a major modification of Algorithm 5.1, which is developed in the next section.

6. NC algorithm for recognizing partial 3-trees. We develop in this section an NC algorithm for recognizing a partial 3-tree that is not necessarily 3-connected. The performance of this algorithm is identical to that developed in § 5 for recognizing and embedding 3-connected partial 3-trees. That is, it requires $O(\log^2 n)$ time and $O(n^4)$ processors. However, its description is somewhat more involved and it depends on some new separation properties that are developed below. First, we need to introduce a new definition.

For two disjoint paths $p_1 = p_1(v, s_1)$ and $p_2 = p_2(v, s_2)$ in $G = (V, E)$, having only vertex v in common, the path $p = p(k, l)$ will be called a *bridge path* between p_1 and p_2 if p originates at some node k in p_1, $k \neq v$, and terminates at node l in p_2, $l \neq v$, but p is otherwise vertex disjoint with p_1 and p_2.

LEMMA 6.1. *Let $S = \{s_1, s_2, s_3\}$ be a separator of $G = (V, E)$ inducing a partition $\{V_1, V_2\}$ of $V\backslash S$ such that no edge in E connects a vertex in V_1 with a vertex in V_2. Assume that for $i = 1, 2$ there exists a vertex $v_i \in V_i$ and three disjoint paths $p(v_i, s_j)$, $j = 1, 2, 3$, joining v_i with s_j, $j = 1, 2, 3$. If for $i = 1, 2$ there exists a bridge path between a pair of paths among the paths $p(v_i, s_j)$, $j = 1, 2, 3$, which is contained in*

$G(V_i \cup S)$, then G is a partial k-tree if and only if $G(V_1 \cup S; K(S))$ and $G(V_2 \cup S; K(S))$ are partial k-trees.

Proof. The existence of a bridge path between some pair of the paths $p(v_i, s_j)$, $j = 1, 2, 3$, for $i = 1, 2$ imply that G is contractible to $G(V_1 \cup S; K(S))$ and $G(V_2 \cup S; K(S))$, see Fig. 6.1. Thus, if G is a partial k-tree, so are $G(V_i \cup S; K(S))$, $i = 1, 2$. On the other hand, Lemma 4.1 implies that if $G(V_i \cup S; K(S))$, $i = 1, 2$, are partial k-trees then G must also be a partial k-tree. □

To illustrate the proof of Lemma 6.1 see Fig. 6.1.

$S = \{s_1, s_2, s_3\}$ is a 3-separator of $G = (V, E)$ in Fig. 6.1, $V_1 = \{v_1, u_1, u_2\}$, $V_2 = \{t_1, t_2, v_2\}$, $p(u_1, u_2)$ is a bridge path in $G(V_1 \cup S)$ and $p(t_1, t_2)$ is a bridge path in $G(V_2 \cup S)$. Contracting, for example, first edges (u_1, s_1), (u_2, s_2), and then (v_1, s_3) would lead to $G(V_2 \cup S; K(S))$.

LEMMA 6.2. *Let $S = \{s_1, s_2, s_3\}$ be a 3-separator of a 2-connected simple graph $G = (V, E)$ that induces a partition $\{V_1, V_2\}$ of $V \backslash S$ such that $2 \leqq |V_i| \leqq \frac{3}{4}|V|$, $i = 1, 2$, and no edge in E connects a vertex in V_1 with a vertex in V_2. Assume that for $i = 1, 2$ there exist vertices $v_i \in V_i$ and three disjoint paths $p(v_i, s_j)$, $j = 1, 2, 3$, joining v_i with vertices in S. If for $i = 1$ or 2 there exists no bridge path between any pair of paths among the paths $p(v_i, s_j)$, $j = 1, 2, 3$, which is contained in $G(V_i \cup S)$ then for some $j \in \{1, 2, 3\}$ and $i \in \{1, 2\}$, $\{v_i, s_j\}$ is a separator of G, which induces a partition $\{V'_1, V'_2\}$ of $V \backslash \{v_i, s_j\}$ such that $\frac{1}{12}|V| - 2 \leqq |V'_1| \leqq \frac{3}{4}|V|$.*

Proof. Assume, without loss of generality, that there is no bridge path between any pair of paths among the paths $p(v_1, s_j)$, $j = 1, 2, 3$, see Fig. 6.2. Since G is connected, for any $v \in V_1$ there exists a node $s(v)$, $s(v) \in S$, and a path from v to $s(v)$, $p(v, s(v))$, in G that does not pass through nodes in $S \backslash \{s(v)\}$. Now, any node v, $v \in V_1$ and $v \neq v_1$, must be in some connected component of $G(V \backslash \{s(v), v_1\})$ that does not contain $S \backslash \{s(v)\}$. Indeed, otherwise there must exist a path, $p(v, \bar{s}(v))$, in G from v to a node $\bar{s}(v)$, $\bar{s}(v) \in S \backslash \{s(v)\}$, which does not pass through either v_1 or $s(v)$. But the paths $p(v, s(v))$ and $p(v, \bar{s}(v))$ in G induce a bridge path in $G(V_1 \cup S)$ between $p(v_1, s(v))$ and $p(v_1, \bar{s}(v))$, which is a contradiction.

Next, for $t = 1, 2, 3$ let C_t denote, respectively, the union of all connected components of $G(V \backslash \{v_1, s_t\})$ that do not contain any node in $S \backslash \{s_t\}$. We claim that the

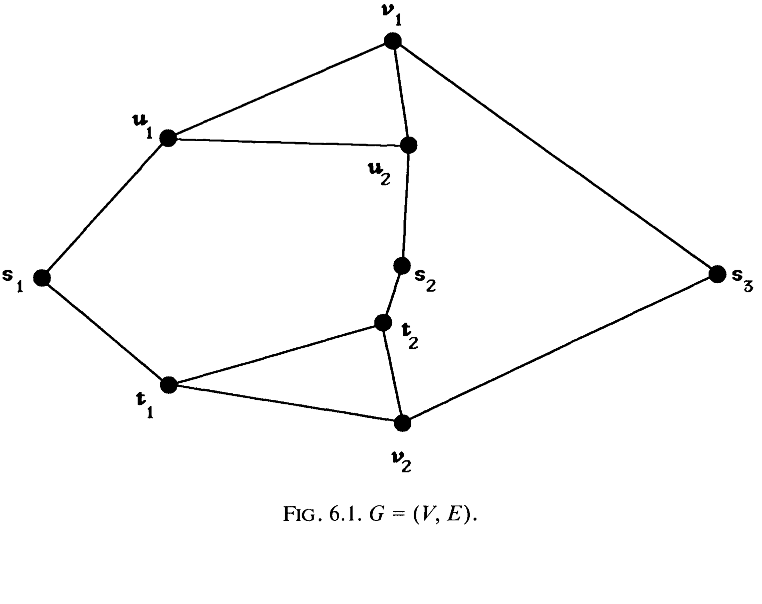


FIG. 6.1. $G = (V, E)$.

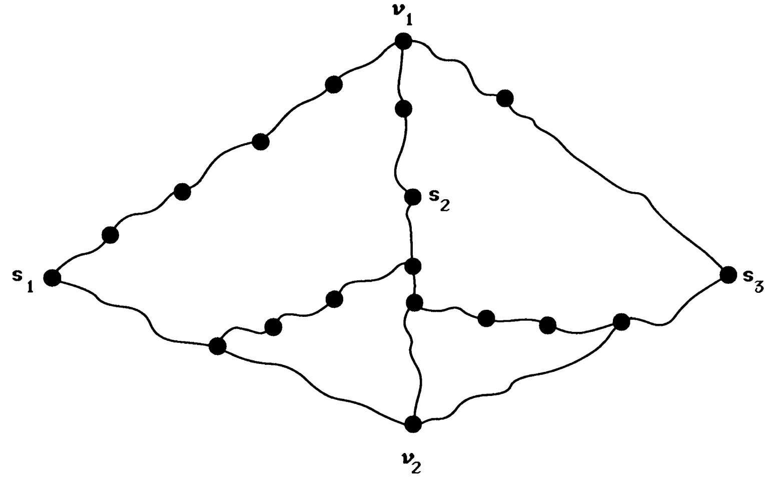


FIG. 6.2. $G = (V, E)$.

sets C_t, $t \in \{1, 2, 3\}$, are mutually exclusive. Indeed, from the 2-connectivity of G and the definition of the sets C_t it follows that if $v \in C_i$, $i \in \{1, 2, 3\}$ and $v \neq v_1$, then there exists a path $p(v, s_i)$, $s_i \in S$, which does not pass through $\{v_1\} \cup S \setminus \{s_i\}$. Thus, if $v \in C_i \cap C_j$, $i, j \in \{1, 2, 3\}$, and $i \neq j$, the paths $p(v, s_i)$ and $p(v, s_j)$ in G would induce a bridge path between $p(v_1, s_i)$ and $p(v_1, s_j)$, which is a contradiction. Thus, we have that $\sum_{t=1}^{3} |V(C_t)| = |V_1| - 1$, where $V(C_t)$ denotes the set of nodes in C_t. Let $|V(C_j)| = \max\{|V(C_t)| : t = 1, 2, 3\}$, and set $V'_1 = V(C_j)$ and $V'_2 = V \setminus (V'_1 \cup \{v_1, s_j\})$. Since S is a 3-separator, $|V_1| \geqq \frac{1}{4}|V| - 3$ and thus, $|V'_1| \geqq \frac{1}{12}|V| - 2$. Clearly, $V(C_t) \subseteq V_1$, $t = 1, 2, 3$, and thus $|V'_1| \leqq \frac{3}{4}|V|$. □

For an illustration of Lemma 6.2 see Fig. 6.2.

$S = \{s_1, s_2, s_3\}$ is a 3-separator of $G = (V, E)$ in Fig. 6.2, $v_1 \in V_1, v_2 \in V_2$; there are bridge paths in $G(V_2 \cup S)$ but there is no bridge path in $G(V_1 \cup S)$. The set $\{v_1, s_1\}$, for example, is the required separator, whose existence is proved in Lemma 6.2.

We use the separation results derived in Lemmas 6.1 and 6.2 and earlier to construct an NC-algorithm for recognizing a partial 3-tree.

```
Algorithm 6.1
Procedure  REC($\bar{G}$, $\alpha$)
Input   A simple graph $\bar{G} = (\bar{V}, \bar{E})$.
Output  $\alpha = 1$ if $\bar{G}$ is a partial 3-tree and $\alpha = 0$ otherwise.
Step 1. Find all biconnected components of $\bar{G}$. If $\bar{G}$ has no biconnected components,
        set $\alpha = 1$ and exit. Otherwise, perform in parallel procedure $REC1(G_i, \alpha_i)$ for
        all biconnected components $G_i = (V_i, E_i)$ of $\bar{G}$. If any $\alpha_i = 0$, set $\alpha = 0$ and
        exit.
Procedure  REC1($G$, $\alpha$)
Input   A simple biconnected graph $G = (V, E)$.
Output  $\alpha = 1$ if $G$ is a partial 3-tree and $\alpha = 0$ otherwise.
Step 0. $\alpha = 1$.
Step 1.
    (1.1) If $|E| > 3n - 6$, set $\alpha = 0$ and exit.
    (1.2) Else, if $|V| < 36$ check, in constant time, whether $G$ is a partial 3-tree. If
          $G$ is not a partial 3-tree, set $\alpha = 0$. Exit.
```

***Step* 2.** Else, for each triple of distinct vertices $S = \{s_1, s_2, s_3\}$ of V, in parallel, find all connected components of $G(V \backslash S)$ and denote their vertex sets by V_j^S, $j = 1, \cdots, r_S$, for some $r_S \geqq 1$.

(2.1) If $|V_j^S| > \frac{3}{4}|V|$ for some $j = 1, \cdots, r_S$, reject S. If all triples S are rejected, set $\alpha = 0$ and exit.

(2.2) Else, if $\frac{1}{4}|V| \leqq |V_j^S| \leqq \frac{3}{4}|V|$ for some S and for some j, $1 \leqq j \leqq r_S$, set $M_1 = V_j^S \cup S$ and $M_2 = (V \backslash V_j^S)$.

(2.3) Else, for an unrejected S and for all $j = 1, \cdots, r_S$, compute $q_l = \sum_{j \leqq l} |V_j^S|$, $l = 1, \cdots, r_S$, and apply a binary search to find an η for which $\frac{1}{4}|V| \leqq q_\eta \leqq \frac{3}{4}|V|$. Set $M_1 = S \cup (\cup V_j^S : j \leqq \eta)$ and $M_2 = (V \backslash M_1) \cup S$.

***Step* 3.** Find, in parallel, connected components of $G(M_t \backslash \{s_i, s_j\})$ that do not contain $S \backslash \{s_i, s_j\}$ for $t = 1, 2$, $i \neq j$, $i, j \in \{1, 2, 3\}$. Denote by $C_{i,j}^t$ the union of all such connected components, and let $C_{i,j} = C_{i,j}^1 \cup C_{i,j}^2$ for all $i, j \in \{1, 2, 3\}$, $i \neq j$.

(3.1) If $C_{1,2} \neq \phi$, $C_{2,3} \neq \phi$, and $C_{1,3} \neq \phi$, let $G_1 = G(M_1; K(S))$ and $G_2 = G(M_2; K(S))$.

(3.2) Else, if there exist at least two distinct pairs $\{i, j\}$ and $\{p, q\}$, $\{i, j\} \cup \{p, q\} = \{1, 2, 3\}$ such that $C_{i,j} = \phi$ and $C_{p,q} \neq \phi$ proceed as follows:

(a) If $|V(C_{1,2}^1)| + |V(C_{2,3}^1)| + |V(C_{1,3}^1)| < |M_1 \backslash S|$ and $|V(C_{1,2}^2)| + |V(C_{2,3}^2)| + |V(C_{1,3}^2)| < |M_2 \backslash S|$, where $V(C_{i,j}^t)$ is the vertex set of component $C_{i,j}^t$, then let $G_1 = G(M_1; K(S))$ and $G_2 = G(M_2; K(S))$.

(b) Else, if for some $t \in \{1, 2\}$, $|V(C_{1,2}^t)| + |V(C_{2,3}^t)| + |V(C_{1,3}^t)| = |M_t \backslash S|$ find $|V(C_{k,l}^t)| = \max\{|V(C_{1,2}^t)|, |V(C_{1,3}^t)|, |V(C_{2,3}^t)|\}$ and let $G_1 = G(V(C_{k,l}^t) \cup \{s_k, s_l\}; K(\{s_k, s_l\}))$ and $G_2 = G((V \backslash V(C_{k,l}^t)) \cup \{s_k, s_l\}; K(\{s_k, s_l\}))$.

(3.3) Else, if $C_{1,2} = \phi$, $C_{2,3} = \phi$, and $C_{1,3} = \phi$, find, in parallel, connected components of $G(V \backslash \{s_t, v\})$, for all $t \in \{1, 2, 3\}$ and $v \in V$, and denote their vertex sets by $V_j(v, s_t)$, $j = 1, \cdots, m(v, s_t)$, for some $m(v, s_t) \geqq 1$. If $|V_j(v, s_t)| > 11/12|V|$ for some j, $j \in \{1, \cdots, m(v, s_t)\}$, reject the pair $\{v, s_t\}$.

(a) If all pairs $\{v, s_t\}$, $t \in \{1, 2, 3\}$ and $v \in V$, were rejected let $G_1 = G(M_1; K(S))$ and $G_2 = G(M_2; K(S))$.

(b) Else, if $1/12|V| - 2 \leqq |V_j(v, s_t)|$ for some $j \in \{1, \cdots, m(v, s_t))\}$, set $M_1' = V_j(v, s_t) \cup \{v, s_t\}$ and $M_2' = V \backslash V_j(v, s_t)$ and let $G_1 = G(M_1'; K(\{v, s_t\}))$ and $G_2 = G(M_2'; K(\{v, s_t\}))$.

(c) Else, for an unrejected pair $\{v, s_t\}$ compute $q_l = \sum_{i \leqq l} |V_i(v, s_t)|$, $l = 1, \cdots, m(v, s_t)$, and apply a binary search to find η for which $\frac{1}{12}|V| - 2 \leqq q_\eta \leqq \frac{3}{4}|V|$. Set $M_1' = \{v, s_t\} \cup (\cup V_j(v, s_t) : j \leqq \eta)$, $M_2' = (V \backslash M_1') \cup \{v, s_t\}$, $G_1 = G(M_1'; K(\{v, s_t\}))$ and $G_2 = G(M_2'; K(\{v, s_t\}))$.

(3.4) Call in parallel $REC1(G_1, \alpha_1)$ and $REC1(G_2, \alpha_2)$. If either $\alpha_1 = 0$ or $\alpha_2 = 0$, set $\alpha = 0$. Exit.

THEOREM 6.1. *Algorithm* 6.1 *recognizes whether a simple graph* $\bar{G} = (\bar{V}, \bar{E})$ *with* $|\bar{V}| = n$ *is a partial* 3-*tree in* $O(\log^2 n)$ *time using* $O(n^4)$ *processes*.

Proof. We will first prove the validity of Algorithm 6.1. If, in Procedure $REC(\bar{G}, \alpha)$, $\bar{G}$ is found not to contain biconnected components, then it could be decomposed by zero or one separators into subgraphs of cardinality less than or equal to 4. Since every graph on four nodes is a partial 3-tree, Lemma 4.1 implies that $\bar{G}$ is a partial 3-tree. The validity of Step 1 in $REC1(G, \alpha)$ was explained in Algorithm 5.1. If all triples were rejected in Step 2, G does not contain a 3-separator and by Theorem

3.1, G is not a partial 3-tree. Otherwise, in Step 2, Algorithm 6.1 finds a 3-separator. In Step 3, if $C_{i,j} \neq \phi$ for some $i, j, i \neq j$, then $\{s_i, s_j\}$ is a separator of G, and by the biconnectivity of G and Lemma 4.2 it follows that G is a partial 3-tree if and only if G, augmented with edge (s_i, s_j), is a partial 3-tree. Thus, by Lemma 4.3, if (3.1) holds, then G is a partial 3-tree if and only if $G(M_1; K(S))$ and $G(M_2; K(S))$ are partial 3-trees. If (3.2) holds then, by Lemma 4.2, we can augment G with edge (p, q). Therefore, if (3.2a) is valid, there must exist vertices v_i, $v_i \in M_i \backslash S$, $i = 1, 2$, and three disjoint paths from v_i to all $s \in S$ in $G(M_i)$, $i = 1, 2$. Thus, by Lemma 6.1, since (p, q) is a bridge path both in $G(M_1)$ and $G(M_2)$, G is a partial 3-tree if and only if $G(M_1; K(S))$ and $G(M_2; K(S))$ are partial 3-trees. On the other hand, if (3.2b) holds then, by Lemma 4.2, G is a partial 3-tree if and only if $G(V(C^t_{k,l}) \cup \{s_k, s_l\}; K(\{s_k, s_l\}))$ and $G((V \backslash V(C^t_{k,l})) \cup \{s_k, s_l\}; K(\{s_k, s_l\}))$ are partial 3-trees. If (3.3) holds, then for each $i = 1, 2$ and for each $v \in M_i$, there exist three disjoint paths in $G(M_i)$ from v to all nodes in S. Therefore, by Lemma 6.2, if all pairs are rejected in (3.3a), there must exist a bridge path *both* in $G(M_1)$ and $G(M_2)$ satisfying the stipulations in Lemma 6.2. Then, by Lemma 6.1, G is a partial 3-tree if and only if $G(M_1; K(S))$ and $G(M_2; K(S))$ are partial 3-trees. Otherwise, by Lemma 4.2, G is a partial 3-tree if and only if $G(M'_1; K(\{v, s_t\}))$ and $G(M'_2; K(\{v, s_t\}))$ are partial 3-trees.

Next, we will show that Algorithm 6.1 requires $O(\log^2 n)$ time using $O(n^4)$ processors. All biconnected components can be found in Step 0, using the parallel algorithm of Tarjan and Vishkin [19], in $O(\log n)$ time using $O(n)$ processors. In Step 1, (1.2) can be carried out in constant time using, for example, the sequential algorithm for recognizing partial 3-trees of Matousek and Thomas [14]. The connected components in Step 2 can be found by the parallel algorithm of Shiloach and Vishkin [18] in $O(\log n)$ time with $O(n + m)$ processors. Since there are $O(n^3)$ triples S to be considered simultaneously, Step 2 requires $O(\log n)$ time and $O(n^4)$ processors. It takes $O(\log n)$ time using $O(n^2)$ processors to compute, in Step (2.3), partial sums q_l and perform a binary search on them. In Step 3, connected components of $G(M_t \backslash \{s_i, s_j\})$ that do not contain $S \backslash \{s_i, s_j\}$ can be found in $O(\log n)$ time using $O(n)$ processors. Similarly, connected components of $G(V \backslash \{s_t, v\})$ can be found in $O(\log n)$ time using $O(n)$ processors, and since there are $O(n)$ pairs $\{s_t, v\}$ to be considered simultaneously, Step (3.3) requires $O(\log n)$ time and $O(n^2)$ processors. Furthermore, (3.3c) takes $O(\log n)$ time using $O(n^2)$ processors. Finally, observe that in (2.2) $|V^S_j| \geqq \frac{1}{4}|V|$, in (2.3) $q_\eta \geqq \frac{1}{4}|V|$ and, since S is a 3-separator, in Step (3.2b) $|V(C^t_{k,l})| \geqq \frac{1}{12}|V|$. Furthermore, in Step (3.3), $|V_j(v, s_t)| \geqq \frac{1}{12}|V| - 2$ and $q_\eta \geqq \frac{1}{12}|V| - 2$. Therefore, we can have at most $O(\log n)$ nested calls of $REC1(G, \alpha)$, and Algorithm 6.1 would terminate in $O(\log^2 n)$ time using $O(n^4)$ processors. □

Acknowledgment. The authors are grateful to Sundaran Gopalakrishnan for some useful comments made on an earlier version of this paper.

REFERENCES

[1] S. ARNBORG, *On the complexity of multivariable query evaluation*, FOA Rapport C 20292-D8, National Defense Research Institute, Stockholm, Sweden, 1979.

[2] S. ARNBORG, D. CORNEIL, AND A. PROSKUROWSKI, *Complexity of finding embeddings in a k-tree*, SIAM J. Algebraic Discrete Methods, 8 (1987), pp. 277–284.

[3] S. ARNBORG, A. PROSKUROWSKI, AND D. CORNEIL, *Forbidden minors characterizations of partial 3-trees*, CIS-TR-86-07, University of Oregon, Eugene, Oregon, July, 1986.

[4] S. ARNBORG AND A. PROSKUROWSKI, *Characterization and recognition of partial 3-trees*, SIAM J. Algebraic Discrete Methods, 1986, pp. 305–314.

[5] ———, *Linear time algorithms for* NP-*hard problems restricted to partial k-trees*, Discrete Appl. Math., 23 (1989), pp. 11–24.

[6] H. L. Bodlaender, *Improved self-reduction algorithms for graphs with bounded treewidth*, RUV-CS-88-29, Rijksuniversiteit Utrecht, the Netherlands, September, 1988.

[7] ———, *Dynamic programming on graphs with bounded treewidth*, RUV-CS-87-22, Rijksuniversiteit Utrecht, the Netherlands, November, 1987.

[8] ———, NC-*Algorithms for graphs with small treewidth*, RUV-CS-88-4, Rijksuniversiteit Utrecht, The Netherlands, February, 1988.

[9] N. Chandrasekharan and S. T. Hedetniemi, *Fast parallel algorithms for tree decomposing and partial k-trees*, Proceedings of the 26th Annual Allerton Conference on Communication, Control and Computing, 1989, pp. 283–292.

[10] A. M. Farley, *Networks immune to isolated failures*, Networks, 11 (1981), pp. 255–268.

[11] A. M. Farley and A. Proskurowski, *Networks immune to isolated line failures*, Networks, 12 (1982), pp. 393–403.

[12] D. Granot and D. Skorin-Kapov, *On some optimization problems on k-trees and partial k-trees*, Working Paper No. 1283, University of British Columbia, Vancouver, British Columbia, Canada, July, 1988.

[13] R. Hassin and A. Tamir, *Efficient algorithms for optimization and selection on SP graphs*, SIAM J. Algebraic Discrete Methods, 7 (1986), pp. 379–389.

[14] J. Matousek and R. Thomas, *Algorithms finding tree-decompositions of graphs*, manuscript, 1988.

[15] E. M. Neufeldt and C. J. Colbourn, *The most reliable series-parallel networks*, Networks, 15 (1985), pp. 27–32.

[16] N. Robertson and P. D. Seymour, *Graph minors* II: *Algorithmic aspects of tree width*, J. Algorithms, 7 (1986), pp. 309–322.

[17] ———, *Disjoint paths—A survey*, SIAM J. Algebraic Discrete Methods, 6 (1985), pp. 300–305.

[18] Y. Shiloach and U. Vishkin, *An $O(\log n)$ parallel connecting algorithm*, J. Algorithms, 3 (1982), pp. 57–67.

[19] R. E. Tarjan and U. Vishkin, *An efficient parallel biconnectivity algorithm*, SIAM J. Comput., (1985), pp. 866–874.

[20] A. Wald and C. J. Colbourn, *Steiner trees, partial* 2-*trees and minimum IFI networks*, Networks, 13 (1983), pp. 159–167.

SIAM J. DISC. MATH.
Vol. 4, No. 3, pp. 355-368, August 1991

005

RIGIDITY MATROIDS*

JACK E. GRAVER†

Abstract. This paper begins with a short discussion of the general principles of Rigidity Theory. The main interest is the combinatorial part of this subject: generic rigidity. While generic rigidity has several combinatorial characterizations in dimensions one and two, these characterizations have not been able to be extended to characterizations of generic rigidity in higher dimensions. In fact, no "purely combinatorial" characterization is presently known for generic rigidity in dimensions three and up. The concept of an abstract rigidity matroid is introduced and, in the context of matroid theory, the present status of the characterization problem is discussed.

Key words. rigidity, infinitesimal rigidity, generic rigidity, matroid theory

AMS(MOS) subject classifications. 05B35, 05C10

1. Introduction to rigidity matroids. We will use the term *framework* to denote a triple $(V, E, \mathbf{p})$ where (V, E) is a finite graph and $\mathbf{p}$ is an embedding (injection) of V into real d-space. We will identify V with the set of the first n positive integers and write $\mathbf{p}_i$ instead of $\mathbf{p}(i)$. Given a framework in d-space for d equal to 1, 2, or 3, we could construct a physical model by actually joining the pairs of points which correspond to the edges by rods hinged at their endpoints. The resulting physical model will either be rigid or admit some motion which alters the distance between some pair of points. The fundamental problem in rigidity theory is developing a method for predicting rigidity without building a model. The concept of rigidity is partly combinatorial and partly geometric in nature. In this paper we will concentrate on the combinatorial aspects of this concept.

Keeping the physical model in mind, it is very easy to see that a framework $(V, E, \mathbf{p})$ in dimension one is rigid if and only if the graph (V, E) is connected. Throughout the paper, we will use the one-dimensional case to illustrate our definitions and results.

Let $(V, E, \mathbf{p})$ be a framework in d-space with $|V| = n$. A direct attack on the rigidity question for this framework could be carried out as follows: coordinatize d-space, write down the distance equations for the pairs of points given by E with the coordinates of the points replaced by dn distinct variables, and then ask if all other "nearby" solutions to this system are congruent to the original framework. Clearly, if the framework admits a deformation, there will be arbitrarily close frameworks which are solutions to this system of quadratic equations but which are not congruent to the original framework. Quadratic systems are not easy to work with, and the problem is often linearized by considering the initial velocity vectors of a deformation at vertices of the framework. This leads to the concept of infinitesimal rigidity. While rigidity and infinitesimal rigidity are not identical concepts, they are closely related and, most importantly for us, the combinatorial aspects of each are identical. Hence, while rigidity is the more natural of the two concepts, we will take the computationally simpler infinitesimal approach in this paper.

Let $(V, E, \mathbf{p})$ be a framework in d-space with $|V| = n$. A second function $\mathbf{u}$ mapping V into $\mathbb{R}^d$ is called a *loading* of $(V, E, \mathbf{p})$ and we interpret $\mathbf{u}_i = \mathbf{u}(i)$ as a vector attached to the point $\mathbf{p}_i$. We think of the image of $\mathbf{u}$ as the set of initial velocities of some motion of the set of points $\mathbf{p}(V)$. Of particular interest are those $\mathbf{u}$ which correspond to the initial velocities of motions of $\mathbf{p}(V)$ which do not stretch or compress the "rods" corresponding

* Received by the editors October 18, 1989; accepted for publication (in revised form) September 26, 1990.

† Department of Mathematics, Syracuse University, Syracuse, New York 13244-1150.

to E. We define $\mathbf{u}$ to be an *infinitesimal motion* of $(V, E, \mathbf{p})$ if, for every pair (i, j) in E, we have

$$(\mathbf{p}_i - \mathbf{p}_j) * (\mathbf{u}_i - \mathbf{u}_j) = 0,$$

where $*$ denotes the usual inner product in real d-space. This is precisely the condition that the components of $\mathbf{u}_i$ and $\mathbf{u}_j$ in the direction $\mathbf{p}_i - \mathbf{p}_j$ are identical, and therefore, do not stretch or compress the rod joining $\mathbf{p}_i$ and $\mathbf{p}_j$. Clearly $\mathbb{V}$, the collection of all infinitesimal motions of a framework $(V, E, \mathbf{p})$, is the solution set of a system of $|E|$ homogeneous linear equations in dn variables and hence is a subspace of real dn-space. Replacing E by the set K, the edge set of the complete graph on V, yields a second solution set. Obviously, this second solution set, which we will denote by $\mathbb{D}$, is a subspace of $\mathbb{V}$. It is not difficult to show that the vector assignments in $\mathbb{D}$ correspond to the initial velocities of the points in $\mathbf{p}$ under the direct isometries or rigid motions of d-space. The vector assignments in $\mathbb{D}$ are called the *trivial* infinitesimal motions of $(V, E, \mathbf{p})$. An infinitesimal motion of $(V, E, \mathbf{p})$ that is not in $\mathbb{D}$ is called an *infinitesimal flex* of $(V, E, \mathbf{p})$. We say that a framework $(V, E, \mathbf{p})$ is *infinitesimally rigid* if the two subspaces $\mathbb{V}$ and $\mathbb{D}$ are equal. That is to say, $(V, E, \mathbf{p})$ is rigid if each infinitesimal motion of $(V, E, \mathbf{p})$ is trivial.

It is not our purpose in this paper to give a general introduction to infinitesimal rigidity. There are several fine papers that do this. The interested reader might start by reading Roth [8]. In this section we will state and prove those results about infinitesimal rigidity that motivate the definition of an abstract rigidity matroid. In the next section, we will usually state, without proof, the standard results about infinitesimal and generic rigidity that relate to the characterization problem. The few proofs that are included are mentioned because they indicate how we may generalize the results to higher dimensions.

Before we go on, we must introduce some additional notation. Given the vertex set $V = \{1, 2, \cdots, n\}$, we will, as previously stated, let K denote the edge set of the complete graph on V. For any subset $U \subseteq V$, and any subset $E \subseteq K$, we use $E(U)$ to denote the set of edges in E with both endpoints in U; for any subset $E \subseteq K$, we use $V(E)$ to denote the *support* of E, i.e., the set of all vertices that is an endpoint of some edge in E. Thus, $(U, K(U))$ is the complete subgraph on the vertex set U and $(V(E), E)$ is the subgraph of (V, K) induced by the edge set E.

Consider an embedding $\mathbf{p}$ of V in $\mathbb{R}^d$. Each edge ij in K determines a linear functional

$$(1) \qquad (\mathbf{p}_i - \mathbf{p}_j) * (\mathbf{u}_i - \mathbf{u}_j)$$

on the dn-dimensional space of loadings. Thinking of the coordinates of a vector in $\mathbb{R}^{dn}$ as the concatenation of n banks of d coordinates each, we may associate the edge ij with the vectors having all banks zero except the ith bank which is occupied by $\mathbf{p}_i - \mathbf{p}_j$ and the jth bank which is occupied by $\mathbf{p}_j - \mathbf{p}_i$. We could then rewrite (1) as an inner product in $\mathbb{R}^{dn}$:

$$(2) \qquad (0, \cdots, 0, \mathbf{p}_i - \mathbf{p}_j, 0, \cdots, 0, \mathbf{p}_j - \mathbf{p}_i, 0, \cdots, 0) * (\mathbf{u}_1, \cdots, \mathbf{u}_i, \cdots, \mathbf{u}_n).$$

The embedding $\mathbf{p}$ may then be represented by the $(n(n-1)/2)$ by (dn) matrix whose rows, indexed by the edges in K, are the left-hand vectors of (2). This matrix $\mathbf{R}(\mathbf{p})$ is called the *rigidity matrix* of the embedding $\mathbf{p}$. We note that the space $\mathbb{D}$ is the orthogonal complement of the space spanned by the rows of $\mathbf{R}(\mathbf{p})$ and that $\mathbb{V}$ is the orthogonal complement of the space spanned by the rows of $\mathbf{R}(\mathbf{p})$ corresponding to the edges of E. Consider the determinant of each minor of $\mathbf{R}(\mathbf{p})$ as $\mathbf{p}$ varies over all of $\mathbb{R}^{dn}$. A specific embedding $\mathbf{p}$ is defined as *generic* if all the nontrivial minors of $\mathbf{R}(\mathbf{p})$, i.e., the minors with determinants that are not identically zero, have nonzero determinants. It is obvious from this definition that the generic embeddings form a dense open subset of $\mathbb{R}^{dn}$. The next result is also obvious from these definitions.

LEMMA 1. *Let a graph (V, E) and a positive integer d be given, and let $\mathbf{p}$ and $\mathbf{q}$ be any two generic embeddings of V into d-space. Then either $(V, E, \mathbf{p})$ and $(V, E, \mathbf{q})$ are both either rigid or not rigid.*

Another important, but not so obvious, fact is that if $\mathbf{p}$ is a generic embedding of V into d-space, then the concepts of rigidity and infinitesimal rigidity coincide for all frameworks $(V, E, \mathbf{p})$ (Asmow and Roth [1], [2]). In light of Lemma 1 and this fact, we define an edge set $E \subseteq K$ to be *d-rigid* (or simply *rigid*, if the value of d is understood) when the frameworks corresponding to E under the generic embeddings of $V(E)$ into d-space are rigid. It is clear from Lemma 1 that the concept of d-rigid is a "purely combinatorial" concept in that it depends only on the structure of the graph $(V(E), E)$. However it is only for d equal to one or two that d-rigid has a "purely combinatorial" characterization, i.e., a characterization in terms of the structure of the graph $(V(E), E)$ alone. The purpose of this paper is to describe the major attempts that have been made to find a purely combinatorial characterization of "d-rigid" for higher dimensions. We will be particularly interested in dimension three.

We begin our investigation by taking a careful look at infinitesimal rigidity. Actually it is the concept of infinitesimal independents that will play the more fundamental role. Given a finite set V and an embedding $\mathbf{p}$ of V into d-space, a subset $E \subseteq K$ is said to be *infinitesimally independent* (relative to $\mathbf{p}$) if the corresponding functionals are independent (or, equivalently, if the corresponding rows of $\mathbf{R}(\mathbf{p})$ are independent). Clearly the infinitesimally independent sets relative to $\mathbf{p}$ are the independent set of a matroid on the set K. We call this matroid a *d-dimensional infinitesimal rigidity matroid on K* or, specifically, *the infinitesimal rigidity matroid on K defined by* $\mathbf{p}$. We denote the infinitesimal rigidity matroid on K defined by $\mathbf{p}$ by $\mathcal{M}(\mathbf{p})$. If $\mathbf{p}$ is a generic embedding, the structure of $\mathcal{M}(\mathbf{p})$ is uniquely determined by the dimension d and the cardinality of V. In the generic case, we call $\mathcal{M}(\mathbf{p})$ the *generic rigidity matroid on K of dimension d* and denote it by $\mathcal{M}(d, n)$. A set $E \subseteq K$ that is an independent set of $\mathcal{M}(d, n)$ is said to be *d-independent*.

There is one important relation between the infinitesimal rigidity matroid on K given by an arbitrary embedding $\mathbf{p}$ of V into d-space and the generic rigidity matroid on K of dimension d: Clearly if a set of rows of $\mathbf{R}(\mathbf{p})$ is independent for some embedding $\mathbf{p}$ then that set of rows is independent for all generic embeddings. Thus, we have the following lemma.

LEMMA 2. *Let V be a finite set and let $E \subseteq K$. If, for any embedding $\mathbf{p}$ of V in $\mathbb{R}^d$, E is infinitesimally independent, then E is d-independent.*

In dimension one it is not difficult to show that all embeddings are generic: Let $\mathbf{A}$ be the edge-vertex adjacency matrix of the direct graph (V, K) where the edge ij is directed from i to j when $i < j$. The rigidity matrix of an embedding $\mathbf{p}$ is obtained from $\mathbf{A}$ by multiplying the ij row by the constant $\mathbf{p}_i - \mathbf{p}_j$. Since this is an embedding, $\mathbf{p}_i - \mathbf{p}_j \neq 0$, and the nonzero determinants of $\mathbf{R}(\mathbf{p})$ correspond to the nonzero determinants of $\mathbf{A}$, independent of the choice of $\mathbf{p}$. It follows from this observation that there is only one 1-dimensional infinitesimal rigidity matroid on K, namely, the generic rigidity matroid on K of dimension one. We easily check that an independent set of rows of $\mathbf{A}$, and hence $\mathbf{R}(\mathbf{p})$, corresponds to a set of edges which is circuit-free in the undirected graph. Hence the independent sets of $\mathcal{M}(1, n)$ are the forests of (V, K) and $\mathcal{M}(1, n)$ is the usual (cycle) matroid on (V, K).

We will not need to refer to the rigidity matrix any further and, in order to facilitate our remaining computations, we will again re-interpret the expressions in (1) above. We interpret the points $\mathbf{p}_j$ of $\mathbf{p}(V)$ as d-dimensional column vectors and the vectors $\mathbf{u}_1, \cdots,$

$\mathbf{u}_n$ of a loading $\mathbf{u}$ of $\mathbf{p}(V)$ as row vectors. Thus the inner products in (1) and (2) above may be replaced by the matrix product $(\mathbf{u}_j - \mathbf{u}_k)(\mathbf{p}_j - \mathbf{p}_k)$. This product may be thought of as the *strain* induced on the edge jk by the loading $\mathbf{u}$; when this product is positive, we think of the edge jk as being stretched by the loading and, when it is negative, we think of the edge as being compressed by the loading. When $\mathbf{p}$ is fixed, we will denote this induced strain by u_{jk}:

$$(3) \qquad u_{jk} = (\mathbf{u}_j - \mathbf{u}_k)(\mathbf{p}_j - \mathbf{p}_k).$$

Given any matroid on K, we have the following usual matroid concepts:

For $E \subseteq K$ the *rank* of E, $r(E)$, is the cardinality of a maximum independent subset of E;

For $E \subseteq K$ the span or *closure* of E, $c(E)$, is defined by

$$c(E) = \{ij \in K \mid r(E \cup \{ij\}) = r(E)\}.$$

The next lemma contains a useful characterization of the closure operator for the infinitesimal rigidity matroid defined by an embedding $\mathbf{p}$.

LEMMA 3. *Let* $\mathbf{p}$ *be an embedding of* $V = \{1, 2, \cdots, n\}$ *into* $\mathbb{R}^d$ *and consider the closure operator* c *of the infinitesimal rigidity matroid defined by* $\mathbf{p}$.

(a) *If* $E \subseteq K$ *and* $hk \in K$, *then* hk *is not in the closure of* E *if and only if there exists an infinitesimal flex* $\mathbf{u}$ *of* E *such that* $u_{hk} \neq 0$.

(b) *If* $E \subseteq K$, *then* $c(E) \subseteq K(V(E))$.

Proof. To prove part (a), assume first that the edge hk is in $c(E)$ and let $\mathbf{u}$ by any infinitesimal motion of E. Since $hk \in c(E)$, the linear functional associated with hk is a linear combination of the linear functionals associated with the edges in E. Thus for any loading $\mathbf{u}$, u_{hk}, the value of the linear functional associated with hk at $\mathbf{u}$ is a linear combination of the values u_{ij} of the linear functionals associated with the edges ij in E. But $u_{ij} = 0$, for $ij \in E$, and we conclude that $u_{hk} = 0$.

Conversely, assume that hk is not in $c(E)$. Since the linear functional associated with hk is not in the span of the linear functionals associated with the edges in E, there exists a loading $\mathbf{u}$ in $\mathbb{R}^{dn}$ at which all the functionals associated with E are zero and at which the functional associated with hk is not zero. This loading is then an infinitesimal flex of E with $u_{hk} \neq 0$.

Turning to part (b), suppose that hk does not belong to $K(V(E))$. Without loss of generality, we may assume that h is not in $V(E)$. We define the loading $\mathbf{u}$ on $\mathbf{p}(V)$ to be the zero vector at all points of $\mathbf{p}(V)$ except h and to be $\mathbf{u}_h$ at h where $\mathbf{u}_h$ is any nonzero vector not perpendicular to $(\mathbf{p}_h - \mathbf{p}_k)$. Clearly, $\mathbf{u}$ is an infinitesimal flex of E with $u_{hk} \neq 0$. By part (a), hk is not in $c(E)$. □

We may now characterize infinitesimal rigidity in terms of the closure operator.

THEOREM 1. *Let* $\mathbf{p}$ *be an embedding of* $V = \{1, 2, \cdots, n\}$ *into* $\mathbb{R}^d$ *and let* c *be the closure operator of* $\mathcal{M}(\mathbf{p})$. *Then* $E \subseteq K$ *is infinitesimally rigid with respect to* $\mathbf{p}$ *if and only if* $c(E) = K(V(E))$.

Proof. Assume that $c(E) = K(V(E))$ and let $\mathbf{u}$ be any infinitesimal motion of E. Next, let hk be any edge in $K(V(E))$; it follows from Lemma 3(a) that $u_{hk} = 0$. Thus, $\mathbf{u}$ is trivial and we conclude that E is infinitesimally rigid.

Conversely, assume that $c(E) \neq K(V(E))$. We conclude from Lemma 3(b) that $c(E)$ is a proper subset of $K(V(E))$. Let hk belong to $K(V(E)) - c(E)$ and apply Lemma 3(a) to get a loading $\mathbf{u}$ of $p(V)$ which is an infinitesimal motion of E and for which $u_{hk} \neq 0$, i.e., an infinitesimal flex of E. Thus, E is not infinitesimally rigid. □

In order to avoid many special cases, we will now restrict our discussion to frameworks $(V, E, \mathbf{p})$ where the points $\mathbf{p}(V)$ are in general position. An embedding $\mathbf{p}$ of V is said to be *general* if the set $\mathbf{p}(V)$ is in general position, i.e., no three points of $\mathbf{p}(V)$ lie on a line, no four lie on a plane, ..., no set of $d+1$ points lie on a $(d-1)$-dimensional hyperplane. It is not difficult to show that all generic embeddings are general embeddings. Hence, this restriction will not affect our consideration of generic rigidity.

THEOREM 2. *Let* $\mathbf{p}$ *be a general embedding of* $V = \{1, 2, \cdots, n\}$ *in* $\mathbb{R}^d$ *and let* c *denote the closure operator of the infinitesimal matroid determined by* $\mathbf{p}$. *Let* $E, F \subseteq K$ *and suppose that* $|V(E) \cap V(F)| < d$. *Then* $c(E \cup F) \subseteq K(V(E)) \cup K(V(F))$.

Proof. Let $S = V(E) \cap V(F)$. If $|S| < (d-1)$ we may add $d-1-|S|$ points to $\mathbf{p}(V)$ to get a larger set, also in general position. Let H be the unique $(d-2)$-dimensional hyperplane that contains S and the additional points. We note that, since the enlarged set is in general position, any $(d-1)$-dimensional hyperplane which contains H can contain at most one other point of $\mathbf{p}(V)$.

Consider a direction of rotation about H. Let $\mathbf{U}$ denote the vector field that assigns to each point $\mathbf{q}$ of $\mathbb{R}^d$ the vector $\mathbf{U}(\mathbf{q})$ which is perpendicular to the $(d-1)$-dimensional hyperplane containing H and $\mathbf{q}$ in the direction of rotation and having length equal to the distance from $\mathbf{q}$ to H. The vector field $\mathbf{U}$ may be thought of as the initial velocities of a rotation of $\mathbb{R}^d$ with H as axis. We note that, for any $\mathbf{q}_1$ and $\mathbf{q}_2$ in $\mathbb{R}^d$, we have

$$(4) \qquad (\mathbf{U}(\mathbf{q}_1)-\mathbf{U}(\mathbf{q}_2))(\mathbf{q}_1-\mathbf{q}_2)=0,$$

where $\mathbf{q}_i$ is written as a column vector while $\mathbf{U}(\mathbf{q}_i)$ is written as a row vector. Define the loading $\mathbf{u}$ by setting $\mathbf{u}_i$ to be the zero vector for all i in $V - V(F)$ and setting $\mathbf{u}_i$ equal to $\mathbf{U}(\mathbf{p}_i)$ for all i in $V(F)$. By (1) above, $\mathbf{u}$ is an infinitesimal motion of F. Since $\mathbf{U}$ assigns the zero vector to each point in H, $\mathbf{u}$ assigns the zero vector to each point in $V(E)$ and is, therefore, an infinitesimal motion of E. We conclude that $\mathbf{u}$ is an infinitesimal motion of $E \cup F$.

Suppose hk does not belong to $K(V(E)) \cup K(V(F))$; then either hk does not belong to $K(V(E) \cup V(F))$ or either h or k belongs to $V(E) - S$ while the other belongs to $V(F) - S$. In the former case, we may apply Lemma 3(b) and the fact that $V(E) \cup V(F) = V(E \cup F)$ to conclude that hk does not belong to $c(E \cup F)$. Turning to the second case, we assume that $h \in (V(E) - S)$ and $k \in (V(F) - S)$. Since $\mathbf{u}_h$ is the zero vector, since $\mathbf{u}_k$ is perpendicular to the $(d-1)$-dimensional hyperplane containing H and $\mathbf{p}_k$, and since $\mathbf{p}_h$ does not lie in that hyperplane,

$$u_{hk} = (\mathbf{p}_h - \mathbf{p}_k) * (\mathbf{u}_h - \mathbf{u}_k) \neq 0.$$

Thus by Lemma 3(a), hk does not belong to $c(E \cup F)$. □

THEOREM 3. *Let a finite set* V *and a general embedding* $\mathbf{p}$ *be given. Let* F, E *be a subsets of* K. *If* E *and* F *are infinitesimally rigid and if* $|V(E) \cap V(F)| \geqq d$, *then* $E \cup F$ *is infinitesimally rigid.*

Proof. We have that $c(E) = K(V(E))$, $c(F) = K(V(F))$, and we must show that $c(E \cup F) = K(V(E \cup F))$. One of the standard results about the closure operator is that it preserves inclusion. Thus

$$K(V(E)) \cup K(V(F)) = c(E) \cup c(F) \subseteq c(E \cup F).$$

Hence if $j, k \in V(E)$ or $j, k \in V(F)$, then $jk \in c(E \cup F)$. We also have that

$$K(V(E \cup F)) = K(V(E) \cup V(F)).$$

Clearly then, it remains only to show that, if $j \in (V(E) - S)$ and $k \in (V(F) - S)$ where $S = V(E) \cap V(F)$, then $jk \in c(E \cup F)$.

Without loss of generality, we may assume that $V = \{0, 1, \cdots, n\}$, $S \supseteq \{1, \cdots, d\}$, $j = 0$, and $k = n$. Let $\mathbf{u}$ be any infinitesimal motion of $E \cup F$. Since E is rigid, $\mathbf{u}$ is a trivial infinitesimal motion when restricted to $V(E)$. Hence $\mathbf{u}$ agrees with the vector field of a rigid motion of d-space on $V(E)$. Subtracting the value of this vector field from $\mathbf{u}$ at all points in $V(E) \cup V(F)$, we may assume that $\mathbf{u}_i$ is the zero vector for all i in $V(E)$. To apply Lemma 3(a) to deduce that $0_n \in C(E \cup F)$, we must show that $u_{0n} = 0$. Since $0 \in V(E)$, $\mathbf{u}_0 = 0$. Since $\{1, \cdots, d, n\} \subseteq V(F)$ and $\mathbf{u}_i = 0$, for each $i = 1, \cdots, d$, we have that $\mathbf{u}_n$ must be perpendicular to each of the vectors $\mathbf{p}_i - \mathbf{p}_n$, for $i = 1, \cdots, d$. Since $\mathbf{p}(V)$ is in general position, these vectors are independent and span d-space; thus, $\mathbf{u}_n$ must be the zero vector. □

Let (V, K) be a complete graph; a matroid $\mathcal{M}$ on K is a *d-dimensional abstract rigidity matroid on* $\mathbf{K}$ if its closure operator satisfies the following two conditions. For $E, F \subseteq K$:

(a) if $|V(E) \cap V(F)| < d$, then $c(E \cup F) \subseteq K(V(E)) \cup K(V(F))$;

(b) if $|V(E) \cap V(F)| \geqq d$, $c(E) = K(V(E))$, and $c(F) = K(V(F))$, then $c(E \cup F) = K(V(E \cup F))$.

Given the complete graph (V, K) and an abstract rigidity matroid $\mathcal{M}$ on K, note that, by taking $F = \emptyset$ in (a), we have that $c(E) \subseteq K(V(E))$. We define a set $E \subseteq K$ to be *rigid* (in the matroid $\mathcal{M}$) if $c(E) = K(V(E))$. Having made this definition, condition (b) could be restated:

(b) If both E and F are rigid and $|V(E) \cap V(F)| \geqq d$, then $E \cup F$ is rigid.

Let the finite set V and any general embedding $\mathbf{p}$ of V into $\mathbb{R}^d$ be given. By Theorem 1, the concept of rigidity as defined above for abstract rigidity matroids coincides with the original definition of infinitesimal rigidity for infinitesimal rigidity matroids. By Theorems 3 and 4, $\mathcal{M}(\mathbf{p})$ is a d-dimensional abstract rigidity matroid on K. It is interesting to note that there are examples of abstract rigidity matroids which are not infinitesimal rigidity matroids. Such examples arise in the theory of splines and in the theory of hyperconnectivity (see Whiteley [10] and Kalai [5]). Also, there are matroids that arise elsewhere in rigidity theory which are not abstract rigidity matroids; for example, the matroids which correspond to bar and body frameworks (see Whiteley [9]).

Our purpose in the remainder of this paper is to prove some fundamental results valid for all d-dimensional abstract rigidity matroids and to try to discover further properties which may be used to characterize the d-dimensional generic rigidity matroids. The natural problem of characterizing the d-dimensional infinitesimal rigidity matroids will not be considered in this paper.

2. Properties of rigidity matroids.

2. Properties of rigidity matroids. We begin this section with two fundamental results about abstract rigidity matroids. These are generalizations to abstract rigidity matroids of well-known results for infinitesimal rigidity.

THEOREM 4. *Let $\mathcal{M}$ be a d-dimensional abstract rigidity matroid for V and let $E \subseteq K$.*

(a) *If the graph $(V(E), E)$ has a vertex of valence d or less and if F is the edge set of the subgraph obtained by deleting that vertex and the edges containing it, then E is independent if and only if F is independent.*

(b) *If $|V(E)| \leqq (d+1)$, then E is independent.*

Proof. (a) Clearly if E is independent, then F is independent. Assume then that F is independent. We denote the vertices in V by $1, \cdots, n$; we assume that n is the vertex of valence d or less; and we assume that n is adjacent to vertices $1, \cdots, e$ $(e \leqq d)$. Assume that $e \geqq 1$ and let $G = E - \{ne\}$. If $e = 1$, then G equals F and is independent; otherwise we assume by the induction hypothesis that G is independent. Note that $G = F \cup H$, where $H = \{n1, \cdots, n(e-1)\}$, and that $|V(F) \cap V(H)| = (e-1) < d$. By part (a) of the definition of an abstract rigidity matroid, $c(G) \subseteq K(V(F)) \cup K(V(H))$

and, since $ne \notin K(V(F)) \cup K(V(H))$, $ne \notin c(G)$. We conclude that $E = G \cup \{ne\}$ is independent.

Turning to part (b), we note that since the empty edge set is independent, $K(U)$, where $|U| \leqq 1$ is independent. If we assume that $U = \{1, \cdots, j\}$ where $2 \leqq j \leqq (d+1)$, we note that the vertex j of $(U, K(U))$ has valence less than or equal to d and that edge set of the subgraph obtained by deleting j is $K(\{1, \cdots, (j-1)\})$. Thus part (b) of this Theorem follows by induction from part (a). □

THEOREM 5. *Let $\mathcal{M}$ be a d-dimensional abstract rigidity matroid for V. Let $U \subseteq V$ and let $E \subseteq K$. Then*:

$$\text{(a) } r(K(U)) = \begin{cases} |U|(|U|-1)/2, & \textit{if } |U| \leqq (d+1), \\ d|U| - d(d+1)/2, & \textit{if } |U| \geqq d; \end{cases}$$

(b) *If E is independent, then for all $F \subseteq E$ with $|V(F)| \geqq d$ we have $|F| \leqq d|V(F)| - d(d+1)/2$.*

Proof. If $|U| \leqq (d+1)$, we conclude from Theorem 4(b) that $K(U)$ is independent. Thus $r(K(U)) = |K(U)| = |U|(|U|-1)/2$. Next assume that $U = S \cup \{1, \cdots, k\}$ where $S \cap \{1, \cdots, k\} = \varnothing$ and $|S| = d$. Let $U_i = S \cup \{i\}$, let $K_i = K(U_i)$, and let $E = K_1 \cup \cdots \cup K_k$. Since E can be partitioned into $K(S)$ and $K_1 - K(S), \cdots, K_k - K(S)$, we have

$$|E| = |S| + dk = (d(d-1)/2) + d(|U| - d) = d|U| - d(d+1)/2.$$

The second line of part (a) will follow if we can show that E is both independent and rigid. To this end let $E_i = K_1 \cup \cdots \cup K_i$, for $i = 1, \cdots, k$; note that $E_1 = K_1$ and $E = E_k$. Also note that E_1 is rigid by definition and independent by Theorem 4(b). We proceed by induction to show that E_i is both rigid and independent and rigid for $i = 1, \cdots, k$.

Assume then that $i > 1$ and that $E_{(i-1)}$ is both rigid and independent. First note that $E_i = E_{(i-1)} \cup K_i$ and that $|E_{(i-1)} \cap K_i| = d$; thus by part (a) of the definition of an abstract rigidity matroid, E_i is rigid. Second, note that the vertex i has valence d in $(V(E_i), E_i)$ and that the edge set of the subgraph obtained by deleting this vertex and the edges containing it is $E_{(i-1)}$; thus by Theorem 4(a), E_i is independent.

Turning to part (b), we note that if E is independent then any subset F of E is also independent. If $|V(F)| \geqq d$, we then have

$$|F| = r(F) \leqq r(K(V(F))) = d|V(F)| - d(d+1)/2. \qquad \square$$

Next we state, without proof, two standard results on infinitesimal and generic rigidity. The first is a useful extension of Lemma 2 relating infinitesimal rigidity matroids and generic rigidity matroids.

THEOREM 6. *Let the finite set V be given and let $E \subseteq K$. If, for some general embedding $\mathbf{p}$ of V, E is independent (respectively, rigid) in $\mathcal{M}(\mathbf{p})$, then E is independent (respectively, rigid) in $\mathcal{M}(d, |V|)$.*

The next result is a first step toward finding additional conditions which may serve to distinguish the generic rigidity matroid among the abstract rigidity matroids.

THEOREM 7. *Let the complete graph (V, K) be given and consider $\mathcal{M}(d, |V|)$ on K. Let $E \subseteq K$ and suppose that $(V(E), E)$ has a vertex of valence $d+1$, let S be the set of vertices adjacent to that vertex and, finally, let F be the edge set of the subgraph obtained by deleting that vertex and the $d+1$ edges containing it. Then E is independent in $\mathcal{M}(d, |V|)$ if and only if there is a pair jk of vertices from S so that the edge jk is not in F and the set $F \cup \{jk\}$ is independent in $\mathcal{M}(d, |V|)$.*

These fundamental results give the background needed to discuss the various attacks on the problem of characterizing generic rigidity. In the remainder of this section, we will consider four characterizations of generic rigidity in dimension two. The first of these characterizations is due to Laman [6] and was the first to be proved. Unfortunately, of the four characterizations that we will discuss, it is the only one which is known not to generalize to higher dimensions.

Consider the condition stated in Theorem 5(b). Let d be a fixed positive integer and let V be a fixed finite set. An edge set $E \subseteq K = K(V)$ is said to satisfy *Laman's condition* (*for dimension* d) if, for each subset $F \subseteq E$ with $|V(F)| \geqq d$, we have

$$|F| \leqq d|V(F)| - d(d+1)/2.$$

Theorem 5(b) states that the independent sets of a d-dimensional abstract rigidity matroid satisfy Laman's condition for dimension d. In 1970 Laman [6] gave the first characterization of generic rigidity in dimension two. We restate his result in our terminology.

LAMAN'S THEOREM. *Let the complete graph* (V, K) *be given. Then,* $E \subseteq K$ *is* 2-*independent if and only if* E *satisfies Laman's condition for dimension two.*

The analogous result is valid in dimension one; it is easy to see that E satisfies Laman's condition for dimension one if and only if it contains no cycle, i.e., if and only if it is the edge set of a forest.

As we indicated, Laman's result does not extend to dimension three or higher. Consider the graph (V, E) illustrated in Fig. 1, and assume that $\mathcal{M}$ is any three-dimensional abstract rigidity matroid for V. We easily check that E satisfies Laman's condition for dimension three and the equality $|E| = 3|V(E)| - 6$. On the other hand, since E is the union of two edge sets whose supports intersect in a set of two vertices, E is not rigid (in particular, the pair consisting of the rightmost and leftmost vertices is not in the closure of E). Thus,

$$r(E) < r(K(V(E))) = 3|V(E)| - 6 = |E|;$$

and hence E is not independent. By adding d-3 vertices to V and all edges between any two of these new vertices or between a new vertex and an old vertex, we get a graph which shows that Laman's Theorem could not extend to d-space, for any d greater than three.

The remaining characterization of generic rigidity in 2-space which we will discuss here *may* be extendible to 3-space. That is, there are no known examples analogous to the one in Fig. 1 that show the extensions to 3-space to be false.

The next characterization of the two-dimensional generic rigidity matroids was given by Henneberg [7]. Before we can state the Henneberg result, we must introduce the concept of isostatic sets: an edge set $E \subseteq K$ is d-isostatic if it is both d-independent and d-rigid. We easily verify that each d-independent set is a subset of some

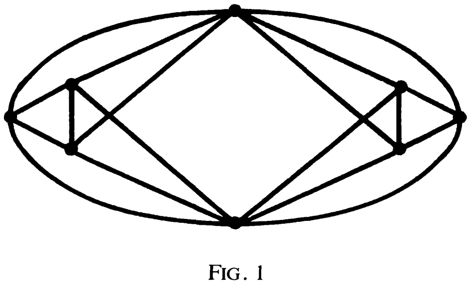


FIG. 1

d-isostatic set with the same support. Thus the d-isostatic sets of $\mathcal{M}(d, n)$ completely determine $\mathcal{M}(d, n)$. Another important observation is that, if E is d-isostatic, then $|E| = d|V(E)| - d(d+1)/2$. A simple counting argument then gives that the average valence in the graph $(V(E), E)$ is less than $2d$. On the other hand if $|V| > d$, it follows from the fact that E is d-rigid and from condition (a) of the definition of an abstract rigidity matroid that every vertex of $(V(E), E)$ has valence at least d. We have proved the following lemma.

LEMMA 4. *Let the finite set V be given and let $E \subseteq K$ be d-isostatic. Then either $E = K(U)$ for some $U \subseteq V$ with $|U| \leqq d$ or the graph $(V(E), E)$ contains only vertices with valence d or more and at least one vertex with valence less than $2d$.*

In 1911 Henneberg [4] described a method for constructing all 2-isostatic sets. A sequence of graphs $(V_1, E_1), \cdots, (V_n, E_n)$ is called a *2-dimensional Henneberg sequence* if (V_1, E_1) consists of a single edge and its endpoints and if, for each index $j\ (2 \leqq j \leqq n)$, (V_j, E_j) is obtained from (V_{j-1}, E_{j-1}) by attaching a new vertex by two edges, or by deleting an edge from (V_{j-1}, E_{j-1}) and attaching a new vertex by three edges in such a way that both endpoints of the deleted edge are adjacent to the new vertex.

THEOREM 8. *Let the complete graph (V, K) be given. An edge set $E \subseteq K$ is 2-isostatic if and only if there is a two-dimensional Henneberg sequence $(V_1, E_1), \cdots, (V_n, E_n)$ such that $(V(E), E) = (V_n, E_n)$.*

We include a proof of this result since it contains the ideas necessary to consider extending this construction to higher dimensions.

Proof. Assume that there is a two-dimensional Henneberg sequence $(V_1, E_1), \cdots, (V_n, E_n)$ with $(V(E), E) = (V_n, E_n)$. We prove inductively that the edge set of each graph in the sequence is 2-isostatic. Clearly, E_1 is 2-isostatic. Assume, then, that E_j is 2-isostatic and apply either Theorem 4(a) or Theorem 7 to conclude that E_{j+1} is 2-isostatic.

Conversely, assume that E is 2-isostatic and prove inductively that any edge set with smaller support and which is 2-isostatic is the edge set of a terminal graph in a two-dimensional Henneberg sequence. By Lemma 4, $(V(E), E)$ contains a vertex n of valence two or three. If the valence is two, let F be the edge set obtained by deleting the two edges containing the vertex n. By Theorem 4(a) F is 2-independent. If the valence is three, apply Theorem 7 and let F be the 2-independent set obtained by deleting the three edges containing the vertex n and adding the appropriate edge between neighbors of n. By a simple counting argument, we see that $|F| = 2|(V(F))| - 3$ and, hence, that F is actually 2-isostatic. Applying the induction hypothesis to $(V(F), F)$ and appending $(V(E), E)$ yields the required sequence. □

We should note that the Henneberg approach does work for dimension one: The 1-isostatic sets are simply the edge set of trees and a one-dimensional Henneberg sequence starts with an isolated vertex and attaches a pendant vertex at each stage.

The next characterization of generic rigidity in dimension two is due to Dress and is based on properties of closed sets. If E is any edge set, the *cliques* of E are the edge sets of the maximal complete subgraphs of the graph $(V(E), E)$.

LEMMA 5. *Let $\mathcal{M}$ be any abstract rigidity matroid of dimension d. If E is a closed set in $\mathcal{M}$ and if $K_1, \cdots, K_k$ are the cliques of E, then*

(1) $E = K_1 \cup \cdots \cup K_k$;

(2) $V(E) = V(K_1) \cup \cdots \cup V(K_k)$;

(3) $|V(K_i) \cap V(K_j)| < d$, *for all distinct i and j.*

Proof. Conclusions 1 and 2 are easily seen to be true for any graph $(V(E), E)$. Now suppose that, for i and j distinct, we have $|V(K_i) \cap V(K_j)| \geqq d$. Since K_i and K_j are rigid we have, by part (b) of the definition of an abstract rigidity matroid, that $K_i \cup$

K_j is rigid. Thus $K(V(K_i) \cup V(K_j))$ is a subset of E, contradicting the assumption that K_i and K_j are distinct cliques of E. □

At a conference in Montreal in 1987, Dress pointed out that, if d is one or two and $\mathcal{M}$ is generic, then we have the formula $r(E) = r(K_1) + \cdots + r(K_k)$, for any closed set E. He also pointed out that the natural extension of this formula to higher dimensions avoids the difficulties which occur in attempting to extend Laman's characterization, and he conjectured that the natural extension of this formula would hold in dimension three. A counterexample to the natural extension of this formula to dimension four was produced at the conference. We will discuss in detail the extensions of this formula to higher dimensions in the next section. Here we show that this formula is not only valid in dimension two but actually characterizes generic rigidity in 2-space.

THEOREM 9. *Let $\mathcal{M}$ be a 2-dimensional abstract rigidity matroid on K. $\mathcal{M}$ is generic if and only if, for every closed set E of $\mathcal{M}$,*

$$r(E) = r(K_1) + \cdots + r(K_k),$$

where $K_1, \cdots, K_k$ are the cliques of E.

Proof. Assume that $\mathcal{M}$ is generic. Let E be a closed set and let $K_1, \cdots, K_k$ be its cliques. For each index i, let F_i be a maximal independent set in K_i and let $F = F_1 \cup \cdots \cup F_k$. Since $F \subseteq E$ and $c(F_i) = K_i$, for each index i, we conclude that $c(F) = E$. So $r(E) = r(F)$. Furthermore,

$$|F| = |F_1| + \cdots + |F_k| = r(K_1) + \cdots + r(K_k).$$

It remains only to show that $r(F) = |F|$, i.e., that F is independent.

Suppose that F is dependent and let H be a minimal dependent subset of F (i.e., a cycle). We have by Laman's Theorem that $|H| > 2|V(H)| - 3$ and $|G| \leqq 2|V(G)| - 3$, for each proper subset G of H. Taking $G = H - \{hj\}$, for some edge hj in H, we may conclude from the two inequalities that $V(G) = V(H)$ and $|G| = 2|V(G)| - 3$. Thus, G is rigid and both G and H lie entirely within some $K(U_i)$. But then $H \subseteq F_i$, which is impossible.

Conversely, assume that for every closed set of $\mathcal{M}$, $r(E) = r(K_1) + \cdots + r(K_k)$. We wish to prove that every set which satisfies Laman's condition is independent. Assume that F is such a set; we proceed by induction on $|F|$. Since single edges are independent, we assume that $|F| > 2$ and, inductively, that all proper subsets of F are independent. Let $E = c(F)$ and let $K_1, \cdots, K_k$ be the cliques of E. If $k = 1$, we note that $E = K_1 = K(V(F))$. Thus we have: $r(F) = r(E) = r(K_1) = 2|V(F)| - 3 \geqq |F|$. But $|F| \geqq r(F)$ with equality only if F is independent. Hence F is independent. Now assume that $k > 1$ and let $F_i = F \cap K_i$, for all i. Since by (3) of Lemma 5 the K_i are disjoint, the F_i are pairwise disjoint. Therefore each F_i is a proper subset of F and hence independent. Thus for each i we have that $r(K_i) \geqq r(F_i) = |F_i|$. We then have

$$r(F) = r(E) = r(K_1) + \cdots + r(K_k) \geqq |F_1| + \cdots + |F_k| = |F|.$$

And, as above, we conclude that F is independent. □

The last characterization that we will consider is based on Theorem 6. If $\mathcal{M}_1$ and $\mathcal{M}_2$ are both matroids on K, we say that $\mathcal{M}_1$ *majorizes* $\mathcal{M}_2$ and write $\mathcal{M}_1 > \mathcal{M}_2$ if each independent set in $\mathcal{M}_2$ is also independent in $\mathcal{M}_1$. Theorem 6 characterizes $\mathcal{M}(d, n)$ as the unique maximal d-dimensional infinitesimal rigidity matroid on K (where $V = \{1, \cdots, n\}$) relative to the relation ">." Since the concept of an infinitesimal rigidity matroid has not been characterized combinatorially, Theorem 6 does not yield a combinatorial characterization of generic rigidity. However as a trivial corollary to Laman's Theorem, we have the following characterization of generic rigidity in dimension two.

COROLLARY TO LAMAN'S THEOREM. *$\mathscr{M}(2, n)$ is the unique maximal two-dimensional abstract rigidity matroid on K (where $V = \{1, \cdots, n\}$) under the relation of majorization.*

We now consider generalizations of these characterizations to dimensions three and higher.

3. Generic rigidity in higher dimensions. If we analyze the proof of Theorem 8, we see that Henneberg sequences were defined to reflect the results of Lemma 4, namely, that each 2-isotropic set (excepting a single edge) has either a vertex of valence two or a vertex of valence three. Thus in defining three-dimensional Henneberg sequences, we must take into account the fact that a 3-isotropic set must have a vertex with valance equal to three, four, or five. In 3-space, Theorem 4(a) tells us that we may attach a vertex of valence three to any 3-isostatic set to get another 3-isostatic set. Theorem 7 describes just how to attach a vertex of valence four. What is needed is a set of conditions under which we may attach a vertex of valence five. If we are to extend the Henneberg construction to higher dimensions, we need necessary and sufficient conditions under which we can attach vertices of valence k for any k $(d \leqq k < 2d)$. A natural set of necessary conditions is easy to describe.

Let the complete graph (V, K) be given and let $\mathscr{M}$ be a d-dimensional abstract rigidity matroid on K and let F and G be subsets of K. We say that G *is independent over* F if $r(F \cup G) = r(F) + |G|$. Note that if G is independent over F, then $G \cap F = \varnothing$ and G is independent; furthermore, if F is independent then so is $F \cup G$. Let $S \subseteq V(F)$. We say that S is *free* in F if, for any $U \subseteq S$ with $|U| > d$, there exists a set $G \subseteq K(U)$ such that $|G| = |U| - d$ and G is independent over F. Note that, if $|S| \leqq d$, then S is trivially free in F.

THEOREM 10. *Let the complete graph (V, K) be given and let $\mathscr{M}$ be a d-dimensional abstract rigidity matroid on K. Let $E \subseteq K$, let $h \in V(E)$, let S be the set of neighbors of h, let H be the set of edges with h as endpoint and let $F = E - H$. Then if H is independent over F, S is free in F.*

Proof. As noted above, the result is trivially valid if the valence of h is d or less. The special case where the valence of h is $d + 1$ and E is independent follows from Theorem 7. However, we do not need that special case here. We assume that h has valence at least $d + 1$ and proceed by induction on the valence of h. In showing that S is free in F, we need only consider the case $U = S$, with the cases with U a proper subset of S being taken care of by the induction hypothesis.

Choose $k \in S$, let $E' = E - \{hk\}$, let $S' = S - \{k\}$ and let $H' = H - \{hk\}$. Clearly, H' is independent over F and $hk \notin c(E')$. Our first task is to show that $K(S)$ is not a subset of $c(E')$. Suppose that it is. We would then have that $K(S' \cup \{h\})$ is a subset of $c(E')$, and that $|V(K(S)) \cap V(K(S' \cup \{h\}))| = |S'| \geqq d$. But this would imply

$$hk \in c(K(S) \cup K(S' \cup \{h\}) \subseteq c(E').$$

We conclude that there is an edge $ij \in K(S)$ so that ij is not in $c(E')$. Let $E'' = E' \cup \{ij\}$ and note that $r(E'') = r(E)$. Let $F' = F \cup \{ij\}$. Since

$$r(E'') = r(E') + 1 = r(F) + |H'| + 1 = r(F') + |H'|,$$

we conclude that H' is independent over F'. By the induction hypothesis, there is a set $G' \subseteq K(S')$ so that $|G'| = |S'| - d$ and G' is independent over F'. We easily check then that $G = G' \cup \{ij\}$ has the correct cardinality and is independent over F, thus demonstrating that S is free in F. □

Let the complete graph (V, K) be given and let $\mathscr{M}$ be a d-dimensional abstract rigidity matroid on K. Let $E \subseteq K$, let $h \in V(E)$, let S be the set of neighbors of h, let H be the set of edges with h as an endpoint and let $F = E - H$. If S is free in F and $G \subseteq K(S)$ such that $|G| = |S| - d$ and G is independent over F, then we say that $(V(E), E)$ is a *(d, k)-extension* of $(V(F \cup G), F \cup G)$. A sequence of graphs $(V_1, E_1), \cdots, (V_n, E_n)$ is called a *d-dimensional Henneberg sequence* if (V_1, E_1) is a complete graph on $d + 1$ vertices and if, for each index $2 \leqq j \leqq n$, (V_j, E_j) is a (d, k)-extension of (V_{j-1}, E_{j-1}) with $d \leqq k < 2d$. We easily verify the following corollary to Theorem 10.

COROLLARY. *Let $\mathcal{M}$ be a d-dimensional abstract rigidity matroid for V. If the edge set $E \subseteq K$ is isostatic, then there is a d-dimensional Henneberg sequence $(V_1, E_1), \cdots, (V_n, E_n)$ such that $(V_n, E_n) = (V(E), E)$.*

We could use Henneberg sequences to characterize generic rigidity in d-space, if we could prove that every d-dimensional Henneberg sequence ends with a d-isostatic edge set, i.e., that for $k = d, \cdots, 2d - 1$, a (d, k)-extension of a d-isostatic set is d-isostatic. By Theorem 4(a) we know this to be true for $k = d$, and Theorem 7 states that it is true for $k = (d + 1)$. Hence in dimension three, only the step that attaches a vertex of valence five is in doubt. In general we wish to know if the necessary conditions given in Theorem 10 are sufficient, specifically, the following: Given the complete graph (V, K), $E \subseteq K$ and a vertex h of valence $d + k$ in $(V(E), E)$ such that F (the set of edges in E not containing h) is d-independent and S (the set of neighbors of h in $(V(E), E)$) is free in F, may we conclude that E is d-independent? Actually, the only case for which the answer to this question is not known is the case $d = 3$ and $k = 2$, i.e., the case needed to attach a vertex of valence five in dimension three. The answers to this question are listed in Table 1 below.

The "Yes" entries in the first column follow from Theorem 4(a), those in the second column from Theorem 7. The "Yes" answers in the first and second rows follow from the Laman characterization in dimensions one and two. The example in Fig. 1 shows that a (3, 3)-extension of a 3-isostatic set need not be 3-isostatic: We easily verify that the set of neighbors of a vertex of valence six in this graph is free in the edge set obtained by deleting the six edges containing that vertex. This example may be altered as follows: Replace the rightmost triangle by a k-circuit and add d-3 vertices attached to all other vertices in the graph. The resulting family of graphs demonstrate that, for $d \geqq 3$ and $k \geqq 3$, a (d, k)-extension of a d-isostatic set need not be d-isostatic. This accounts for all of the "No" entries except those in the second column.

The "No" in the (4, 2) position was demonstrated by Maehara [7] and Woodall [11]. The relevant example is easy to describe. Start with the edge set of the complete graph on six vertices from which one edge has been deleted. This edge set is easily seen to be 4-isostatic. Then we can make seven (4, 2)-extensions deleting the fourteen original

TABLE 1
Is a (d, k)-extension of a d-isostatic set always d-isostatic?

	0	1	2	3	· · ·	k	· · ·
1	Yes	Yes	Yes	Yes	· · ·	Yes	· · ·
2	Yes	Yes	Yes	Yes	· · ·	Yes	· · ·
3	Yes	Yes	???	No	· · ·	No	· · ·
4	Yes	Yes	No	No	· · ·	No	· · ·
⋮	⋮	⋮	⋮	⋮		⋮	
d	Yes	Yes	No	No	· · ·	No	· · ·
⋮	⋮	⋮	⋮	⋮		⋮	

edges and ending with the complete bipartite graph $K_{6,7}$. Bloker and Roth [3] have shown that $K_{6,7}$ is not 4-isostatic, so at some step in the construction, a (4, 2)-extension of a 4-isostatic set did not result in a 4-isostatic set. By attaching additional vertices this example may be extended to higher dimensions. Hence the only open case is the one which would enable us to deduce that each 3-dimensional Henneberg sequence ends in a 3-isostatic set. Henneberg believed this to be true but did not prove it; hence we have the following conjecture.

THE HENNEBERG CONJECTURE. *Any* (3, 2)*-extension of a* 3*-isostatic set is* 3*-isostatic and, therefore, any three-dimensional Henneberg sequence terminates in an edge set which is* 3*-isostatic.*

We turn next to the Dress conjecture in dimension three and higher. As stated in the previous section, this conjecture was put forward by Dress at the 1987 Montreal Conference. Dress pointed out that, for a closed set E with cliques $K_1, \cdots, K_m$, the rank of E seems to be given by the "inclusion-exclusion" formula based on rank instead of cardinality:

$$r(E) = -\Sigma_{J\subseteq I}(-1)^{|J|} r\left(\bigcap_{j\in J} K_j\right). \tag{5}$$

In particular, if we consider the closure of the edge set of the "standard counter example graph" in Fig. 1 equality does hold. However, it was discovered at the meeting that the edge set E of the complete bipartite graph $K_{6,7}$ does not satisfy this equality in dimension four: As we noted above, this graph is not 4-isostatic even though it has $42 = 4(13) - 10$ edges. Thus it is not rigid. By symmetry, if one of the edges between two vertices of one of the vertex set were to belong to the closure of E, then all of the edges between vertices in this vertex set would have to belong to $c(E)$, which would imply that E were rigid. We conclude that E is closed and that the cliques of E are its individual edges. In this case then, the right-hand side of (1) sums to 42 while $r(E)$ is less than 42. Thus it is in dimension three that the Dress conjecture is of interest, and in dimension three it has a simpler form.

Let (V, K) be given and consider $\mathcal{M}(3, |V|)$. Let $E \subseteq K$ be a closed set and let $K_i, \cdots, K_m$ be the cliques of E. Then by Lemma 5, any two cliques of E are disjoint or meet in exactly one edge. For each edge ij in E, let $\wp(ij)$ denote one less than the number of cliques containing ij. Reformulating (1) we have the following conjecture.

THE DRESS CONJECTURE. *Let* (V, K) *be given and consider the matroid* $\mathcal{M}(3, n)$ *on* K. *Let* $E \subseteq K$ *be a closed set and let* $K_i, \cdots, K_m$ *be the cliques of* E. *Then*

$$r(E) = \Sigma_{i=1,\cdots,m} r(K_i) - \Sigma_{ij\in E}\wp(ij).$$

The third and final conjectured characterization that we discuss here follows.

THE MAXIMAL CONJECTURE. $\mathcal{M}(d, n)$ *is the unique maximal d-dimensional abstract rigidity matroid on* K *(where* $V = \{1, \cdots, n\}$*) under the relation of majorization.*

It follows directly from the corollary to Theorem 10 that the Henneberg conjecture implies the maximal conjecture for dimension three. But the maximal conjecture could be valid in dimension three even if the Henneberg conjecture were false.

In spite of considerable effort on the part of several researchers, the Henneberg, Dress, and maximal conjectures remain unresolved. We do not yet have a "combinatorial" definition for the matroid $\mathcal{M}(3, n)$. While resolving these conjectures is the central problem in this area, there are several other interesting questions to consider:

(1) How are the Henneberg and Dress conjectures related?

(2) What are the appropriate necessary and *sufficient* conditions for (d, k)-extendibility, for $d \geqq 3$ and $k \geqq 2$?

(3) Is the Maerhara counterexample due to the structure of 4-dimensional abstract matroids or to the structure of real 4-space?

(4) Is there a d-dimensional abstract rigidity matroid ($d \geqq 3$) for which the Dress (inclusion-exclusion) formula holds?

(5) Is there a unique maximal d-dimensional abstract rigidity matroid, perhaps other than $\mathcal{M}(d, n)$, for $d \geqq 3$?

REFERENCES

[1] L. Asmow and B. Roth, *Rigidity of graphs*, Trans. Amer. Math. Soc., 245 (1978), pp. 279–289.

[2] ———, *Rigidity of graphs* II, SIAM J. Appl. Math., 68 (1979), pp. 171–190.

[3] E. Bloker and B. Roth, *When is a bipartite graph a rigid framework?*, Pacific J. Math., 90 (1980), pp. 27–43.

[4] L. Henneberg, *Die Graphische Statik*, Liepzig, 1911.

[5] G. Kalai, *Hyperconnectivity of graphs*, Graphs Combin., 1 (1985), pp. 65–79.

[6] G. Laman, *On graphs and the rigidity of planar skeletal structures*, J. Engrg. Math., 4 (1970), pp. 331–340.

[7] H. Maehara, *On Graver's conjecture concerning the rigidity problem of graphs*, to appear.

[8] B. Roth, *Rigid and flexible frameworks*, Amer. Math. Monthly, 88 (1981), pp. 6–20.

[9] W. Whiteley, *Union of matroids and rigidity of frameworks*, SIAM J. Discrete Math., 1 (1988), pp. 237–255.

[10] ———, *The analogy between algebraic splines and hinged panel structures*, to appear.

[11] D. Woodall, private communication, 1990.

SIAM J. DISC. MATH.
Vol. 4, No. 3, pp. 369-384, August 1991

006

DYNAMIC STEINER TREE PROBLEM*

MAKOTO IMASE† AND BERNARD M. WAXMAN‡

Abstract. This paper proposes a new problem called the *dynamic Steiner tree problem*. Interest in the dynamic Steiner tree problem is motivated by multipoint routing in communication networks, where the set of nodes in the connection changes over time. This problem, which has its basis in the Steiner tree problem on graphs, can be divided into two cases: one in which rearrangement of existing routes is not allowed, and a second in which rearrangement is allowed.

For the nonrearrangeable version, it is shown that the worst-case performance for any algorithm is at least $\frac{1}{2}\lg n$ times the cost of an optimum solution with complete rearrangement. Here n is the maximum number of nodes to be connected. In addition, a simple, polynomial time algorithm is present that has worst-case performance within two times this bound. In the rearrangeable case, a polynomial time algorithm is presented with worst-case performance bounded by a constant times optimum.

Key words. Steiner tree problem, multipoint connection, multipoint routing, approximation algorithm, worst case, communication networks

AMS(MOS) subject classifications. 68Q25, 68M10, 05C05

1. Introduction.

1.1. Background and motivation.

With the growth of interest in flexible multipoint communication networks supporting a wide class of applications, the importance of routing techniques for multipoint connections is being emphasized [9], [10]. Routing a multipoint connection is typically treated as the problem of finding a minimum cost tree connecting a set of nodes. This set of nodes is called the terminal set, and the elements of the terminal set are called terminal nodes. If the terminal set is known in advance, this problem is the classical problem in graph theory known as the *Steiner tree problem on graphs* (ST). ST has been studied extensively [4], [13] including the investigation of distributed algorithms [5].

To support some services, for example, video broadcasts and multiperson conferences [9], we need facilities for adapting to changes in the terminal set. There are relatively few studies dealing with this problem [11] in spite of its practical importance.

The model for multipoint routing assumed in this paper is a connection oriented communication network that can be represented by an undirected graph with edge costs. In addition, since bandwidth is reserved for each connection, communication delay is not the main consideration, but instead the main criteria for route selection is the minimal usage of network resources.

The graph theoretic version of the multipoint routing problem, called the *dynamic Steiner tree problem* (DST), comes in two flavors. In the first version, as new nodes are either added to or removed from a connection, rearrangement of existing routes is not allowed. In the second version, rearrangement is allowed, and we consider not only the cost of the trees generated by an algorithm, but also the number of rearrangements. Of course, if complete rearrangement is allowed whenever the terminal set changes, DST reduces to ST. For the nonrearrangeable version, when a node is removed from a con-

* Received by the editors October 10, 1989; accepted for publication (in revised form) July 25, 1990. This research was supported by Bell Communications Research, Bell Northern Research, Italtel SIT, Nippon Electric Corporation, and National Science Foundation grant DCI-8600947.

† NTT Laboratories, Tokyo, Japan.

‡ Department of Computer Science, Southern Illinois University at Edwardsville, Edwardsville, Illinois 62026-1653.

nection, no link may be added to the connection, and when a node is added, no link may be removed. If rearrangements are allowed, we use a measure of the deviation from the nonrearrangeable case. This measure, called the number of rearrangements, will be made precise in the next section.

In a connection oriented network, it is important to limit rearrangements as a connection evolves. For example, rearranging a large multipoint connection may be time consuming and may require significant use of network resources in the form of CPU time, especially if an attempt is made to reroute the entire connection. Another problem occurs when the network is heavily loaded. In a large network, we assume that control and routing functions will be distributed as opposed to centralized. Thus, rearrangement of a connection may result in the blocking of some parts of the connection as rearrangement proceeds. Note that blocking occurs when there is insufficient bandwidth to support a branch of the connection.

1.2. Overview. Section 2 gives formal definitions for the two versions of DST as well as a definition for the worst-case performance ratio. The definition of worst-case performance presented in this paper is based on the usage of the term *worst-case performance* as applied to approximation algorithms. This measure of performance has been selected not only because it is useful, but also because worst-case measures are generally easier to evaluate than average-case or other probabilistic measures.

Section 3 presents results for the nonrearrangeable version of DST. The primary result presented in this section is given by Theorems 1 and 3. Let n be the maximum cardinality of the terminal sets for an instance of the nonrearrangeable version of DST. If only add requests are allowed, any algorithm, not just polynomial time algorithms, for this problem will have a worst-case performance ratio that is at least $\frac{1}{2}$ lg n times the cost of an optimum solution, i.e., $\frac{1}{2}$ lg n times the cost of a minimum Steiner tree. If, in addition, remove requests are allowed, then no upper bound on the worst-case performance ratio exists. This section also presents a greedy algorithm which has performance within two times the best possible bound for nonrearrangeable algorithms.

Section 4 presents results for the rearrangeable version of DST. A class of algorithms, based on the concept of an edge-bounded tree, is presented here. An algorithm based on two edge-bounded trees is presented and is shown to have a worst-case performance ratio of eight. In addition, this algorithm reduces the number of rearrangements required when compared to standard ST approximation algorithms.

A brief discussion of open questions and areas for further investigation are presented in the concluding section.

2. Definitions. In this paper, all graphs $G = (V, E)$ are both undirected and connected. When it is necessary for clarity, we use the notation $V(G)$ and $E(G)$ to indicate the nodes and edges of graph G. Associated with each graph G is a cost function cost : $E \rightarrow \mathscr{R}^+$ (positive reals), and the distance function dist : $V \times V \rightarrow \mathscr{R}^+$ mapping each pair of nodes from V into $\mathscr{R}^+$. The function dist returns the length of a shortest path between each pair of nodes in V.

An instance of DST consists of a graph $G = (V, E)$ with a cost function and a sequence of requests $R = \{r_0, r_1, \cdots, r_K\}$ where each r_i is a pair (v_i, ρ_i), $v_i \in V$, $\rho_i \in \{add, remove\}$. Request r_i may be viewed as a call of the form $add(v_i)$ or $remove(v_i)$. We let

$$S_i = \{v \mid (v, add) = r_j \text{ for some } j, 0 \leqq j \leqq i \text{ and } (v, remove) \neq r_l \text{ for all } l, j < l \leqq i\}.$$

The set S_i, called the terminal set at step i, is simply the collection of nodes which are to be connected with a Steiner tree after request r_i.

The objective of DST is to find a minimum cost tree connecting each terminal set S_i without knowledge of request r_j for any $j > i$. Thus, DST belongs to the category of on-line problems, which have recently attracted significant interest. DST is divided into two cases. In the first case, once a particular set of edges has been used in a route, no rearrangement is allowed as the algorithm proceeds, while in the second case, rearrangement is allowed.

For the remainder of this paper we let $T_0 = (\{v_0\}, \emptyset)$.

PROBLEM 1 (DST-N). *Given an instance* $I = (G, \text{cost}, R)$, *find a sequence of trees* $\{T_1, T_2, \cdots, T_K\}$ *satisfying the following conditions and minimizing a function of* $\{\text{cost}(T_i) \mid i = 1, 2, \cdots, K\}$. *Here* $\text{cost}(T_i)$ *is* $\sum_{e \in E(T_i)} \text{cost}(e)$.

1. *Each* T_i *spans* S_i.
2. *If* r_i *is an add request,* T_i *includes* T_{i-1} *as a subgraph.*
3. *If* r_i *is a remove request,* T_{i-1} *includes* T_i *as a subgraph.*

Conditions 2 and 3 imply that edges and nodes are added to a tree only for an add request and that they are removed only for a remove request. Even though we will use a specific function, the definition given for DST does not specify this function since there are several which are appropriate. The function used here has been chosen based on its simplicity and its relationship to the usual measure of worst-case performance for approximation algorithms. Given an instance I of DST-N and an algorithm A, let $A(S_i)$ be the cost of the tree generated by algorithm A for terminal set S_i and let $OPT(S_i)$ be the cost of a minimum Steiner tree for S_i. Then the performance of algorithm A on instance I is given by

$$A(I) = \max_{0 < i \leq K} \frac{A(S_i)}{OPT(S_i)}.$$

For all instances I, where the maximum terminal set size $|I|$ is equal to n, we define the worst-case performance ratio of algorithm A to be

$$C_A(n) = \sup \{A(I) \text{ over all instance } I, |I| = n\}.$$

In the *rearrangeable dynamic Steiner tree* problem (DST-R), the restrictions on the relationship between T_{i-1} and T_i is relaxed. Instead, the number of rearrangements required to derive T_i from T_{i-1} is restricted. The number of rearrangements α_i is the number of connected components in $T_{i-1} \cap T_i$ less one, i.e., $\text{comp}(T_{i-1} \cap T_i) - 1$. (Graph intersection is defined in the obvious way.) This definition is motivated by the concept of point-to-point routing. Informally, α_i is the number of point-to-point connections required to derive T_i from T_{i-1}. The network resources required to rearrange a connection are likely to be related to the number of new point-to-point *like* connections. In addition, this is a reasonable method for measuring the difference between two consecutive trees.

We note that there is a variant of DST which allows cycles in a connection. Even though we do not consider this problem here, the definition for the number of rearrangements has a simple extension to this problem [12].

To summarize we give the following definition for DST-R.

PROBLEM 2 (DST-R). *Given an instance* (G, cost, R) *of* DST, *find a sequence of trees* $\{T_1, T_2, \cdots, T_K\}$ *where each* T_i *spans* S_i *and minimize a function of* $\{\text{cost}(T_i) \mid i = 1, 2, \cdots, K\}$ *while not exceeding an upper bound B on the number of rearrangements.*

In § 4, we consider algorithms where an upper bound B is established for the value of $\sum_{i=1}^{K} \alpha_i$. In § 5, we suggest an open question regarding algorithms where each α_i is bounded above by a constant, for example, $\alpha_i \leqq 2$ for all i. Depending on the requirements of a specific application, it is possible to first fix an upper bound on the worst-case performance ratio and then consider bounds achievable on the number of rearrangements, or to first fix an upper bound on the number of rearrangements and then try to minimize the worst-case performance ratio. For example, with polynomial time algorithms it is possible to achieve a worst-case performance ratio of two if no restriction is placed on rearrangements. (Apply one of the well-known Steiner tree heuristics.) It is an open question whether or not this worst-case performance is still achievable while at the same time restricting the number of rearrangements.

3. Nonrearrangeable case. In this section, we consider the worst-case performance ratio for DST-N. Most of the analysis deals with the restriction of DST-N to the case where each request r_i is an add operation. In the more general case, as will be shown in Theorem 3, the worst-case performance of any algorithm for DST-N is unbounded. We should note that in spite of the results presented in this section, experimental studies [11], [12] indicate that the algorithm presented here yields reasonably good average-case performance.

3.1. Performance with add requests. In Theorem 1 we prove that $1 + 1/2\lfloor \lg (n-1) \rfloor$ is a lower bound on $C_A(n)$ for any algorithm A for DST-N if each request r_i is an add request. We prove this result by creating a sequence of instances I_k based on a sequence of graphs G_k. We use an adversary argument to show that instances I_k can be created to yield this result for any algorithm A.

We begin by defining graphs $G_k = (V_k, E_k)$, $k \in Z_0^+$ (nonnegative integers) and a constant cost function c_k on the edges of G_k. G_0 is the complete graph with two nodes and single edge with cost equal to one. The two nodes in G_0, v_0 and v_1, are called level zero nodes. Graph G_k, for $k > 0$, is defined recursively in terms of G_{k-1}. For each edge (u, v) in G_{k-1}, a distinct pair of nodes α, β, is introduced, and the edge (u, v) is replaced with two paths (u, α, v) and (u, β, v). We refer to the nodes α and β as level k sister nodes. Each edge in G_k is then assigned a cost of 2^{-k}. Note that v_0 and v_1 will be connected by (simple) paths of cost 1. (See Fig. 1.)

We say that two nodes $u, v \in V_k$ are i-adjacent, $0 \leqq i \leqq k$ if the level of both u and v is no more than i and there is a path from u to v which has no intermediate node from level j, $j \leqq i$. That is, the corresponding nodes in graph G_i would actually be adjacent. Note that exactly one of two i-adjacent nodes must be a level i node and that the distance between i-adjacent nodes is 2^{-i}.

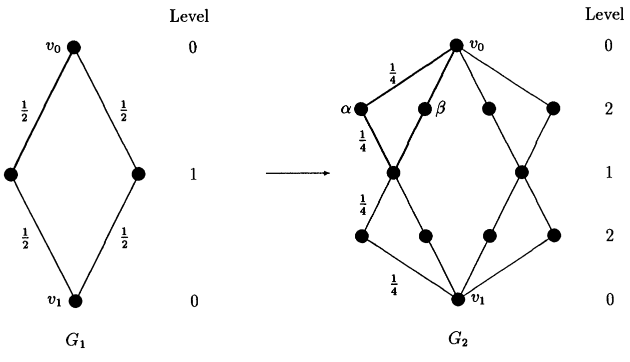


FIG. 1. *Example of* G_k.

LEMMA 1. *In graph G_i, $i > 1$,*

(i) *Each level i node x is adjacent to exactly two nodes, node v at level $i - 1$ and node u at level j, $0 \leqq j < i - 1$. In addition node x has a sister node y at level i which is also adjacent to both u and v.*

(ii) *Each level $i - 1$ node v is adjacent to exactly four level i nodes, consisting of two sister pairs.*

Thus, in graph G_k, $k \geqq i$, (i) *and* (ii) *hold if the term adjacent is replaced by i-adjacent.*

Proof. This lemma follows by a simple induction on i. Figure 2 illustrates the properties described here for two sister pairs α, β and α', β' in graph G_i. □

A sequence of node sets $N = \{N_0, N_1, \cdots, N_k\}$ is called a k-sequence for graph G_k if there exists a path p between v_0 and v_1 such that each N_i is the set of level i nodes in path p. Note that $N_0 = \{v_0, v_1\}$, that $|N_i| = 2^{i-1}$ for $0 < i \leqq k$, and that $\cup_{i=0}^{i=k} N_i = V(p)$. Here $V(p)$ is used to denote the nodes in path p.

We are now ready to present an informal explanation for the main result of this section. The graphs G_k will be used to demonstrate the existence of instances of DST-N which give a lower bound on the worst-case performance for any arbitrary algorithm. As an algorithm A for DST-N proceeds, a k-sequence is generated in a way that will force the cost of a final solution to be as large as possible. More formally, given an instance of DST-N with G_k as the underlying graph, we prove the existence of a k-sequence N which yields the required result.

Finally, we give the definition of a minimal tree sequence $\hat{T} = \{\hat{T}_0, \hat{T}_1, \cdots, \hat{T}_k\}$ for graph G_k, with respect to a k-sequence N. $\hat{T}_0$ is any tree that spans the nodes in N_0 such that no proper subgraph of $\hat{T}_0$ also spans N_0. $\hat{T}_i$, $0 < i \leqq k$ must contain tree $\hat{T}_{i-1}$ as a subgraph and span all of the nodes in N_i from the k-sequence N. The requirement that $\hat{T}_i$ is minimal means that no subgraph of $\hat{T}_i$ also satisfies these properties.

In Lemma 2, we consider a minimal tree sequence $\hat{T}$ that is constructed by an algorithm A which generates each $\hat{T}_i$ based only on knowledge of $\hat{T}_{i-1}$ and N_i. ($\hat{T}$ is a subsequence of the entire sequences of trees generated by A.) We show the existence of node sets N_i, based on $\hat{T}_{i-1}$, $i > 0$, which force the cost of each $\hat{T}_i$ to be large enough to prove Lemma 2. Restricting N to a k-sequence insures that there exists a minimum Steiner tree for the entire terminal set with cost one.

LEMMA 2. *Given the graph G_k, $k \in Z_0^+$, if algorithm A constructs a minimal tree sequence $\{\hat{T}_0, \hat{T}_1, \cdots, \hat{T}_k\}$ for any instance of* DST-N *based on G_k, then there exists a*

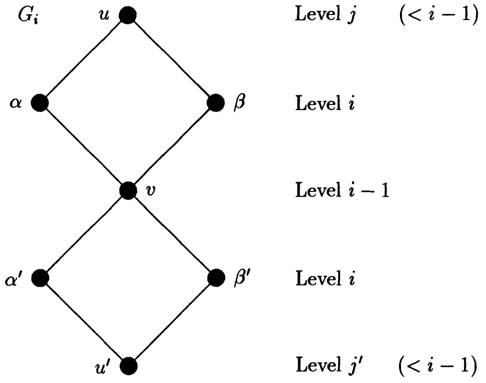


FIG. 2. *Level i sister nodes.*

sequence of requests on G_k corresponding to a k-sequence N such that

$$\text{cost}\,(\hat{T}_i) \geqq 1 + \tfrac{1}{2}i \tag{1}$$

for all i, $0 \leqq i \leqq k$.

Proof. We prove this lemma for each $\hat{T}_i$, $0 \leqq i \leqq k$ by induction on i. Inequality (1) clearly holds for $i = 0$. If we choose N_1 so that it contains the level one node not in $\hat{T}_0$, then cost $(\hat{T}_1) \geqq 1.5$. Note that there is a path of cost one in G_k that contains all nodes in N_0 and N_1, so that $\{N_0, N_1\}$ is an initial segment of some k-sequence.

Informally what follows can be couched in terms of an adversary which selects nodes for N_i based on the choice made by algorithm A in order to ensure the desired result. Assume that (1) holds for every j, $0 \leqq j < i$, where $1 < i \leqq k$, and assume that $\{N_0, N_1, \cdots, N_{i-1}\}$ is an initial segment for some k-sequence. Since $\hat{T}_{i-1}$ is a minimal tree, each leaf node must be in one of the N_j, and there are no cycles in $\hat{T}_{i-1}$. Consider a node v from N_{i-1}. Node v is i-adjacent to exactly four nodes at level i consisting of two sister pairs by Lemma 1(ii). If $\hat{T}_{i-1}$ contains both nodes of a level i sister pair then $\hat{T}_{i-1}$ has a cycle or a leaf node at a level greater than $i - 1$. But this contradicts the minimality of $\hat{T}_{i-1}$, hence $\hat{T}_{i-1}$ can contain at most one of the nodes from each sister pair i-adjacent to v.

For each node $v \in N_{i-1}$ we select one node from each sister pair of v, that is not in $\hat{T}_{i-1}$, to place in N_i. Note that if there is a path of cost one which contains all nodes from every N_j, $0 \leqq j < i$ then there is a path of cost one which also contains the nodes in N_i. Furthermore, N_i will contain all level i nodes in this path by Lemma 1(i). Thus, $\{N_0, N_1, \cdots, N_i\}$ is an initial segment for some k-sequence.

The cost of a shortest path from each of the nodes in N_i to a node in T_{i-1} will be 2^{-i}. Since $|N_{i-1}| = 2^{i-2}$, we will have selected 2^{i-1} level i nodes to include in N_i. Thus, it follows that cost $(\hat{T}_i) \geqq$ cost $(\hat{T}_{i-1}) + \frac{1}{2}$. □

Using Lemma 2 we derive a lower bound for the best possible worst-case performance ratio given any algorithm for DST-N.

THEOREM 1. *Given any algorithm A for* DST-N, *there is an instance* (*G*, cost, *R*) *such that for all i*, $0 < i \leqq K$

$$\frac{A(S_i)}{OPT(S_i)} \geqq 1 + \frac{1}{2}\lfloor \lg (n_i - 1) \rfloor, \tag{2}$$

where $n_i = |S_i|$. *Furthermore, this bound holds even if each request is restricted to node addition.*

Proof. Consider an instance of DST-N (G_k, c_k, R), where $R = \{(u_i, add), 0 \leqq i \leqq K = 2^k\}$, all u_i are distinct with $u_0 = v_0$, $u_1 = v_1$ and for $i > 1$, $u_i \in N_j$, $j = \lceil \lg i \rceil$ for a k-sequence N. Then, by Lemma 2, there exists a k-sequence N which yields the result in (2). □

3.2. Dynamic greedy algorithm. Next, we present the dynamic greedy algorithm (DGA) for DST-N that has performance within two times the best possible bound in the case where each request is restricted to node addition. For each add request, DGA joins the new node by a shortest path to a nearest node already in the connection. In the case of a remove request, a terminal node is dropped by simply deleting the portion of the connection which serves only that terminal node. See Fig. 3 for complete details.

Theorem 2 along with Theorem 1 implies that DGA has a worst-case performance ratio within two of an optimal algorithm for DST-N. In order to prove Theorem 2, we use the following lemma which follows directly from a result due to Rosenkrantz, Stearns, and Lewis [8, Lemma 1].

```
$T_0 := (\{v_0\}, \varnothing)$;   $S_0 := \{v_0\}$;   $i = 1$
for $i \leqq K$ →
  if $r_i$ is an add request →
    Choose the shortest path $p_i$ from $v_i$ to $T_{i-1}$
    $T_i := T_{i-1} \cup p_i$
    $S_i := S_{i-1} \cup \{v_i\}$
  | $r_i$ is a remove request →
    $S_i := S_{i-1} - \{v_i\}$
    $T_i := T_{i-1}$
    do $V(T_i) - S_i$ contains node $w$ with degree 1 →
      $T_i := T_i - w$
    od
  fi
rof
```

FIG. 3. *Dynamic greedy algorithm* (DGA).

LEMMA 3. *Let $G(V, E)$ be a complete graph with a cost function $C: E \rightarrow \mathscr{R}^+$ satisfying the triangle inequality, and let S be any nonempty subset of V with $|S| = i$. If $2P$ is the cost of an optimal tour for S and l is a function $l: V \rightarrow \mathscr{R}^+$ satisfying the following conditions*:

1. $\text{dist}(u, v) \geqq \min(l(u), l(v))$ *for all nodes* $u, v \in S$, *and*
2. $l(v) \leqq P$ *for all* $v \in S$,

then

$$\left(\sum_{v \in S} l(v)\right) - \max_{v \in S} l(v) \leqq (\lceil \lg i \rceil) P.$$

Proof. This lemma follows from an intermediate result that is derived in the proof of Lemma 1 of [8] since the proof holds not only when $S = V$, but for any nonempty subset $S \subseteq V$. □

THEOREM 2. *Let I be any instance of* DST-N *with requests $R = \{r_0, r_1, \cdots, r_K\}$ and let $n_i = |S_i|$ for each terminal set S_i. If each r_i is an add request, then*

$$\frac{\text{DGA}(S_i)}{OPT(S_i)} \leqq \lceil \lg (n_i) \rceil \tag{3}$$

holds for all i, $0 < i \leqq K$.

Proof. In the construction of tree T_i for terminal set S_i, node v_i is connected by a shortest path from v_i to a node in T_{i-1}. Thus, if we let $l(v_i) = \min_{0 \leqq j < i} \text{dist}(v_i, v_j)$ for $1 \leqq i \leqq K$, then the cost of the path selected by DGA to join v_i to T_{i-1} is less than or equal to $l(v_i)$. Let $l(v_0) = \max_{1 \leqq j \leqq K} \text{dist}(v_0, v_j)$, so that $l(v_0) \geqq \max_{0 \leqq j \leqq i} l(v_j)$. Note that $l(v_j) \leqq OPT(S_i)$ for all j, $0 \leqq j \leqq i$, and that $\text{DGA}(S_i) \leqq (\sum_{j=0}^{i} l(v_j)) - l(v_0)$.

Now consider any pair of nodes v_h, v_j in tree T_i and assume, without loss of generality, that $h < j$. It then follows that $l(v_j) \leqq \text{dist}(v_h, v_j)$, so that (1) of Lemma 3 holds. Note that a tour of set S_i can be constructed from a Steiner tree for S_i such that the cost of the tour is no more than twice the cost of the Steiner tree. Thus $P \leqq OPT(S_i)$. Since $l(v_j) \leqq P$ for all j, $0 \leqq j \leqq k$, (2) of Lemma 3 also holds, and the theorem follows. □

A slightly better bound of $\lg(n_i)$ can be proved [3] at the expense of a more complex proof.

3.3. Performance with remove requests. We now consider the general case of DST-N where we allow both the addition and removal of nodes. We show that any algorithm for DST-N has worst-case performance that is unbounded as a function of the number of terminal nodes in the solution tree.

THEOREM 3. *Let A be an algorithm for* DST-N. *For any pair M, $l \in Z^+$, there exists an instance* (G, cost, R) *of* DST-N *and a positive integer j such that*

$$\frac{A(S_i)}{OPT(S_i)} \geqq M \tag{4}$$

for $j < i \leqq j + l$ independent of the number of terminal nodes in S_i.

Proof. Let graph G contain, as a subgraph, an $M + 2$ node cycle C_{M+2} where each edge has cost 1, and let each of the remaining $M + 2$ nodes be connected to a distinct node in cycle C_{M+2} by an edge of cost ε. Let the set R consist of an initial sequence of $M + 2$ add requests, one for each node in C_{M+2}. Thus, T_{M+1} will be a path containing exactly those nodes in the cycle C_{M+2}. Let the next M steps remove each node of degree two in T_{M+1} to create T_{2M+1}. Then, $T_{M+1} = T_{M+2} = \cdots = T_{2M+1}$ so that the cost of T_{2M+1} is $M + 1$. Since S_{2M+1} contains only two nodes from C_{M+2} the cost of an optimal solution is just one. For the remaining steps, alternately add and remove one of the nodes connected to a leaf node of T_{2M+1}, i.e., one of the nodes in G connected by edge of cost ε to C_{M+2}. The value of ε can be made sufficiently small so that (4) holds. □

4. Rearrangeable case.

In this section, we present a class of algorithm for DST-R, called edge bounded algorithms. Each specific algorithm (EBA(δ)) is determined by a positive real value $\delta \geqq 1$. We prove that, for $\delta \geqq 2$, each algorithm has a worst-case performance ratio bounded above by 2δ, and given a sequence of requests $\{r_0, r_1, \cdots, r_K\}$, the total number of rearrangements is $O(K^{3/2})$. For any algorithm which allows complete rearrangement, the total number rearrangements may be as large as $\Omega(K^2)$. In constructing EBA(δ), we first established a bound on the worst-case performance ratio and then attempted to minimize the number of rearrangements.

4.1. Edge-bounded trees and extension trees.

If the number of rearrangements is not restricted, DST-R is equivalent to ST for each instance (G, cost, S_i). As a starting point, we apply the minimum spanning tree approximation algorithm (MSTA) for ST [6]. MSTA is one of the most well-known approximation algorithms for ST, because, in spite of its simplicity, it has the best worst-case behavior among all known polynomial time approximation algorithms. The cost, MSTA(S), of T_{MST}, a tree generated by MSTA for node set S, is never more than twice optimal. However, if we apply MSTA to DST-R, the number of rearrangements can be very large. For the graph shown in Fig. 4, the number of rearrangements for each step i is $i - 1$, which yields $\Omega(K^2)$ rearrangements.

The proposed algorithm is based on properties of both the δ edge-bounded trees and the extension trees defined below.

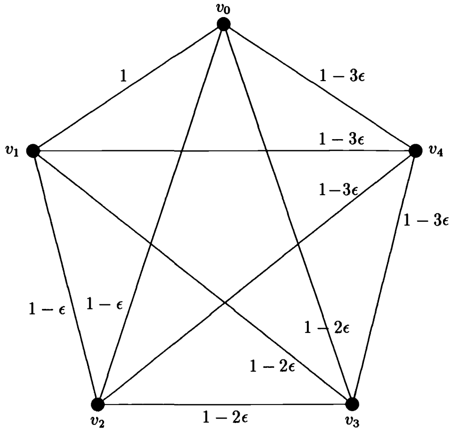
DEFINITION 1. Let u and v be nodes in tree T. If u and v satisfy the following condition, they are called a δ edge-bounded pair. For any $e \in p(u, v, T)$,

$$\text{cost}(e) \leqq \delta \cdot \text{dist}(u, v), \tag{5}$$

where $p(u, v, T)$ is the set of edges on the path between u and v in T. Furthermore, if every pair of nodes in T is δ edge-bounded, T is called a δ edge-bounded tree.

The trees T_{MST} generated by MSTA are 1 edge-bounded trees. Thus, the δ edge-bounded tree can be considered to be a generalization of T_{MST}.

In order to simplify the proofs that follow, for the remainder of this section, we restrict the underlying graph G to a *distance graph*, i.e., a complete graph with a cost function obeying the triangle inequality. For a given graph $G' = (V', E')$ and a cost



$cost(v_i, v_j) = 1-(i-1)\epsilon$ where $i > j$ and $\epsilon \ll 1$.

$R = (v_0, add), (v_1, add), \ldots, (v_K, add)$.

FIG. 4. *Example for maximum number of rearrangements.*

function cost′, we can construct a distance graph $G = (V, E)$. In G, let $V = V'$, $E = \{(u, v) | u, v \in V\}$ and $\text{cost}((u, v)) = \text{dist}(u, v, G')$, the distance between u and v in G'. (Since G is a complete graph we will use the notation $\text{cost}(u, v)$ to indicate the cost of the edge joining the pair u, v in G.) Even though we assume that the underlying graph G is a distance graph, the results obtained in this section remain valid in general.

Note that the cost of an optimum Steiner tree for a node set $S \subseteq V$ in G is the same as cost of an optimum Steiner tree for S in G', and that, in G, MSTA simply constructs a minimum spanning tree for the subgraph induced by S.

DEFINITION 2. For a node set S, if tree $T = (V, E)$ satisfies the following conditions, T is called an extension tree for S.

- $S \subseteq V$.
- For any node x in $V - S$, the degree, in T, of x is greater than two.

LEMMA 4. *If $T = (S, E)$ is a δ edge-bounded tree, then*

$$\text{cost}(T) \leqq \delta \cdot \text{MSTA}(S) \leqq 2\delta \cdot OPT(S). \tag{6}$$

Proof. In (6) the right inequality is valid from [6]. Let $T_{\text{MST}} = (S, E_{\text{MST}})$ be the tree generated by MSTA for set S. Since T and T_{MST} are trees and their node sets are the same, $|E| = |E_{\text{MST}}|$. Assume that there exists a one-to-one function f from E_{MST} to E such that if $f(e) = e'$, the edge e' is contained in $p(u, v, T)$, where u and v are the two endpoints of e. Since T is a δ edge-bounded tree, $\text{cost}(f(e)) \leqq \delta \cdot \text{cost}(e)$. Thus,

$$\text{cost}(T) = \sum_{e' \in E} \text{cost}(e') = \sum_{e \in E_{\text{MST}}} \text{cost}(f(e)) \leqq \sum_{e \in E_{\text{MST}}} \delta \cdot \text{cost}(e) = \delta \cdot \text{MSTA}(S). \tag{7}$$

We complete this proof by showing the existence of f.

For an edge $e = (u, v) \in E_{\text{MST}}$, let $\Gamma(e)$ be the set of edges on the path between u and v in T, that is, $p(u, v, T)$. From Hall's theorem [2, Thm. 5.1.1, p. 45], there exists a one-to-one function f if and only if

$$|\Gamma(X)| \geqq |X|, \quad \forall X \subseteq E_{\text{MST}}, \tag{8}$$

where $\Gamma(X) = \cup_{e \in X} \Gamma(e)$.

Let X be an arbitrary subset of E_{MST}. Then $|V(X)| \leqq |V(\Gamma(X))|$ since every node in $V(X)$ is in $V(\Gamma(X))$, where $V(X)$ is the set of nodes incident with an edge in X. Every pair of nodes $u, v \in V(X)$, connected by an edge in X, is also connected by a path consisting of edges in $\Gamma(X)$. Hence, (8) holds, since neither set of edges contains any cycles. □

LEMMA 5. *If $T = (V, E)$ is a δ edge-bounded extension tree for S, then the following holds*:

$$(9) \qquad \mathrm{cost}\,(T) \leqq 2\delta \cdot \mathrm{MSTA}(S) \leqq 4\delta \cdot OPT(S).$$

Proof. Let $T_{\mathrm{MST}} = (S, E_{\mathrm{MST}})$ be the tree generated by MSTA for set S, and assume that there exists a function g from E_{MST} to the power set 2^E satisfying the following three conditions.

1. If $e' \in g((a, b))$, then $e' \in p(a, b, T)$.
2. For all $e \in E_{\mathrm{MST}}$, $|g(e)| \leqq 2$.
3. For all $e' \in E$, there is some $e \in E_{\mathrm{MST}}$ such that $e' \in g(e)$.

If $e' \in g(e)$, then cost $(e') \leqq \delta \cdot$cost (e) follows from condition 1 and the fact that T is a δ edge-bounded tree. Let cost $(g(e)) = \sum_{e' \in g(e)}$ cost (e'), then cost $(g(e)) \leqq 2\delta \cdot$cost (e) follows from condition 2. In addition, $E \subseteq \bigcup_{e \in E_{\mathrm{MST}}} g(e)$ follows from condition 3. Therefore,

$$(10) \quad \mathrm{cost}\,(T) = \sum_{e' \in E} \mathrm{cost}\,(e') \leqq \sum_{e \in E_{\mathrm{MST}}} \mathrm{cost}\,(g(e)) \leqq \sum_{e \in E_{\mathrm{MST}}} 2\delta \cdot \mathrm{cost}\,(e) = 2\delta \cdot \mathrm{MSTA}(S).$$

Next, we now show the existence of a function g by induction on $|S|$.

If $|S| = 2$, T_{MST} has only one edge, denoted by e. It is clear that $T = T_{\mathrm{MST}}$ by Definition 2. Thus, $g(e) = \{e\}$ satisfies conditions 1, 2, and 3.

Let $|S| = n + 1 \geqq 3$ and let $T = (V, E)$ be a δ edge-bounded extension tree for S. Then, there is a node $v \in S$ with degree one in $T_{\mathrm{MST}}(\deg(v, T_{\mathrm{MST}}) = 1)$. Let $S' = S - \{v\}$, $T' = (V', E')$ be a δ edge-bounded extension tree for S', and $T'_{\mathrm{MST}} = (S', E'_{\mathrm{MST}})$ be an MSTA tree for S'. We show how to construct T' for S', and a function g mapping E_{MST} to 2^E from a function g' mapping E'_{MST} to $2^{E'}$. There are three cases to consider depending on $\deg(v, T)$. Without loss of generality, assume that $E'_{\mathrm{MST}} = E_{\mathrm{MST}} - \{(v, w)\}$ where w is the unique node adjacent to v in T_{MST}. (The tree with the edge set E'_{MST} can be generated by MSTA.)

Case 1 ($\deg(v, T) > 2$). Since $u \in V - S$ implies that $\deg(u, T) > 2$ and since $\deg(v, T) > 2$, it follows that $T' = T$ is an extension tree for S'. From the inductive hypothesis, there exists a function $g' : E'_{\mathrm{MST}} \to 2^E$. The function $g : E_{\mathrm{MST}} \to 2^E$ is

$$g(e) = \begin{cases} g'(e) \text{ if } e \neq (v, w), \\ \varnothing \quad \text{ if } e = (v, w). \end{cases}$$

Case 2 ($\deg(v, T) = 2$). Let x and y be the nodes adjacent to v in T. Without loss of generality, assume that the path from v to w goes through x. (It is possible that $x = w$.) Let $V' = V - \{v\}$ and $E' = (E - \{(v, x), (v, y)\}) \cup \{(x, y)\}$. Thus, $\deg(u, T') > 2$ for every node $u \in V' - S'$ since $V' - S' = V - S$ and $\deg(u, T') = \deg(u, T)$. It follows that T' is an extension tree for S', and there exists a function g': $E'_{\mathrm{MST}} \to 2^{E'}$. Now define the function g as follows:

$$g(e) = \begin{cases} g'(e) & \text{if } e \neq (v, w) \text{ and } (x, y) \notin g'(e), \\ \{(v, x)\} & \text{if } e = (v, w), \\ (g'(e) - \{(x, y)\}) \cup \{(v, y)\} & \text{if } (x, y) \in g'(e). \end{cases}$$

Function g clearly satisfies conditions 2 and 3 since each of the three cases in the definition of g are mutually exclusive. If $e = (v, w)$, condition 1 is satisfied, since the path between v and w goes through x. If a path $p(s, v, T')$ contains edge (x, y), then $p(s, v, T)$ contains (v, x) and (v, y). Thus, condition 1 holds when $(x, y) \in g'(e)$. Condition 1 holds in the remaining case when $g(e) = g'(e)$.

Case 3 ($\deg(v, T) = 1$). Let x be the node adjacent to v in T. If $x \notin S'$, then $x \notin S$ which implies $\deg(x, T) > 2$. We proceed by considering two subcases.

Case 3.1 ($x \in S'$ or $\deg(x, T) \geqq 4$). Let $V' = V - \{v\}$, $E' = E - \{(v, x)\}$, and let

$$g(e)=\begin{cases} g'(e) & \text{if } e\neq(v,w),\\ \{(v,x)\} & \text{if } e=(v,w).\end{cases}$$

Case 3.2 ($x \notin S'$ and $\deg(x, T) = 3$). Let y, z be the two nodes adjacent to x in addition to v, and assume that the path between v and w goes through y. Let $V' = V - \{v, x\}$, $E' = (E - \{(v, x), (x, y), (x, z)\}) \cup \{(y, z)\}$, and let

$$g(e)=\begin{cases} g'(e) & \text{if } e\neq(v,w) \text{ or } (y,z)\notin g'(e),\\ \{(v,x),(x,y)\} & \text{if } e=(v,w),\\ (g'(e)-\{(y,z)\})\cup\{(x,z)\} & \text{if } (y,z)\in g'(e).\end{cases}$$

We can verify that g satisfies conditions 1, 2, and 3 in a manner similar to that used for Case 2. □

4.2. Algorithm. This section presents the algorithm EBA(δ) for DST-R which generates δ edge-bounded extension trees for each S_i. Figure 5 presents the details of EBA(δ). We remind the reader that the underlying graph G is a distance graph so that G is a complete graph. To apply EBA(δ) to an instance on an arbitrary graph G', first construct the equivalent distance graph G, apply EBA(δ) to the instance based on G, and then convert the solution trees in G to solution trees in G' by converting each edge (u, v) to a shortest path from u to v in G'. (This is reminiscent of MSTA [6].)

For a remove request (v_i, *remove*), if $\deg(v_i, T_{i-1}) > 2$, then $T_i = T_{i-1}$. Otherwise v_i is deleted from T_{i-1} to form T. If $\deg(v_i, T_{i-1}) = 1$ all remaining leaf nodes of T not in S_i are also deleted from T. If $\deg(v_i, T_{i-1}) = 2$, then the two components of T are joined by adding an edge e to T. The edge e is selected to minimize the cost of the maximum cost edge in $p(w_0, w_1, T)$ where w_0 and w_1 are the nodes adjacent to v_i in T_{i-1}. Furthermore, if the resulting tree T has a nonterminal node with degree two, delete one of these nodes, and repeat the step for the case $\deg(v_i, T_{i-1}) = 2$. Continue until all nonterminal nodes have degree three or more.

For an add request (v_i, *add*), EBA(δ) joins v_i to T_{i-1} by a shortest edge to create T. Then it determines if every pair v_i and $u \in V(T_{i-1})$ is δ edge-bounded in T. If not, the maximum cost edge in $p(u, v_i, T)$ is replaced by the edge (u, v_i). If, at this point, tree T is not an extension tree for S_i, T is modified by the *remove* procedure above.

The trees generated by EBA(δ) at each step i are extension trees for S_i. Thus, if we can show that these trees are δ edge-bounded trees, Lemma 5 will give us an upper bound of 4δ on the worst-case performance ratio for EBA(δ).

LEMMA 6. *For any $\alpha \geqq 1$ and $\delta \geqq 1$, if a pair of nodes $x, y \in S_i$ is α edge-bounded in an intermediate tree T generated by* EBA(δ), *then the pair is α edge-bounded in any intermediate tree generated after T as* EBA(δ) *constructs T_i from T_{i-1}.*

Proof. Let T be an intermediate tree, and let T' be the tree constructed from T by one of the four elementary operations performed by EBA(δ) on the intermediate tree T

```
EBA($\delta$)($G$, cost, $R$)
  $T_0 := (\{v_0\}, \emptyset)$; $S_0 := \{v_0\}$; and $i = 1$
  for $i \leqq K$ →
    if $r_i$ is an add request →
      $S_i := S_{i-1} \cup \{v_i\}$;   $T_i$ :=add($v_i$, $T_{i-1}$,$S_i$)
    | $r_i$ is a remove request →
      $S_i := S_{i-1} - \{v_i\}$;   $T_i$ :=remove($T_{i-1}$,$S_i$)
    fi
  rof
end EBA($\delta$)

add($v$,$T$,$S$)
{ Let $U$ be the set of edges between $v$ and $T$. }
  Select the minimum cost edge ($u$, $v$) from $U$
  $T := T + (u, v)$ and $U := U - (u, v)$
  do $U \neq \emptyset$ →
    Select the minimum cost edge ($u$, $v$) from $U$
    $U := U - (u, v)$
    Find a maximum cost edge, $d$, in $p(u, v, T)$
    if cost($d$) > $\delta$·cost($u$, $v$) → $T := T - d + (u, v)$  fi
  od
  return(remove($T$,$S$))
end add

remove($T$,$S$)
  $W := V(T) - S$ where V(T) is the node set of $T$
  for $w \in W$ →
    if deg($w$,$T$) = 1 → $T := T - w$
    | deg($w$,$T$) = 2 →
      Let $w_0$ and $w_1$ be the nodes adjacent to $w$
      Let $C_0$ and $C_1$ be the connected components of $T - w$
      Select two nodes $u_0 \in C_0$ and $u_1 \in C_1$ which minimizes $h(u_0, u_1)$*
      $T := T - w + (u_0, u_1)$
    fi
  rof
  return($T$)
end remove
* $h(u_0, u_1) = \max\{\text{cost}(e) \mid e \in p(w_0, w_1, T - w + (u_0, u_1))\}$
```

FIG. 5. *Edge-bounded algorithm* (EBA(δ)).

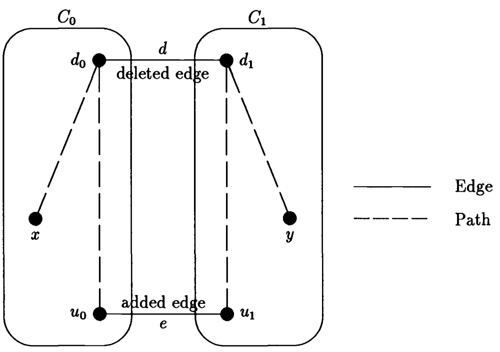
during step i. Note that elementary operations 1 and 2 may be executed only in response to an add request, while elementary operations 3 and 4 may be executed in response to either an add or remove request.

1. Add the node v_i to T by joining v_i to T with a minimum cost edge e.

2. If a pair of nodes u_0, u_1 is not δ edge-bounded in T, remove a maximum cost edge $d = (d_0, d_1)$ in $p(u_0, u_1, T)$ and join the two components by the edge $e = (u_0, u_1)$. (One of the nodes u_0, or u_1 will be v_i.)

3. Remove node v, with deg $(v, T) = 1$, along with the incident edge. (The first time this operation is performed in response to a remove request v will be v_i.)

4. Remove node v with deg $(v, T) = 2$ and the two incident edges, (v, w_0) and (v, w_1). Then join the two components created, by an edge $e = (u_0, u_1)$ that minimizes $h(u_0, u_1)$ where the function $h(u_0, u_1)$ returns the cost of the maximum edge in $p(w_0, w_1, T - v + (u_0, u_1))$. (The first time this operation is performed in response to a remove request, v will be node v_i.)



$T = C_0 \cup C_1 \cup (d_0, d_1) \qquad T' = C_0 \cup C_1 \cup (u_0, u_1)$

FIG. 6. *Elementary operation* 2.

We assume that x and y form an α edge-bounded pair in T and that both nodes are in T'. We then show that they form an α edge-bounded pair in T' for each of the elementary operations.

Operation 1. Since the path between x and y in T' is unchanged, x and y form an α edge-bounded pair in T'.

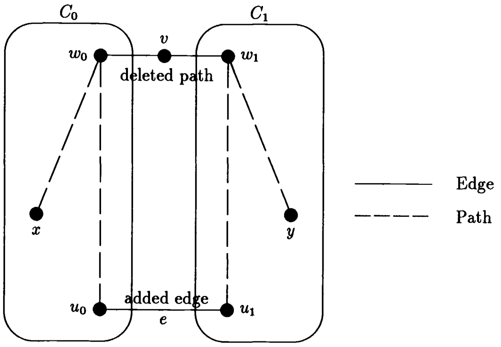
Operation 2. If $d \notin p(x, y, T)$ the path remains unchanged. If $d \in p(x, y, T)$, then $p(x, y, T')$ is a subset of $p(x, y, T) \cup p(u_0, u_1, T) \cup \{e\}$ (see Fig. 6). It suffices to show that for any edge z in $p(x, y, T) \cup p(u_0, u_1, T) \cup \{e\}$,

$$\text{cost}(z) \leqq \alpha \cdot \text{cost}(x, y). \tag{11}$$

If $z \in p(x, y, T)$, (11) holds since the pair x and y is α edge-bounded in T. If $z \in p(u_0, u_1, T)$, (11) holds since d is the maximum cost edge in $p(u_0, u_1, T)$. Finally, (11) holds for edge e since $\delta \cdot \text{cost}(e) < \text{cost}(d)$.

Operation 3. As in the case of Operation 1, the path between x and y is unchanged.

Operation 4. The two components of $T - v$ are denoted by C_0 and C_1. If nodes x and y are in the same component, the path from x to y is unchanged and we are done. If x and y are in different components, we can assume that x, w_0, and u_0 are contained in C_0 and y, w_1, and u_1 are contained in C_1 (see Fig. 7).



$T = C_0 \cup C_1 \cup \{(w_0, v), (v, w_1)\} \qquad T' = C_0 \cup C_1 \cup (u_0, u_1)$

FIG. 7. *Elementary operation* 4.

From the choice of (u_0, u_1), the following inequality holds:

$$\begin{aligned}\max\{\operatorname{cost}(b) \mid b \in p(w_0, w_1, T')\}\\ \leqq \max\{\operatorname{cost}(b) \mid b \in p(x, w_0, C_0) \cup \{(x, y)\} \cup p(y, w_1, C_1)\}.\end{aligned} \tag{12}$$

Since x and y form an α edge-bounded pair in T, the right side of (12) is not larger than $\alpha \cdot \operatorname{cost}(x, y)$. Thus, $\operatorname{cost}(b) \leqq \alpha \cdot \operatorname{cost}(x, y)$ for any

$$b \in p(w_0, w_1, T') \cup p(x, w_0; C_0) \cup p(y, w_1, C_1).$$

On the other hand, $p(x, y, T')$ is a subset of $p(w_0, w_1, T') \cup p(x, w_0; C_0) \cup p(y, w_1; C_1)$. Consequently, any edge in $p(x, y, T')$ is not larger than $\alpha \cdot \operatorname{cost}(x, y)$. □

Lemma 6 is useful for estimating the number of rearrangements in addition to showing the following theorems. In the theorems and lemmas that follow r, assume that an instance of DST-R is given with $R = \{r_0, r_1, \cdots, r_K\}$.

THEOREM 4. *Any tree T_i generated by* EBA(δ) *is a δ edge-bounded extension tree for S_i. Hence the inequality*

$$\operatorname{cost}(T_i) \leqq 2\delta \cdot (\operatorname{MSTA}(S_i)) \leqq 4\delta \cdot (OPT(S_i)) \tag{13}$$

for $0 \leqq i \leqq K$.

Proof. Since T_i is clearly an extension tree for S_i, it is sufficient to show that T_i is δ edge-bounded. Certainly T_0 is δ edge-bounded. Assume that T_{i-1} is δ edge-bounded. From Lemma 6, every pair of nodes in T_i which are also in T_{i-1} is δ edge-bounded. Now considered a node pairs v_i and $w \in T_{i-1}$ when $r_i = (v_i, add)$. EBA(δ) examines each of these pairs. If a pair v_i, w is not δ edge-bounded, then EBA(δ) modifies the tree so that the pair becomes 1 edge-bounded. Thus, T_i is a δ edge-bounded tree and the theorem follows from Lemma 5. □

THEOREM 5. *If every r_i is an add request, then every tree T_i generated by* EBA(δ) *satisfies the inequality*

$$\operatorname{cost}(T_i) \leqq \delta \cdot (\operatorname{MSTA}(S_i)) \leqq 2\delta \cdot (OPT(S_i)). \tag{14}$$

Proof. If every request is an add request, the set of nodes in T_i is S_i. Since T_i is δ edge-bounded, the theorem follows from Lemma 4. □

4.3. Total number of changes. We now determine an upper bound on the total number of rearrangements for EBA(δ) when $\delta \geqq 2$. Note that if an edge (u, v) is contained in T_i, then the pair u and v is a 1 edge-bounded pair in T_i since the path from u to v is just (u, v). We proceed by finding other 1 edge-bounded pairs.

If r_i is an add request, let L_i be the set of endpoints for the edges added at step i incident with v_i. The cardinality of L_i is one more than the number of edges added since v_i is one of the endpoints for each edge. Therefore, the number of rearrangements β_i at step i due to elementary operation 2 is $|L_i| - 2$. If r_i is a remove request, let $L_i = \varnothing$.

LEMMA 7. *Every pair of nodes in L_i is* 1 *edge-bounded in T_j, $i \leqq j \leqq K$, as long as neither node has been removed by any request r_m, $i < m \leqq j$.*

Proof. Let $L_i = \{v_i, w_1, w_2, \cdots, w_l\}$. Since T_i contains edge (v_i, w) for all $w \in L_i$, each pair v_i, w is 1 edge-bounded in all $T_j, j \geqq i$, from Lemma 6 (assuming neither node has been removed). Next consider a pair w, w' in L_i, and assume that $\operatorname{cost}(v_i, w) \leqq \operatorname{cost}(v_i, w')$. Let T and T' be the intermediate trees just before and after the edge (v_i, w') is added by elementary operation 2. Then $T' = T - \{d\} + (v_i, w')$ where d is the edge deleted. Note that T and T' both contain the edge (v_i, w).

Since the pair of w and w' is δ edge-bounded in T and d is an edge in $p(w, w', T)$, $\operatorname{cost}(d) \leqq \delta \cdot \operatorname{cost}(w, w')$. Since d is replaced with (v_i, w'), it follows that $\delta \cdot \operatorname{cost}(v_i, w') < \operatorname{cost}(d)$. Hence both $\operatorname{cost}(v_i, w') < \operatorname{cost}(w, w')$ and $\operatorname{cost}(v_i, w) <$

cost (w, w') hold. Since the path from w to w' in T_i is (w, v_i, w'), the pair of nodes w and w' is 1 edge-bounded in all $T_j, j \geqq i$. □

LEMMA 8. *If $\delta \geqq 2$, for all i and j, $0 < i < j \leqq K$ and each add request adds a different node, the following holds:*

$$|L_i \cap L_j| \leqq 1.$$

Proof. We assume $|L_i \cap L_j| \geqq 2$ and derive a contradiction. If either r_i or r_j is a remove request the intersection is empty. Assume both r_i and r_j are add requests. Let w and w' be nodes in $L_i \cap L_j$, and assume $j > i$. Note that edges (v_j, w) and (v_j, w') are added at step j since $r_j = (v_j, add)$. Without loss of generality, we can assume that

$$(15) \qquad \text{cost}(v_j, w) \leqq \text{cost}(v_j, w').$$

Then from the triangle inequality, $\text{cost}(w, w') \leqq \text{cost}(w, v_j) + \text{cost}(w', v_j) \leqq 2\,\text{cost}(w', v_j)$. Consider the elementary operation which adds (v_j, w') to an intermediate tree T. It is clear that

$$(16) \qquad p(v_j, w', T) \subseteq \{(v_j, w)\} \cup p(w, w', T).$$

Since $w, w' \in L_i$, the pair w, w' is 1 edge-bounded by Lemma 7. Thus, for any $e \in p(w, w', T)$,

$$(17) \qquad \text{cost}(e) \leqq \text{cost}(w, w') \leqq 2\,\text{cost}(v_j, w').$$

From (15), (16), and (17), pair v_j and w' is 2 edge-bounded, which is a contradiction since EBA(2) adds edge (v_j, w') only if (v_j, w') is not 2 edge-bounded. □

The next theorem derives the bound $O(K^{3/2})$ on the number of rearrangements needed by EBA(δ) for $\delta \geqq 2$. In this theorem, let K_a be the number of add requests in R and K_r be that of remove requests.

THEOREM 6. *For any instance $(G$, cost, $R)$, if $\delta \geqq 2$, then the upper bound on the total number of rearrangements required by* EBA(δ) *is given by*

$$(18) \qquad \sum_{i=0}^{K} \alpha_i \leqq \frac{1}{2} K_a(\sqrt{4K_a - 3} - 3) + K_r.$$

Proof. Let R_a and R_r be the sets of add and remove requests in R, respectively. The total number of rearrangements due to elementary operations 3 and 4 is bounded above by the number of remove requests, i.e., $K_r = |R_r|$, since one of these rearrangements occurs only for those nodes deleted from tree T_{i-1} in generating T_i. A node is deleted only if it is a node in one of the remove requests r_j, $0 < j \leqq i$.

In order to determine $\sum_{r_i \in R_a} \beta_i$, the number of rearrangements due to elementary operation 2, we consider the bipartite graph $B = (V_1, V_2; E)$ where $V_1 = R_a$, $V_2 = \bigcup_{i=1}^{K} L_i$ and $E = \{(r_i, v) | v \in L_i, 1 \leqq i \leqq K\}$. It is clear that $|V_1| = |V_2| = K_a$. From Lemma 8, this graph does not contain a complete bipartite graph $K_{2,2}$, i.e., a cycle of length four. We assume, without loss of generality, that each add request is for a distinct node in order to apply Lemma 8. From Theorem 10 of [1, p. 74] we have that $z(n, n) \leqq \frac{1}{2} n(1 + \sqrt{4n - 3})$ where $z(n, n)$ is the maximum number of edges in a bipartite graph with n nodes in each partition and with no 4-cycles. Since the number of edges in G is equal to $\sum |L_i|$ and since each $\beta_i = |L_i| - 2$,

$$(19) \qquad \sum_{r_i \in R_a} \beta_i \leqq \tfrac{1}{2} K_a(\sqrt{4K_a - 3} - 3). \qquad □$$

We suspect that $O(K^{3/2})$ growth given by Theorem 6 is not tight, and conjecture that the total number of rearrangements is not larger than $K = K_a + K_r$.

5. Conclusions. In this paper we have presented a new problem called the dynamic Steiner tree problem. For the nonrearrangeable version (DST-N), we have presented the polynomial time algorithm DGA with worst-case performance within two times optimal of any nonrearrangeable algorithm. For the rearrangeable problem (DST-R), we have presented a polynomial time algorithm EBA(2) whose performance is within eight times optimal.

Numerous questions related to DST still remain open. For example, is DGA optimal among all algorithms for DST-N? We guess that it is, and that its worst-case performance ratio is actually $\frac{1}{2}\lg n$. In the case of the rearrangeable version, EBA(δ) has relatively good performance, but it potentially requires a large number of rearrangements. Is the bound given in this paper for the maximum number of rearrangements tight? We do not believe that it is. An open problem here is to find an algorithm for DST-R where the number of rearrangements at each step is bound above by a small constant, e.g., two, and yet has a worst-case bound within a constant of optimal. Another interesting question involves the extension of DST to a problem in which the interconnections are allowed to contain cycles. With this extension, does there exist an algorithm with worst-case performance bounded above by a function of n in the case where remove operations are allowed?

Other areas of interest include: average case, and probabilistic performance of algorithms for DST, distributed implementation of algorithms for DST, and the application of these algorithms to multipoint communication networks. Finally, since algorithms for DST can be classified as *on-line algorithms*, it would be of interest to cast DST in a form so that it could be analyzed using the concept of *competitive algorithms* [7].

REFERENCES

[1] B. Bollobás, *Graph Theory*, Springer-Verlag, New York, 1979.

[2] P. Hall, *On representatives of subsets*, J. London Math. Soc., 10 (1935), pp. 26–30.

[3] M. Imase and B. M. Waxman, *Dynamic Steiner tree problem*, Tech. Report 11, Washington University, St. Louis, MO, 1989.

[4] K. Bharath-Kumar and J. M. Jaffe, *Routing to multiple destinations in computer networks*, IEEE Trans. Comm., 31 (1983), pp. 343–351.

[5] J. M. Jaffe, *Distributed multi-destination routing: the constrains of local information*, SIAM J. Comput., 14 (1985), pp. 875–888.

[6] L. Kou, G. Markowsky, and L. Berman, *A fast algorithm on Steiner trees*, Acta Inform., 3 (1977), pp. 141–145.

[7] M. S. Manasee, L. A. McGeoch, and D. D. Sleator, *Competitive algorithms for on-line problems*, in Proc. 20th Annual ACM Symposium on the Theory of Computing, Association for Computing Machinery, New York, 1988, pp. 322–333.

[8] D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis, *An analysis of several heuristics for the Traveling Salesman Problem*, SIAM J. Comput., 6 (1977), pp. 563–581.

[9] J. S. Turner, *The challenge of multipoint communication*, in Proceedings of the ITC Seminar on Traffic Engineering, ISDN Design and Planning, 1988, pp. 263–279.

[10] ———, *New directions in communications*, IEEE Comm. Magazine, 24 (1986), pp. 8–15.

[11] B. M. Waxman, *Routing of multipoint connections*, IEEE J. Select. Areas Comm., 6 (1988), pp. 1617–1622.

[12] ———, *Evaluation of algorithms for multipoint routing*, Ph.D. thesis, Washington University, St. Louis, MO, 1989.

[13] P. Winter, *Steiner problem in networks: a survey*, Networks, 17 (1987), pp. 129–167.

SIAM J. DISC. MATH.
Vol. 4, No. 3, pp. 385-396, August 1991

OPTIMAL PARALLEL ALGORITHMS FOR REGION LABELING AND MEDIAL AXIS TRANSFORM OF BINARY IMAGES*

SUNG KWON KIM†

Abstract. Given a $\sqrt{n} \times \sqrt{n}$ binary image, region labeling labels each 1 of the image so that two 1's have the same label if and only if they are in the same region (i.e., connected) and medial axis transform finds for each 1 of the image the largest square subimage having it as top-left corner and consisting only of 1's. Both can be solved in $\theta(n)$ sequential time. $O(\log n)$ time, $n/\log n$ processor parallel algorithms for both problems in the EREW PRAM are presented.

Key words. binary image, EREW PRAM, medial axis transform, parallel algorithm, region labeling

1. Introduction. This paper addresses two problems on binary images, namely, *region labeling* and *medial axis transform*, and presents optimal parallel algorithms for them. Given a $\sqrt{n} \times \sqrt{n}$ binary image, region labeling is to label each 1 of the image so that two 1's have the same label if and only if they are in the same region. We say that two 1's are *neighboring* if one is immediately above or to the left of the other and that two 1's are *in the same region* (or *connected*) if there is a path of 1's between them in which every two consecutive 1's are neighboring. Region labeling is often called the connectivity problem for binary images. Medial axis transform (MAT) is defined as finding for each 1 of the image the largest square subimage having it as top left corner and consisting entirely of 1's. We present parallel algorithms for these two problems. Our model of parallel computation is the EREW PRAM (exclusive-read exclusive-write parallel random access machine), which is a shared memory machine with no two processors simultaneously allowed to access (read from or write into) the same memory location. Our parallel algorithms are optimal in the sense that their processor-time products equal the sequential lower bound, $\Omega(n)$, of the problems.

Region labeling can be solved in $\theta(n)$ sequential time by applying graph connectivity algorithms such as depth first search [1]. Various parallel connectivity algorithms can be used to solve it in parallel (e.g., Shiloach and Vishkin [16]). Agrawal, Nekludova, and Lim [2] and Cypher, Sanz, and Snyder [10] directly solved the problem to give $O(\log n)$ time, n processor EREW PRAM algorithms. Phillips [15] solved it indirectly by presenting a randomized $O(\log n)$ time, $n/\log n$ processor EREW PRAM algorithm for connectivity of bounded-degree planar graphs. We give an optimal $O(\log n)$ time, $n/\log n$ processor EREW PRAM algorithm for region labeling. We actually solve a more general problem for computing connectivity of planar straight-line graphs (PSLGs) in $O(\log n)$ time using n processors (using $n/\log n$ processors if PSLGs are of bounded-degree or if adjacency lists of vertices are circularly presorted by angles). A PSLG is a planar graph embedded on the plane so that each edge is a straight line segment and no two edges intersect. Recently, Alnuweiri and Kumar [3] independently gave a region labeling algorithm that matches our algorithm in performance (ours is more general).

For the MAT, Guibas and Lipton [11] posted a restricted problem open, in which only the largest square subimage consisting only of 1's was to be computed. Vo [19] gave an algorithm with running time $O(kn)$, where $k \times k$ is the size of the largest such square image, and then Stout [17] presented a linear time algorithm. Both algorithms solved

* Received by the editors September 13, 1989; accepted for publication (in revised form) September 26, 1990. This research was supported in part by National Science Foundation grants CCR-8703196 and CCR-8907960.

† Department of Computer Science and Engineering, University of Washington, Seattle, Washington 98195.

the MAT problem and then found the maximum of these squares obtained. We give another linear time algorithm and show that it can be implemented in the EREW PRAM to run in $O(\log n)$ time using $n/\log n$ processors. Chandran and Mount [6] claimed to give an $O(\log n \log \log n)$ time, n processor CREW PRAM algorithm for the MAT problem. Unfortunately, their algorithm appears to have serious errors.

In the next section, we briefly review some useful parallel techniques. Sections 3 and 4 discuss the region labeling and MAT problems, respectively.

2. Preliminaries. We review some parallel techniques published in the literature.

Prefix Computation. Given an array of numbers $a_1, \cdots, a_n$ and an associative operation $\star$, compute $b_j = a_1 \star \cdots \star a_j$ for $1 \leqq j \leqq n$. See Kruskal, Rudolph, and Snir [13] and Ladner and Fisher [14] for $O(\log n)$ time, $n/\log n$ processor EREW PRAM algorithms. Note that $\star$ may be $+$ and MIN among others.

List Ranking. Given a linear linked list of length n, compute the rank (the distance from the end of the list) of each cell. Anderson and Miller [4] and Cole and Vishkin [8] gave $O(\log n)$ time, $n/\log n$ processor EREW PRAM algorithms. List ranking can be used to do prefix computations over a linear linked list by using the ranks to reorganize the list into an array and applying known prefix computation algorithms on the array.

Extended List Ranking. Given several linear linked lists with a total of n cells, compute the rank of each cell in its own list. Cole and Vishkin [9] introduced the problem and showed that the list ranking algorithms by Anderson and Miller [4] and by Cole and Vishkin [8] could be used within the same bounds.

Cycle Cutting. Given several circular linked lists (i.e., disjoint cycles), find a representative cell for each cycle. Each cell v has a field NEXT (v), which points to the next cell in its cycle. The significance of this problem is that if a representative for each cycle is known, then we can cut each cycle to obtain the same number of linear linked lists and apply extended list ranking algorithms to compute some functions on the cycles.

Since Anderson and Miller's list ranking algorithm can easily be adapted to solve the cycle cutting problem, we first briefly look at their algorithm. It is composed of three major steps.

1. Remove cells from the list so that only $O(n/\log n)$ cells remain. This step is central and rather involved; we refer the reader to their paper [4] for details.
2. Apply Wyllie's standard list ranking algorithm [22] to the reduced list of length $O(n/\log n)$.
3. Compute the ranks of cells that were removed by backtracking the operations performed by each processor at step 1.

Our algorithm for cycle cutting has two steps that are similar to the first two of Anderson and Miller's.

1. Remove cells until $O(n/\log n)$ cells remain. Whenever a cell v is removed, we check to see if NEXT (v) $= v$ (i.e., if v is the last cell remaining in its list). If so, mark v. Note that values of NEXT (v) vary as the algorithm proceeds and thus cells are removed. After this step, fewer cycles of possibly shorter lengths remain (some cycles may entirely disappear) and a representative cell of each disappeared cycle is marked.
2. Apply pointer doubling [22] to the remaining cycles to mark a representative cell of each cycle surviving step 1. Since there are only $O(n/\log n)$ cells and the same number of processors, this step can be done in $O(\log n)$ time.

Euler Tour Technique. The Euler tour technique of Tarjan and Vishkin [18] has many applications for computing simple tree functions such as pre-order and postorder numberings, the level or height of each vertex, and the number of descendants of each vertex. An Euler tour of a tree with n vertices can be found in $O(\log n)$ time using n processors if each vertex knows its parent only or using $n/\log n$ processors if each vertex knows its children as well as its parent. The technique computes these functions by reducing the problems to list ranking, for which $O(\log n)$ time, $n/\log n$ processor EREW PRAM algorithms are known [4], [7].

3. Region labeling. Recall that region labeling assigns a label to each 1 of the image so that two 1's are assigned the same label if and only if they are in a common region. In § 3.1, we introduce some definitions. Section 3.2 presents a parallel connectivity algorithm for PSLGs and § 3.3 discusses its applications to related problems, including region labeling.

3.1. Definitions. A *line* will refer to an undirected edge, while an *edge* will refer to a directed edge. A *circuit* will refer to an undirected cycle, while a *cycle* will refer to a directed cycle. A *vertex* will be used in both directed and undirected graphs. Given two vertices u and v, $\{u, v\}$ denotes the line between them, while (u, v) denotes the edge from u to v and (v, u) the edge from v to u. We say that (u, v) and (v, u) are the *antiparallel* edges of $\{u, v\}$.

A vertex v of a PSLG is assumed to be associated with a coordinate $(x(v), y(v))$. A vertex u is *lower* (respectively, *higher*) than another vertex v if $y(u) < y(v)$ (respectively, $y(u) > y(v)$). Given a vertex v, a line $\{u, v\}$ is a lower (respectively, higher) line of v if u is lower (respectively, higher) than v. Given a vertex v and its lower edges, draw a horizontal line, not passing through v, intersecting all lower edges of v. A lower line $\{u_1, v\}$ is to the *left* (respectively, *right*) of another line $\{u_2, v\}$ if the intersection of $\{u_1, v\}$ with the horizontal line is to the left (respectively, right) of that of $\{u_2, v\}$. Similar definitions are made for edges.

A PSLG partitions the plane into several *faces*. The PSLG in Fig. 1 has four faces; three of them are bounded and the remaining one is unbounded. Any PSLG has exactly one unbounded face, called the *external face* (all other faces are called *internal faces*).

Two vertices in an undirected graph are said to be *connected* if there is a path between them. The problem of computing connectivity of an undirected graph is to assign a label to each vertex so that every two vertices in the same connected component have the same label. Our example PSLG has two *connected components*.

3.2. Connectivity of PSLGs. We give a parallel algorithm for the following problem: Given a PSLG G with n vertices, compute its connectivity.

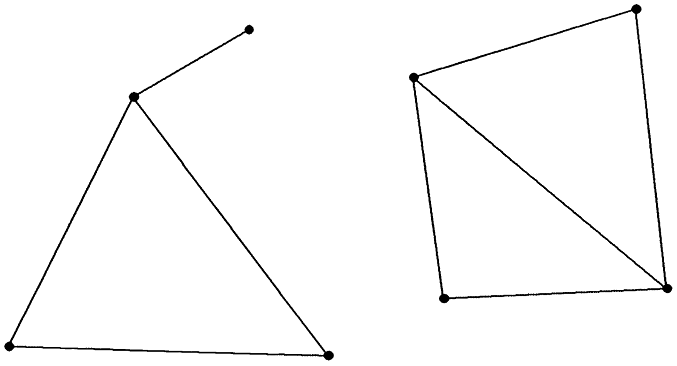


FIG. 1. *An example* PSLG.

We begin by replacing each line of G by its antiparallel edges to obtain a directed graph G'. For each vertex $v \in G'$, order its incident edges in clockwise order. Let $e_{v,1}$, $e_{v,2}, \cdots, e_{v,\deg(v)}$ be the ordered incoming edges to v and $\bar{e}_{v,1}, \bar{e}_{v,2}, \cdots, \bar{e}_{v,\deg(v)}$, the ordered outgoing edges from v, where $e_{v,i}$ and $\bar{e}_{v,i}$ are antiparallel. Assign for each vertex $v \in G'$ in parallel,

$$\text{SUCC}\,(e_{v,i}) \leftarrow \bar{e}_{v,i+1} \text{ for } 1 \leqq i \leqq \deg(v) - 1,$$

and

$$\text{SUCC}\,(e_{v,\deg(v)}) \leftarrow \bar{e}_{v,1}.$$

Each incoming edge $e_{v,i}$ points to the clockwise next outgoing edge $\bar{e}_{v,i+1}$. We call this procedure the *circular assignment*. Figure 2 illustrates this procedure. Then SUCC $(\cdot)$'s partition the edges of G' into cycles. See Fig. 3.

Circular assignment can easily be done in $O(\log n)$ time with $n/\log n$ processors if for each vertex its adjacency list is presorted by angles or if G is of bounded degree. If this is not the case, n processors are needed to sort the adjacent edges of each vertex.

There are two types of cycles according to their orientation, namely *clockwise* (CW) cycles and *counterclockwise* (CCW) cycles. If the interior of a cycle is on the right-hand side when one walks along the cycle, then it is CW. Otherwise, it is CCW. It is easy to see that there is a CCW cycle for each internal face and a CW cycle for each connected component. In Fig. 3 we have three CCW and two CW cycles.

Let the *head vertex* of a cycle be the highest vertex on the cycle. If there is more than one highest vertex, then choose one with minimum x-coordinate. To find the head vertex for each cycle, we apply cycle cutting of § 2 to the cycles to obtain the same number of linear lists and then apply extended list ranking to the resulting lists. This can be done in $O(\log n)$ time using $n/\log n$ processors.

Consider a vertex v and its lower edges. Let $e_{v,l}$ and $\bar{e}_{v,r}$ be the leftmost lower incoming edge and rightmost lower outgoing edge of v, respectively. The following lemma distinguishes two types of cycles.

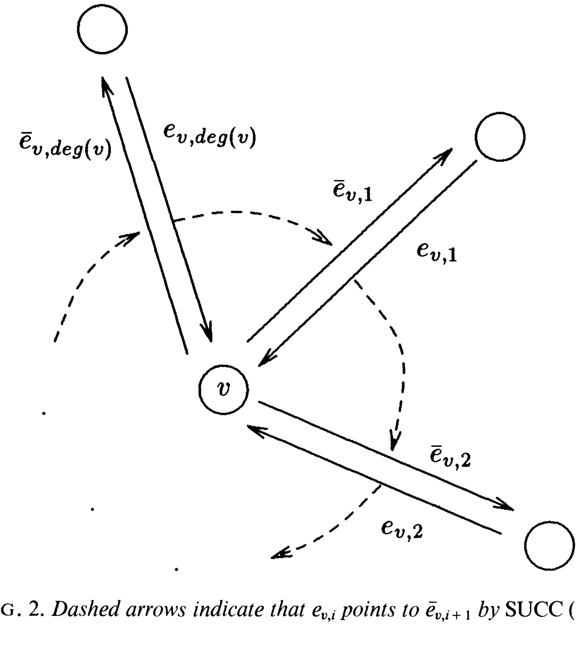


FIG. 2. *Dashed arrows indicate that $e_{v,i}$ points to $\bar{e}_{v,i+1}$ by* SUCC $(\cdot)$.

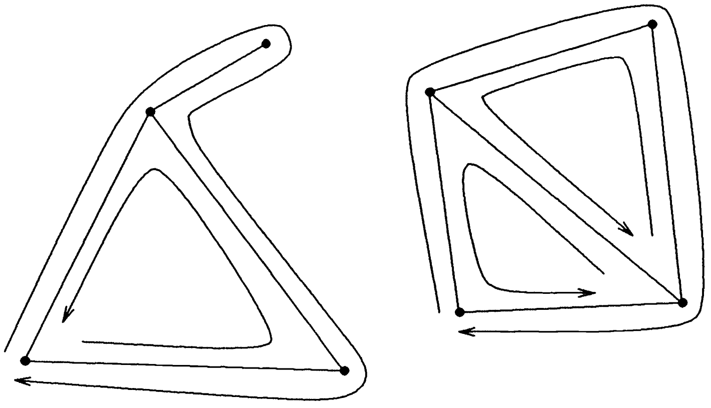


FIG. 3. *After the circular assignment.*

LEMMA 3.1. *A cycle C with head vertex v is* CW *if and only if* SUCC $(e_{v,l}) = \bar{e}_{v,r}$ *and $e_{v,l}$ is on C.*

Proof. $\Rightarrow$) Since C is a CW cycle and v is its head vertex, all edges incident to v are lower edges of v. Otherwise, C would contain higher edges of v, which would imply that v is not the head vertex of C. We have SUCC $(e_{v,l}) = \bar{e}_{v,r}$ because $e_{v,l}$ and $\bar{e}_{v,r}$ are circularly consecutive. If $e_{v,l}$ were not on C, then C could not orient CW. So, $e_{v,l}$ is on C.

$\Leftarrow$) If C oriented CCW, then there would be a vertex in C that is higher than v. This is a contradiction to the definition of head vertex. So, C orients CW. □

Find all CCW cycles by Lemma 3.1. For each CCW cycle C with head vertex v, let the *head edge* of C be the rightmost lower incoming edge of v which is on C. Find head edges of all CCW cycles. Undirect them to get *head lines* in G. Delete all head lines from G. Let F be the resulting graph. The following lemma shows that F is a spanning tree of G.

LEMMA 3.2. *F is a forest that preserves the connectivity of G.*

Proof. Let R be the set of head lines obtained above. We will show that the head lines are all distinct. Consider two CCW cycles C_1 and C_2 in G'. Suppose that their corresponding circuits in G share a head line r and a head vertex v. Consider two antiparallel edges of r. One of them is incoming to and the other outgoing from v. By definition, a head edge is an incoming edge to a head vertex. Since C_1 and C_2 are disjoint, one of them has to have an outgoing edge from v as its head edge. This is a contradiction. So, all head lines are distinct. Thus $|R| = f$, where f is the number of CCW cycles (i.e., internal faces) of G.

Next, we will show that connectivity is preserved. Let $R = \{r_1, \cdots, r_j\}$. Let $C_1, \cdots, C_j$ be the circuits obtained by undirecting the CCW cycles. Let r_i be the head line of C_i for $1 \leqq i \leqq f$. Suppose that we delete r_i one by one. Let G_i be the resulting graph after deleting $r_1, \cdots, r_{i-1}$ from G for $1 \leqq i \leqq f + 1$. We show by induction on i that r_j for $i \leqq j \leqq f$ is contained in a circuit of G_i.

($i = 1$) G_1 has C_j, which contains r_j for $1 \leqq j \leqq f$.

Induction step. Assume that the claim holds for $1 \leqq i \leqq j - 1$. Let D_k, $j - 1 \leqq k \leqq f$, be a circuit of G_{j-1} containing r_k. Define, for $j \leqq k \leqq f$,

$$D'_k = \begin{cases} D_k & \text{if } r_{j-1} \text{ is not contained in } D_k, \\ D_{j-1} \oplus D_k & \text{otherwise,} \end{cases}$$

where $D_{j-1} \oplus D_k$ is the exclusive OR of D_{j-1} and D_k, i.e., $D_{j-1} \cup D_k - (D_{j-1} \cap D_k)$. Then D'_k for $j \leqq k \leqq f$ has a circuit of G_j containing r_k. Note that $D_{j-1} \oplus D_k$ may consist of several circuits of G_j.

Consider Euler's formula $n - e + f = c$ for a PSLG with n vertices, e lines, f internal faces, and c connected components. Note that deletion of any r_i causes both e and f to decrease by one, because each G_i has a circuit containing it and, thus, deletion of r_i joins either two internal faces or an internal face to the external face. Since n remains unchanged, so does c. Thus connectivity is preserved.

Let f' be the number of internal faces in F. Since F has n vertices, $e - f$ lines, and c connected components, we have $n - (e - f) + f' = c$, which gives $f' = 0$. Therefore, F has no internal faces; so F is a forest. □

Identifying all CCW cycles can be done in $O(\log n)$ time. Finding a head edge for each CCW cycle can also be done in $O(\log n)$ time. Both can be done with $n/\log n$ processors if each of the adjacency lists of G is presorted by angles or if G is of bounded-degree.

The connected components of F can be found in $O(\log n)$ time as follows. We already know that F is a spanning forest of G. The problem now is how to assign a label $CC(v)$ to each vertex v so that the vertices in the same component have the same label. Replacing each line of F by two antiparallel edges, we apply the circular assignment to the resulting graph F'. Then each connected component of F is associated with a CW cycle. Let C be a cycle. Compute a label $D(e) \leftarrow \text{MIN}\ \{v \mid v \text{ is a vertex in } C\}$ for every edge $e \in C$. Then all edges in the same connected component have the same label. Finally, assign $CC(v) \leftarrow D(e_{v,1})$ for each vortex v, where $e_{v,1}$ is the first incoming edge to v. Then $CC(v)$ is the smallest vertex in the connected component to which v belongs. Computing $D(\cdot)$ for each vertex of each cycle in F' can be done by cycle cutting.

THEOREM 3.1. *Given a* PSLG *with n vertices, its connected components can be computed in* $O(\log n)$ *time using n processors in the* EREW PRAM. *If the* PSLG *is of bounded degree or if for each vertex its adjacency list is presorted by angles, then* $n/\log n$ *processors suffice.*

3.3. Applications. Since a 1 in a binary image can have at most four neighbors, our PSLG connectivity algorithm solves the region labeling problem in $O(\log n)$ time using $n/\log n$ processors in the EREW PRAM, which is optimal and deterministic ([15] gave a randomized algorithm with the same bounds) and is a $\log n$ processor factor improvement over [2] and [10].

THEOREM 3.2. *Given a* $\sqrt{n} \times \sqrt{n}$ *binary image, its region labeling can be done in* $O(\log n)$ *time using* $n/\log n$ *processors in the* EREW PRAM.

Our algorithm also provides solutions for some related problems such as finding a spanning forest and the bridges of PSLGs. Note that F in our connectivity algorithm is a spanning forest of G. To find the bridges of G, we apply the following lemma, which can be checked in $O(\log n)$ time.

LEMMA 3.3. *A line in G is a bridge if and only if its two antiparallel edges are on a common cycle in G' after the circular assignment.*

Proof. $\Rightarrow$) Let b be a bridge. Let e and $\bar{e}$ be its antiparallel edges. Suppose that e is on a cycle C and $\bar{e}$ is not. Then the undirected version of C has a circuit containing b, contradicting the definition of a bridge.

$\Leftarrow$) Suppose that a line b is not a bridge, i.e., that b is on a circuit in G. Let e and $\bar{e}$ be two antiparallel edges of b. Two different faces are incident to b. One of them must be an internal face and must be entirely contained in the polygon determined by the circuit. Since the CCW cycle of the internal face contains only one of e and $\bar{e}$, no cycle contains both e and $\bar{e}$. □

4. Medial axis transform. The MAT is an important representation of binary images. Wu, Bhaskar, and Rosenfeld [20], [21] gave sequential and parallel algorithms for computing several geometric properties from the MAT. In § 4.1, a problem called the *all nearest blue dominators* (ANBD) is defined and a parallel algorithm for the problem is given. Section 4.2 gives our linear time sequential algorithm for the MAT. In § 4.3, we implement the sequential algorithm in the EREW PRAM using our parallel algorithm for the ANBD problem. Section 4.4 discusses applications of our algorithm.

4.1. The ANBD problem. The ANBD problem is formally defined as follows. Let $A = (a_1, \cdots, a_n)$ and $B = (b_1, \cdots, b_n)$ be two arrays of real numbers, where $0 < a_1 < \cdots < a_n$ and $b_i > 0$ for all i. The problem is to find for each i the smallest index $j \geqq i$, if one exists, such that $a_i \leqq b_j$ and $b_k < a_i$ for all k where $i \leqq k < j$.

Geometrically, this problem can be interpreted as follows: Draw a red vertical line segment connecting two points $(2i - 1, a_i)$ and $(2i - 1, 0)$ for each a_i and a blue vertical line segment connecting $(2i, b_i)$ and $(2i, 0)$ for each b_i and find for each red segment its *nearest blue dominator*, i.e., the first blue segment to its right that is taller than it.

In the following, a_i (respectively, b_i) will be used to denote the red (respectively, blue) segment as well as its height. We denote by $\mathrm{ALT}_{A,B}(a_i)$ the nearest blue dominator in B of a_i in A. If no such blue dominator exists, $\mathrm{ALT}_{A,B}(a_i) = 0$. ALT implies that the red and blue segments are *alternating*.

A simple linear time algorithm works as follows: Assume that $a_{n+1} = b_{n+1} = \infty$. Initially, set $p \leftarrow q \leftarrow 1$. If $a_p \leqq b_q$, then $\mathrm{ALT}_{A,B}(a_p) \leftarrow b_q$; $p \leftarrow p + 1$ and if $p = q + 1$ then $q \leftarrow q + 1$. Otherwise, $q \leftarrow q + 1$. Repeat this until $p > n$. After this, if $\mathrm{ALT}_{A,B}(a_i) = b_{n+1}$, then set $\mathrm{ALT}_{A,B}(a_i) \leftarrow 0$.

Following is a variation of the ANBD problem that will appear in the description of our parallel algorithm for the ANBD problem. In this problem, the red segments a_i are entirely to the left of the blue segments b_i and both sets of segments are increasing in height. This problem can be solved by merging two sets according to height. We use $\mathrm{SEP}_{A,B}(a_i)$ to denote the nearest blue dominator in B of a_i in A in this problem. SEP implies that the red and blue segments are *separate*.

Another problem appearing in our algorithm is, given the blue segments b_i only, to compute the nearest dominator of each segment. This problem is called the *all tallest neighbors problem* and can be solved in $O(\log n)$ time using $n/\log n$ processors in the EREW PRAM [12]. $N_B(b_i)$ will be used to denote the nearest dominator of $b_i \in B$ in this problem. Let T be the tree whose vertex set is B and in which b_j is the parent of b_i if $b_j = N_B(b_i)$. It can be assumed that T is a tree. Otherwise, introduce $b_{n+1} = \infty$.

An $O(\log n)$ time, $n/\log n$ processor EREW PRAM algorithm for the ANBD problem will be presented. We first give an $O(\log n)$ time, n processor algorithm and then reduce the processor bound to $O(n/\log n)$.

4.1.1. *n* Processor algorithm. The $\sqrt{n}$-divide-and-conquer approach is used. From now on, $\mathrm{ALT}(\cdot)$ will replace $\mathrm{ALT}_{A,B}(\cdot)$. We begin by partitioning A into $A_1, \cdots, A_{\sqrt{n}}$ and B into $B_1, \cdots, B_{\sqrt{n}}$, each of size $\sqrt{n}$, where $A_i = \{a_{(i-1)\sqrt{n}+1}, \cdots, a_{i\sqrt{n}}\}$ and $B_i = \{b_{(i-1)\sqrt{n}+1}, \cdots, b_{i\sqrt{n}}\}$. Recursively solve the problem for each pair of A_i and B_i in parallel. The "marriage" starts. After the recursive calls, each $a_j \in A_i$ has $\mathrm{ALT}_{A_i,B_i}(a_j)$, its nearest blue dominator in B_i. If $\mathrm{ALT}_{A_i,B_i}(a_j) \neq 0$, then it is the final solution, i.e., $\mathrm{ALT}(a_j) = \mathrm{ALT}_{A_i,B_i}(a_j)$. If $\mathrm{ALT}_{A_i,B_i}(a_j) = 0$, then we need to check to see if $B_{i+1} \cup \cdots \cup B_{\sqrt{n}}$ contains the nearest blue dominator of a_j and, if so, compute it. Below, we show how the "marriage" can be done in $O(\log n)$ time using n processors.

Let $C_i = \{a_j \in A_i \mid \mathrm{ALT}_{A_i,B_i}(a_j) = 0\}$, the set of segments in A_i which have no blue dominator in B_i. Since A_i is increasing in height, C_i is a suffix of A_i, i.e., either $C_i = \emptyset$ or $C_i = \{a_k, a_{k+1}, \cdots, a_{i\sqrt{n}}\}$ for some $(i-1)\sqrt{n} + 1 \leqq k \leqq i\sqrt{n}$. In other words, for a_x

and a_y in A_i with $x < y$, if $\mathrm{ALT}_{A_i,B_i}(a_x) = 0$ then $\mathrm{ALT}_{A_i,B_i}(a_y) = 0$. Let $q_i = a_{i\sqrt{n}}$, the tallest segment in A_i. Let $Q = (q_1, \cdots, q_{\sqrt{n}})$. Let p_i be the shortest segment in C_i if $C_i \neq \emptyset$. If $C_i = \emptyset$, then $p_i = q_i$. Let $P = (p_1, \cdots, p_{\sqrt{n}})$. Let r_i be the tallest segment in B_i. Let $R = (r_1, \cdots, r_{\sqrt{n}})$. Note that since A is increasing in height, $p_1 \leqq q_1 < \cdots < p_{\sqrt{n}} \leqq q_{\sqrt{n}}$ and the intervals $[p_1, q_1], \cdots, [p_{\sqrt{n}}, q_{\sqrt{n}}]$ are disjoint.

Solve the ANBD problem for P and R and for Q and R, i.e., compute $\mathrm{ALT}_{P,R}(p_i)$ and $\mathrm{ALT}_{Q,R}(q_i)$ for each i. Since $|P| = |Q| = |R| = \sqrt{n}$ and n processors are available, this can be done in $O(\log n)$ time in the EREW PRAM.

Currently, only $\mathrm{ALT}_{P,R}(p_i)$ and $\mathrm{ALT}_{P,R}(q_i)$ are known. We show how this information can be used to determine $\mathrm{ALT}(p_i)$ and $\mathrm{ALT}(q_i)$. Once $\mathrm{ALT}(p_i)$ and $\mathrm{ALT}(q_i)$ are determined, computing $\mathrm{ALT}(a_j)$ for $a_j \in C_i - \{p_i, q_i\}$ becomes rather easy. Note that p_i and q_i are the first and last segments in C_i.

Consider p_i for some i. Let $\mathrm{ALT}_{P,R}(p_i) = r_j$. Then it is clear that $\mathrm{ALT}(p_i) \in B_j$, because no segment in $B_i \cup \cdots \cup B_{j-1}$ is taller than p_i. Note that p_i is taller than $r_i, \cdots, r_{j-1}$ and shorter than r_j. More precisely, $\mathrm{ALT}(p_i) \in B'_j$ where

$$B'_j = \{b_{(j-1)\sqrt{n}+1}, N_B(b_{(j-1)\sqrt{n}+1}), N_B(N_B(b_{(j-1)\sqrt{n}+1})), \cdots, r_j\}.$$

$B'_j \subseteq B_j$ consists of blue segments on the path of T, starting at the first segment of B_j and ending at the tallest segment in B_j. See Fig. 4.

To compute $\mathrm{ALT}(p_i)$ for all $p_i \in P$, divide P into $\sqrt{n}$ subsets $P_1, \cdots, P_{\sqrt{n}}$ where $P_j = \{p_i \in P \mid \mathrm{ALT}_{P,R}(p_i) = r_j\}$, the set of segments in P having the same $\mathrm{ALT}_{P,R}(\cdot) = r_j$, which can easily be obtained from P. This is because p_i's having the same $\mathrm{ALT}_{P,R}(\cdot)$ appear consecutively in P, which is due to the increase property of P. Since P_j and B'_j are separate and both increasing in height, we have $\mathrm{ALT}(p_i) = \mathrm{SEP}_{P_j,B'_j}(p_i)$ for all $p_i \in P_j$, which can be obtained by merging P_j and B'_j according to the height of the segments in $P_j \cup B'_j$. This can be done in $O(\log n)$ time using $|P_j| + |B'_j|$ processors. Similarly, compute $\mathrm{ALT}(q_i)$ for $q_i \in Q$.

Having determined $\mathrm{ALT}(p_i)$ and $\mathrm{ALT}(q_i)$, we explain how to use these to compute $\mathrm{ALT}(a_j)$ for $a_j \in C_i - \{p_i, q_i\}$. Note that either $\mathrm{ALT}(q_i) = \mathrm{ALT}(p_i)$ or $\mathrm{ALT}(q_i)$ is an ancestor of $\mathrm{ALT}(p_i)$ in T, i.e., that $\mathrm{ALT}(q_i) = N_B^{(k)}(\mathrm{ALT}(p_i))$ for $k \geqq 0$, where

$$N_B^{(k)}(x) = \underbrace{N_B(N_B(\cdots(N_B}_{k}(x))\cdots)).$$

Let $I_i = \{\mathrm{ALT}(p_i), N_B(\mathrm{ALT}(p_i)), N_B(N_B(\mathrm{ALT}(p_i))), \cdots, \mathrm{ALT}(q_i)\}$ for $1 \leqq i \leqq$

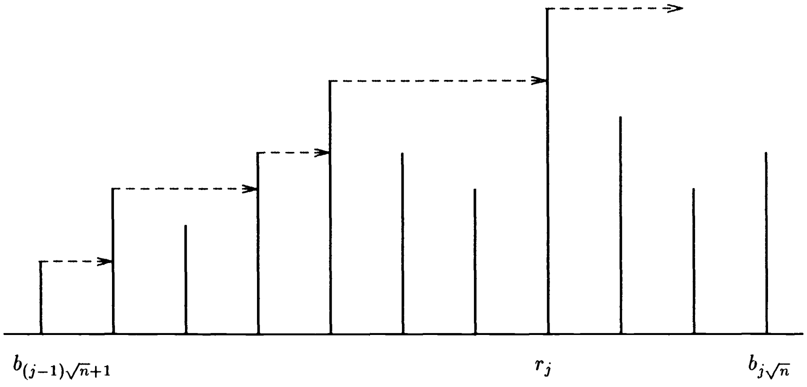


FIG. 4. *B_j has* 11 *segments and B'_j has* 5 *segments, each of which has a dashed arrow.*

$\sqrt{n}$, the set of blue segments on the path of T between ALT (p_i) and ALT (q_i). Then ALT $(a_j) \in I_i$ for all $a_j \in C_i - \{p_i, q_i\}$. Again, since P_i and I_i are separate and both increasing in height, it is clear that ALT $(a_j) = \text{SEP}_{C_i,I_i}(a_j)$ for $a_j \in C_i - \{p_i, q_i\}$.

Since computing $\text{SEP}_{C_i,I_i}(\cdot)$ is easy once I_i is known, we describe how to compute I_i for all i. If ALT (p_i) = ALT (q_i), then ALT (a_j) = ALT (p_i) for all $a_j \in C_i$. So, we can exclude such i's from further consideration. We can now assume that $|I_i| \geqq 2$ for all i. Then $|I_i \cap I_{i+1}| \leqq 1$ and $I_i \cap I_j = \emptyset$ for $|i-j| > 1$. To prove this, we first show that $|I_i \cap I_j| \leqq 1$ for any i and j. Suppose that $|I_i \cap I_j| \geqq 2$ for some i and j. Since I_i and I_j are paths in T, $I_i \cap I_j$ is also a path in T. Then, $I_i \cap I_j$ is a prefix of either I_i or I_j, which implies either $q_i \in [p_j, q_j]$ or $q_j \in [p_i, q_i]$, which in turn implies that $[p_i, q_i]$ and $[p_j, q_j]$ are not disjoint. Therefore, $|I_i \cap I_j| \leqq 1$ for any i and j. Next, we show that $I_i \cap I_j = \emptyset$ for $|i-j| > 1$. Suppose that $I_i \cap I_j \neq \emptyset$ for some i and $j > i+1$. Then $|I_i \cap I_j| = 1$, as proved above, which implies that $|I_{i+1}| = 1$. This contradicts our assumption that $|I_i| \geqq 2$ for all i. We have proven that $|I_i \cap I_{i+1}| \leqq 1$ and $I_i \cap I_j = \emptyset$ for $|i-j| > 1$. Furthermore, if $|I_i \cap I_{i+1}| = 1$, then $I_i \cap I_{i+1} = \{\text{ALT}(p_{i+1})\}$.

Define $I'_i = I_i - \{\text{ALT}(p_i)\}$ for $1 \leqq i \leqq \sqrt{n}$. Then I'_i's are mutually disjoint. To compute I'_i, we partition T into $\sqrt{n}+1$ trees $T_1, \cdots, T_{\sqrt{n}+1}$, by deleting edges between ALT (q_i) and its parent. If the root of T is ALT (q_i) for some i, then $T_{\sqrt{n}+1} = \emptyset$ because in this case only $\sqrt{n}-1$ edges are removed. Assume that ALT (q_i) is the root of T_i for $1 \leqq i \leqq \sqrt{n}$. Then I'_i is a path in T_i between $N_B(\text{ALT}(p_i))$ and the root of T_i. It is easy to find I'_i from T_i in $O(\log n)$ time using $|T_i|$ processors in the EREW PRAM by the Euler tour technique as in [12]. Compute $I_i = I'_i \cup \{\text{ALT}(p_i)\}$ for all i.

We have shown that each step of the "marriage" runs in $O(\log n)$ time using n processors. Let $T(n)$ be the running time of our algorithm on problems of size n. Then $T(n) \leqq T(\sqrt{n}) + O(\log n)$, which gives $T(n) = O(\log n)$.

THEOREM 4.1. *The* ANBD *problem can be solved in* $O(\log n)$ *time using n processors in the* EREW PRAM.

4.1.2. $n/\log n$ processor algorithm. Partition A into $A_1, \cdots, A_{n/\log n}$ and B into $B_1, \cdots, B_{n/\log n}$, each of size $\log n$, where $A_i = \{a_{(i-1)\log n+1}, \cdots, a_{i\log n}\}$ and $B_i = \{b_{(i-1)\log n+1}, \cdots, b_{i\log n}\}$. Solve the problem for each pair of A_i and B_i simultaneously with one processor per pair by our linear time sequential algorithm (it takes $O(\log n)$ time). Compute C_i for $1 \leqq i \leqq n/\log n$. Compute P, Q, and R. Note that $|P| = |Q| = |R| = n/\log n$. Compute $\text{ALT}_{P,R}(p_i)$ and $\text{ALT}_{Q,R}(q_i)$ for all i. Since $n/\log n$ processors are available, this can be done in $O(\log n)$ time by our linear processor parallel algorithm.

Compute P_j and $B'_j = \{b_{(j-1)\log n+1}, N_B(b_{(j-1)\log n+1}), \cdots, r_j\}$ for $1 \leqq i \leqq n/\log n$. Compute ALT (p_i) for all $p_i \in P_j$ by merging P_j and B'_j, which can be done in $O(\log n)$ time using $|P_j|$ processors [5]. Since $\sum_j |P_j| = n/\log n$, we have enough processors for these mergings. Similarly, compute ALT (q_i) for $q_i \in Q$.

I_i is defined as before for $1 \leqq i \leqq n/\log n$. We show how to compute I_i for all i in $O(\log n)$ time using only $n/\log n$ processors. Solve the all tallest neighbors problem for R, i.e., compute $N_R(r_i)$ for all $r_i \in R$. This can be done in $O(\log n)$ time because R is of size $n/\log n$ and $n/\log n$ processors are available. Let

$$J_i = \{\text{ALT}_{P,R}(p_i), N_R(\text{ALT}_{P,R}(p_i)), \cdots, \text{ALT}_{Q,R}(q_i)\} \text{ for } 1 \leqq i \leqq n/\log n.$$

Then J_i for all i can be found in $O(\log n)$ time (in a similar way as we found the I_i's in § 4.1.1) by constructing a tree T' with vertex set R and edge set $\{(r_i, r_j) | r_j = N_R(r_i)\}$. We can easily prove that $|J_i \cap J_j| \leqq 1$ for any i and j by a similar technique used to prove $|I_i \cap I_i| \leqq 1$ for any i and j in § 4.1.1. Therefore, $\sum_i |J_i| \leqq 2n/\log n$.

Now we compute I_i from J_i. Assume $J_i = \{r_{\alpha_i(1)}, \cdots, r_{\alpha_i(|J_i|)}\}$. Then ALT $(p_i) \in B'_{\alpha_i(1)}$ and ALT $(q_i) \in B'_{\alpha_i(|J_i|)}$. Let

$$K_{i,1} = \{\text{ALT}\,(p_i), N_B(\text{ALT}\,(p_i)), \cdots, r_{\alpha_i(1)}\},$$

$$K_{i,j} = \{t_{\alpha_i(j)}, N_B(t_{\alpha_i(j)}), \cdots, r_{\alpha_i(j)}\} \text{ for } 2 \leqq j \leqq |J_i| - 1, \text{and}$$

$$K_{i,|J_i|} = \{t_{\alpha_i(|J_i|)}, N_B(t_{\alpha_i(|J_i|)}), \cdots, \text{ALT}\,(q_i)\},$$

where $t_{\alpha_i(j)}$ is the shortest segment in $B'_{\alpha_i(j)}$ that is taller than $r_{\alpha_i(j-1)}$. Recall that r_i is the tallest segment in B_i and B'_i. Then $I_i = K_{i,1} \cup \cdots \cup K_{i,|J_i|}$, because $N_B(r_{\alpha_i(j-1)}) = t_{\alpha_i(j)}$.

To compute I_i's, assign a processor to each member of $r_{\alpha_i(j)}$ of J_i (a total of $|J_i|$ processors to J_i). A processor assigned to $r_{\alpha_i(j)}$ sequentially scans $B'_{\alpha_i(j)}$ to determine $K_{i,j}$. Note that $K_{i,j} \subseteq B'_{\alpha_i(j)}$. This clearly takes $O(\log n)$ time using $n/\log n$ processors because $\sum_i |J_i| \leqq 2n/\log n$. A problem here is that there may be read conflicts, i.e., more than one processor may be assigned to the same r_i and these processors may simultaneously scan the same B'_i. This is because J_i's may not be disjoint.

One way to avoid these potential read conflicts is to assign a copy of $B_{\alpha_i(j)}$ to $r_{\alpha_i(j)}$ and for the processor assigned to $r_{\alpha_i(j)}$ to scan its own copy of $B_{\alpha_i(j)}$ to determine $K_{i,j}$. To do this, let β_i be the number of r_i's in $J_1 \uplus \cdots \uplus J_{n/\log n}$, where $\uplus$ is the set concatenation (i.e., duplicate elements are allowed). This can be done by sorting and applying parallel prefix computation in $O(\log n)$ time, because there are no more than $2n/\log n$ items. Assign β_i processors to B'_i and make β_i copies of B'_i, which can be done in $O(\log n)$ time, because $|B'_i| \leqq \log n$. Finally, assign a copy of B'_i to each of the $\beta_i r_i$'s.

Compute ALT $(a_j) = \text{SEP}_{C_i,I_i}(a_j)$ for all $a_j \in C_i - \{p_i, q_i\}$ by merging C_i and I_i, which can be done in $O(\log n)$ time using $|J_i|$ processors [5], because $|C_i| \leqq \log n$ and $|I_i| \leqq |J_i| \cdot \log n$.

THEOREM 4.2. *The* ANBD *problem can be solved in* $O(\log n)$ *time using* $n/\log n$ *processors in the* EREW PRAM.

4.2. Sequential algorithm. This section presents a linear time algorithm for the MAT representation of a $\sqrt{n} \times \sqrt{n}$ binary image $H = (h_{i,j})$, $1 \leqq i, j \leqq \sqrt{n}$, where $h_{1,1}$ is top-left, $h_{1,\sqrt{n}}$ top-right, $h_{\sqrt{n},1}$ bottom-left, and $h_{\sqrt{n},\sqrt{n}}$ bottom-right. We first give the basic idea behind our algorithm. The MAT problem can be interpreted as follows: Given a $\sqrt{n} \times \sqrt{n}$ grid, H, with grid points labeled either 0 or 1, compute for each grid point $h_{i,j} = 1$ the L_∞-distance $z_{i,j}$ to a nearest grid point labeled 0 lying in the southeast quadrant of $h_{i,j}$. Recall that the L_∞ distance between two points is $\max\{|x_1 - x_2|, |y_1 - y_2|\}$. A southeast quadrant is divided into an east-southeast (ESE) octant and a south-southeast (SSE) octant by a line of slope $-45°$. Finding a nearest 0 in the southeast quadrant of a 1 is equivalent to finding a nearest 0 in its ESE octant and a nearest 0 in its SSE octant and taking the nearer one. Quadrants and octants are assumed to include their boundaries. Note that the L_∞-distance between two points, one lying in the other's ESE (respectively, SSE) octant, is $|x_1 - x_2|$ (respectively, $|y_1 - y_2|$).

We now present our algorithm. Augment H by adding a 0-column vector to its left and right and a 0-row vector above and below it. This augmentation increases the input size by $4\sqrt{n} + 4$ and the asymptotic behavior of our algorithm does not change.

Compute $x_{i,j}$ for all i and j, where $x_{i,j}$ is the integer such that $h_{i,x_{i,j}} = 0$ is the nearest 0 in the ith row to the left of $h_{i,j}$ (i.e., $h_{i,k} = 1$ for all k where $x_{i,j} + 1 \leqq k \leqq j$). If $h_{i,j} = 0$ then $x_{i,j} = j$. Similarly, compute $y_{i,j}$ for all i and j where $y_{i,j}$ is the integer such that $h_{y_{i,j},j} = 0$ is the nearest 0 in the jth column above $h_{i,j}$ (i.e., $h_{k,j} = 1$ for all k where $y_{i,j} + 1 \leqq k \leqq i$). It is easy to compute all $x_{i,j}$ and $y_{i,j}$ in linear time. Following is a simple lemma that is crucial to our algorithm.

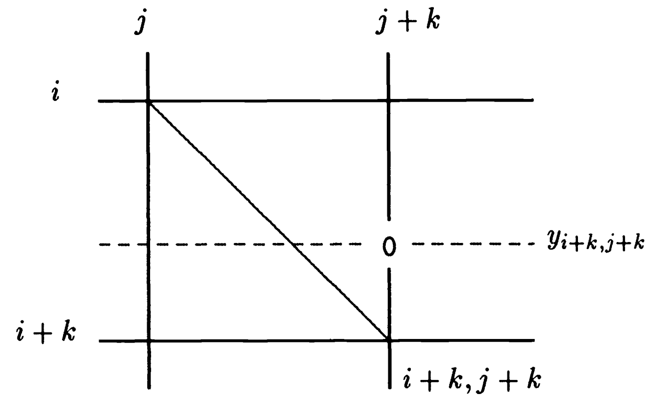


FIG. 5. *An illustration for a proof of Lemma* 4.1.

LEMMA 4.1. *Given i and j,*

(i) *the $(j+k)$th column of H for $k \geqq 1$ has a* 0 *in the* ESE *octant of $h_{i,j}$ if and only if $y_{i+k,j+k} \geqq i$; and*

(ii) *the $(i+k)$th row of H for $k \geqq 1$ has a* 0 *in the* SSE *octant of $h_{i,j}$ if and only if $x_{i+k,j+k} \geqq j$.*

See Fig. 5 for a proof idea.

By Lemma 4.1, the L_∞-distance of $h_{i,j} = 1$ to an ESE-nearest 0 (i.e., a nearest 0 in the ESE octant of $h_{i,j}$) is

$$\Delta x_{i,j} = \min \{ k \mid y_{i+k,j+k} \geqq i \text{ for } k \geqq 1 \}.$$

Analogously, the L_∞-distance of $h_{i,j} = 1$ to an SSE-nearest 0 is

$$\Delta y_{i,j} = \min \{ k \mid x_{i+k,j+k} \geqq j \text{ for } k \geqq 1 \}.$$

Then the L_∞-distance of $h_{i,j} = 1$ to an SE-nearest 0 is

$$z_{i,j} = \min \{ \Delta x_{i,j}, \Delta y_{i,j} \}.$$

To show how to compute $\Delta x_{i,j}$ for all i and j in linear time, consider a diagonal

$$h_{1,i}, h_{2,i+1}, \cdots, h_{\sqrt{n}-i+1,\sqrt{n}} \text{ for some } i.$$

Solve the ANBD problem with

$$(a_1, a_2, \cdots, a_{\sqrt{n}-i+1}) = (1, 2, \cdots, \sqrt{n}-i+1),$$

and

$$(b_1, b_2, \cdots, b_{\sqrt{n}-i+1}) = (y_{1,i}, y_{2,i+1}, \cdots, y_{\sqrt{n}-i+1,\sqrt{n}}).$$

Then $\Delta x_{k,i+k-1} = m - k$ if ALT $(a_k) = b_m$ for $1 \leqq k \leqq \sqrt{n} - i + 1$, which can be computed in $O(\sqrt{n} - i)$ time. Similarly, $\Delta x_{j+k-1,k}$ for $1 \leqq k \leqq \sqrt{n} - j + 1$ can be computed in $O(\sqrt{n} - j)$ time. Therefore, $\Delta x_{i,j}$ for all i and j can be computed in linear time. In an analogous manner, $\Delta y_{i,j}$ for all i and j can be computed in linear time.

Finally, computing $z_{i,j}$ for all i and j is trivial. So, we have a linear time algorithm for the MAT problem.

4.3. Parallel implementation. This section implements our sequential algorithm in the EREW PRAM to run in $O(\log n)$ time using $n/\log n$ processors. Computing $x_{i,j}$ and $y_{i,j}$ for all i and j can be done in $O(\log n)$ time using $n/\log n$ processors by parallel prefix computations. Computing $\Delta x_{i,j}$ and $\Delta y_{i,j}$ for all i and j are instances of the ANBD problem as explained in § 4.2, which can be solved in $O(\log n)$ time using $n/\log n$ processors.

THEOREM 4.3. *Given a* $\sqrt{n} \times \sqrt{n}$ *binary image, its* MAT *representation can be computed in* $O(\log n)$ *time using* $n/\log n$ *processors in the* EREW PRAM.

4.4. Applications. Our MAT algorithm can be used to solve the problem of finding for each 1 of the image the largest L_1- (L_∞-) disk centered at it. Recall that an L_1-disk is a rhombus and an L_∞-disk is a square. For each $h_{i,j} = 1$, find the nearest 0 in each of its four quadrants and take the nearest one. Finding an L_∞-nearest 0 in a quadrant has already been explained in previous subsections. It can easily be modified to find an L_1-nearest 0.

Acknowledgments. I would like to thank Professor Walter L. Ruzzo for his helpful discussion and the referee for several constructive suggestions and for bringing [3] to my attention.

REFERENCES

[1] A. V. AHO, J. E. HOPCROFT, AND J. D. ULLMAN, *Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

[2] A. AGRAWAL, L. NEKLUDOVA, AND W. LIM, *A parallel* $O(\log N)$ *algorithm for finding connected components in planar images*, Proc. International Conference on Parallel Processing, 1987, pp. 783–786.

[3] H. ALNUWEIRI AND V. K. PRASANNA KUMAR, *Parallel architectures and algorithms for image component labeling*, Report No. 253, Institute for Robotics and Intelligent Systems, University of Southern California, Los Angeles, CA, 1990.

[4] R. J. ANDERSON AND G. L. MILLER, *Deterministic parallel list ranking*, Proc. 3rd Aegean Workshop on Computing, Lecture Notes in Computer Science, Vol. 319, Springer-Verlag, Berlin, New York, 1988, pp. 81–90.

[5] G. BILARDI AND A. NICOLAU, *Adaptive bitonic sorting: an optimal parallel algorithm for shared-memory machines*, SIAM J. Comput., 18 (1989), pp. 216–228.

[6] S. CHANDRAN AND D. MOUNT, *Shared memory algorithms and the medial axis transform*, Proc. IEEE Workshop on Computer Architecture for PAMI, Seattle, WA, 1987, pp. 44–50.

[7] R. COLE AND U. VISHKIN, *Deterministic coin tossing with applications to optimal parallel list ranking*, Inform. and Control, 70 (1986), pp. 32–53.

[8] ———, *Approximate parallel scheduling. Part* I: *The basic technique with applications to optimal parallel list ranking in logarithmic time*, SIAM J. Comput., 17 (1988), pp. 128–142.

[9] ———, *The accelerated centroid decomposition technique for optimal parallel tree evaluation in logarithmic time*, Algorithmica, 3 (1988), pp. 329–346.

[10] R. CYPHER, J. L. C. SANZ, AND L. SNYDER, *An* EREW PRAM *algorithm for image component labeling*, IEEE Trans. Pattern Anal. Mach. Intell., 11 (1989), pp. 258–261.

[11] L. GUIBAS AND R. LIPTON, ⟨*Problem* 80-4⟩, J. Algorithms, 1 (1980), p. 108.

[12] S. K. KIM, *Optimal parallel algorithms on sorted intervals*, Proc. 27th Annual Allerton Conference Comm. Control, and Comput., Allerton, IL, 1989, pp. 766–775.

[13] C. P. KRUSKAL, L. RUDOLPH, AND M. SNIR, *The power of parallel prefix*, IEEE Trans. Comput., 34 (1985), pp. 965–968.

[14] R. LADNER AND M. FISCHER, *Parallel prefix computation*, J. ACM., 27 (1980), pp. 831–838.

[15] C. A. PHILLIPS, *Parallel graph contraction*, Proc. ACM Symposium Parallel Algorithms Arch., Santa Fe, NM, 1989, pp. 148–157.

[16] Y. SHILOACH AND U. VISHKIN, *An* $O(\log n)$ *parallel connectivity algorithm*, J. Algorithms, 3 (1982), pp. 57–67.

[17] D. F. STOUT, *An improved solution to* ⟨*Problem* 80-4⟩, J. Algorithms, 4 (1983), p. 177.

[18] R. E. TARJAN AND U. VISHKIN, *An efficient parallel connectivity algorithm*, SIAM J. Comput., 14 (1985), pp. 862–874.

[19] K. P. VO, *A solution to* ⟨*Problem* 80-4⟩, J. Algorithms, 3 (1982), pp. 366–367.

[20] A. Y. WU, S. K. BHASKAR AND A. ROSENFELD, *Computation of geometric properties from the medial axis transform in* $O(n \log n)$ *time*, Comput. Vision Graphics Image Process., 34 (1986), pp. 76–92.

[21] ———, *Parallel computation of geometric properties from the medial axis transform*, Comput. Vision Graphics Image Process., 41 (1988), pp. 323–332.

[22] J. C. WYLLIE, *The complexity of parallel computations*, Ph.D. dissertation, Computer Science Department, Cornell University, Ithaca, NY, 1981.

SIAM J. DISC. MATH.
Vol. 4, No. 3, pp. 397-408, August 1991

008

ON THE COMPLEXITY OF COLOURING BY VERTEX-TRANSITIVE AND ARC-TRANSITIVE DIGRAPHS*

GARY MACGILLIVRAY†

Abstract. Let H be a fixed directed graph whose vertices are called colours. An *H-colouring* of a digraph G is an assignment of these colours to the vertices of G such that if x is adjacent to y in G, then *colour* (x) is adjacent to *colour* (y) in H (i.e., a *homomorphism* $G \to H$). In this paper the complexity of the H-colouring problem, when the directed graph H is vertex-transitive or arc-transitive, is investigated. In both instances a complete classification is obtained.

Key words. graph colouring, graph homomorphism, complexity, NP-completeness, polynomial algorithm, vertex-transitive graphs

AMS(MOS) subject classifications. 05C15, 05C20, 68Q25

1. Introduction. Let D and H be directed graphs (respectively, graphs). A *homomorphism of D to H* is a function $f:V(D) \to V(H)$ such that $f(x)f(y)$ is an arc (respectively, edge) of H whenever xy is an arc (respectively, edge) of D. Observe that an n-colouring of a graph G is a homomorphism of G to K_n. In analogy with this terminology, the term *H-colouring* of G has been used to describe a homomorphism of a digraph (respectively, graph) G to a fixed digraph (respectively, graph) H.

The *H-colouring problem* may be formally stated as follows:

H-COL (*H*-colouring)
Instance: A digraph (respectively, graph) D.
Question: Does there exist an H-colouring of D?

The H-colouring problem clearly belongs to NP for any fixed digraph (respectively, graph) H.

The complexity of the H-colouring problem has received considerable attention in the literature [2]–[6], [10]–[12], [14]–[16]. Several developments are worthy of special attention. Hell and Nešetřil [14] have settled the undirected case by proving that the H-colouring problem is NP-complete for any fixed graph H that contains an odd cycle, and is polynomial otherwise. When H is a directed graph the "dividing line" does not seem clear. The H-colouring problem is polynomial if H is an oriented path, but there is an oriented tree T for which the T-colouring problem is NP-complete [10]–[12]. The problem is known to be NP-complete for a large collection of digraphs with at least two directed cycles [3]–[6], [15], [16]. It has been conjectured that if H is a connected digraph in which each vertex has in-degree at least one and out-degree at least one, then the H-colouring problem is NP-hard unless H admits a retraction to a directed cycle [3]. Another formulation of this conjecture is given in [5], where it is proved to be equivalent to the following statement. If G is a connected digraph in which each vertex has in-degree at least one and out-degree at least one, and which does not admit a homomorphism to a directed cycle, then the H-colouring problem is NP-hard whenever G is a subdigraph of H. The reader is referred to [2] for a survey of results in direction H-colouring.

In this paper we prove that the aforementioned conjecture of Hell and Bang-Jensen is true for vertex-transitive digraphs, and arc-transitive digraphs. For vertex-transitive

* Received by the editors January 29, 1990; accepted for publication (in revised form) September 26, 1990. This research was supported by the Natural Sciences and Engineering Research Council of Canada.

† Department of Mathematics and Statistics, University of Regina, Regina, Saskatchewan, Canada, S4S 0A2.

digraphs we prove that the H-colouring problem is NP-complete unless H admits a retraction to a directed cycle, and is polynomial otherwise (cf. Theorem 3.4). As a corollary, we deduce a similar classification of arc-transitive digraphs (cf. Corollary 3.5). We also give necessary and sufficient conditions for a Cayley graph of a finite group to admit a retraction to a directed cycle (i.e., for the associated H-colouring problem to be polynomial).

2. Preliminaries. The purpose of this section is to review the background material needed for § 3.

We use the terminology of [8], with a few exceptions and additions. Let D be a directed graph, and let x, y be vertices of D. If both xy and yx are arcs of D, we say that x and y are joined by a *double-edge*, and denote this situation by $[x, y]$. The *undirected part of* D, *undir*(D), is the subdigraph induced by the set of double-edges. Note that *undir* (D) is the equivalent digraph of an undirected graph. We do not distinguish between *undir* (D) and its underlying simple graph. In particular, we talk about undirected paths or cycles in D, or about whether *undir* (D) is bipartite (equivalently, colourable by C_2). In our figures double edges will be drawn as undirected edges.

We say that a digraph D is *connected* if, for any two vertices u and v, there exists an oriented (u, v)-walk in D. If D is not connected, we call each maximal connected subdigraph a *connected component* of D.

We say that D is a *strong digraph* if, for any two vertices u and v, there exists a directed (u, v)-path and a directed (v, u)-path.

A *source* of a directed graph D is a vertex of in-degree zero. A *sink* is a vertex of out-degree zero. We use the term *smooth* to describe a directed graph with no sources and no sinks.

Let I be a fixed digraph, and let u, v be distinct vertices of I. The *indicator construction with respect to* (I, u, v) transforms a given directed graph H into the directed graph H^*, defined to have the same vertex set as H, and to have as its arc set the set of all pairs xy such that there is a homomorphism of I to H which maps u to x and v to y. The triple (I, u, v) is called an *indicator*, and if the digraph H^* is loopless (no homomorphism of I into H maps u and v to the same vertex), it is called a *good indicator*. A *symmetric indicator* is an indicator (I, u, v) such that some automorphism of I exchanges u and v. Symmetric indicators are important because the result of the indicator construction with respect to a symmetric indicator is an undirected graph.

LEMMA 2.1 [14]. *If H^*-COL problem is* NP-*complete, then so is H-COL.*

In applying the indicator construction care must be taken to assure that H^* has no loops (i.e., that (I, u, v) is a good indicator), otherwise there is a polynomial time algorithm for H^*-colouring: map all vertices to a vertex with a loop.

The following result is an immediate consequence of the indicator construction (use a double edge as the indicator).

LEMMA 2.2 [5], [14]. *If undir (H) is loopless and contains an odd cycle, then H-COL is* NP-*complete.*

Let G and H be directed graphs, and suppose that H is a subdigraph of G. A *retraction* of G to H is a homomorphism $f{:}G \to H$ such that $f(h) = h$ for all vertices h of H. If there exists a retraction of G to H, the directed graph H is called a *retract* of G. A digraph is *retract-free* (or a *core* [3], [6], [12], or a *minimal graph* [10], [14], [16]) if it does not admit a retraction to a proper subdigraph. Every directed graph G contains a unique (up to isomorphism) minimal retract-free subdigraph H which admits a retraction $G \to H$. We refer to such an H as *the core of* G. Observe that if G' is a retract of G then G and G' have the same core. If H is the core of G, then there are homomorphisms

$i: H \rightarrow G$ (the inclusion) and $r: G \rightarrow H$ (a retraction); thus a graph D is G-colourable if and only if it is H-colourable.

Let $h_1, h_2, \cdots, h_t$ be specified vertices of a retract-free digraph H, and let J be a fixed digraph, with specified vertices x and $j_1, j_2, \cdots, j_t$. The *subindicator construction with respect to* $(J, x, j_1, j_2, \cdots, j_t)$, *and* $h_1, h_2, \cdots, h_t$ transforms H to its subdigraph $H^{\sim}$ induced by the vertex set $V^{\sim}$ defined as follows: Let W be the digraph obtained from the disjoint union of J and H by identifying j_i and h_i, $i = 1, 2, \cdots, t$. A vertex v of H belongs to $V^{\sim}$ just if there is a retraction of W to H which maps x to v. The structure $(J, x, j_1, j_2, \cdots, j_t)$ is called a *subindicator*.

LEMMA 2.3 [14]. *Let H be a retract-free digraph. If $H^{\sim}$-COL is* NP-*complete, then so is H-COL.*

The directed cycle of length n is denoted by C_n, and assumed to have vertex set $\{0, 1, \cdots, n-1\}$ and arc set $\{i(i+1): i = 0, 1, \cdots, n-1$, where addition is modulo $n\}$. Similarly, P_n, the directed path of length n, is assumed to have vertex set $\{0, 1, \cdots, n\}$, and arc set $\{i(i+1): i = 0, 1, \cdots, n-1\}$. Note that an isolated vertex is a path of length zero but not a cycle of length zero. A loop at vertex v creates a cycle of length one.

LEMMA 2.4 [16]. *For every positive integer n, both C_n-COL and P_n-COL are polynomial.*

Thus the H-colouring problem is polynomial whenever a directed path or a directed cycle is a retract of H.

We conclude this section by stating a result concerning the cycle structure of strong digraphs. (Note that a connected vertex-transitive digraph is strong.)

LEMMA 2.5 [5]. *Let D be a strong digraph. There is a homomorphism of D to C_n if and only if the length of every directed cycle in D is divisible by n.*

Thus a strong digraph does not admit a homomorphism to C_n if and only if it has a directed cycle whose length is not divisible by n, and does not admit a homomorphism to any directed cycle of length greater than one if and only if it has a collection $C^1, C^2, \cdots, C^k$ of directed cycles such that $gcd\{|V(C^i)|: i = 1, 2, \cdots, k\} = 1$.

3. Results.

The following three lemmas are essential to the proof of our main result.

LEMMA 3.1. *The core of a vertex-transitive digraph is vertex-transitive.*

Proof. Let H be the core of G. Then there is a retraction $r: G \rightarrow H$. Let x and y be vertices of H, and let f be an automorphism of G such that $f(x) = y$. Then $r \circ f$ is a homomorphism of H to H and, as H is retract-free, an automorphism of H. Since $r(f(x)) = r(y) = y$, we have that H is vertex-transitive. □

LEMMA 3.2. *Let H be a directed graph and let (I, u, v) be an indicator. Let H^* be the digraph that results from applying the indicator construction with respect to (I, u, v) to H. Then Aut (H) is a subgroup of Aut (H^*).*

Proof. Since $Aut\ (H)$ is a group it suffices to prove that $Aut\ (H^*)$ contains $Aut\ (H)$. Let f be an automorphism of H and let ab be an arc of H^*. Then there is a homomorphism $h: I \rightarrow H$ such that $h(u) = a$ and $h(v) = b$. The function $f \circ h$ is also a homomorphism of I to H, and $f(h(u)) = f(a)$ and $f(h(v)) = f(b)$. Hence $f(a)f(b)$ is also an arc of H^*. Since f is a one-to-one arc preserving map, it is an automorphism of H^*. □

By Lemma 3.2, the digraph that results from applying an indicator construction to a vertex-transitive digraph is also vertex-transitive.

We now define a special type of indicator that plays a central role in the proof of Theorem 3.4. A *z-indicator* is an indicator (I, u, v) such that there is a vertex z that is the only neighbour of v (the vertex z may be an in-neighbour of v or an out-neighbour of v). If z is an in-neighbour of v, we sometimes call (I, u, v) an *in-z-indicator* and, similarly, if z is an out-neighbour of v, we sometimes call (I, u, v) an *out-z-indicator*. These special indicators are important in our work on vertex-transitive digraphs because of the following lemma.

LEMMA 3.3. *Let H be a vertex-transitive digraph and let (I, u, v) be an in-z-indicator (respectively, out-z-indicator). If there exists a vertex x of H, and homomorphisms h_1 and h_2 of I to H such that $h_1(u) = h_2(u) = x$ and $h_1(z) \neq h_2(z)$, then either*

(a) $|E(H^*)| > |E(H)|$, *or*

(b) $E(H^*)| = |E(H)|$, *and* $N_H^+(h_1(z)) = N_H^+(h_2(z))$ *(respectively,* $N_H^-(h_1(z)) = N_H^-(h_2(z))$*).*

Proof. Since H is vertex-transitive, every vertex is a homomorphic image of the vertex z. Thus, for every vertex a, there is a vertex b such that $N_{H^*}^+(b) \supseteq N_H^+(a)$. Hence the out-degree of a vertex does not decrease. (Every vertex v of a vertex-transitive digraph has $d^+(v) = d^-(v) = c$ for some constant c.) Therefore H^* has at least as many arcs as H. Suppose that equality holds. Let r be the out-degree of every vertex of H and H^*. Since $N_{H^*}^+(x)$ contains both $N_H^+(h_1(z))$ and $N_H^+(h_2(z))$, the vertex x has $d_{H^*}^+(x) = r$ only if these two r-sets are equal. □

We note that if H is loopless and $N_H^+(x) = N_H^+(y)$ (respectively, $N_H^-(x) = N_H^-(y)$), then x and y cannot be neighbours.

THEOREM 3.4. *Let H be a vertex-transitive digraph. Then the H-colouring problem is* NP-*complete unless H admits a retraction to a directed cycle. In the latter case H-COL is polynomial.*

Proof. We have previously noted the second statement. The first assertion is proved by contradiction. Let H be a counterexample with the minimum number of vertices and, within all counterexamples on $|V(H)|$ vertices, one with the maximum number of arcs. That is, H is a vertex-transitive digraph that does not admit a retraction to a directed cycle, and for which the H-colouring problem is not NP-complete. Note that, in particular, H has no loop, and that by Lemma 2.2, H is not the equivalent digraph of a complete graph. Furthermore, the minimality of $|V(H)|$ implies that H is retract-free. (Otherwise the core H' of H is a vertex-transitive digraph with fewer vertices than H, and which does not retract to a directed cycle. By our choice of H, the H'-colouring problem is NP-complete. Consequently, *H-COL* is also NP-complete.) Therefore H is connected. The result is known if H has at most four vertices [10], [16], hence we may assume that H has at least five vertices. Since H is not a directed cycle, each vertex has out-degree at least two. We make the following sequence of assertions about the digraph H.

(1) H does not admit a homomorphism to a directed cycle of length greater than one. Assume H maps to a directed cycle of length greater than one and let k be the largest positive integer such that $H \rightarrow C_k$. The integer k exists because H is strong and therefore contains a directed cycle. (The integer k lies between 1 and the length of the shortest directed cycle in H.) Since the core of H is not a directed cycle, C_k is not a subdigraph of H (since C_k is retract-free, it is a retract of a given directed graph G if and only if it is both a subdigraph of G and a homomorphic image of G). Thus $(P_k, 0, k)$ is a good indicator. Let H^* denote the result of applying the indicator construction with respect to $(P_k, 0, k)$ to H. By Lemma 3.2, H^* is vertex-transitive. Since H is strong, each colour class of the C_k-colouring induces a connected component of H^*. Thus H^* has precisely k isomorphic connected components, and so the core of H^* is a vertex-transitive digraph with fewer vertices than H.

We claim that the core of H^* is not a directed cycle. By the choice of k, the digraph H has a collection $C^1, C^2, \cdots, C^m$ of directed cycles such that $gcd\{|V(C^i)|: i = 1, 2, \cdots, m\} = k$. Each C^i gives rise to a directed cycle C^{i*} in H^* of length $(1/k)|V(C^i)|$. Hence $gcd\{|V(C^{i*})|: i = 1, 2, \cdots, m\} = 1$, and H^* does not map to a directed cycle of length greater than 1. Therefore H^* does not retract to a directed cycle. Thus the H^*-colouring problem is NP-complete (by the choice of H), and so *H-COL* is also NP-complete. This completes the proof of (1).

By (1), H is not an orientation of a bipartite graph.

In the remainder of this section, we omit from our proofs the observation that the digraph which results from applying an indicator construction to a vertex-transitive digraph is itself vertex-transitive.

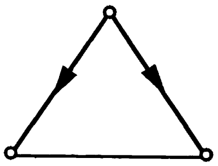
(2) Every vertex of H is incident with a double-edge. Since H is vertex-transitive, it suffices to prove that H has a double-edge. Suppose not. Then $(P_2, 0, 2)$ is a good in-z-indicator. Let H^* be the result of applying the indicator construction with respect to $(P_2, 0, 2)$ to H.

We claim that H^* does not retract to a directed cycle. Since H is strong and does not map to a directed cycle of length greater than 1, it has a collection C^1, C^2, $\cdots$, C^m of directed cycles such that $gcd\{|V(C^i)|: i = 1, 2, \cdots, m\} = 1$. Each C^i gives rise to a directed cycle C^{i*} in H^* (two such cycles if $|V(C^i)|$ is even). If $|V(C^i)|$ is odd, $|V(C^{i*})| = |V(C^i)|$, and $|V(C^{i*})| = (\frac{1}{2})|V(C^i)|$ otherwise. Therefore $gcd\{|V(C^{i*})|: i = 1, 2, \cdots, m\} = 1$. This proves the claim.

By Lemma 3.3, H^* has at least as many arcs as H. If $|E(H^*)| > |E(H)|$, then the H^*-colouring problem is NP-complete by the maximality of $|E(H)|$, and so *H-COL* is also NP-complete, which is again a contradiction. Suppose that equality holds. Let x, y, z be vertices of H such that x and y are both adjacent from z. By Lemma 3.3(b), $N_H^+(x) = N_H^+(y)$. Similarly, it follows from considering the indicator construction with respect to the out-z-indicator $(P_2, 2, 0)$ that $N_H^-(x) = N_H^-(y)$. Then x and y are non-adjacent, there is a retraction $H \to H - x$ which maps x to y, contradicting the fact that H is retract-free. This completes the proof of (2).

(3) H contains T_1 (see Fig. 3.1). Assume H does not contain T_1. Then the symmetric indicator (I, u, v) shown in Fig. 3.2 is good. Let H^* be the result of applying the indicator construction with respect to (I, u, v) to H. Then H^* is loopless and undirected. Since *undir*(H) is spanning, H^* contains the equivalent digraph of the underlying graph corresponding to H. Since H is not an orientation of a bipartite graph, H^* has an odd cycle, whence the H^*-colouring problem is NP-complete. Therefore *H-COL* is also NP-complete, which is a contradiction.

(4) Every vertex of H is incident with at least two double-edges. Suppose to the contrary that *undir*(H) is a disjoint union of double edges. Then the indicator (I_1, u, v) shown in Fig. 3.3 is a good in-z-indicator. Let H^* be the result of applying the indicator construction with respect to (I_1, u, v) to H. It is not hard to see that H^* contains a transitive triple and, therefore, does not admit a retraction to a directed



T_1

FIG. 3.1

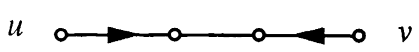


FIG. 3.2

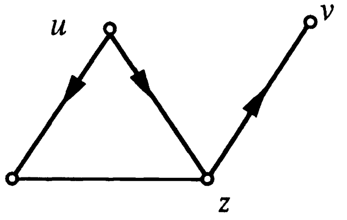


FIG. 3.3

cycle. By Lemma 3.3 the digraph H^* has at least as many arcs as does H. If $|E(H^*)| > |E(H)|$, then the H^*-colouring problem is NP-complete because of our choice of H and, consequently, H-COL is also NP-complete. Suppose that $|E(H^*)| = |E(H)|$. Let $[x, y]$ be a double edge. Then Lemma 3.3 asserts that $N^+_H(x) = N^+_H(y)$, which is a contradiction. This completes the proof of (4).

(5) *H* contains C_3^* (see Fig. 3.4). Suppose not. Then the z-indicators (I_1, u, v) and (I_2, u, v) shown in Figs. 3.5(a) and 3.5(b), respectively, are good. Let H^* and H^{**} denote the result of applying the indicator construction with respect to (I_1, u, v) and (I_2, u, v), respectively, to H. Both $E(H^*)$ and $E(H^{**})$ contain $E(H)$. If either containment is proper, we reach the contradiction that the H-colouring problem is NP-complete by Lemma 2.1 and our choice of H.

Suppose that $E(H) = E(H^*) = E(H^{**})$, and let x, y, z be an undirected path of length two in H. Then, by Lemma 3.3, $N^+_H(z) = N^+_H(x)$ and $N^-_H(z) = N^-_H(x)$. Therefore there is a retraction $H \rightarrow H - z$ which maps z to x, contradicting the fact that H is retract-free.

(6) *H* contains A_1 or A_2 (see Figs. 3.6(a) and 3.6(b), respectively). Suppose not. Then the z-indicators (I_1, u, v) and (I_2, u, v) shown in Figs. 3.7(a) and 3.7(b), respectively, are good. Let H^* and H^{**} denote the result of applying the indicator construction with respect to $(I_1, u,.v)$ and (I_2, u, v), repetitively, to H. By (5), both $E(H^*)$ and $E(H^{**})$ contain $E(H)$. If either containment is proper, we have a contradiction. Suppose that $E(H) = E(H^*) = E(H^{**})$. Consider a homomorphic image of (I_1, u, v) in H, such that u maps to x, and z maps to $y \neq x$ (the vertex y exists by (4)). By Lemma 3.3, $N^+_H(x) = N^+_H(y)$. Since there also exists a homomorphism of (I_2, u, v) to H such that u maps to x and z maps to y, we also have $N^-_H(x) = N^-_H(y)$. But then there is a retraction $H \rightarrow H - x$ that maps x to y, which is a contradiction. This completes the proof of (6).

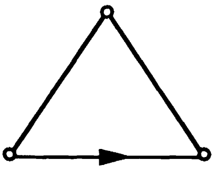


FIG. 3.4. C_3^*.

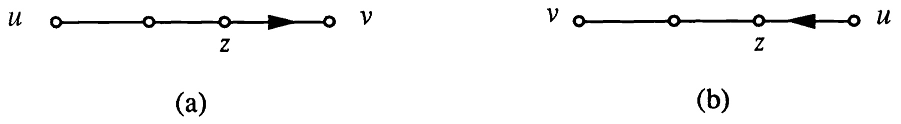


FIG. 3.5

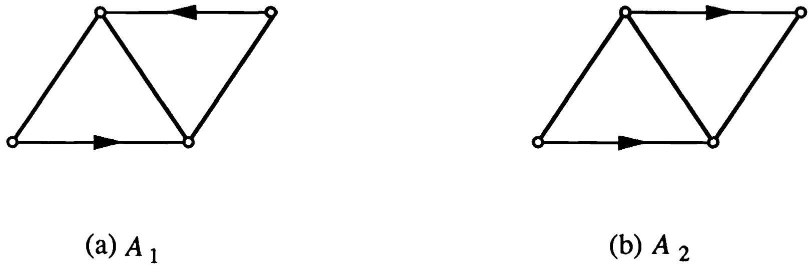


FIG. 3.6

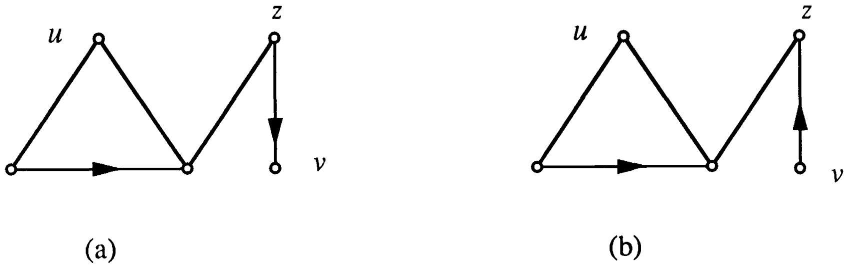


FIG. 3.7

(7) **H contains at least one of X_1, X_2, X_3, X_4, X_5 (see Figs. 3.8(a)–(e), respectively).** Suppose first that H contains A_1, but none of X_1, X_2, X_3. Then the indicators (I_1, u, v) and (I_2, u, v) shown in Figs. 3.9(a) and (b), respectively, are good. The remaining details are similar to those in (5), and the reader should have little difficulty in completing the proof.

Similarly, if H contains A_2 but none of X_3, X_4, X_5, the indicators (I_3, u, v) and (I_4, u, v) shown in Figs. 3.10(a) and (b), respectively, are good. The details are again left to the reader.

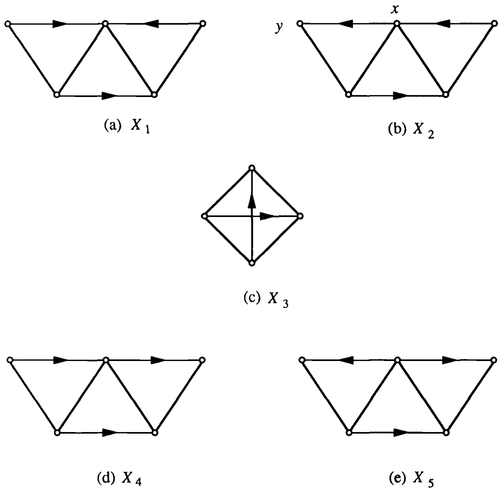


FIG. 3.8

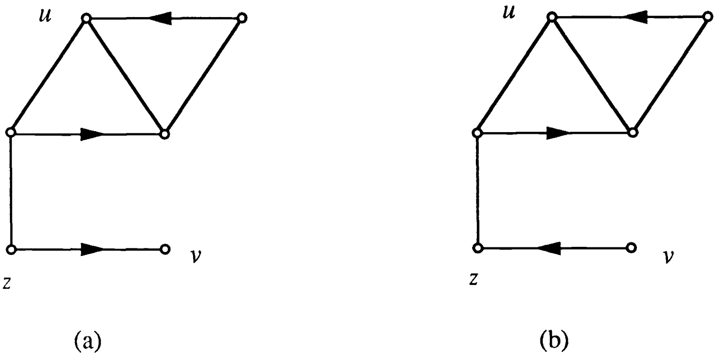


FIG. 3.9

(8) H contains neither X_1 nor X_5. We prove that if H contains X_1 or X_5, then the H-colouring problem is NP-complete. Since X_5 is the converse of X_1, it suffices to prove the result when H contains X_1.

Let x and y be vertices of H as shown in Fig. 3.11(a). Let (I_1, u, v) and (I_2, u, v) be the z-indicators shown in Figs. 3.11(b) and 3.11(c), respectively. Suppose first that both indicators are good, and let H^* and H^{**} denote the result of applying the indicator construction with respect to (I_1, u, v) and (I_2, u, v), respectively, to H. Both $E(H^*)$ and $E(H^{**})$ contain $E(H)$. If either containment is proper, the result follows from Lemma 2.1 and our choice of H. Hence assume $E(H) = E(H^*) = E(H^{**})$. Then, by Lemma 3.3, $N_H^+(x) = N_H^+(y)$ and $N_H^-(x) = N_H^-(y)$. This contradicts the fact that H is retract-free.

Now suppose that one of the indicators is not good. We may assume that *undir*(H) is bipartite, otherwise H-COL is NP-complete. Let C be a connected component of *undir*(H) and let (R, B) be a two-colouring of C. Then $H[R]$ is a vertex-transitive digraph with fewer vertices than H. If there exists a homomorphism of either I_1 or I_2 to H such that u and v map to the same vertex, then $H[R]$ contains a transitive triple. Therefore $H[R]$ does not map to a directed cycle of length greater than 1. By our choice of H, $H[R]$-COL is NP-complete. Let $r \in R$. There exists an even integer k such that for every vertex x in R, there is an undirected (r, x)-walk in H of length k. Let P be (the equivalent digraph of) an undirected path of length k, with origin a and terminus b. Let $H^{\sim}$ be the result of the applying the subindicator construction with respect to (P, a, b) and r to H. Then $H^{\sim} = H[R]$, and so the result follows from Lemma 2.3.

(9) H does not contain X_3. Suppose H contains X_3. We show that H-COL is NP-complete. Let x and y be vertices of H as shown in Fig. 3.12(a). Let (I_1, u, v) be the in-

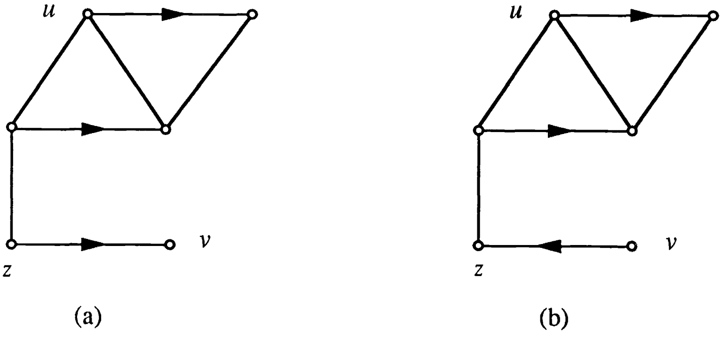


FIG. 3.10

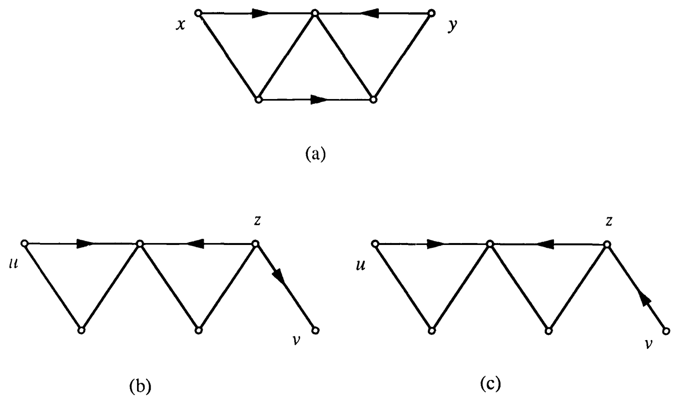


FIG. 3.11

z-indicator shown in Fig. 3.12(b), and let H^* be the result of applying the indicator construction with respect to (I_1, u, v) to H. Then $E(H^*)$ contains $E(H)$.

Suppose that $E(H^*) = E(H)$. Then, by Lemma 3.3, $N_H^+(x) = N_H^+(y)$. xy is an arc of H, therefore H has a loop at y, which is a contradiction.

Thus $E(H^*)$ properly contains $E(H)$. If H^* has no loops, then the H-colouring problem is NP-complete by Lemma 2.1 and our choice of H. Hence we may assume that H^* has a loop. Thus H contains an undirected triangle or the graph shown in Fig. 3.13(a). In the former case H-COL is NP-complete by Lemma 2.2. In the latter case, let (I_2, u, v) be the indicator shown in Fig. 3.13(b), and let H^{**} be the result of applying the indicator construction with respect to (I_2, u, v) to H. Note that $E(H^{**})$ contains $E(H)$.

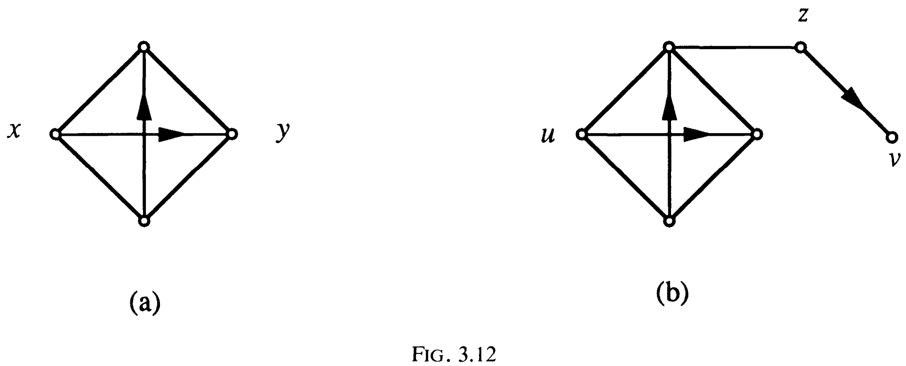


FIG. 3.12

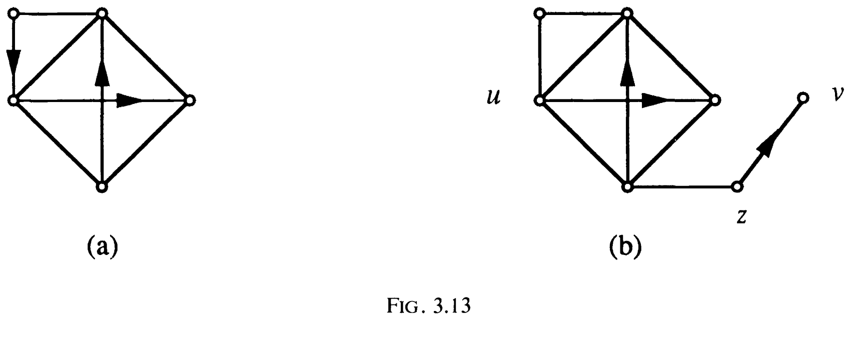


FIG. 3.13

Suppose that $E(H^{**}) = E(H)$. Then, as above, we see that H has a loop at y, which is a contradiction.

Thus $E(H^{**})$ properly contains $E(H)$. If H^{**} has no loops, the H-colouring problem is NP-complete by Lemma 2.1 and our choice of H. Hence we may assume that H^{**} has a loop. Then H contains an undirected triangle, the digraph X_1, or the digraph shown in Fig. 3.14(a). In the first case *H-COL* is NP-complete. The second case contradicts (8). It remains to consider the last case. Let (I_3, u, v) be the symmetric indicator shown in Fig. 3.14(b). We may assume that H does not contain an undirected three-cycle; otherwise *H-COL* is NP-complete. Therefore (I_3, u, v) is good. Let H^{***} be the digraph that results from applying the indicator construction with respect to (I_3, u, v) to H. It may be directly verified that H^{***} contains an undirected five-cycle. This completes the proof of (9).

(10) ***H* does not contain X_2.** Suppose to the contrary that H contains X_2. We show that *H-COL* is NP-complete. It may be assumed that H does not contain X_1, X_3, or X_5. Let x and y be vertices of H as shown in Fig. 3.8(b). Let (I, u, v) be the indicator shown in Fig. 3.9(a), and let H^* be the result of applying the indicator construction with respect to (I, u, v) to H. Since neither X_1 nor X_3 is a subdigraph of H, the digraph H^* is loopless, unless H contains an undirected triangle, in which case we are done by Lemma 2.2. Note that $E(H^*)$ contains $E(H)$. If the containment is proper, the result follows from Lemma 3.3 and our choice of H. Hence assume that $E(H^*) = E(H)$. Then by Lemma 3.3, $N_H^+(x) = N_H^+(y)$. This is a contradiction.

(11) ***H* does not contain X_4.** The proof is similar to (10). The indicator needed is shown in Fig. 3.9(b). The details are omitted.

Hence the digraph H cannot exist. This completes the proof of Theorem 3.4. □

COROLLARY 3.5. *Let H be an arc-transitive digraph. Then the H-colouring problem is* NP-*complete, unless H admits a retraction to C_n, P_0, or P_1. In the latter case H-COL is polynomial.*

Proof. We have already noted the second statement. Let H be an arc-transitive digraph with at least one arc. We may assume without loss of generality that H has no isolated vertices. Then either H is smooth, or every vertex of H is a source or a sink. In the former case H is vertex-transitive, so the result follows from Theorem 3.4. In the latter case P_1 is a retract. This completes the proof. □

Since a connected vertex-transitive digraph H is strong, it admits a retraction to a directed cycle if and only if the length of every directed cycle in H is divisible by the directed girth of H. When H is a Cayley digraph of a finite group we are able to give another characterisation.

Let Γ be a finite group. We denote by $\Gamma(S)$ the Cayley digraph with symbol S. That is, the digraph with vertex-set Γ and arc-set $E(\Gamma(S)) = \{xy \mid yx^{-1} \in S\}$.

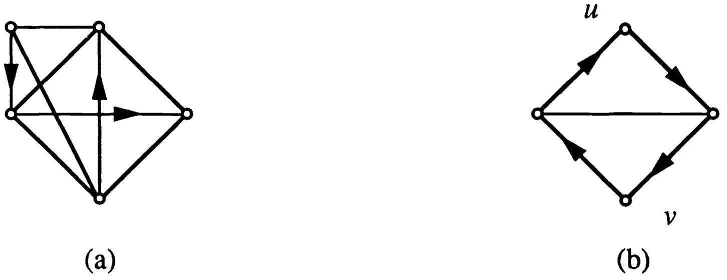


FIG. 3.14

It is well known that a Cayley digraph $\Gamma(S)$ is connected only if the set S generates Γ. Since Cayley digraphs are vertex-transitive (because, for any $a \in \Gamma$, the mapping $x \rightarrow xa$ is an automorphism), a connected Cayley digraph is strong.

LEMMA 3.6. *Suppose that S generates Γ and H is a nontrivial normal subgroup of Γ of index h. If S is a union of cosets of H, then the Cayley digraph $\Gamma/H(S/H)$ is a retract of $\Gamma(S)$ (where $S/H = \{Hx\colon Hx$ is a subset of $S\}$).*

Proof. Let $S = Hx_1 \cup Hx_2 \cup \cdots \cup Hx_k$, and let the collection of all cosets of H be $Hx_1, Hx_2, \cdots, Hx_h$. We first show that $\Gamma/H(S/H)$ is an induced subdigraph of $\Gamma(S)$. There is an arc from x_i to x_j in $\Gamma(S)$ if and only if $x_jx_i^{-1} \in Hx_m$, for some m between 1 and k; equivalently, x_ix_j is an arc if and only if $Hx_jHx_i^{-1} = Hx_m$. Therefore $\{x_1, x_2, \cdots, x_h\}$ induces a copy T of $\Gamma/H(S/H)$ in $\Gamma(S)$. It remains to show that there is a retraction $f\colon\Gamma(S) \rightarrow T$. For each $g \in \Gamma$, let $f(g)$ be the unique vertex x_i such that g is in Hx_i. Then f fixes $V(T)$. Let ab be an arc of $\Gamma(S)$ and $f(a) = x_s$ and $f(b) = x_t$. It is not hard to see that $Hba^{-1} = Hx_tx_s^{-1}$. Therefore $x_tx_s^{-1}$ is in S, and x_sx_t is an arc of T. Hence f is a homomorphism. This completes the proof. □

LEMMA 3.7. *Suppose that S generates Γ. There is a homomorphism of $\Gamma(S)$ to C_h if and only if S is contained in a coset of a normal subgroup of Γ with index h.*

Proof. ($\Rightarrow$) Suppose that $f\colon\Gamma(S) \rightarrow C_h$. Without loss of generality the identity e of Γ has $f(e) = 1$. Let $H = f^{-1}(1)$. Let a, b be in H. Since $\Gamma(S)$ is connected, there is a directed (e, a)-path of length zero modulo h, and a directed (e, b)-path of length zero modulo h. Consequently, there is a directed (b, ab)-path of length zero modulo h, and a directed (e, ab)-walk of length zero modulo h. Since a C_h-colouring of a connected digraph is completely determined by the colour assigned to a single vertex, we deduce that $f(ab) = 1$, that is, $ab \in H$. Similarly, $a^{-1} \in H$. Hence H is a subgroup.

Let $g \in \Gamma$ and let $x \in H$. There exists a directed (e, x)-path of length zero modulo h, a directed (e, g)-path of length r modulo h, and a directed (e, g^{-1})-path of length $(-r)$ modulo h (because there is a directed (g^{-1}, e)-path of length r modulo h and a closed directed walk containing both e and g^{-1} has length zero modulo h). Therefore there is a directed $(e, g^{-1}xg)$-walk of length zero modulo h, that is $f(g^{-1}xg) = 1$. Thus H is normal.

Let s be in S. The automorphism $x \rightarrow xs$ maps each $f^{-1}(i)$ to $f^{-1}(i+1)$, $i = 1, 2, \cdots, h$, with addition modulo h. Hence each colour class of the C_h-colouring is a coset of H. Since there are h cosets and S is contained in $f^{-1}(2)$, the proof of the implication is complete.

($\Leftarrow$) Without loss of generality $S = Hx$. The result follows from Lemma 3.6 (the graph $\Gamma/H(S/H)$ is connected because $\Gamma(S)$ is strong). □

COROLLARY 3.8. *Suppose that S generates Γ. The core of $\Gamma(S)$ is a directed cycle if and only if S is contained in a coset of a normal subgroup of index equal to the directed girth of $\Gamma(S)$.* □

We conclude this section by mentioning the group-theoretic analogue of Lemma 3.6: If H is a normal subgroup of a finite group Γ, then Γ/H is cyclic if and only if there exists an $x \in \Gamma$ such that $\Gamma = \langle Hx \rangle$.

Acknowledgments. The author thanks Jørgen Bang-Jensen and Pavol Hell for helpful conversations, and the referee for valuable suggestions.

REFERENCES

[1] M. ALBERTSON, P. CATLIN, AND L. GIBBONS, *Homomorphisms of 3-chromatic graphs* II, Congr. Numer., 47 (1985), pp. 19–28.

[2] J. Bang-Jensen, *On the complexity of generalized colourings by directed graphs*, preprint 1989, No. 3, Institut for Matematik og Datalogi, Odense Universitet, Odense, Denmark.

[3] J. Bang-Jensen and P. Hell, *The effect of two cycles on the complexity of colourings by directed graphs*, Discrete Appl. Math., 26 (1990), pp. 1–23.

[4] J. Bang-Jensen, P. Hell, and G. MacGillivray, *The complexity of colourings by semicomplete digraphs*, SIAM J. Discrete Math., 1 (1988), pp. 281–298.

[5] ———, *Hereditarily hard colouring problems*, submitted.

[6] ———, *On the complexity of colouring by superdigraphs of bipartite graphs*, submitted.

[7] G. Bloom and S. Burr, *On unavoidable digraphs in orientations of graphs*, J. Graph Theory, 11 (1987), pp. 453–462.

[8] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, Macmillan, Great Britain, 1976.

[9] A. E. Brouwer and H. J. Veldman, *Contractability and* NP-*completeness*, J. Graph Theory, 11 (1987), pp. 71–79.

[10] W. Gutjahr, *Farbung durch gerichtete Graphen*, Diplomarbeit, Institute for Information Processing, IIG, Technical University Graz, Graz, Austria, 1988.

[11] ———, Ph.D. thesis, Technical University Graz, Graz, Austria, 1989 (in preparation).

[12] W. Gutjahr, E. Welzl, and G. Woeginger, *Polynomial graph colourings*, Tech. Report B-88-06 (Informatik), Freie Universtät Berlin, Berlin, Germany.

[13] R. Häggkvist, P. Hell, D. J. Miller, and V. Neuman Lara, *On multiplicative graphs and the product conjecture*, Combinatorica, 8 (1988), pp. 63–74.

[14] P. Hell and J. Nešetřil, *On the complexity of H-colouring*, J. Combin. Theory Ser. B, 48 (1990), pp. 92–110.

[15] G. MacGillivray, *The complexity of generalised colourings*, Ph.D. thesis, Simon Fraser University, Burnaby, BC, Canada, 1989.

[16] H. A. Maurer, J. H. Sudborough, and E. Welzl, *On the complexity of the general colouring problem*, Inform. and Control, 51 (1981), pp. 123–145.

[17] J. Nešetřil and A. Pultr, *On classes of relations determined by subobjects and factorobjects*, Discrete Math., 22 (1978), pp. 287–300.

[18] E. Welzl, *Colour families are dense*, Theoret. Comput. Sci., 17 (1982), pp. 29–41.

SIAM J. DISC. MATH.
Vol. 4, No. 3, pp. 409–412, August 1991

009

A LOWER BOUND ON PROBABILISTIC ALGORITHMS FOR DISTRIBUTIVE RING COLORING*

MONI NAOR†

Abstract. Suppose that n processors are arranged in a ring and can communicate only with their immediate neighbors. It is shown that any probabilistic algorithm for 3 coloring the ring must take at least $\frac{1}{2}\log^* n - 2$ rounds, otherwise the probability that all processors are colored legally is less than $\frac{1}{2}$. A similar time bound holds for selecting a maximal independent set. The bound is tight (up to a constant factor) in light of the deterministic algorithms of Cole and Vishkin [*Inform. and Control*, 70 (1986), pp. 32–53] and extends the lower bound for deterministic algorithms of Linial [Proc. 28th IEEE Foundations of Computer Science Symposium, 1987, pp. 331–335].

Key words. distributed computation, probabilistic algorithms, graph coloring

AMS(MOS) subject classifications. 68M10, 68Q20, 68R05, 68R10

1. Introduction. In [L] Linial considered the following problem: n processors are connected in a ring and can communicate with their immediate neighbors. They wish to decide on an assignment of one of three colors to each processor, such that no two neighboring processors are assigned the same color (a legal coloring). The question is what is the radius of the neighborhood around each processor which must be considered in order to decide on the coloring. The system is assumed to be completely synchronous, the communication reliable, and there are no limitations on the internal computation of each processor or on the length of the messages sent. The processors are identical, except that each one has a unique id in the range $\{1 \cdot\cdot n\}$. The id's are assigned in some arbitrary manner, not known initially to the processors. The radius of the neighborhood that affects how a processor is colored is exactly the number of rounds it takes to execute the algorithm.

Linial [L] has shown a lower bound of $\frac{1}{2}\log^* n - 4$ rounds on any deterministic algorithm for coloring the ring with three colors. This bound is tight up to a constant factor, since Cole and Vishkin [CV] and Goldberg, Plotkin, and Shannon [GPS] have provided an $O(\log^* n)$ round algorithm for achieving it.

In this paper we consider probabilistic algorithms for that task. Each processor is equipped with a perfect source of randomness, and the processor's actions can depend in any way on its coin flips. The performance of an algorithm is now measured in terms of the probability of success as a function of the number of rounds. We show that allowing the processors to flip coins does not help: any algorithm that runs in less than $\frac{1}{2}\log^* n - 2$ rounds has a high probability of failure, i.e., there will be at least two adjacent nodes whose color is the same.

The 3-coloring problem is closely related to the maximal independent set problem: Each processor should decide if it is in the set or not, no two adjacent processors are allowed to be in the set, and for every processor not in the set, one of its neighbors must be in the set. Any algorithm for 3-coloring a ring can be translated with two additional rounds into one that finds a maximal independent set and vice versa. Thus, a lower bound on the 3-coloring problem provides a similar lower bound for the maximal independent set problem. Cole and Vishkin [CV] provided an algorithm for the maximal independent set, and Goldberg, Plotkin, and Shannon [GPS] have generalized it to colorings of various degree bounded graphs.

* Received by the editors June 20, 1990; accepted for publication (in revised form) September 4, 1990.

† IBM Research Division, Almaden Research Center, San Jose, California 95120.

In [BNN] the number of bits of communication required to achieve 3-coloring is investigated. (That is, messages are 1-bit long.) It is shown that in any deterministic algorithm it must be $\Omega(\log n)$ which is tight by the [CV] algorithm. Interestingly, for randomized algorithms it is $\Theta(\sqrt{\log n})$.

2. The lower bound.

THEOREM 2.1. *Let $k = n^{1/3}$. Any probabilistic algorithm for 3-coloring a ring of n processors that takes less than $t = \frac{1}{2}\log^* n - b - 2$ rounds, has probability at most $(1 - (1/\log^{(b)} n))^{k/2t} + 2t/k$ to produce a legal coloring.*

Proof. In any probabilistic algorithm it can be assumed that the processors first make their random choices and from then on act deterministically. Since the processors' actions are determined by the order of the id's and the random numbers selected in the system, an algorithm that runs in t rounds can be simulated by one where the processors send to each other their id number and their random selections. After t rounds each processor knows the random numbers selected by $2t + 1$ processors: itself and the $2t$ processors that are of distance at most t from it. Based on this information it decides on a color. Let D be the range from which the processors make their random selection and let $R = D \times \{1 \cdots n\}$. Any $r \in R$ corresponds to a selection for the random choices of a processor concatenated with its id. After t rounds the information any processor has corresponds to a vector $(r_1, r_2, \cdots, r_{2t+1})$ where $r_i \in R$. Thus, any t-round algorithm induces a 3-coloring of the vectors $\{(r_1, r_2, \cdots, r_{2t+1}) | r_i \in R\}$ by associating a vector with the color that the algorithm assigns a processor with neighborhood information represented by the vector.

We concentrate on a segment of $k + 2t$ consecutive processors on the ring. Suppose that the adversary assigns each processor an id by choosing it independently from $\{1 \cdots n\}$. With probability at least $1 - 2t/k$ all the id's in the segment are unique. This is true, since the probability that at least two processors choose the same id is bounded by $\binom{k+2t}{2}$ times the probability that two specific processors chose the same id which is $1/n$ and $\binom{k+2t}{2} \cdot 1/n \leqq 2t/k$. If the id's chosen are not unique, we consider it as if the algorithm "won." The lower bound of the theorem will follow if we can bound by $(1 - (1/\log^{(b)} n))^{k/2t}$ the probability that the processors of the segment choose a legal coloring in case each processor i selects at random $r_i \in \{1, \cdots, R\}$. This is true since for any two events A and B, $\Pr[A \mid B] \leqq \Pr[A] + \Pr[\bar{B}]$. In our case A is the event that the algorithm succeeds and B is the event that the adversary assigns unique id's to the segment.

Consider the directed graph $G_{R,2t+1}$: each node corresponds to a vector $(r_1, r_2, \cdots, r_{2t+1})$ such that $r_i \in R$; node $(r_1, r_2, \cdots, r_{2t+1})$ is connected to node $(s_1, s_2, \cdots, s_{2t+1})$ if and only if $r_i = s_{i+1}$ for $2 \leqq i \leqq 2t$. The edge in this case is called $(r_1, r_2, \cdots, r_{2t+1}, s_{2t+1})$ (or, equivalently, $(r_1, s_1, \cdots, s_{2t+1})$).

This graph was used in the lower bound proof for deterministic algorithms in [L]. It was shown that any algorithm that colors the ring must define a legal coloring of $G_{R,2t+1}$ and by deriving a bound on the chromatic number of $G_{R,2t+1}$ as a function of t, the lower bound was shown. Here the situation is more complicated, since the ring coloring algorithm does not necessarily define a legal coloring of $G_{R,2t+1}$: the probability of selecting an edge with similarly colored endpoints might be small. (We call such an edge monochromatic.) Instead, we will show a lower bound on the fraction of monochromatic edges.

The process of selecting the random numbers by the $k + 2t$ processors in the segment corresponds to selecting a (not necessarily simple) path of length k in the graph $G_{R,2t+1}$: if the random numbers selected are $r_1, r_2, \cdots, r_{k+2t}$, then the path selected is $v_1, v_2, \cdots, v_{k+2t}$ where $v_i = (r_{i-t}, \cdots, r_i, \cdots, r_{i+t})$. Let $z_1, z_2, \cdots, z_k$ be the edges of this path. Each z_i is uniformly distributed over the edges of $G_{R,2t+1}$, and z_i is independent of all z_j

for j such that $|j-i| \geqq 2(t+1)$. Therefore, we have $k/2(t+1)$ random variables z_1, $z_{2t+2}, \cdots, z_k$ that are mutually independent and each is a random choice of an edge in $G_{R,2t+1}$.

For any coloring (not necessarily legal) of $G_{R,2t+1}$ we call an edge *monochromatic* if both of its endpoints are assigned the same color. Let p be the probability that an edge chosen at random in $G_{R,2t+1}$ is monochromatic. For a randomly chosen path of length k in $G_{R,2t+1}$,

Prob [some edge is monochromatic]

$$\geqq \text{Prob [at least one of } \{z_1, z_{2t+2}, \cdots, z_k\} \text{ is monochromatic]} \geqq 1-(1-p)^{k/(2t+2)}.$$

If we show that for $t = \frac{1}{2}\log^* n - b - 2$, for all three colorings of $G_{R,2t+1}$, $p \geqq 1/\log^{(b)}$, then the probability that any t-round algorithm succeeds is at most $(1-(1/\log^{(b)} n))^{k/2t} + 2t/k$.

Consider the series of graphs $G_{R,1}, G_{R,2}, \cdots, G_{R,2t+1}$, where $G_{R,i}$ is defined similarly to $G_{R,2t+1}$. Let $c_t = 3$ and $c_i = 2^{c_{i+1}}$. Define $p_1, p_2, \cdots, p_t$ by setting $p_1 = 1/c_1$ and $p_{i+1} = p_i^2/(2c_{i+1})$. We will show that p_i is such that for every coloring of $G_{R,i}$ with c_i colors Prob [random edge in $G_{R,i}$ is monochromatic] $> p_i$.

PROPOSITION 2.1. *For any coloring of $G_{R,1}$ with c_1 colors,*

$$\text{Prob [\textit{random edge in } } G_{R,1} \text{ \textit{is monochromatic}]} \geqq p_1 = \frac{1}{c_1}.$$

Proof. $G_{R,0}$ is actually a complete graph with self loops. Therefore, to minimize the probability that two nodes have the same colors, all color classes should be of the same size, and we get that $p_1 = 1/c_1$. □

LEMMA 2.1. *Assume that for any coloring of $G_{R,i}$ with c_i colors, the probability that a random edge is monochromatic is at least p_i. Then for any coloring of $G_{R,i+1}$ with c_{i+1} colors*

$$\text{Prob [\textit{random edge in } } G_{R,i+1} \text{ \textit{is monochromatic}]} \geqq p_{i+1} = \frac{p_i^2}{2 \cdot c_{i+1}}.$$

Proof. The nodes of $G_{R,i+1}$ correspond naturally to the edges of $G_{R,i}$. Selecting a random edge in $G_{R,i+1}$ corresponds to selecting a path of length two in $G_{R,i}$. If we can show that for every coloring of the edges of $G_{R,i}$ with c_{i+1} colors the probability that two edges in a random path have the same color is at least $p_i^2/(2 \cdot c_{i+1})$, then we are done.

Given a coloring of the edges of $G_{R,i}$ with c_{i+1} colors we define a corresponding coloring of the nodes of $G_{R,i}$ with $c_i = 2^{c_{i+1}}$ colors by the following procedure.

For a node v call a color c *frequent* for v if at least a fraction $f_{i+1} = p_i/2c_{i+1}$ of the edges starting at v are colored c. An edge $e = (v, u)$ whose color is frequent for v is called *frequent*. Otherwise, it is called *infrequent*. Let S_v be the set of frequent colors of v and let $C_v \in \{0, 1\}^{c_{i+1}}$ be the characteristic vector of S_v. Node v is assigned the color C_v.

This is a refinement of the coloring used in [L], where the color a node is assigned is the characteristic vector of the set of all colors that meet that node.

CLAIM 2.1. *The fraction of infrequent edges is at most $f_{i+1} \cdot c_{i+1} = p_i/2$.*

Proof. For every node, at most f_{i+1} of the edges starting from it are colored by any color not in S_v. Thus, the fraction of infrequent edges is at most $c_{i+1} \cdot f_{i+1} = p_i/2$. □

CLAIM 2.2. *For every edge coloring of $G_{R,i+1}$ with c_{i+1} colors and the corresponding node coloring with $c_i = 2^{c_{i+1}}$ colors, at least $p_i/2$ of the edges are both frequent and monochromatic.*

Proof. By assumption, p_i of the edges are monochromatic and by the previous claim at most $p_i/2$ of the edges are infrequent. Thus, at least $p_i/2$ of the edges are both monochromatic and frequent. □

Fix a coloring of the edges of $G_{R,i}$ and its corresponding node coloring. Suppose that a path of length 2 is randomly selected at $G_{R,i}$. If the first edge $e = (v, u)$ is monochromatic and frequent, then the color of e is frequent at u as well as at v, since (v, u) being monochromatic means that the lists of frequent colors at v and u are the same. Therefore, there is probability at least f_{i+1} that the second edge (starting from u) will be colored as $e = (v, u)$. Thus the probability that both events occur is at least $p_i/2 \cdot f_{i+1} = p_i/2 \cdot p_i/c_{i+1} = p_i^2/(2c_{i+1})$, concluding the proof of the lemma. □

Applying the lemma t times we get that

$$\text{Prob}\,[\text{random edge } z_j \text{ in } G_{R,2t+1} \text{ is monochromatic}] > p_t.$$

By definition

$$p_t > \left(\frac{p_1}{2c_1}\right)^{2^t} = \frac{1}{(2c_1^2)^{2^t}}.$$

Now,

$$c_1 = \left.2^{2^{2^{\cdot^{\cdot^{\cdot^{2^3}}}}}}\right\} 2t+1.$$

Thus, if $t = \frac{1}{2} \log^* n - b - 2$ for some $b > 0$, then $p_t > 1/\log^{(b)} n$ and we get our theorem. □

A different proof for the fact that there are many monochromatic edges was suggested by Alon (personal communication, 1990): It relies on the fact that there is lower bound on the chromatic number of $G_{R,i}$, and thus for any large enough subset of $\{1 \cdots R\}$, the induced subgraph contains at least one edge which is monochromatic.

Acknowledgments. Stimulating discussions with Richard Karp, Vijaya Ramachandran, and Danny Soroker are gratefully acknowledged. I thank the two Josephs, Yossi Azar, and Seffi Naor, and the anonymous referee for carefully reading the manuscript.

REFERENCES

[BNN] A. Bar-Noy, J. Naor, and M. Naor, *One-bit algorithms*, Distributed Computing, 4 (1990), pp. 3–8.

[CV] R. Cole and U. Vishkin, *Deterministic coin tossing with applications to optimal parallel list ranking*, Inform. and Control, 70 (1986), pp. 32–53.

[GPS] A. Goldberg, S. Plotkin, and G. Shannon, *Efficient parallel algorithms for* $(\Delta + 1)$*-coloring and maximal independent set problems*, Proc. 19th ACM Symp. on Theory of Computing, 1987, pp. 315-324.

[L] N. Linial, *Distributive graphs algorithms—global solutions from local data*, Proc. 28th IEEE Foundations of Computer Science Symposium, 1987, pp. 331–335.

SIAM J. DISC. MATH.
Vol. 4, No. 3, pp. 413–422, August 1991

010

GENERATING THE LINEAR EXTENSIONS OF CERTAIN POSETS BY TRANSPOSITIONS*

GARA PRUESSE† AND FRANK RUSKEY‡

Abstract. For poset $\mathscr{P}$ define the graph $G(\mathscr{P})$ whose vertices are the linear extensions of $\mathscr{P}$ and where two vertices are connected by an edge if the corresponding linear extensions differ by a transposition. Let $\mathscr{P}$ be a poset and M a subset of its minimal elements for which $G(\mathscr{P} - M)$ has a Hamilton path (cycle). If no element of $\mathscr{P} - M$ has exactly one descendant in M then $G(\mathscr{P})$ also has a Hamilton path (cycle). Given only $\mathscr{P}$ and a constant average time algorithm for building a path (cycle) in $G(\mathscr{P} - M)$, a Hamilton path (cycle) can also be constructed in $G(\mathscr{P})$ in constant average time.

As an application of the results stated above it is proved that the linear extensions of any ranked poset in which every nonmaximal element has at least two (upper) covers can be generated by transpositions in constant average time.

Key words. linear extension, transposition, Hamiltonian graph

AMS(MOS) subject classifications. 05C45, 06A05, 06A06, 68Q25

1. Introduction. The set of linear extensions of a poset are of great combinatorial interest. The linear extensions of posets, depending on the poset, give rise to permutations, alternating permutations, Young tableaux, multiset permutations, and so on. General information about posets and linear extensions may be found in Aigner [1], Knuth [7], and Stanley [16]. For computer scientists the linear extensions are the so called "topological sortings" of the poset. Several algorithms have been published for listing all topological sortings (see Kalvin and Varol [6]). In this paper we consider the problem of listing all linear extensions of a poset so that successive linear extensions differ by a transposition; if such a listing can be constructed then we say that the linear extensions can be *generated by transpositions*. In general such a listing is not possible; for example, if a poset consists of two disjoint chains of lengths $n > 1$ and $m > 1$, then the linear extensions can be generated by transpositions if and only if n and m are both odd (see Eades, Hickey, and Read [3], Buck and Wiedemann [2], or Ruskey [10]).

If the poset relation is empty then the well-known algorithm of Steinhaus [17], Johnson [5], and Trotter [18] can be used to generate permutations by adjacent transpositions. This algorithm is described in several books including the chapter "Combinatorial Card Problems" of Gardner [4]. In a sense we are trying to extend this algorithm to other classes of posets. Related questions were considered by Lehmer [9] and Knuth [8].

Let us review some definitions and introduce our notation. A *poset* $\mathscr{P}$ is a partial order $R(\mathscr{P})$ on a set $S(\mathscr{P})$. The number of elements in $S(\mathscr{P})$ is denoted $|\mathscr{P}|$. If $(a, b) \in R(\mathscr{P})$ then we write $a \leqq b$ or $b \geqq a$; if also $a \neq b$ then we write $a < b$ and say that a is a *descendant* of b. For elements $a, b \in S(\mathscr{P})$ we say that b *covers* a if $a < b$ and there is no c such that $a < c < b$. An element is *minimal* if it covers no element. A permutation $a_1 a_2 \cdots a_n$ of $S(\mathscr{P})$ is a *linear extension* of $\mathscr{P}$ if $a_i < a_j$ implies $i < j$. The

* Received by the editors November 28, 1988; accepted for publication (in revised form) May 8, 1990.

† Department of Computer Science, University of Toronto, Toronto, Ontario, M5S 1A1 Canada. This research was supported in part by a summer Undergraduate Scholarship from the Natural Science and Engineering Research Council at the University of Victoria.

‡ Department of Computer Science, University of Victoria, Victoria, British Columbia, V8W 2Y2 Canada. This research was supported in part by Natural Sciences and Engineering Research Council of Canada grant A3379.

set of all linear extensions of $\mathscr{P}$ is denoted $E(\mathscr{P})$, and the number of linear extensions is denoted $e(\mathscr{P})$. A poset is *ranked* if there is a function ρ from $S(\mathscr{P})$ to the natural numbers such that $\rho(a) = 0$ if a is minimal, and $\rho(b) = \rho(a) + 1$ if b covers a.

Define a graph $G(\mathscr{P})$ that has vertex set $E(\mathscr{P})$ and edges joining those linear extensions that differ by a transposition. The graph $G(\mathscr{P})$ is called the *transposition graph* of $\mathscr{P}$. It is connected and bipartite. If the sizes of the two partite sets are equal then $\mathscr{P}$ is said to be *balanced*. The subgraph of $G(\mathscr{P})$ that has the same vertex set but where adjacent vertices are joined only by adjacent transpositions is denoted $G'(\mathscr{P})$ and is called the *adjacent transposition graph*.

A conjecture of Ruskey [11] states that the transposition graph of every balanced poset has a Hamilton path. The results of this paper provide further evidence in support of this conjecture.

Our main result is the theorem stated below. It is proven in the next section.

THEOREM. *The linear extensions of any ranked poset in which every nonmaximal element has at least two upper covers can be generated by transpositions.*

An algorithm for generating the linear extensions of a poset is said to run in *constant average time* if the total amount of computation is $O(e(\mathscr{P}))$, independent of $|\mathscr{P}|$. There is no constant average time algorithm for generating all linear extensions of an arbitrary poset. However, we show in the third section of this paper that the linear extensions of a poset satisfying the conditions of the theorem can be generated in constant average time. Thus we add to the growing list of combinatorial objects that can be generated in constant average time.

2. The proofs. In this section we prove the theorem. We first prove that a certain type of generation of all unrestricted permutations is possible.

LEMMA 1. *There exists a list $L_n = p_1, p_2, \cdots, p_{n!}$ of the permutations of $\{1, 2, \cdots, n\}$ for $n \geqq 2$, such that* (a) *if i is odd then p_i and p_{i+1} differ by the transposition of the first two elements, and* (b) *if i is even then p_i and p_{i+1} differ by a transposition, and* (c) *the permutations p_1 and $p_{n!}$ differ by a transposition.*

Proof. If $n = 2$ then $L_2 = 12, 21$ provides such a listing. For $n > 2$, assume that such a listing L_{n-1} is available for the permutations of $\{1, 2, \cdots, n-1\}$, and that $p_i = a_1a_2a_3\cdots a_{n-1}$ and $p_{i+1} = a_2a_1a_3\cdots a_{n-1}$ for i odd are successive elements of L_{n-1}. We now insert the n in all possible ways into p_i and p_{i+1} as shown below. The final permutation depends on whether n is even or odd. The first list corresponds to n odd and the second list to n even. The list may be generated from top to bottom, or in reverse order from bottom to top. In going from the top permutation to the bottom permutation we will say that the n is moving from right to left:

$$\begin{array}{ll}
a_1a_2\cdots a_{n-1}n & a_1a_2\cdots a_{n-1}n \\
a_2a_1\cdots a_{n-1}n & a_2a_1\cdots a_{n-1}n \\
a_2a_1\cdots na_{n-1} & a_2a_1\cdots na_{n-1} \\
a_1a_2\cdots na_{n-1} & a_1a_2\cdots na_{n-1} \\
\vdots \quad \vdots & \vdots \quad \vdots \\
a_1a_2n\cdots a_{n-1} & a_2a_1n\cdots a_{n-1} \\
a_2a_1n\cdots a_{n-1} & a_1a_2n\cdots a_{n-1} \\
a_2na_1\cdots a_{n-1} & a_1na_2\cdots a_{n-1}
\end{array}$$

$$
\begin{array}{ll}
na_2a_1\cdots a_{n-1} & na_1a_2\cdots a_{n-1}\\
na_1a_2\cdots a_{n-1} & na_2a_1\cdots a_{n-1}\\
a_1na_2\cdots a_{n-1} & a_2na_1\cdots a_{n-1}.
\end{array}
$$

The n will move from right to left through p_1, p_2, then from left to right through p_3, p_4, then from right to left through p_5, p_6, and so on, with the direction changing for each pair of permutations of $\{1, 2, \cdots, n-1\}$. We now need to discuss what happens at the interfaces of the pairs of permutations, say between p_{i+1} and p_{i+2}. For n even the a_1 and a_2 are in different positions in the initial and final permutations of the list above so there is no problem at the interface; simply transpose the elements that were transposed in going from p_{i+1} to p_{i+2}. However, if n is odd then we must be a little more careful in our argument. Observe that (no matter what the parity of n) p_i and p_{i+3} differ by a transposition if i is odd, namely the transposition that takes p_{i+1} to p_{i+2}. Thus the following partial list of permutations of $\{1, 2, \cdots, n\}$ can be generated by transposition, where p_i, $1 \leqq i \leqq n!$ are as in L_n:

$$L'_n = p_1, p_4, p_5, p_8, p_9, \cdots, p_{4j-3}, p_{4j}, p_{4j+1}, p_{4j+4}, \cdots, p_{n!-3}, p_{n!}.$$

Let Γ be the function that transposes the first two elements of a permutation. Also, for a list of permutations $L = q_1, \cdots, q_m$, let $\Gamma(L)$ denote the list $\Gamma(q_1), \cdots, \Gamma(q_m)$. Note that $L_n - L'_n = \Gamma(L'_n)$.

In the generation of L_n for n odd, we are in effect expanding the list L'_{n-1} as follows: for each permutation p of L'_{n-1}, generate $\Gamma(p)$, and insert n in all possible ways into p and $\Gamma(p)$ as shown above. The final permutation in the expansion induces the permutation p on $\{1, 2, \cdots, n-1\}$; thus we may make the transposition that takes p to its successor in L'_{n-1}, and continue.

The initial permutation is $12\cdots n$ and the final permutation is 21, 132, 1432 for $n = 2, 3, 4$, and for $n \geqq 5$ is $143256\cdots n$. Thus the first and last permutations differ by a transposition. The reason that the pattern regularizes for $n \geqq 5$ is because $n = 5$ is the smallest value of n for which $(n-1)!$ is divisible by four. □

There is no other published algorithm for generating permutations that has the properties (a), (b), (c) of Lemma 1. Most known permutation algorithms are surveyed in Sedgewick [14].

DEFINITION. A *B-poset* is a poset $\mathscr{P}$ where

$$S(\mathscr{P}) = \{a_1, a_2, \cdots, a_n, b_1, b_2, \cdots, b_m\}$$

and $R(\mathscr{P})$ is the transitive closure of

$$X \cup \{(a_i, a_{i+1}) \mid i = 1, 2, \cdots, n-1\} \cup \{(b_i, b_{i+1}) \mid i = 1, 2, \cdots, m-1\},$$

where X is a set of relations of the form $a_i < b_j$. An example is shown in Fig. 1.

We call m and n the *parameters* of a B-poset. A B-poset is characterized by the sequence $m \geqq l_1 \geqq l_2 \geqq \cdots \geqq l_n \geqq 0$, where $l_i = |\{b_j : a_i < b_j\}|$. Thus the number of B-posets with parameters m and n is $(n+m)!/(n!m!)$. A poset is a B-poset if and only if it has jump number less than or equal to 1.

As is standard, $G(\mathscr{P}) \times e$ denotes the Cartesian product of $G(\mathscr{P})$ and an edge; in other words, it is two copies of $G(\mathscr{P})$ together with a perfect matching joining corresponding vertices in the two copies. The linear extensions in one copy will be preceded with a plus (+) and in the other copy with a minus (−). The *canonical* linear extension of a B-poset is $\mathbf{c} = a_1a_2\cdots a_nb_1b_2\cdots b_m$. In the statement of the following lemma a single edge is regarded as having a Hamilton cycle. For any poset of width two, such as a B-

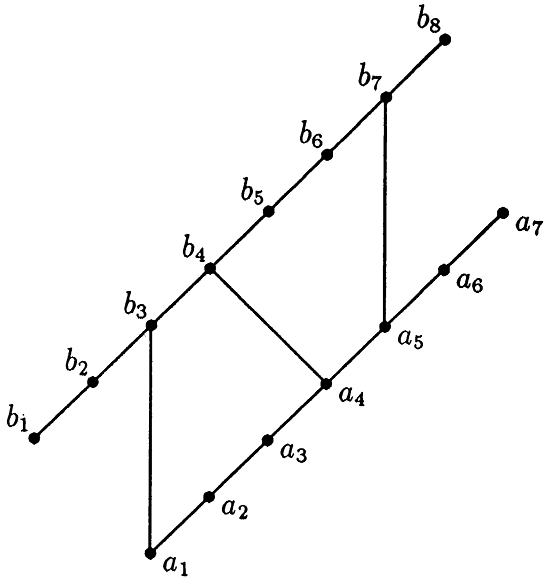


FIG. 1. *A B-poset.*

poset, $G(\mathcal{P}) = G'(\mathcal{P})$. We use G' in the statement of the following lemma to remind the reader that all transpositions are adjacent transpositions.

LEMMA 2. *For any B-poset $\mathcal{P}$, as defined above, there is a Hamilton cycle T in $G'(\mathcal{P}) \times e$. Furthermore, the cycle T contains the edge joining vertices $\pm\mathbf{c}$.*

Proof. Without loss of generality, we may assume that $\mathcal{P}$ has no maximum or minimum element. The proof then proceeds by induction on m and n. The base case is $m = 0$ or $n = 0$, where $G(\mathcal{P}) \times e$ consists of a single edge.

If $m, n > 0$ then we show how to construct a list $T(\mathbf{c})$ that includes every vertex of $G(\mathcal{P}) \times e$, and that starts at $+\mathbf{c}$ and ends at $-\mathbf{c}$. It will also prove convenient to have a notation $S(\mathbf{c})$ for the list T up to but not including $-\mathbf{c}$. Thus the final vertex of the list S is $-f$, where $f = a_1 \cdots a_{n-1} b_1 a_n b_2 \cdots b_m$. The list $S^R(\mathbf{c})$ is the reverse of the list $S(\mathbf{c})$.

In the algorithm of § 3 we use a list Y that includes all the linear extensions of T except $\pm\mathbf{c}$. The Y list starts and ends at the extensions $\pm f$. Clearly, lists equivalent to the S and T lists can be constructed from Y lists.

Our proof classifies the vertices of $G(\mathcal{P})$ according to the number of b's that are to the left of a_n. This leads to the notation

$$a_1 a_2 \cdots a_{n-1} b_1 b_2 \cdots b_j \overline{a_n b_{j+1} \cdots b_m}$$

for the subgraph H of $G(\mathcal{P}) \times e$ induced by those vertices that have suffix $a_n b_{j+1} \cdots b_m$. Note that this subgraph is the adjacent transposition graph of a B-poset of smaller parameters. The notation $T(\mathbf{c})$ is extended so that

$$T(a_1 a_2 \cdots a_{n-1} b_1 b_2 \cdots b_j \overline{a_n b_{j+1} \cdots b_m})$$

denotes a listing of the vertices of H that begins and ends at

$$\pm a_1 a_2 \cdots a_{n-1} b_1 b_2 \cdots b_j a_n b_{j+1} \cdots b_m.$$

The notation for $S(\mathbf{c})$ is similarly extended.

Here is how to construct the list $T(\mathbf{c})$ when m is even:

$$+a_1a_2\cdots a_{n-1}a_nb_1b_2b_3b_4\cdots b_{m-1}b_m$$

$$S(a_1a_2\cdots a_{n-1}b_1\overline{a_nb_2b_3b_4\cdots b_{m-1}b_m})$$

$$S^R(a_1a_2\cdots a_{n-1}b_1b_2\overline{a_nb_3b_4\cdots b_{m-1}b_m})$$

$$S(a_1a_2\cdots a_{n-1}b_1b_2b_3\overline{a_nb_4\cdots b_{m-1}b_m})$$

$$S^R(a_1a_2\cdots a_{n-1}b_1b_2b_3b_4\overline{a_n\cdots b_{m-1}b_m})$$

$$\vdots$$

$$S(a_1a_2\cdots a_{n-1}b_1b_2b_3b_4\cdots b_{m-1}\overline{a_nb_m})$$

$$S^R(a_1a_2\cdots a_{n-1}b_1b_2b_3b_4\cdots b_{m-1}b_m\overline{a_n})$$

$$-a_1a_2\cdots a_{n-1}b_1b_2b_3b_4\cdots b_{m-1}b_ma_n$$

$$-a_1a_2\cdots a_{n-1}b_1b_2b_3b_4\cdots b_{m-1}a_nb_m$$

$$\vdots$$

$$-a_1a_2\cdots a_{n-1}b_1b_2b_3b_4a_n\cdots b_{m-1}b_m$$

$$-a_1a_2\cdots a_{n-1}b_1b_2b_3a_nb_4\cdots b_{m-1}b_m$$

$$-a_1a_2\cdots a_{n-1}b_1b_2a_nb_3b_4\cdots b_{m-1}b_m$$

$$-a_1a_2\cdots a_{n-1}b_1a_nb_2b_3b_4\cdots b_{m-1}b_m$$

$$-a_1a_2\cdots a_{n-1}a_nb_1b_2b_3b_4\cdots b_{m-1}b_m.$$

Here is how to construct the list $T(\mathbf{c})$ when m is odd:

$$+a_1a_2\cdots a_{n-1}a_nb_1b_2b_3b_4\cdots b_{m-1}b_m$$

$$S(a_1a_2\cdots a_{n-1}b_1\overline{a_nb_2b_3b_4\cdots b_{m-1}b_m})$$

$$S^R(a_1a_2\cdots a_{n-1}b_1b_2\overline{a_nb_3b_4\cdots b_{m-1}b_m})$$

$$S(a_1a_2\cdots a_{n-1}b_1b_2b_3\overline{a_nb_4\cdots b_{m-1}b_m})$$

$$S^R(a_1a_2\cdots a_{n-1}b_1b_2b_3b_4\overline{a_n\cdots b_{m-1}b_m})$$

$$\vdots$$

$$S(a_1a_2\cdots a_{n-1}b_1b_2b_3b_4\cdots \overline{a_nb_{m-1}b_m}$$

$$S^R(a_1a_2\cdots a_{n-1}b_1b_2b_3b_4\cdots b_{m-1}\overline{a_nb_m})$$

$$T(a_1a_2\cdots a_{n-1}b_1b_2b_3b_4\cdots b_{m-1}b_m\overline{a_n})$$

$$-a_1a_2\cdots a_{n-1}b_1b_2b_3b_4\cdots b_{m-1}a_nb_m$$

$$\vdots$$

$$-a_1a_2\cdots a_{n-1}b_1b_2b_3b_4a_n\cdots b_{m-1}b_m$$

$$-a_1a_2\cdots a_{n-1}b_1b_2b_3a_nb_4\cdots b_{m-1}b_m$$

$$-a_1a_2\cdots a_{n-1}b_1b_2a_nb_3b_4\cdots b_{m-1}b_m$$

$$-a_1a_2\cdots a_{n-1}b_1a_nb_2b_3b_4\cdots b_{m-1}b_m$$

$$-a_1a_2\cdots a_{n-1}a_nb_1b_2b_3b_4\cdots b_{m-1}b_m. \qquad \square$$

LEMMA 3. *Let $\mathcal{P}$ be a poset and M a subset of its minimal elements for which $G(\mathcal{P} - M)$ has a Hamilton path (cycle). If no element of $\mathcal{P} - M$ has exactly one descendant in M then $G(\mathcal{P})$ also has a Hamilton path (cycle).*

Before proving this lemma let us make a few observations regarding the conditions of the lemma. First, the condition that no element of $\mathcal{P} - M$ has exactly one descendant in M is not equivalent to the condition that no element of $\mathcal{P} - M$ covers exactly one element of M. An element can cover exactly one element of M and yet have more than one descendant in M.

Secondly, note that if $|M| > 1$, then $G(\mathcal{P})$ must be balanced; a perfect matching in $G(\mathcal{P})$ can be defined by transposing the leftmost two elements of M in any linear extension. In fact, if $|M| > 1$ then the condition that no element of $\mathcal{P} - M$ has exactly one descendant in M is equivalent to the condition that for every linear extension of $\mathcal{P}$ another linear extension is obtained by transposing the leftmost two elements of M.

We now present the proof of Lemma 3.

Proof. Let $\mathcal{P}$ and M be as described in the statement of the lemma. Let M have n elements and $\mathcal{P} - M$ have m elements. If $n = 1$ the single element a of M is not comparable to any element of $S(\mathcal{P})$. Thus the Steinhaus–Johnson–Trotter idea of sweeping the a back and forth through the permutations along the Hamilton path (cycle) in $G(\mathcal{P} - M)$ produces a Hamilton path (cycle) in $G(\mathcal{P})$. From now on we assume that $n > 1$.

We construct the Hamilton path in three stages. Each stage produces a list of linear extensions, and each successive stage expands this list. In the first stage we list the vertices along the Hamilton path (cycle) in $G(\mathcal{P} - M)$. Denote this list by

$$q_1, q_2, \cdots, q_s, \quad \text{where } s = e(\mathcal{P} - M).$$

In the second stage we replace each q_j by a list of all linear extensions of $\mathcal{P}$ of the form pq_j, where p is a permutation of the elements of M. Let $p_1, p_2, \cdots, p_r$, where $r = n!$, be the list produced by Lemma 1. Then the stage two construction iteratively, for $i = 1, 2, \cdots, s$, produces the linear extensions

$$p_1q_i, p_2q_i, \cdots, p_rq_i \quad \text{if } i \text{ is odd;}$$

$$p_rq_i, \cdots, p_2q_i, p_1q_i \quad \text{if } i \text{ is even.}$$

Note that each successive linear extension differs by a transposition. At the end of stage two there are $n!e(\mathcal{P} - M)$ linear extensions in the list. The first linear extension is p_1q_1. If $e(\mathcal{P} - M)$ is odd, then the final linear extension is p_rq_s. If $e(\mathcal{P} - M)$ is even, then the final linear extension is p_1q_s. Thus if there is a Hamilton cycle in $G(\mathcal{P} - M)$ then the first and final linear extensions at the end of stage two also differ by a transposition.

In stage three we intermingle the elements of $M = \{a_1, a_2, \cdots, a_n\}$ and $\mathcal{P} - M = \{b_1, b_2, \cdots, b_m\}$. Let

$$\mathcal{Q} = \mathcal{P}[a_1a_2\cdots a_n; b_1b_2\cdots b_m]$$

denote the extension of $\mathcal{P}$ where $S(\mathcal{Q}) = S(\mathcal{P})$ and $R(\mathcal{Q})$ is the transitive closure of

$$R(\mathcal{P}) \cup \{(a_i, a_{i+1}) \mid i = 1, 2, \cdots, n-1\} \cup \{(b_i, b_{i+1}) \mid i = 1, 2, \cdots, m-1\}.$$

The poset $\mathcal{Q}$ is a B-poset. Now consider two successive linear extensions from the second stage that differ by a transposition of the first two elements. Call them $p = a_1a_2a_3\cdots a_nb_1b_2\cdots b_m$ and $p' = a_2a_1a_3\cdots a_nb_1b_2\cdots b_m$. Let

$$\mathcal{P}_1 = \mathcal{P}[a_1a_2a_3\cdots a_n;b_1b_2\cdots b_m] = \mathcal{P}[p];$$

$$\mathcal{P}_2 = \mathcal{P}[a_2a_1a_3\cdots a_n;b_1b_2\cdots b_m] = \mathcal{P}[p'].$$

Let us use $<_1$ ($<_2$) to refer to the relations in $\mathcal{P}_1$ ($\mathcal{P}_2$). We claim that $\mathcal{P}_1$ and $\mathcal{P}_2$ are isomorphic as B-posets, and the isomorphism is a transposition function. Only relations involving a_1 or a_2 and an element of $\mathcal{P} - M$ are of concern. If $a_1 <_1 b_j$ then $a_2 <_1 b_j$, because otherwise b_j would have only one descendant in M. If $a_2 <_1 b_j$ then clearly $a_1 <_1 b_j$. Similarly, $a_1 <_2 b_j$ if and only if $a_2 <_2 b_j$. Thus the two posets are isomorphic, by the function that transposes a_1 and a_2. Note that a_1 is not covered by an element of $\mathcal{P} - M$ in $\mathcal{P}_1$ and a_2 is not covered by an element of $\mathcal{P} - M$ in $\mathcal{P}_2$. Thus the a_1 and a_2 elements can be transposed in any linear extension of $\mathcal{P}_1$ to obtain a corresponding linear extension of $\mathcal{P}_2$, and vice versa. Hence the subgraph of the transposition graph $G(\mathcal{P})$ induced by the vertices of $E(\mathcal{P}_1) \cup E(\mathcal{P}_2)$ is the graph $G(\mathcal{P}_1) \times e$.

We now apply Lemma 2 to replace the two successive linear extensions p and p' by a list of all linear extensions of $\mathcal{P}_1$ or $\mathcal{P}_2$. This list starts at p and ends at p'. Stage three does this replacement for every successive pair of linear extensions that differ by a transposition of the leftmost two elements, taken from the list at the end of stage two. The proof is complete. □

The theorem now follows as an easy consequence of Lemma 3.

THEOREM. *The linear extensions of any ranked poset in which every nonmaximal element has at least two upper covers can be generated by transpositions.*

Proof. Starting with the minimal elements we proceed up the poset one rank at a time. The linear extensions of the minimal elements can be generated by transpositions by using the Steinhaus–Johnson–Trotter algorithm. Let $\mathcal{P}$ be the subposet consisting of those elements with rank r or less and let M be the set of all elements of rank r. Assume that all linear extensions of elements with rank less than r have been generated (e.g., there is a Hamilton path in $G(\mathcal{P} - M)$). By the dual version of Lemma 3 the linear extensions of $\mathcal{P}$ can be generated by transpositions. □

An alternate version of the theorem is given below.

THEOREM. *The linear extensions of any ranked poset in which every nonminimal element has at least two lower covers can be generated by transpositions.*

This is not just the dual of the previous theorem since the dual of a ranked poset is not necessarily ranked. However, it does follow from an inductive argument similar to the one used to prove the original theorem.

We now list some immediate corollaries of the theorem.

COROLLARY 1. *For n odd, alternating permutations can be generated by transpositions.*

This was first proven by Ruskey [12]. A permutation $a_1a_2\cdots a_n$ is *alternating* if $a_1 < a_2 > a_3 < a_4\cdots$; they are counted by the Euler numbers (see Stanley [16]). The inverses of alternating permutations arise as the linear extensions of the so called "fence" poset.

COROLLARY 2. *The linear extensions of a crown can be generated by transpositions.*

This is listed as an open problem in [12]. A *crown* is a poset whose Hasse diagram (as a graph) is an even length cycle and where every element is on one of two ranks.

COROLLARY 3. *The linear extensions of the Boolean algebra lattice, the lattice of subspaces of a finite-dimensional vector space over $GF(q)$, and the partition lattice can all be generated by transpositions.*

These orders are discussed in Aigner [1]. It is interesting that the transposition graph of the Boolean algebra lattice has a Hamilton cycle but the number of vertices it has is unknown (Sha and Kleitman [15]). One would suspect that this is true of the other examples mentioned in Corollary 3 as well.

If there is a Hamilton path (cycle) in $G'(\mathscr{P} - M)$, where M is the set of all maximal elements, then the only complication in obtaining a Hamilton path (cycle) in $G'(\mathscr{P})$ is in the proof of Lemma 1, where nonadjacent transpositions are possible. It was recently shown by Ruskey and Savage [13] that it is possible to generate permutations by adjacent transpositions and satisfy conditions (a) and (b) of Lemma 1. Since condition (c) is not used in the proof of the theorem, it holds for adjacent transpositions as well. However, there is no implementation of the permutation generation method of [13] that runs in constant average time.

3. The algorithms. In this section we discuss the efficient implementation of the algorithms implicit in the proofs of Lemmas 1 and 2, and the theorem of § 2. In each case the algorithm can be implemented to run in *constant average time*. This means that the total amount of computation divided by the number of permutations generated is bounded by a constant. The input is the poset $\mathscr{P}$. Only $O(|\mathscr{P}|)$ additional storage is necessary.

We first discuss the algorithm corresponding to Lemma 1. The algorithm is recursive and follows the proof. Three global arrays `a`, `ai`, and `dir` are maintained. Array `a` is the permutation, `ai` is its inverse, and `dir` is a directions array indicating whether an element is moving from right to left (−1) ; or from left to right (+1) . Initially `a` = `ai` = $1, 2, \cdots, n$ and all directions are −1 . A Pascal procedure to do the generation is shown in Fig. 3. The initial call is `Perm( 2 )`. The effect of the procedure call `Swap( k1, k2 )` is to transpose the elements `k1` and `k2` in the permutation as well as their inverses. Procedure call `SetSmaller( k, i1, i2 )` returns the indices of the leftmost elements in the permutation that are less than `k`. See Fig. 2.

Let c_n denote the number of calls to `SetSmaller`. We see that $c_2 = 0$ and for $n > 2$ that

$$c_n = c_{n-1} + \tfrac{1}{2}(n-1)!.$$

From this recurrence relation it follows that $c_n < (n-1)!$. Since each call to `SetSmaller` is $O(n)$, the total amount of computation used in the calls to `SetSmaller` is $O(n!)$. Other than the computation done by `SetSmaller`, the amount of computation is proportional to the number of recursive calls. The number of recursive calls is less than $n!$. Hence the complete algorithm is $O(n!)$.

```
procedure SetSmaller( k : integer; var i1, i2 : integer );
begin
   i1 := 1;
   while a[i1] >= k do i1 := i1 + 1;
   i2 := i1 + 1;
   while a[i2] >= k do i2 := i2 + 1;
end {of SetSmaller};
```

FIG. 2. *Pascal procedure* `SetSmaller`.

```
procedure Perm ( k : integer ) ;
var i : integer;
begin
   if k > n then begin
      PrintIt; Swap( a[1] , a[2] ) ; PrintIt;
   end else begin
      if dir[k] = +1 then begin
         Perm( k+1 ) ;
         SetSmaller( k, i1, i2 ) ; Swap( a[ i1] , a[ i2] ) ;
      end;
      Perm( k+1 ) ;
      for i := 1 to k-2 do begin
         Swap( k, a[ai[k]+dir[k]] ) ; Perm( k+1 ) ;
      end;
      if (dir[k] = -1) and (k > 2) then begin
         SetSmaller( k, i1, i2 ) ; Swap( a[ i1] , a[ i2] ) ;
         Perm( k+1 ) ;
      end;
      dir[k] := -dir[k];
   end;
end {of Perm};
```

FIG. 3. *Pascal procedure to generate all permutations.*

We now implement the algorithm of Lemma 2 to generate a Hamilton cycle in $G(\mathscr{P}) \times e$, where $\mathscr{P}$ is a B-poset with parameters n and m. The basic approach is the same as in Lemma 2 but some of the details vary. The procedure `Y ( n )` of Fig. 4 generates a list of all linear extensions of $G(\mathscr{P}) \times e$ except for the canonical extensions $\pm\mathbf{c}$; the list starts at $+f$ and ends at $-f$ (recall that f was defined in the proof of Lemma 2 to be $a_1 \cdots a_{n-1} b_1 a_n b_2 \cdots b_m$). In the proof of Lemma 2 we assumed that there was no maximum element in $G(\mathscr{P})$ and thus a_n was free to move to the rightmost position of the linear extension. However, our algorithm simply moves a_n to the right as far as possible. This is the reason that m is not a parameter of `Y`.

The Boolean function `Right ( n )` returns true only if element a_n can be transposed with the element to its right. We assume that `Right` takes time $O(1)$. This may require

```
procedure Y ( n : integer ) ;
var mr, j : integer;
begin
   mr := 0;
   while Right ( n ) do begin
      if odd ( mr ) then Move ( n, +1 )
      else
         if Right ( n-1 ) {#1} then begin
            Move ( n-1, +1 ) ; Y ( n-1 ) ; Move ( n, +1 ) ;
            Y ( n-1 ) ; Move ( n-1, -1 )
         end else Move ( n, +1 ) ;
      mr := mr + 1
   end {while};
   if odd( mr ) or not Right ( n-1 ) then Switch
   else begin Move ( n-1, +1 ) ; Y ( n-1 ) ; Move ( n-1, -1 ) end;
   for j := 1 to mr do Move ( n, -1)
end {of Y};
```

FIG. 4. *Pascal procedure to generate a Hamilton path in* $G'(\mathscr{P}) \times e$.

some preprocessing, depending on how $\mathcal{P}$ is specified. Procedure `Switch` changes the `(+)` prefix to `(−)`, and vice versa. Procedure `Move( n, +1 )` moves element a_n one position to the right, and `Move( n, −1 )` moves it one position left. The variable `mr` counts the number of times that a_n has moved to the right.

The calling sequence first determines whether a_n can be moved to the right. If not, then `Switch` is called. Otherwise, execute `Move( n, +1 )`, `Y( n )`, and `Move( n, −1 )`. Within the while loop, the call to `Right( n−1 ) {#1}` always returns the same value.

Procedure `Y` runs in constant average time because the running time is determined by the number of recursive calls to `Y`, and every call to `Y` is preceded or followed by a `Move`, and every `Move` creates a new linear extension.

We now need to put all the pieces together to implement the proof of the theorem. We first modify the terminating case of `Perm` so that instead of printing, swapping the first two elements, and printing, that it executes the calling sequence as described above. Furthermore, `Switch` is modified so that, instead of changing signs, it swaps a_1 and a_2. And finally, we must alternate the directions in which `Perm` produces its list of permutations.

```
while all of $L(\mathcal{P} - M)$ has not been generated do begin
    $b_1 b_2 \cdots b_m \leftarrow$ next element of $L(\mathcal{P} - M)$;
    $a_{n+1} \cdots a_{n+m} \leftarrow b_1 b_2 \cdots b_m$;
    Initialize for call to Perm;
    Perm( 2 );
end {while};
```

Acknowledgments. We wish to thank the referees for their helpful comments that led to improvements in the presentation.

REFERENCES

[1] M. AIGNER, *Combinatorial Theory*, Springer-Verlag, Berlin, New York, 1979.

[2] M. BUCK AND D. WIEDEMANN, *Gray codes with restricted density*, Discrete Math., 48 (1984), pp. 163–171.

[3] P. EADES, M. HICKEY, AND R. C. READ, *Some Hamilton paths and a minimal change algorithm*, JACM, 31 (1984), pp. 19–29.

[4] M. GARDNER, *Time Travel*, W. H. Freeman, New York, 1988.

[5] S. M. JOHNSON, *Generation of permutations by adjacent transpositions*, Math. Comp., 17 (1963), pp. 282–285.

[6] A. D. KALVIN AND Y. L. VAROL, *On the generation of all topological sortings*, J. Algorithms, 4 (1983), pp. 150–162.

[7] D. E. KNUTH, *Sorting and Searching*, Addison-Wesley, New York, 1973.

[8] ———, *Lexicographic permutations with restrictions*, Discrete Appl. Math., 1 (1979), pp. 117–125.

[9] D. H. LEHMER, *Permutation by adjacent interchanges*, Amer. Math. Monthly, 72 (1965), pp. 36–46.

[10] F. RUSKEY, *Adjacent interchange generation of combinations*, J. Algorithms, 9 (1988), pp. 162–180.

[11] ———, *Research problem* 90, Discrete Math., 70 (1988), pp. 111–112.

[12] ———, *Transposition generation of alternating permutations*, Order, 6 (1989), pp. 227–233.

[13] F. RUSKEY AND C. SAVAGE, *Hamilton cycles which extend transposition matchings in Cayley graphs of* S_n, Technical report, University of Victoria, DCS-132-IR, submitted.

[14] R. SEDGEWICK, *Permutation generation methods*, Computing Surveys, 9 (1977), pp. 137–164.

[15] J. SHA AND D. J. KLEITMAN, *The number of linear extensions of subset ordering*, Discrete Math., 63 (1987), pp. 271–278.

[16] R. P. STANLEY, *Enumerative Combinatorics*, Vol. I, Wadsworth, Monterey, 1986.

[17] H. STEINHAUS, *One Hundred Problems in Elementary Mathematics*, Basic, New York, 1964.

[18] H. F. TROTTER, *Algorithm* 115: *Perm*, Comm. ACM, 5 (1962), pp. 434–435.

SIAM J. DISC. MATH.
Vol. 4, No. 3, pp. 423–435, August 1991

011

ON THE POWER OF THRESHOLD CIRCUITS WITH SMALL WEIGHTS*

KAI-YEUNG SIU† AND JEHOSHUA BRUCK‡

Abstract. Linear threshold elements (LTEs) are the basic processing elements in artificial neural networks. An LTE computes a function that is a sign of a weighted sum of the input variables. The weights are arbitrary integers; actually, they can be very big integers—exponential in the number of input variables. However, in practice, it is very difficult to implement big weights. So the natural question that may be asked is whether there is an efficient way to simulate a network of LTEs with big weights by a network of LTEs with small weights. The following results are proved: (1) every LTE with big weights can be simulated by a depth-3, polynomial size network of LTEs with small weights; and (2) every depth-d, polynomial size network of LTEs with big weights can be simulated by a depth-$(2d+1)$, polynomial size network of LTEs with small weights. To prove these results, tools from harmonic analysis of Boolean functions are used. The technique is quite general; it provides insights to some other problems. For example, the best known results on the depth of a network of threshold elements that computes the *COMPARISON*, *ADDITION*, and *PRODUCT* of two n-bits numbers, and the *MAXIMUM* and the *SORTING* of n n-bit numbers are improved.

Key words. threshold circuits, linear threshold functions, neural networks, polynomial bounded weights, circuit depth

AMS(MOS) subject classifications. 68Q15, 68Q05, 68Rxx

1. Introduction.

Linear threshold functions. A linear threshold function $f(X)$ is a Boolean function such that

$$f(X) = \operatorname{sgn}(F(X)) = \begin{cases} 1 & \text{if } F(X) \geqq 0, \\ -1 & \text{if } F(X) < 0 \end{cases}$$

where

$$F(X) = \sum_{i=1}^{n} w_i \cdot x_i + w_0.$$

Throughout this paper, a *Boolean function* will be defined as $f : \{1, -1\}^n \to \{1, -1\}$; namely, 0 and 1 are represented by 1 and -1, respectively. Without loss of generality, we can assume $F(X) \neq 0$ for all $X \in \{1, -1\}^n$. The coefficients w_i are commonly referred to as the *weights* of the threshold function. We denote the class of all linear threshold functions by LT_1.

$\widehat{LT}_1$ functions. In this paper, we shall study a subclass of LT_1, which we denote by $\widehat{LT}_1$. Each function $f(X) = \operatorname{sgn}(\sum_{i=1}^{n} w_i \cdot x_i + w_0)$ in $\widehat{LT}_1$ is characterized by the property that the weights w_i are integers and bounded by a polynomial in n, i.e., $|w_i| \leqq n^c$ for some constant $c > 0$.

* Received by the editors February 21, 1990; accepted for publication (in revised form) September 26, 1990. This work was done while the first author was a research student associate at IBM Almaden Research Center and was supported in part by the Joint Services Program at Stanford University (U.S. Army, U.S. Navy, U.S. Air Force) under contract DAAL03-88-C-0011, and the Department of the Navy (NAVELEX) under contract N00039-84-C-0211, NASA Headquarters, Center for Aeronautics and Space Information Sciences under grant NAGW-419-S6.

† Information Systems Laboratory, Stanford University, Stanford, California, 94305.

‡ IBM Research Division, Almaden Research Center, K54/802, 650 Harry Road, San Jose, California, 95120-6099.

Threshold circuits. A *threshold circuit* [5], [11] is a Boolean network in which every gate computes an $\widehat{LT}_1$ function. The *size* of a threshold circuit is the number of $\widehat{LT}_1$ elements in the circuit. Let $\widehat{LT}_k$ denote the class of threshold circuits of *depth k* with the *size bounded by a polynomial* in the number of inputs. We define LT_k similarly except that we allow each gate in LT_k to compute an LT_1 function.

Although the definition of (LT_1) linear threshold function allows the weights to be real numbers, it is known [13] that we can replace each of the real weights by integers of $O(n \log n)$ bits, where n is the number of input Boolean variables. So in the rest of the paper, we shall assume without loss of generality that all weights are integers. However, this still allows the magnitudes of the weights to increase exponentially fast with the size of the inputs. It is natural to ask if this is necessary. In other words, is there a linear threshold function that must require exponentially large weights? Since there are $2^{\Omega(n^2)}$ linear threshold functions in n variables [9], [15], [16], there exists at least one which requires $\Omega(n^2)$ bits to specify the weights. By the pigeonhole principle, at least one weight of such a function must need $\Omega(n)$ bits, and thus is exponentially large in magnitude, i.e.,

$$\widehat{LT}_1 \subsetneqq LT_1.$$

The result above was proved in [10] using a different method by explicitly constructing an LT_1 function and proving that it is not in $\widehat{LT}_1$. In the following section, we shall show that the *COMPARISON* function (to be defined later) also requires exponentially large weights. We will refer to this function later on in the proof of our main results.

Some motivation. The motivation for this work comes from the area of neural networks, where a linear threshold element is the basic processing element. Many experimental results in this area have indicated that the magnitudes of the coefficients in the threshold elements grow very fast with the size of the inputs and therefore limit the practical use of the network. One natural question to ask is the following. How limited is the computational power of the network if we restrict ourselves to threshold elements with only "small" growth in the coefficients? We answer this question by showing that we can trade off an exponential growth with a polynomial growth in the magnitudes of coefficients by increasing the depth of the network by a factor of almost two and a polynomial growth in the size.

Main results. The fact that we can simulate a linear threshold function with exponentially large weights in a "constant" number of layers of elements with "small" weights follows from the results in [4] and [12]. Their results show that the sum of n n-bit numbers is computable in a constant number of layers of "counting" gates, which in turn can be simulated by a constant number of layers of threshold elements with "small" weights. However, it is not explicitly stated how many layers are needed in each step of their construction and direct application of their results would yield a constant such as 13. In this paper, we shall reduce the constant to 3 by giving a more "depth"-efficient algorithm and by using harmonic analysis of Boolean functions [2], [3], [6]. We then generalize this result to higher depth circuits and show how to simulate a threshold circuit of depth-d and exponentially large weights in a depth-$(2d+1)$ threshold circuit of "small" weights, i.e., $LT_d \subseteq \widehat{LT}_{2d+1}$.

As another application of harmonic analysis, we also show that the *COMPARISON* and *ADDITION* of two n-bit numbers is computable with only two layers of elements with "small" weights, while it was only known to be computable in three layers [5]. We also indicate how our "depth"-efficient algorithm can be applied to show that the product of two n-bit numbers can be computed in $\widehat{LT}_4$. In addition, we show that the *MAXIMUM* and *SORTING* of n n-bit numbers can be computed in $\widehat{LT}_3$ and $\widehat{LT}_4$, respectively.

The remainder of this paper is divided into four major sections. In § 2, we introduce the basic notions of harmonic analysis of Boolean functions and show that the *COMPARISON* function of two n-bit numbers is not an $\widehat{LT}_1$ function but is computable in $\widehat{LT}_2$. In § 3, we present a "depth"-efficient algorithm and use the results of § 2 to show that $LT_1 \subsetneq \widehat{LT}_3$ and $LT_d \subseteq \widehat{LT}_{2d+1}$. In § 4, we give some extensions and applications of our techniques. In the concluding remarks, we indicate some open problems and possible extensions of our results.

2. Harmonic analysis and the comparison function. In this section, we shall focus on the *COMPARISON* function of two n-bit numbers. This LT_1 function has the interesting property that it is not an $\widehat{LT}_1$ function, i.e., it separates the complexity class $\widehat{LT}_1$ from LT_1. The best-known result about this function is that it belongs to $\widehat{LT}_3$ [5]. Using tools from harmonic analysis [2], [3], we obtain the depth-optimal result by showing that *COMPARISON* $\in \widehat{LT}_2$. Later on in the proof of our main result we shall see how any arbitrary LT_1 function can be reduced to the *COMPARISON* function using one layer of $\widehat{LT}_1$ elements. First, we give a definition of the *COMPARISON* function.

DEFINITION. Let $X = (x_1, \cdots, x_n)$, $Y = (y_1, \cdots, y_n) \in \{1, -1\}^n$. We consider X and Y as two n-bit numbers representing $\sum_{i=1}^n x_i \cdot 2^i$ and $\sum_{i=1}^n y_i \cdot 2^i$, respectively.

The *COMPARISON* function is defined as

$$C(X, Y) = 1 \quad \text{iff } X \geqq Y.$$

In other words,

$$C(X, Y) = \operatorname{sgn}\left\{\sum_{i=1}^{n} 2^i(x_i - y_i) + 1\right\}.$$

LEMMA 1.

$$COMPARISON \notin \widehat{LT}_1.$$

Proof. We write the values of the function $C(X, Y)$ in the form of a 2^n by 2^n matrix in such a way that each row corresponds to the values of the function over the variables y_i's for a fixed value of X. For example, taking $n = 2$, we have

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & 1 \end{bmatrix}.$$

The rows are arranged from top to bottom in the following order: $X = (1, 1)$, $(1, -1)$, $(-1, 1)$, $(-1, -1)$. The columns are arranged from left to right in the same order for the Y's. Observe that every row in the matrix is distinct. This is true for general n. Suppose we realize $C(X, Y)$ by any other linear threshold function $\operatorname{sgn}(w_0 + \sum_{i=1}^n (w_i x_i + \tilde{w}_i y_i))$ with integer weights. The fact that there are 2^n distinct rows in the matrix implies that there are 2^n distinct values of $\sum_{i=1}^n w_i x_i$. This is only possible if some w_i is exponentially large. □

On the other hand, using harmonic analysis [3], we can show the following lemma.

LEMMA 2.

$$COMPARISON \in \widehat{LT}_2.$$

Before we proceed to the proof of Lemma 2, we need to introduce the tools from harmonic analysis.

Spectral representation of Boolean functions. Recently, harmonic analysis has been found to be a powerful tool in studying the *computational complexity* of Boolean functions [2], [3], [7]. The idea is that every Boolean function $f : \{1, -1\}^n \to \{1, -1\}$ can be represented as a polynomial over the field of rational numbers as follows:

$$f(X) = \sum_{\alpha \in \{0,1\}^n} a_\alpha X^\alpha \quad \text{where } X^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}.$$

Such representation is unique and the coefficients of the polynomial $\{a_\alpha \mid \alpha \in \{0, 1\}^n\}$ are called the *spectral coefficients* of f. The coefficients a_α's are computed as follows. Let P_{2^n} denote the vector of the 2^n values of $f(X)$, and let A_{2^n} denote the vector of the spectral coefficients a_α's. Then

$$A_{2^n} = \frac{1}{2^n} H_{2^n} P_{2^n}$$

where H_{2^n} denotes the *Sylvester-type Hadamard* matrix of order 2^n [8]. H_{2^n} can be defined recursively as follows:

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

$$H_{2^{n+1}} = \begin{bmatrix} H_{2^n} & H_{2^n} \\ H_{2^n} & -H_{2^n} \end{bmatrix}.$$

We shall define the L_1 *spectral norm* of f to be

$$\|f\| = \sum_{\alpha \in \{0,1\}^n} |a_\alpha|.$$

The proof of Lemma 2 is based on the spectral techniques developed in [3]. Using probabilistic arguments, it is proved in [3] that if a Boolean function has L_1 spectral norm which is polynomially bounded, then the function is computable in $\widehat{LT}_2$. We observe (together with Noga Alon) that the techniques in [3] can be generalized to show that any Boolean function with polynomially bounded L_1 spectral norm can even be closely *approximated by a sparse polynomial*. This observation is crucial when we extend our result from a single element to networks of elements with large weights. We shall give the proof of the stronger result here.

LEMMA 3. *Let $f(X) : \{1, -1\}^n \to \{1, -1\}$ such that $\|f\| \leqq n^c$ for some c. Then for any $k > 0$, there exists a sparse polynomial*

$$F(X) = \frac{1}{N} \sum_{\alpha \in S} w_\alpha X^\alpha \quad \textit{such that } |F(X) - f(X)| \leqq n^{-k},$$

where w_α and N are integers, $S \subset \{0, 1\}^n$, the size of S, w_α, and N are all bounded by a polynomial in n. Hence, $f(X) \in \widehat{LT}_2$.

Proof. We use the same probabilistic argument as in [3]. It suffices to show that there exists a *random sparse polynomial* of the above form $F(X)$ such that $|F(X) - f(X)| \leqq n^{-k}$ with probability > 0. Let $f(X) = \sum_{\alpha \in \{0,1\}^n} a_\alpha X^\alpha$ and $L_1 = \|f\| = \sum_{\alpha \in \{0,1\}^n} |a_\alpha|$. Let $\lceil L_1 \rceil$ be the least integer not smaller than L_1. Note that $\lceil L_1 \rceil < \|f\| + 1$ and is polynomially bounded. For $\alpha \in \{0, 1\}^n$, let $p_\alpha = |a_\alpha| / \lceil L_1 \rceil$. We define independent and identically distributed random variables $Z_i(X)$ such that

$$Z_i(X) = \begin{cases} \operatorname{sgn}(a_\alpha) X^\alpha & \text{with probability } p_\alpha, \alpha \in \{0, 1\}^n, \\ 0 & \text{with probability } 1 - \sum_{\alpha \in \{0,1\}^n} p_\alpha. \end{cases}$$

Note that the expected value of $Z_i(X)$ is

$$E[Z_i(X)] = \sum_{\alpha\in\{0,1\}^n} p_\alpha \operatorname{sgn}(a_\alpha)X^\alpha = \sum_{\alpha\in\{0,1\}^n} a_\alpha X^\alpha / \lceil L_1 \rceil = \frac{f(X)}{\lceil L_1 \rceil},$$

and the variance is

$$\operatorname{Var}[Z_i(X)] = E[Z_i^2(X)] - E^2[Z_i(X)] = \frac{\|f\|}{\lceil L_1 \rceil} - \frac{1}{\lceil L_1 \rceil^2} = \Omega(1).$$

Therefore, by the *central limit theorem* and for sufficiently large n, we have

$$\Pr\left\{\left|\sum_{i=1}^{N}\left(Z_i(X) - \frac{f(X)}{\lceil L_1 \rceil}\right)\right| > n^{1/2}N^{1/2}\right\} = \Pr\left\{\left|\frac{\lceil L_1 \rceil}{N}\left(\sum_{i=1}^{N} Z_i(X)\right) - f(X)\right| > n^{-k}\right\}$$
$$= O(e^{-n}) < 2^{-n}$$

for $N = O(\lceil L_1 \rceil^2 n^{2k+1})$ which is polynomially bounded. Now take $F(X) = \lceil L_1 \rceil / N \sum_{i=1}^{N} Z_i(X)$. By union bound, we obtain

$$\Pr\{\,|F(X) - f(X)| > n^{-k} \text{ for some } X \in \{1, -1\}^n\} < 1.$$

Thus,

$$\Pr\{\,|F(X) - f(X)| \leqq n^{-k} \text{ for all } X \in \{1, -1\}^n\} > 0.$$

We can rewrite $F(X) = 1/N \sum_{\alpha\in S} w_\alpha X^\alpha$, where w_α and the size of S are bounded by N. Observe that

$$f(X) = \operatorname{sgn}(F(X)) = \operatorname{sgn}\left(\sum_{\alpha\in S} w_\alpha X^\alpha\right).$$

It was noted in [3] that since each monomial X^α is a symmetric function and thus can be written as a sum of polynomially many $\widehat{LT}_1$ functions [2], [5], it follows that $f(X) \in \widehat{LT}_2$. □

Remark 1. The fact that any Boolean function $f(X)$ with polynomially bounded L_1 spectral norm can be expressed as a sign of a sparse polynomial and thus belongs to $\widehat{LT}_2$ is shown in [3]. Our result stated in Lemma 3 is stronger in the sense that each such function $f(X)$ can even be closely approximated by a sparse polynomial. We make crucial use of this lemma (and its consequence) when we prove Theorem 2 later.

Now we are ready to prove Lemma 2. It suffices to show that *COMPARISON* has a polynomially bounded L_1 spectral norm.

Proof of Lemma 2. We shall write a recursion for the spectral representation of the *COMPARISON* function $C(X, Y)$. If C_n is the polynomial corresponding to the function of $x_n, \cdots, x_1$ and $y_n, \cdots, y_1$, it is easy to see that

$$C_n = \frac{x_n - y_n}{2} + \frac{1 + x_n y_n}{2} C_{n-1}.$$

This shows that the L_1 spectral norm increases by one when we increase n by one. So if we denote the L_1 spectral norm by $\|\cdot\|$ as before, then with $\|C_1\| = 2$, we have by induction that $\|C_n\| = n + 1$ and thus is polynomially bounded. Hence *COMPARISON* $\in \widehat{LT}_2$. □

Remark 2. It follows trivially by taking $-Y$ that

$$\tilde{C}(X, Y) = \operatorname{sgn}\left\{\sum_{i=1}^{n} 2^i (x_i + y_i) + 1\right\} \in \widehat{LT}_2.$$

Remark 3. Suppose the L_1 spectral norm of an arbitrary LT_1 function is polynomially bounded. Then it would follow that $LT_1 \in \widehat{LT}_2$. However, even some simple $\widehat{LT}_1$ function such as sgn $(\sum_{i=1}^{n} x_i)$ has L_1 spectral norm that is not polynomially bounded. We shall show this fact later.

As a consequence of Lemma 3, we obtain the following.

LEMMA 4. *Let* $f(X) : \{1, -1\}^n \rightarrow \{1, -1\}$ *such that* $\|f\| \leqq n^c$ *for some* c. *Then for any* $k > 0$, *there exists a linear combination of* $\widehat{LT}_1$ *functions*

$$F(X) = \frac{1}{N}\sum_{i=1}^{s} w_j t_j(X) \quad \textit{such that } |F(X) - f(X)| \leqq n^{-k},$$

where $t_j(X) \in \widehat{LT}_1$, *and* s, w_j*'s, and* N *are integers bounded by a polynomial in* n.

Proof. The proof follows immediately from Lemma 3 by rewriting every monomial X^α in $F(X)$ as a sum of polynomially many $\widehat{LT}_1$ functions. □

3. Main results. Although most linear threshold functions require exponentially large weights, we can always simulate them by three layers of $\widehat{LT}_1$ elements.

THEOREM 1.

$$LT_1 \subsetneq \widehat{LT}_3.$$

Proof. The proof of Theorem 1 will be divided into three parts. First, we show how any LT_1 function $f(X) = \text{sgn}\,(F(X))$ can be reduced to the *COMPARISON* function in two layers. Second, by Lemma 2, the final result can be obtained using another two layers. Finally, we will see how the second and the third layers can be combined into one layer so that altogether only three layers are needed. The strict inclusion follows from the well-known fact that the XOR function is not computable in LT_1 whereas XOR $\in \widehat{LT}_2$.

(i) *Reduction to COMPARISON*. We shall show how we can reduce $F(X)$ to a sum of two numbers using two layers. Then it follows from Remark 2 of Lemma 2 that we can reduce sgn $(F(X))$ to the *COMPARISON* function in two layers.

First observe that by considering the binary representation of the weights w_i, we can introduce more variables and assign some constant values to the renamed variables in such a way that any linear threshold function can be assumed to be of the following generic form:

$$f(X) = \text{sgn}\,(F(X)),$$

where

$$F(X) = \sum_{i=1}^{n \log n} 2^i (x_{1_i} + x_{2_i} + \cdots + x_{n_i}).$$

We can further assume that n is *odd* by noting that $f(X)$ is not changed if we add 1 to the $F(X)$ and the fact that $2^{n \log n} - 2^{n \log n - 1} - \cdots - 2 - 1 = 1$. For convenience of presentation, we assume $n \log n$ to be an integer, where log denotes logarithm to base two. Let

$$(1) \qquad s_i = x_{1_i} + x_{2_i} + \cdots + x_{n_i} \quad \text{for } i = 1, \cdots, n.$$

Note that $|s_i| \leqq n$. Now partition the sum $F(X) = \sum_{i=1}^{n \log n} s_i 2^i$ into n consecutive blocks of $l = \lceil \log n \rceil$ summands each, so that

$$\begin{aligned} F(X) &= \sum_{i=1}^{l} s_i 2^i + \sum_{i=l+1}^{2l} s_i 2^i + \cdots + \sum_{i=n\log n - l + 1}^{n \log n} s_i 2^i \\ &= \sum_{j=0}^{n-1} \left(\sum_{i=jl+1}^{(j+1)l} s_i 2^i \right) = \sum_{j=0}^{n-1} \left(\sum_{k=1}^{l} s_{jl+k} 2^{k-1} \right) 2^{jl+1} \\ &= \sum_{j=0}^{n-1} \tilde{s}_j 2^{jl+1} \quad \text{where } \tilde{s}_j = \sum_{k=1}^{l} s_{jl+k} 2^{k-1}. \end{aligned}$$

Note that

$$(2) \qquad |\tilde{s}_j| \leqq \sum_{k=1}^{l} |s_{jl+k}| 2^{k-1} \leqq n(2^l - 1) < 2^{2\lceil \log n \rceil}.$$

Furthermore, note that every *odd* number z such that $|z| < 2^{2\lceil \log n \rceil}$ can be expressed in ± 1 *binary representation* with $2\lceil \log n \rceil$ bits:

$$(3) \qquad z = \sum_{i=0}^{2\lceil \log n \rceil - 1} z_i 2^i \quad \text{where } z_i = \pm 1.$$

Since n is assumed to be odd without loss of generality, it follows that $\tilde{s}_j$ is also odd and therefore can be represented as $2\lceil \log n \rceil$ bits of ± 1 as in the above expression (3). Now observe that because of (2), there is no overlapping in the ± 1 binary representation between $\tilde{s}_j 2^{jl+1}$ and $\tilde{s}_{j+2} 2^{(j+2)l+1} = 2^{2\lceil \log n \rceil}(\tilde{s}_{j+2} 2^{jl+1})$. Thus we can compute the ± 1 binary representation of each $\tilde{s}_j$ for j odd in parallel and concatenate the resulting bits together to obtain the ± 1 binary representation of

$$s_{\text{odd}} = \sum_{j \text{ odd}} \tilde{s}_j 2^{jl+1}.$$

We can obtain the ± 1 binary representation of

$$s_{\text{even}} = \sum_{j \text{ even}} \tilde{s}_j 2^{jl+1}$$

in a similar fashion. Obviously, $F(X)$ is the sum of s_{odd} and s_{even}. It remains to show how to compute the ± 1 binary representation of each $\tilde{s}_j$. Observe that each $\tilde{s}_j$ is a polynomially bounded linear combination of $\log n \times n$ input variables. Let b_k be the kth bit in the ± 1 binary representation of $\tilde{s}_j$. Then there exists a set of numbers $\{k_1, \cdots, k_l\}$ such that $b_k = 1$ if and only if $\tilde{s}_j \in \{k_1, \cdots, k_l\}$. Let

$$(4) \qquad y_{k_m} = \operatorname{sgn}\{2(\tilde{s}_j - k_m) + 1\}; \ \tilde{y}_{k_m} = \operatorname{sgn}\{2(k_m - \tilde{s}_j) + 1\} \quad \text{for } m = 1, \cdots, l.$$

Then

$$(5) \qquad b_k = \left\{ \sum_{m=1}^{l} (y_{k_m} + \tilde{y}_{k_m}) - 1 \right\}.$$

Since $\tilde{s}_j$ is polynomially bounded, there are only polynomially many different k_m's. The first layer of our circuit consists of $\widehat{LT}_1$ elements which compute the values y_{k_m}'s and $\tilde{y}_{k_m}$'s. The second layer takes as inputs y_{k_m}'s and $\tilde{y}_{k_m}$'s and outputs $\operatorname{sgn}\{\sum_{m=1}^{l}(y_{k_m} + \tilde{y}_{k_m}) - 1\}$. Hence the ± 1 binary representation of each $\tilde{s}_j$ can be computed in two layers.

Remark 4. It was known [5] that any *symmetric* function is computable in two layers. The construction (4) and (5) above is an immediate generalization of this result.

Remark 5. Similar techniques were applied in [4] and [12] to reduce the computing of a multiple sum to that of a sum of two numbers. Their construction first reduces the sum of n n-bit numbers to that of $\log n$ numbers by applying a "counting" gate to each s_i in (1) above, and then to a sum of $\log \log n$ numbers. The novelty in our ideas is in reducing the multiple sum to sum of two numbers in one step by computing each $\tilde{s}_j$ in (2) above using only two layers.

(ii) *Another two layers to compute COMPARISON*. Now since $F(X)$ can be reduced to a sum of two $O(n \log n)$-bit numbers in ± 1 binary representation, it follows from Remark 2 of Lemma 2 that we only need two more layers to compute the function.

(iii) *Combining the second and third layers*. So far we have used four layers of $\widehat{LT}_1$ elements to simulate an LT_1 element: two layers to reduce the computing of a general LT_1 function to that of a *COMPARISON*, then another two layers to compute the *COMPARISON*. Now we shall see how to combine the second and the third layer together so that $LT_1 \subset \widehat{LT}_3$.

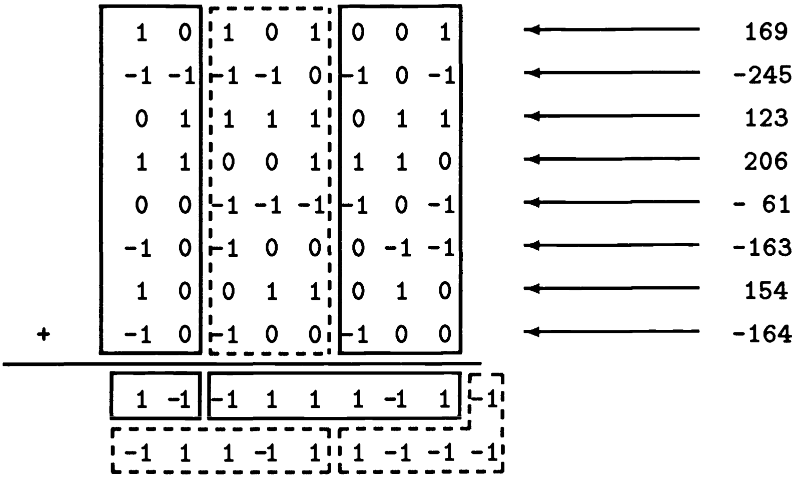
Note from (5) that the inputs b_k to the third layer are linear combinations of the outputs from the first layer. Thus, it is redundant to compute the sgn $(\cdots)$ after computing the linear combination. Therefore, we can directly feed the outputs from the first layer to the third layer without the use of the second layer. □

Example. A small numerical example will be helpful to illustrate the ideas. We take $n = 8$ so that $l = \lceil \log n \rceil = 3$, and let

$$f(X) = \operatorname{sgn}(F(X))$$
$$= \operatorname{sgn}(169x_1 - 245x_2 + 123x_3 + 206x_4 - 61x_5 - 163x_6 + 154x_7 - 164x_8).$$

In Fig. 1, the upper-right and upper-left solid blocks indicate $\tilde{s}_0$ and $\tilde{s}_2$, respectively, whereas the upper-middle dotted block indicates $\tilde{s}_1$. We set all x_i's to be 1. Each row in the summands indicates each "weight" in binary representation and is partitioned into three subblocks. For example, $169x_1 = 2^7 x_1 + (2^5 + 2^3)x_1 + x_1$ and $-245x_2 = (-2^7 - 2^6)x_2 + (-2^5 - 2^4)x_2 + (-2^2 - 1)x_2$. We compute the sum of all the first subblocks to obtain $\tilde{s}_0 = -5$. The other subblocks are summed to obtain $\tilde{s}_1 = -5$ and $\tilde{s}_2 = 1$ in parallel. The bottom-right dotted block, which has sum $= 0$, is added to make the resulting two numbers into the form of the arguments in the *COMPARISON* function.

The result stated in Theorem 1 implies that a depth-d threshold circuit with exponentially large weights can be simulated by a depth-$3d$ threshold circuit with polynomially



```
   1  0 | 1  0  1 | 0  0  1   <-------  169
  -1 -1 |-1 -1  0 |-1  0 -1   <------- -245
   0  1 | 1  1  1 | 0  1  1   <-------  123
   1  1 | 0  0  1 | 1  1  0   <-------  206
   0  0 |-1 -1 -1 |-1  0 -1   <------- - 61
  -1  0 |-1  0  0 | 0 -1 -1   <------- -163
   1  0 | 0  1  1 | 0  1  0   <-------  154
+ -1  0 |-1  0  0 |-1  0  0   <------- -164
----------------------------
   1 -1 |-1  1  1  1 -1  1 |-1
  -1  1  1 -1  1 | 1 -1 -1 -1
```

FIG. 1. *An example to illustrate the computing of the sign of a multiple sum.*

large weights. Using the result of Lemma 3, we can actually obtain a more depth-efficient simulation.

THEOREM 2.

$$LT_d \subseteq \widehat{LT}_{2d+1}.$$

Proof. We shall prove the result by induction on d. The case for $d = 1$ is stated in Theorem 1, where the last two layers of the simulating circuit $\in \widehat{LT}_3$ compute the *COMPARISON* function of the outputs from the first layer. For the inductive step, suppose every circuit $C_{d-1} \in LT_{d-1}$ can be simulated by a circuit $\hat{C}_{2d-1} \in \widehat{LT}_{2d-1}$, where the last two layers of $\hat{C}_{2d-1}$ compute the *COMPARISON* function of the outputs from the $(2d - 3)$th layer. Let $C_d \in LT_d$ and $f_1(X), \cdots, f_m(X)$ be the outputs from the $(d - 1)$th layer and $g(f_1(X), \cdots, f_m(X))$ be the output from the last layer of C. By Theorem 1, the function g can be computed in three layers of $\widehat{LT}_1$ elements with $f_i(X)$'s as inputs to the first layer; each gate in the first layer computes a function

$$h_j(f_1(X), \cdots, f_m(X)) = \operatorname{sgn}\left(\sum_{i=1}^{m} w_i f_i(X) + w_0\right),$$

where $\sum_{i=1}^{k} w_i f_i(X) + w_0 \neq 0$ for all $X \in \{1, -1\}^n$ and $\sum_{i=0}^{m} |w_i| < n^k$ for some $k > 0$. Since $f_i(X) \in LT_{d-1}$, by inductive hypothesis it can be simulated by some $C^i \in \widehat{LT}_{2d-1}$, where the last two layers compute the *COMPARISON* function of the outputs from some circuits in $\widehat{LT}_{2d-3}$. Now apply Lemma 4 to the last two layers of each C^i. It follows that we can express each $f_i(X) = 1/N \sum \tilde{w}_l t_{il}(X) + \varepsilon_i$, where N and $\sum |\tilde{w}_l|$ are polynomially bounded, $|\varepsilon_i| \leqq n^{-k}$, $t_{il}(X) \in \widehat{LT}_{2d-2}$. Let $F_i(X) = 1/N \sum \tilde{w}_l t_{il}(X)$; then

$$h_j(F_1(X), \cdots, F_m(X)) = \operatorname{sgn}\left(\sum_{i=1}^{m} w_i(f_i(X) - \varepsilon_i) + w_0\right) = \operatorname{sgn}\left(\sum_{i=1}^{m} w_i f_i(X) + w_0 - \varepsilon\right)$$

where

$$|\varepsilon| = \left|\sum_{i=1}^{m} w_i \varepsilon_i\right| \leqq n^{-k} \sum_{i=1}^{m} |w_i| < 1.$$

Since by assumption, $\sum_{i=1}^{k} w_i f_i(X) + w_0$ is a nonzero integer, therefore

$$h_j(F_1(X), \cdots, F_m(X)) = \operatorname{sgn}\left(\sum_{i=1}^{m} w_i f_i(X) + w_0\right) = h_j(f_1(X), \cdots, f_m(X)).$$

Thus we can rewrite

$$h_j(f_1(X), \cdots, f_m(X)) = \operatorname{sgn}\left(\sum_{i=1}^{m} w_i F_i(X) + w_0\right) = \operatorname{sgn}\left(\sum_{s} \hat{w}_s t_s(X) + \hat{w}_0\right)$$

where $\hat{w}_s$'s are integers, $\sum |\hat{w}_s|$ is polynomially bounded and $t_s(X) \in \widehat{LT}_{2d-2}$. Hence, $h_j(f_1(X), \cdots, f_m(X)) \in \widehat{LT}_{2d-1}$. Since g can be computed with three layers of $\widehat{LT}_1$ elements with $f_i(X)$ as inputs and each gate in the first layer computes $h_j(f_1(X), \cdots, f_m(X))$, it follows that $g(f_1(X), \cdots, f_m(X)) \in \widehat{LT}_{2d+1}$. Therefore by induction we have shown that any function computable in LT_d can be simulated in $\widehat{LT}_{2d+1}$. □

As another consequence of Lemma 3, we have the following corollary.

COROLLARY 1. *Let $f_1(X), \cdots, f_m(X)$ be functions with polynomially bounded L_1 spectral norms, and let $g(f_1(X), \cdots, f_m(X))$ be an $\widehat{LT}_1$ function with $f_i(X)$'s as in-*

puts, i.e.,

$$g(f_1(X), \cdots, f_m(X)) = \operatorname{sgn}\left(\sum_{i=1}^{m} w_i f_i(X) + w_0\right).$$

Then g can be expressed as a sign of a sparse polynomial in X with polynomially many number of monomial terms X^{α}'s and polynomially bounded integer coefficients. Hence $g \in \widehat{LT}_2$.

Proof. Using Lemma 3, we can replace each $f_i(X)$ by its sparse polynomial approximation $F_i(X)$ with sufficiently small error. By the same argument as in the proof of Theorem 2, this does not change the function g, i.e.,

$$g(f_1(X), \cdots, f_m(X)) = g(F_1(X), \cdots, F_m(X)) = \operatorname{sgn}\left(\sum_{i=1}^{m} \tilde{w}_i F_i(X) + w_0\right).$$

Thus g can be expressed as a sign of a sparse polynomial as claimed. Since each monomial X^{α} can be expressed as a sum of polynomially many $\widehat{LT}_1$ functions, it follows that $g \in \widehat{LT}_2$. □

4. Extensions and applications. In Remark 3, it was mentioned that not all LT_1 functions have polynomially bounded L_1 spectral norms. We shall prove this fact via the following theorem. (As in Lemma 3, by a sparse polynomial we mean a polynomial with only polynomially many monomial terms X^{α}'s.)

THEOREM 3. *The $\widehat{LT}_1$ function MAJORITY*

$$\operatorname{sgn}\left(\sum_{i=1}^{n} x_i\right)$$

cannot be approximated by a sparse polynomial with an error $o(n^{-1})$.

Proof. In [2] it is proved that there is a symmetric function that needs an exponential number of monomial terms X^{α}'s to be expressed as a sign of a polynomial. We shall prove that the existence of an approximation to the *MAJORITY* function (with an error $o(n^{-1})$) by a sparse polynomial will contradict the result in [2]. Note that any $\widehat{LT}_1$ function is a projection of the *MAJORITY* function with an increase of polynomially many variables. If the *MAJORITY* function can be approximated by a sparse polynomial (with error $o(n^{-1})$), then it is easy to see that every $\widehat{LT}_1$ function would have such an approximation. Let $g(X)$ be an arbitrary symmetric function. It was known that [5] $g(X)$ can be computed by a $\widehat{LT}_2$ circuit with the sum of the magnitudes of weights in the second layer bounded by $O(n)$. Let $f_1(X), \cdots, f_m(X)$ be the outputs from the first layer and $\tilde{g}(f_1(X), \cdots, f_m(X))$ be the output from the second layer. If each $f_i(X)$ can be approximated by a sparse polynomial $F_i(X)$ (with error $o(n^{-1})$), then as in the proof of Corollary 1, it is clear that $\tilde{g}(f_1(X), \cdots, f_m(X)) = \tilde{g}(F_1(X), \cdots, F_m(X))$. It follows that $g(X)$ can be expressed as a sign of sparse polynomial with polynomially many monomial terms X^{α}'s. Since $g(X)$ is an arbitrary symmetric function, this contradicts the result in [2]. □

It follows from Lemma 3 and the above theorem that the *MAJORITY* function cannot have a polynomially bounded spectral norm.

It was known [5] that *ADDITION* of two n-bit numbers is computable in $\widehat{LT}_3$, i.e., we can compute each bit of the sum of two n-bit numbers using three layers. It seems intuitive that *ADDITION* is harder than *COMPARISON* with respect to circuit depth. Surprisingly, *ADDITION* can actually be computed in two layers, i.e., *ADDITION* $\in \widehat{LT}_2$. Also note that *ADDITION* $\notin \widehat{LT}_1$ since the least significant bit of the sum is the *XOR* of the least significant bits of the two numbers.

THEOREM 4. *Let x, y be two n-bit numbers. Then*

$$ADDITION(x,y) \in \widehat{LT}_2.$$

Proof. We apply the spectral techniques in [3] again and show that the L_1 spectral norm of each bit of the sum is polynomially bounded.

Let $x_1 = x_{1_{n-1}} x_{1_{n-2}} \cdots x_{1_0}$, $x_2 = x_{2_{n-1}} x_{2_{n-2}} \cdots x_{2_0}$ be the two n-bit binary numbers whose sum is to be computed. The input vector to the function *ADDITION* is arranged as $(x_{1_{n-1}}, x_{2_{n-1}}, \cdots, x_{1_0}, x_{2_0})$. Let $s = s_n s_{n-1} \cdots s_0$ denote the resulting $(n+1)$-bit sum and c_k denote the kth carry bit, for $k = 1, \cdots, n$. Then $s_n = c_n$, $s_k = PARITY(x_{1_k}, x_{2_k}, c_k)$ for $1 \leqq k \leqq n-1$, $s_0 = XOR(x_{1_0}, x_{2_0})$. It follows that $\|s_k\| = \|x_{1_k} x_{2_k} c_k\| = \|c_k\|$. Hence it suffices to show that c_k has polynomially bounded L_1 spectral norm.

As in the proof of Lemma 2, we write a recursion for the spectral representation of the nth-carry bit c_n as a function of $(x_{1_{n-1}}, x_{2_{n-1}}, \cdots, x_{1_0}, x_{2_0})$. Then if $\tilde{C}_n$ is the polynomial corresponding to the nth-carry bit, we have

$$\tilde{C}_n = \frac{x_{1_{n-1}} + x_{2_{n-1}}}{2} + \frac{1 - x_{1_{n-1}} x_{2_{n-1}}}{2} \tilde{C}_{n-1}.$$

This shows that the L_1 spectral norm increases by 1 when we increase n by 1. The same calculation as in the case of *COMPARISON* gives $\|\tilde{C}_n\| = n+1$ for all $n \geqq 1$. It follows that the L_1 spectral norm of each bit of the resulting sum is polynomially bounded and thus $ADDITION \in \widehat{LT}_2$ [3]. □

Another application of our "depth"-efficient algorithm and Theorem 2 yields the following theorem.

THEOREM 5. *The product of two n-bit integers can be computed in* $\widehat{LT}_4$.

Sketch of proof. The product of two n-bit numbers can be reduced to a sum of n $2n$-bit numbers using one layer, and then to the sum of two numbers using another two layers. By Theorem 2, the final sum can be computed using two more layers. As in the proof of Theorem 1, we can combine the third and the fourth layers into one layer so that altogether only four layers are needed. □

Remark 6. In [5], it was shown that at least three layers are needed to compute the product of two n-bit numbers. We give an upper bound of four layers.

Given n n-bit numbers, we would like to compute the maximum or sort the numbers. We now show that these two problems can be computed in $\widehat{LT}_3$ and $\widehat{LT}_4$, respectively.

THEOREM 6. *The MAXIMUM of n n-bit numbers can be computed in* $\widehat{LT}_3$.

Proof. Let $z_i = z_{i_n} z_{i_{n-1}} \cdots z_{i_1}$, for $i = 1, \cdots, n$, denote the input binary numbers. Recall from our convention that 0 and 1 are encoded as 1 and -1, respectively. Define

$$c_{ij} = \begin{cases} -1 & \text{if } z_i \geqq z_j, \\ 1 & \text{otherwise.} \end{cases}$$

Then the kth bit of the maximum number is

$$\bigvee_{1 \leqq i \leqq n} \left(\bigwedge_{1 \leqq j \leqq n} c_{ij} \wedge z_{ik} \right)$$

where $\vee$ and $\wedge$, respectively, denote the OR and AND functions. Note that c_{ij} is essentially the *COMPARISON* function of z_i and z_j. By Lemma 4, we can closely approximate each c_{ij} by a linear combination of $\widehat{LT}_1$ functions. The same argument as in the proof of Theorem 2 shows that $\wedge_{1 \leqq j \leqq n} c_{ij} \wedge z_{ik}$ can be computed in $\widehat{LT}_2$, since the AND function is in $\widehat{LT}_1$. The last layer computes the OR function. □

It was shown in [14] that the *SORTING* of n n-bit numbers can be computed in $\widehat{LT}_5$. By using Corollary 1, we can improve the depth of the sorting circuit presented in [14]. In *SORTING*, we assume that the input is a list of n n-bit binary numbers and the output will be the same list sorted in nondecreasing order. A number which appears m times in the input list will be duplicated m times in the output list.

THEOREM 7. *The SORTING of n n-bit numbers can be computed in $\widehat{LT}_4$.*

Proof. As in Theorem 6, let $z_i = z_{i_n} z_{i_{n-1}} \cdots z_{i_1}$, for $i = 1, \cdots, n$, denote the input binary numbers. Define

$$c_{ij} = \begin{cases} -1 & \text{if } z_i > z_j \text{ or } (z_i = z_j \text{ and } i \geqq j), \\ 1 & \text{otherwise.} \end{cases}$$

Note that for each i, $p_i = \sum_{j=1}^{n} (1 - c_{ij})/2$ is the position of z_i in the sorted list. If we let

$$EQ_m(p_i) = 1 - (\operatorname{sgn}\{p_i - m\} + \operatorname{sgn}\{m - p_i\}) = \begin{cases} -1 & \text{if } p_i = m, \\ 1 & \text{otherwise,} \end{cases}$$

then the kth bit of the mth number in the sorted list is

$$\bigvee_{1 \leqq i \leqq n} (EQ_m(p_i) \wedge z_{ik})$$

where $\vee$ and $\wedge$, respectively, denote the OR and AND functions. Recall again from our convention that 0 and 1 are encoded as 1 and -1, respectively.

We can compute $\operatorname{sgn}\{p_i - m\}$ and $\operatorname{sgn}\{m - p_i\}$ in the first two layers by Corollary 1. The third layer is used to compute

$$(EQ_m(p_i) \wedge z_{ik}) = \operatorname{sgn}\{z_{ik} - (\operatorname{sgn}\{p_i - m\} + \operatorname{sgn}\{m - p_i\}) + 2\}$$

and the fourth layer is used to compute the OR of the outputs $(EQ_m(p_i) \wedge z_{ik})$, $i = 1, \cdots, n$. □

5. Conclusions and future directions.

In this paper, we have shown that any linear threshold (LT_1) function can be simulated by three layers of $\widehat{LT}_1$ elements. We then generalize this result by showing that every depth-d, polynomial size network of linear threshold elements with big weights can be simulated by a depth-$(2d + 1)$, polynomial size network of elements with small weights, i.e., $LT_d \subseteq \widehat{LT}_{2d+1}$. Using the spectral techniques, we also show that both the *COMPARISON* and *ADDITION* of two n-bit numbers are in $\widehat{LT}_2$, whereas they are not computable in $\widehat{LT}_1$. We note here that, recently, Alon and Bruck [1] found explicit constructions of $\widehat{LT}_2$ circuits that compute the *COMPARISON* and the *ADDITION* functions. We also indicate how the *PRODUCT* of two n-bit numbers can be computed in $\widehat{LT}_4$, and show that the *MAXIMUM* and the *SORTING* of n n-bit numbers can be computed in $\widehat{LT}_3$ and $\widehat{LT}_4$, respectively.

There are a few open problems related to the results in this paper:

1) The spectral approach gives an existence proof that functions with "small" L_1 spectral norms are computable in $\widehat{LT}_2$. It will be interesting to have a general method to find constructions for these functions, extending the results of [1].

2) The spectrum of a generic LT_1 function has an exponential L_1 spectral norm. Is it true that $LT_1 \not\subseteq \widehat{LT}_2$?

3) Is it possible to strengthen the result stated in Theorem 2? We conjecture that $LT_d \subseteq \widehat{LT}_{d+2}$.

Acknowledgments. We thank Noga Alon for many helpful discussions and the referees for their useful comments that greatly improved this paper. K.-Y. Siu thanks his advisor Prof. Thomas Kailath for guidance, constant encouragement, and financial support.

REFERENCES

[1] N. ALON AND J. BRUCK, private communication, 1990.

[2] J. BRUCK, *Harmonic analysis of polynomial threshold functions*, SIAM J. Discrete Math., 3 (1990), pp. 168–177.

[3] J. BRUCK AND R. SMOLENSKY, *Polynomial threshold functions, AC^0 functions and spectral norms*, in Proc. 31st Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Washington, DC, 1990.

[4] A. K. CHANDRA, L. STOCKMEYER, AND U. VISHKIN, *Constant depth reducibility*, SIAM J. Comput., 13 (1984), pp. 423–439.

[5] A. HAJNAL, W. MAASS, P. PUDLAK, M. SZEGEDY, AND G. TURAN, *Threshold circuits of bounded depth*, in Proc. 27th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Washington, DC, 1987, pp. 99–110.

[6] R. J. LECHNER, *Harmonic analysis of switching functions*, in Recent Development in Switching Theory, A. Mukhopadhyay, ed., Academic Press, New York, 1971.

[7] N. LINIAL, Y. MANSOUR, AND N. NISAN, *Constant depth circuits, Fourier transforms, and learnability*, in Proc. 30th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Washington, DC, 1989.

[8] F. J. MACWILLIAMS AND N. J. A. SLOANE, *The Theory of Error-Correcting Codes*, North-Holland, New York, 1973.

[9] S. MUROGA AND I. TODA, *Lower bound of the number of threshold functions*, IEEE Trans. Electronic Computers, EC 15, (1966), pp. 805–806.

[10] J. MYHILL AND W. H. KAUTZ, *On the size of weights required for linear-input switching functions*, IRE Trans. Electronic Computers, EC 10, (1961), pp. 288–290.

[11] I. PARBERRY AND G. SCHNITGER, *Parallel computation with threshold functions*, in Structure in Complexity Theory, 1986, pp. 272–290.

[12] N. PIPPENGER, *The complexity of computations by networks*, IBM J. Res. Develop., 31 (1987), pp. 235–243.

[13] P. RAGHAVAN, *Learning in threshold networks: a computation model and applications*, Tech. Report RC 13859, IBM Research, July 1988.

[14] K.-Y. SIU AND J. BRUCK, *Neural computation of arithmetic functions*, in Proc. IEEE, October (1990), pp. 1669–1675.

[15] D. R. SMITH, *Bounds on the number of threshold functions*, IEEE Trans. Electronic Computers, EC 15, (1966), pp. 368–369.

[16] S. YAJIMA AND T. IBARAKI, *A lower bound of the number of threshold functions*, IEEE Trans. Electronic Computers, EC 14, (1965), pp. 926–929.

SIAM J. DISC. MATH.
Vol. 4, No. 3, pp. 436–447, August 1991

ON THE GEOMETRY AND COMPUTATIONAL COMPLEXITY OF RADON PARTITIONS IN THE INTEGER LATTICE*

SHMUEL ONN†

Abstract. The following integer analogue of a Radon partition in affine space $\mathcal{R}^d$ is studied: A partition (S,T) of a set of integer points in $\mathcal{R}^d$ is an *integral Radon partition* if the convex hulls of S and T have an integer point in common. The Radon number $r(d)$ of an appropriate convexity space on the integer lattice $\mathcal{Z}^d$ is then the infimum over those natural numbers n such that any set of n points or more in $\mathcal{Z}^d$ has an integral Radon partition. An $\Omega(2^d)$ lower bound and an $O(d2^d)$ upper bound on $r(d)$ are given, $r(2) = 6$ is proved, and the existence of integral Radon partitions, in lattice polytopes having a 1-skeleton with a large stable set of vertices, is established. The computational complexity of deciding if a given set of points in $\mathcal{Z}^d$ has an integral Radon partition is discussed, and it is shown that if d is fixed, then this problem is in P, while if d is part of the input, it is NP-complete.

Key words. abstract convexity, convexity spaces, geometry of numbers, Radon number, Radon partition, lattice polytopes, integer programming, integer lattice

AMS(MOS) subject classifications. 52A01, 52A25, 52A35, 52A40, 52A43, 68C25, 90C10

1. Introduction.

We study the following integer analogue of a Radon partition in affine space $\mathcal{R}^d$: A partition (S,T) of a set of integer points in $\mathcal{R}^d$ is an *integral Radon partition* if the convex hulls of S and T have an integer point in common. The Radon number $r(d)$ of an appropriate convexity space on the integer lattice $\mathcal{Z}^d$, to be defined below, is then the infimum over those natural numbers n such that any set of n points or more in $\mathcal{Z}^d$ has an integral Radon partition.

As the study of Radon partitions has a natural setting in the context of convexity spaces, we recall a few basic definitions from that theory.

DEFINITION 1.1 (Convexity space). A convexity space is a pair (X, C), where X is a set, $C \subseteq 2^X$, and the following hold:

(1) $\emptyset \in C,\ X \in C$,

(2) For all $F \subseteq C\ \ [\cap F \in C]$.

The members of C are called convex, and the C-hull of a set $A \subseteq X$ is $C(A) = \cap\{B : [B \in C] \wedge [A \subseteq B]\}$. The classic example is $(\mathcal{R}^d, \text{conv})$, where $\text{conv} = \{A : A \subseteq \mathcal{R}^d$ and A is convex$\}$.

DEFINITION 1.2 (Radon partition). A partition (S,T) of a given subset of X, where (X, C) is a convexity space, is called a Radon partition, if the C-hulls of S and T intersect.

Finally, we give the following slightly extended definition.

DEFINITION 1.3 (Radon number). Given a convexity space (X, C) and $F \subseteq 2^X$, we define the F-Radon number $r(F)$ as the infimum over those natural numbers n such that any set $A \in F$ with n elements or more has a Radon partition,

$$\begin{aligned} r(F) \quad = \quad \inf\{n \in N : \quad & \forall A \in F\ [[|A| \geq n] \\ & \rightarrow [\exists S \exists T\ [S \cup T = A] \wedge [S \cap T = \emptyset] \wedge [C(S) \cap C(T) \neq \emptyset]]]\}. \end{aligned}$$

* Received by the editors January 3, 1990; accepted for publication (in revised form) July 25, 1990. This research was partially supported by National Science Foundation grant DMS8706133 through the Center for Applied Mathematics, Cornell University, Ithaca, New York, 14853.

† School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York, 14853.

When $F = 2^X$, this reduces to the usual definition of the Radon number, which is denoted simply by r. When the underlying set of the space is a Cartesian product $X = K^d$, we will use the notation $r(d, F)$, and for $F = 2^X$ we will use $r(d)$.

The classical Radon theorem, which follows easily from the affine structure of $\mathcal{R}^d$, states that the Radon number of the space $(\mathcal{R}^d, \text{conv})$ is r(d)=d+2.

The convexity space that is studied in this paper is the restriction of $(\mathcal{R}^d, \text{conv})$ to the integer lattice, and will be denoted by $(\mathcal{Z}^d, C_d)$. Thus, the C_d-hull of a set $A \subseteq \mathcal{Z}^d$ is $C_d(A) = \text{conv}(A) \cap \mathcal{Z}^d$, and the collection of convex sets is $C_d = \{C_d(A) : A \subseteq \mathcal{Z}^d\}$. It follows that the Radon partitions and the Radon number $r(d)$ of that space coincide with those defined in the beginning of this section.

There are several papers discussing convexity spaces and their Radon numbers and also Helly, Caratheodory, and exchange numbers (to be defined in §3), e.g., Danzer, Grünbaum, and Klee [5], Eckhoff [7], Hammer [11], Levi [14], Tverberg [18]. In particular, the Helly number of our space $(\mathcal{Z}^d, C_d)$ was studied before by Doignon [6]. Also, some recent investigations motivated by integer programming were concerned with Helly properties of the integer lattice (Bell [1], Scarf [15], Schrijver [16, p. 234]), and Caratheodory properties of it (Cook, Fonlupt, and Schrijver [4]).

In contrast with the linear growth rate (as a function of the dimension) of the Radon number of the reals given above, we show in §2 an $\Omega(2^d)$ lower bound on the Radon number $r(d)$ of the integer lattice. It is interesting to note that this lower bound is also in contrast with a linear upper bound on the Radon number of the product space $(\mathcal{Z}, C_1)^d$ (defined on the same ground set $\mathcal{Z}^d$, but having only the collection $\{A_1 \times A_2 \times \cdots \times A_d : A_i \subseteq \mathcal{Z},\ 1 \leq i \leq d\}$ of d-boxes as its family of convex sets), which follows from an upper bound on Radon numbers of product convexity spaces given in Eckhoff [7].

On the other hand, we show in §3 an $O(d2^d)$ upper bound on the Radon number of the integer lattice.

Throughout the paper, by a "polytope" we mean a real convex polytope, i.e., a set $P = \text{conv}(V) \subseteq \mathcal{R}^d$ such that V is finite. If $V \subseteq \mathcal{Z}^d$, then P is called a "lattice polytope." It is easy to see that, in the definition of the Radon number, it is enough to consider subsets $A \subseteq X$ such that $|A| < \infty$ and $A = \text{ext}(P)$, the set of vertices of P, where P is some lattice polytope. Thus, the question about $r(d)$ is, in fact, a question about the existence of Radon partitions of sets of vertices of lattice polytopes, and we ask the same question about subclasses of lattice polytopes. This is the subject of §4, in which we study some properties of polytopes having a 1-skeleton with a large stable set of vertices (in the graph theoretical sense), and establish the existence of a Radon partition for such polytopes. This result is also useful for the proof, given in §5, that $r(2) = 6$, and leads to a simpler and more direct proof, for simple lattice polytopes, of the upper bound.

Finally, in §6, we discuss the computational complexity of deciding if a given set of points in $\mathcal{Z}^d$ has a Radon partition, and show that if d is fixed, than this problem is in P, while if d is part of the input, it is NP-complete.

Before going on, we note that the Radon number of the integer lattice is invariant under affine transformations. More precisely, we have the following observation.

OBSERVATION 1.4. *Consider a set $S \subseteq \mathcal{Z}^d$ such that $\dim(\text{aff}(S)) = k \leq d$. If $|S| \geq r(k)$, then S has a Radon partition.*

The proof is based on the fact that, given such a set, there is an affine bijection from aff(S) to $\mathcal{R}^k$ that preserves integrality (this is a result, say, of Theorem 3 in Gruber and Lekkerkerker [9, p. 19] or Corollary 4.3b in Schrijver [16]).

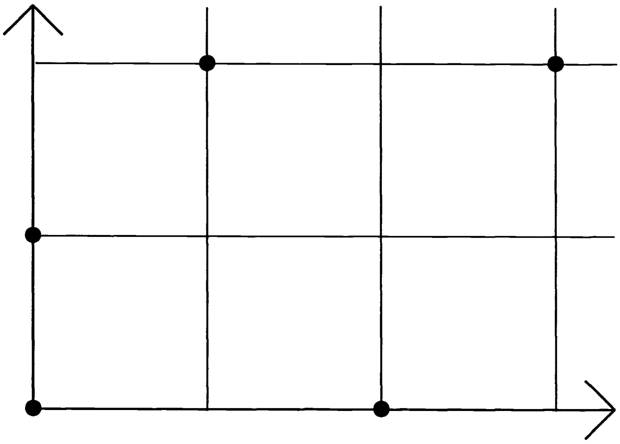


Fig. 1

2. A lower bound. We have the following proposition.

Proposition 2.1. *For $d \geq 1$ we have*

$$r(d+1) \geq 2r(d) - 1.$$

Proof. Assume indirectly that $r(d+1) \leq 2r(d) - 2$. Let $A \subseteq \mathcal{Z}^d$, $|A| = r(d) - 1$ be a set that admits no Radon partition. Let

$$X_i = \{(a, i) : a \in A\} \qquad (i = 0, 1).$$

Then $X_0 \cup X_1 \subseteq \mathcal{Z}^{d+1}$ and $|X_0 \cup X_1| = 2(r(d) - 1) \geq r(d+1)$, so there exists a Radon partition of $X_0 \cup X_1$ into two sets Y_0, Y_1. Let $p \in C_{d+1}(Y_0) \cap C_{d+1}(Y_1)$. Then $p \in \mathcal{Z}^{d+1}$, and $C_{d+1}(Y_0) \cap C_{d+1}(Y_1) \subseteq \{(x_1, \cdots, x_{d+1}) : 0 \leq x_{d+1} \leq 1\}$, so $p_{d+1} \in \{0, 1\}$. Assume, without loss of generality, that $p_{d+1} = 0$. Then, for $i = 0, 1$, let $Z_i = Y_i \cap \{(x_1, \cdots, x_{d+1}) : x_{d+1} = 0\}$. We then have that $p \in C_{d+1}(Z_0) \cap C_{d+1}(Z_1)$. Thus, the sets $\{a : (a, 0) \in Z_0\}$ and $\{a : (a, 0) \in Z_1\}$ form a Radon partition of A, a contradiction. □

The following set of points shows that $r(2) \geq 6$ (see Fig. 1):

$$\{(0,0), (2,0), (0,1), (1,2), (3,2)\}.$$

This, together with Proposition 2.1, yield the following corollary.

Corollary 2.2. *For $d \geq 2$, $r(d) \geq 2^d + 2^{d-2} + 1 = 5 \cdot 2^{d-2} + 1$.*

Proof. The proof is by induction on d. for $d = 2$ this is true since $r(2) \geq 6$. Now,

$$r(d+1) \geq 2r(d) - 1 \geq 2 \cdot (2^d + 2^{d-2} + 1) - 1 = 2^{d+1} + 2^{(d+1)-2} + 1,$$

as required. □

3. An upper bound. In this section, we must recall a few more invariants of a convexity space (X, C).

Definition 3.1 (Caratheodory number). The Caratheodory number c is the infimum over those natural numbers n such that the C-hull of any set $S \subseteq X$ is the union of the C-hulls of subsets $T \subseteq S$ *with* $|T| \leq n$,

$$c \;=\; \inf\{n \in N : \forall S \subseteq X \ [C(S) = \cup\{C(T) : [T \subseteq S] \wedge [|T| \leq n]\}]\}.$$

Definition 3.2 (Helly number). The Helly number h is the infimum over those natural numbers n such that any finite family $L \subseteq 2^X$ with the property that the intersection of the C-hulls of all its members, $\cap\{C(A) : A \in L\}$, is empty, has a subfamily $M \subseteq L$ of size $|M| \leq n$ with the same property,

$$\begin{aligned} h &= \inf\{n \in N : \ \forall L \subseteq 2^X \ [[[|L| < \infty] \wedge [\cap\{C(A) : A \in L\} = \emptyset]] \\ &\qquad \to [\exists M \subseteq L \ \ [|M| \leq n] \wedge [\cap\{C(A) : A \in M\} = \emptyset]]]\}. \end{aligned}$$

Definition 3.3 (Exchange number). The exchange number e is the infimum over those natural numbers n such that for every point $p \in X$ and finite set $A \subseteq X$, if $|A| \geq n$ then the C-hull of A is contained in the union of the C-hulls of sets obtained from A by replacing a point $a \in A$ by the point p,

$$\begin{aligned} e &= \inf\{n \in N : \ \forall p \in X \ \forall A \subseteq X \ [[n \leq |A| < \infty] \\ &\qquad \to [C(A) \subseteq \cup\{C(p \cup (A - a)) : a \in A\}]]\}. \end{aligned}$$

As with the Radon number, when the underlying set of the space is a Cartesian product $X = K^d$, we will use $c(d)$, $h(d)$, $e(d)$. It is well known that for $(\mathcal{R}^d, \text{conv})$, $c(d) = h(d) = e(d) = d + 1$.

The following proposition was proved in Doignon [6].

Proposition 3.4. *The Helly number of the convexity space $(\mathcal{Z}^d, C_d)$ is $h(d) = 2^d$.*

The following proposition follows immediately from the definitions.

Proposition 3.5. *Let (X, C) be a convexity space. Suppose this space has a Caratheodory number c and an exchange number e, and let $Y \subseteq X$. Then $(Y, \{A \cap Y : A \in C\})$ is a convexity space (the restriction to Y) with Caratheodory number $c' \leq c$, and exchange number $e' \leq e$.*

The following proposition follows immediately from the proof of Theorem 3 in Sierksma [17].

Proposition 3.6. *Let (X, C) be a convexity space with finite Caratheodory, Helly and exchange numbers c, h, e. Let a be a nonnegative integer such that $e \leq a$ and $c \leq a$. Then the Radon number satisfies*

$$r \leq (a - 1)(h - 1) + 3.$$

We now give an upper bound on the Radon number $r(d)$ of the convexity space $(\mathcal{Z}^d, C_d)$.

Corollary 3.7. $r(d) \leq d(2^d - 1) + 3$.

Proof. It is known that for $(\mathcal{R}^d, \text{conv})$ the Caratheodory number and the exchange number are both equal to $d + 1$. By Proposition 3.5, we then have, for $(\mathcal{Z}^d, C_d)$, that $c(d) \leq d + 1$ and $e(d) \leq d + 1$. By Proposition 3.4, we have $h(d) = 2^d$, and so by Proposition 3.6 we get

$$r(d) \leq ((d + 1) - 1)(h(d) - 1) + 3 = d(2^d - 1) + 3. \qquad \square$$

4. Stable sets in polytopes. In this section we study in detail some elementary properties of stable sets of vertices of polytopes. We establish the existence of a Radon partition of the set of extreme points (vertices) of a lattice polytope having a large stable set (in the sense defined below) of vertices. As a byproduct, this result yields a simpler and more direct proof of an upper bound for $r(d, SP)$, where $SP = \{A \subseteq \mathcal{Z}^d : A = \text{ext}(P), \ P \text{ a simple polytope}\}$, which is asymptotically the same as the one given in Corollary 3.7 (recall that a d-polytope is called "simple" if each one of its vertices is contained in exactly d 1-faces). It is also helpful in proving, in the next section, that $r(2) = 6$.

We start with some notation. By the "graph" of a polytope we mean the abstract graph having as vertices and edges the 0-faces and 1-faces of the polytope, respectively. For two points $x, y \in \mathcal{R}^d$ we denote by $[x, y] = \text{conv}(\{x, y\})$ the closed line segment joining them. For a vertex $s \in \text{ext}(P)$, we denote its set of neighbors by $N_P(s) = \{v \in \text{ext}(P) : [s, v] \text{ is a 1-face of } P\}$.

We will need the following (Brøndsted [3, Thm. 11.8]) proposition.

PROPOSITION 4.1. *Let P be a d-polytope, v a vertex of P, and H a hyperplane separating v from $N_P(v)$. Then H separates v from* $\text{ext}(P) \setminus \{v\}$.

By a (real) "Minimal Radon Partition" (MRP) we mean a pair (S, T) constituting a real Radon partition of $S \cup T \subseteq \mathcal{R}^d$ (i.e., $S \cap T = \emptyset$ and $\text{conv}(S) \cap \text{conv}(T) \neq \emptyset$), such that no proper subset of $S \cup T$ admits a real Radon partition. The following is easy to derive from the affine structure on $\mathcal{R}^d$, and, in fact, is just the circuit elimination axiom in the more general setting of oriented matroids (for a general reference to oriented matroids, see Björner et al. [2]).

PROPOSITION 4.2. *Let (S_1, T_1) and (S_2, T_2) be two* MRP*'s such that $S_1 \neq S_2$, $S_1 \neq T_2$, and $x \in S_1 \cap T_2$. Then there exists an* MRP *(S_3, T_3) such that $x \notin S_3 \cup T_3$, $S_3 \subseteq S_1 \cup S_2$, and $T_3 \subseteq T_1 \cup T_2$.*

The following is an elementary fact, formulated in terms of MRP's.

PROPOSITION 4.3. *Let P be a d-polytope, $V = \text{ext}(P)$, $u, v \in V$. Then $[u, v]$ is not an edge of P if and only if there exists an* MRP *$(\{u, v\}, T)$, $T \subseteq V \setminus \{u, v\}$.*

We will now establish some properties of polytopes, which will lead to the proof of the main result of this section.

LEMMA 4.4. *Let P be a d-polytope, $u \in V = \text{ext}(P)$, $P' = \text{conv}(V \setminus \{u\})$, and $v \in V \setminus (\{u\} \cup N_P(u))$. If $w \in V \setminus \{u, v\}$ is such that $[v, w]$ is not an edge of P, then it is also not an edge of P'.*

Proof. If $[v, w]$ is not an edge of P then, by Proposition 4.3, there exists $T_1 \subseteq V \setminus \{v, w\}$ such that $(\{v, w\}, T_1)$ is an MRP. If $u \notin T_1$, then $T_1 \subseteq \text{ext}(P')$ and so the claim follows by Proposition 4.3. Suppose then, that $u \in T_1$. Now, $[u, v]$ is not an edge of P, and so there exists $T_2 \subseteq V \setminus \{u, v\}$ such that $(\{u, v\}, T_2)$ is an MRP. Now, eliminating $u \in \{u, v\} \cap T_1$ using Proposition 4.2, we find an MRP (S, T) with $S \subseteq \{v, w\}$ and $T \subseteq T_1 \cup T_2 \setminus \{u\}$. However, v, w are vertices of P, so we must have $S = \{v, w\}$, and $T \subset \text{ext}(P')$, so again by Proposition 4.3 $[v, w]$ is not an edge of P'. □

LEMMA 4.5. *Let P be a d-polytope, $u \in V = \text{ext}(P)$, $P' = \text{conv}(V \setminus \{u\})$, $v \in P \setminus (\{u\} \cup P')$, and $P'' = \text{conv}(V \cup \{v\} \setminus \{u\})$. If $w \in V \setminus \{u\}$ is such that $[u, w]$ is not an edge of P, then $[v, w]$ is not an edge of P''.*

Proof. First, it is clear that $\text{ext}(P'') = V \cup \{v\} \setminus \{u\}$. Now, if $[u, w]$ is not an edge of P, then there exists $T_1 \subseteq V \setminus \{u, w\}$ such that $(\{u, w\}, T_1)$ is an MRP. Also, $v \in P$ implies that there exists $T_2 \subseteq V$ such that $(\{v\}, T_2)$ is an MRP, and since $v \notin P'$, it must be that $u \in T_2$. Eliminating u via Proposition 4.2, we obtain an MRP (S, T) with $S \subseteq \{v, w\}$ and $T \subseteq T_1 \cup T_2 \setminus \{u\}$. Now, w is a vertex of P, $T \subseteq \text{ext}(P')$, and $v \notin P'$, so it must be that $S = \{v, w\}$, and so by Proposition 4.3, $[v, w]$ is not an edge of P''. □

LEMMA 4.6 (Stable set exchange). *Let P be a d-polytope, $V = \text{ext}(P)$, and $S \subset V$ a stable subset of vertices (i.e., no two vertices in S lie on a common 1-face). Let $x \in \text{conv}(S) \setminus (S \cup \text{conv}(V \setminus S))$. Then there exists a vertex $s \in S$ such that in the polytope $P' = \text{conv}(V \cup \{x\} \setminus \{s\})$, we have $\text{ext}(P') = V \cup \{x\} \setminus \{s\}$ and $S' = S \cup \{x\} \setminus \{s\}$*

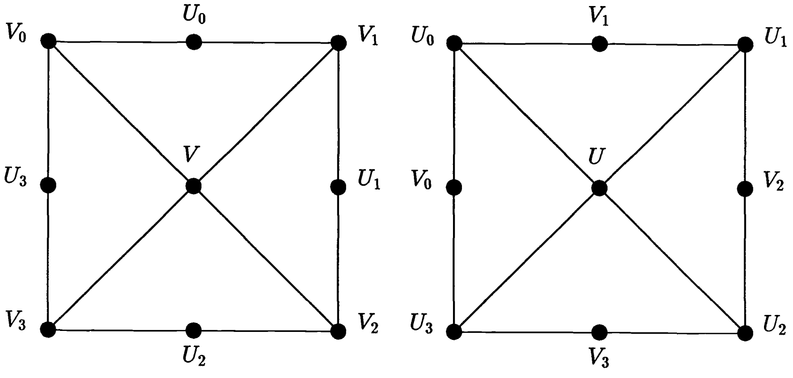


FIG. 2

is a stable set.

Proof. Let H be a hyperplane separating x from $V \setminus S$. Since $x \in \text{conv}(S)$, there is a vertex $s \in S$, which is also separated from $V \setminus S$ by H. S is stable, however, so $N_P(s) \subseteq V \setminus S$, and so H separates s from $N_P(s)$. Then, by Proposition 4.1, it separates s, and hence x, from $V \setminus \{s\}$. Thus $\text{ext}(P') = V \cup \{x\} \setminus \{s\}$.

Now, if $t \in S \setminus \{s\}$, then $[s,t]$ is not an edge of P, and so by Lemma 4.5, $[x,t]$ is not an edge of P'. If $u \in S \setminus \{s,t\}$, then $[t,u]$ is not an edge of P; so by Lemma 4.4, $[t,u]$ is not an edge of $\text{conv}(V \setminus \{s\}) \subseteq P'$, and so is not an edge of P' either. Therefore, S' is a stable set in P', as claimed. □

We can now prove the following theorem.

THEOREM 4.7. *Let P be a d-lattice polytope such that $S \subset V = \text{ext}(P)$ is a stable set of vertices and $|S| = 2^d + 1$. Then $(S, V \setminus S)$ is a Radon partition of V, i.e., $C_d(S) \cap C_d(V \setminus S) \neq \emptyset$.*

Proof. S is a set of integer points in $\mathcal{Z}^d$ of size $2^d + 1$, so there are two points $y, z \in S$ having the same parity on all coordinates. The point $x = (y+z)/2$ is an integer point in $\text{conv}(S) \setminus S$. If $x \in \text{conv}(V \setminus S)$, then we are done. If not, then let the point $s \in S$, the polytope P', and the set S' be as defined in Lemma 4.6. Then, by the lemma, P' and S' satisfy the hypothesis of the current theorem and we can repeat the above argument. Now $|\text{conv}(S') \cap \mathcal{Z}^d| < |\text{conv}(S) \cap \mathcal{Z}^d|$ because $s \in \text{conv}(S) \setminus \text{conv}(S')$, and $|\text{conv}(S) \cap \mathcal{Z}^d| < \infty$, so after finitely many applications of the above argument, we obtain a polytope P'' and a set S'' such that $(S'', V \setminus S)$ is a partition of $\text{ext}(P'')$ such that there exists an integer point $x'' \in \text{conv}(S'') \cap \text{conv}(V \setminus S)$. But, by the construction, $\text{conv}(S'') \subseteq \text{conv}(S)$, so in fact we have $x'' \in C_d(S) \cap C_d(V \setminus S)$. □

Example 4.8. Consider, for $k \geq 3$, the graph $G(k) = (V(k), E(k))$ which is the union of two homeomorphs of the k-wheel (see Fig. 2 for $k = 4$), defined as follows (we call it the "Bicycle wheel with $2k$ alternating spokes"):

$$\begin{aligned} V(k) &= \{v, v_0, \cdots, v_{k-1}, u, u_0, \cdots, u_{k-1}\}, \\ E(k) &= \{\{v_i, u_i\} : 0 \leq i \leq k-1\} \cup \{\{u_i, v_{i+1\ (mod\ k)}\} : 0 \leq i \leq k-1\} \\ &\cup\ \{\{v, v_i\} : 0 \leq i \leq k-1\} \cup \{\{u, u_i\} : 0 \leq i \leq k-1\}. \end{aligned}$$

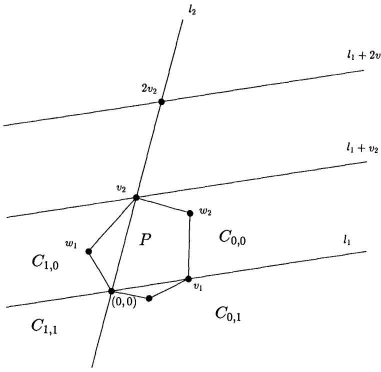


FIG. 3

The graph $G(k)$ is planar and 3-connected, so by Steinitz's theorem (see, e.g., Grünbaum [10, §13.2]) there exists a (convex) lattice 3-polytope $P(k)$ having $G(k)$ as its graph. Note, for example, that the 3-cube C^3 has $G(3)$ as its graph. Now, for $k \geq 9$ and $d \geq 3$, each one of the polytopes $P(k) \times C^{d-3} \subseteq \mathcal{R}^d$ satisfies the hypothesis of Theorem 4.7, and so admits a Radon partition. Note that for $k = 9$ or 10 we cannot deduce this from Corollary 3.7.

As a corollary to Theorem 4.7 we get again the upper bound, for simple polytopes, and so we have the following corollary.

COROLLARY 4.9. $2^d + 2^{d-2} + 1 \leq r(d, SP) \leq d2^d + 2$.

Proof. The lower bound is the same as the one given in Corollary 2.2, as the proof in §2 remains valid for simple polytopes. For the upper bound, recall that the graph of a simple polytope is d-regular, so if the polytope has at least $d2^d + 2$ vertices, it has a stable set of size at least $\lceil ((d2^d + 2) - 1)/d \rceil = 2^d + 1$. Then, by Theorem 4.7, the set $\mathrm{ext}(P)$ has a Radon partition. □

5. The two dimensional case. In §2 we have shown that $r(2) \geq 6$. We now prove the converse. Recall our notation that, for $S \subseteq \mathcal{Z}^2$, $C_2(S) = \mathrm{conv}(S) \cap \mathcal{Z}^2$. We will use the following simple fact (Gruber and Lekkerkerker [9, Thm. 4, p. 20]).

PROPOSITION 5.1. *Let v_1, v_2 be two linearly independent points in $\mathcal{Z}^2$. If*

$$C_2(\{(0,0), v_1, v_2\}) = \{(0,0), v_1, v_2\},$$

then the set $\{v_1, v_2\}$ is a lattice basis of $\mathcal{Z}^2$.

LEMMA 5.2. *Let $V \subseteq \mathcal{Z}^2$ be such that $|V| = 6$ and $P = \mathrm{conv}(V)$ is a convex hexagon. Let $S \subseteq V$ be a stable subset with $|S| = 3$. Then $C_2(S) \setminus S \neq \emptyset$.*

Proof. Without loss of generality we may assume that $S = \{(0,0), v_1, v_2\}$. Consider the two lines $l_i = lin(v_i)$ $(i = 1, 2)$. Denote the four connected components of $\mathcal{R}^2 \setminus (l_1 \cup l_2)$ by $c_{i,j}$ $(i, j = 0, 1)$ (see Fig. 3).

There must be two points in $V \setminus S$ lying on adjacent components, say $w_1 \in c_{1,0}$, $w_2 \in c_{0,0}$.

Now, suppose indirectly that $C_2(S) = S$. Then, by Proposition 5.1, $\{v_1, v_2\}$ is a basis of the lattice $\mathcal{Z}^2$. Now for $i = 1, 2$, $w_i \in \mathcal{Z}^2$ so lies on the line $l_1 + k_i v_2$ for some integer $k_i \geq 1$. But then we have $v_2 \in \mathrm{conv}(\{(0,0), w_1, w_2\})$, a contradiction. □

THEOREM 5.3. $r(2) = 6$.

Proof. Let $U \subseteq \mathcal{Z}^2$ be such that $|U| \geq 6$. We may assume that there is a subset $V \subseteq U$ such that $|V| = 6$ and $P = \mathrm{conv}(V)$ is a convex hexagon. It is enough to show that V admits a Radon partition. Let $S \subseteq V$ be a stable set with $|S| = 3$. By Lemma 5.2, $C_2(S) \setminus S \neq \emptyset$. Then either $C_2(S) \cap C_2(V \setminus S) \neq \emptyset$, in which case we are done, or there exists a point $x \in C_2(S) \setminus (S \cup \mathrm{conv}(V \setminus S))$. In that case, we can do a stable set exchange and obtain a new set S' and polytope P' as defined in Lemma 4.6, and in particular $C_2(S') \subset C_2(S)$.

Repeating the same argument as above, we eventually get, as in the proof of Theorem 4.7, a set S'' such that $|S''| = 3$, $P'' = \mathrm{conv}((V \setminus S) \cup S'')$ is a convex hexagon, S'' is stable in P'', and $C_2(S'') \cap C_2(V \setminus S) \neq \emptyset$. But then we are done, since $\mathrm{conv}(S'') \subseteq \mathrm{conv}(S)$ and so $(S, V \setminus S)$ is a Radon partition of V. □

6. Computational complexity aspects.

Consider the following decision problems.

RADON(d):
Instance: $n \in N$ and $A \subseteq \mathcal{Z}^d$ such that $|A| = n$.
Question: Does A admit an (integral) Radon partition?

RADON:
Instance: $d, n \in N$ and $A \subseteq \mathcal{Z}^d$ such that $|A| = n$.
Question: Does A admit an (integral) Radon partition?

Note that in RADON(d), the dimension d is fixed and is not part of the input. We show that RADON(d) is decidable in polynomial time while, in contrast, RADON is NP-complete.

PROPOSITION 6.1. *RADON(d) is decidable in polynomial time.*

Proof. Given an input to RADON(d), if $n \geq r(d)$ then there exists a Radon partition. Otherwise, consider all possible pairs (S, T) of disjoint nonempty subsets of the input set A such that $|S|, |T| \leq d + 1$ and the points in S (respectively T) are affinely independent in $\mathrm{aff}(S)$ (respectively $\mathrm{aff}(T)$). Since the Caratheodory number of the reals $c(d) \leq d + 1$, there will be such a pair with $C_d(S) \cap C_d(T) \neq \emptyset$ if and only if the input set A has a Radon partition. The number of such pairs is bounded by a polynomial function of n, and in fact, $n < r(d) < \infty$ (the second inequality from Corollary 3.7), so it is bounded by a finite constant which depends on d only. For each pair (S, T) with $|S| \leq d+1, |T| \leq d+1$, do the following. Find a system of linear equalities defining $\mathrm{aff}(S)$, and check affine independence, i.e., $|S| = 1 + \dim(\mathrm{aff}(S))$ (otherwise, move on to the next pair).

Next, add inequalities to this system to get a description of $\mathrm{conv}(S)$. This is easy, as $\mathrm{conv}(S)$ is a simplex in $\mathrm{aff}(S)$, so any $(|S| - 1)$-subset of S spans a facet of $\mathrm{conv}(S)$ in $\mathrm{aff}(S)$.

It is easily verified that the bit size of such a description is bounded above by a polynomial function in the bit size of the input, and so this step could be done in polynomial time.

Similarly, find a linear inequalities and equalities description of $\mathrm{conv}(T)$.

Now, let L be the union of the collections of linear equalities and inequalities describing $\mathrm{conv}(S)$ and $\mathrm{conv}(T)$. Apply to it an algorithm that checks if its solution set contains an integer point, e.g., the integer programming algorithm given in Lenstra [13], which runs in polynomial time when the dimension d is not a part of the input. Since the solution set of L is $\mathrm{conv}(S) \cap \mathrm{conv}(T)$, this algorithm checks if $C_d(S) \cap C_d(T) \neq \emptyset$, which is what we need. □

Next, we show that, in contrast, RADON is NP-complete. The following problem is known to be NP-complete (see Garey and Johnson [8, p. 223]).

PARTITION:
Instance: $n \in N$ and a set of nonnegative integers $B = \{a_1, \cdots, a_{2n}\}$.
Question: Is there a balanced partition, namely a partition (I, J) of $[2n] = \{1, \cdots, 2n\}$ such that $|I| = |J| = n$ and $\sum_{i \in I} a_i = \sum_{j \in J} a_j$?

THEOREM 6.2. RADON *is* NP-*complete.*

Proof. To show that RADON is in NP, suppose the instance in question has a Radon partition (S, T). Then there exists $x \in C_d(S) \cap C_d(T)$, and so the triple (x, S, T) is a witness of size bounded by a polynomial in the size of the input, to the existence of a Radon partition, which can be checked in polynomial time.

We now show that PARTITION is polynomial time reducible to RADON. Given an input (n, B) to PARTITION, let $d = 2\binom{2n}{2}+1, k = n+n^2$. We construct a set A of $2n+2\binom{2n}{2}$ distinct points in $\mathcal{Z}^d$ such that A has a Radon partition if and only if there exists a balanced partition for (n, B). For convenience, we index the first coordinate of a point by 0, and for each pair i, j such that $1 \leq i < j \leq 2n$, a point will have two coordinates, one indexed by $1, i, j$ and the other by $2, i, j$, so a point will be written as

$$x = (x_0, x_{1,1,2}, x_{2,1,2}, x_{1,1,3}, x_{2,1,3}, \cdots, x_{1,2n-1,2n}, x_{2,2n-1,2n}).$$

Now, for each $i \in [2n]$, we will have one point x^i, and for each pair i, j such that $1 \leq i < j \leq 2n$ we will have two points $y^{i,j}, z^{i,j}$, so

$$A = \{x^1, \cdots, x^{2n}, y^{1,2}, z^{1,2}, \cdots, y^{2n-1,2n}, z^{2n-1,2n}\},$$

where the points are defined as follows:

for all $i \in [2n]$ let $x_0^i = ka_i$.

for all $i, j \in [2n]$ such that $i < j$, let

$$x_{1,i,j}^i = y_{1,i,j}^{i,j} = -1, \quad x_{1,i,j}^j = z_{1,i,j}^{i,j} = 1, \quad y_{2,i,j}^{i,j} = z_{2,i,j}^{i,j} = k.$$

All other entries of all points will be set to zero.

Clearly the set A defined in this way can be constructed from B in polynomial time.

Now, suppose first that (I, J) is a balanced partition for (n, B). Define $S, T \subseteq A$ as follows:

$$\begin{aligned} S &= \{x_i : i \in I\} \cup \{z_{i,j} : 1 \leq i < j \leq 2n, i \in I, j \in J\} \\ &\quad \cup \{y_{i,j} : 1 \leq i < j \leq 2n, i \in J, j \in I\}, \\ T &= \{x_i : i \in J\} \cup \{y_{i,j} : 1 \leq i < j \leq 2n, i \in I, j \in J\} \\ &\quad \cup \{z_{i,j} : 1 \leq i < j \leq 2n, i \in J, j \in I\}. \end{aligned}$$

It is easy to see that $S \cap T = \emptyset$ and $|S| = |T| = k$, and that $\sum_{s\in S} \frac{1}{k}s = \sum_{t\in T} \frac{1}{k}t \in \mathcal{Z}^d$, so $C_d(S) \cap C_d(T) \neq \emptyset$, and so, for instance, $(S, A \setminus S)$ is a Radon partition.

Conversely, suppose (S,T) is a Radon partition of A, i.e., for every point $x \in A$ there exists a real number $c(x)$, $0 \leq c(x) \leq 1$, such that

$$\sum_{s\in S} c(s)s = \sum_{t\in T} c(t)t \in \mathcal{Z}^d \text{ and } \sum_{s\in S} c(s) = \sum_{t\in T} c(t) = 1.$$

Let $1 \leq i < j \leq 2n$, and consider the $(2,i,j)$th coordinate. It is clear then, that if $c(y^{i,j}) > 0$ or $c(z^{i,j}) > 0$, then in fact $c(y^{i,j}) = c(z^{i,j})$ and $y^{i,j}, z^{i,j}$ are on opposite sides of the partition. If both coefficients above are zero then, moving $y^{i,j}$ to the other side if necessary, we may assume again that it is on the opposite side of $z^{i,j}$. So, we may assume that for all $1 \leq i < j \leq 2n$ we have $c(y^{i,j}) = c(z^{i,j})$ and $y^{i,j}, z^{i,j}$ are on opposite sides of the partition.

Now let $1 \leq i < j \leq 2n$. We can make the following claim.

CLAIM. *If the points x^i and x^j are on the same side of the partition, then $c(y^{i,j}) = c(z^{i,j}) = 0$ and $c(x^i) = c(x^j)$, whereas if they are on opposite sides of the partition, then $c(x^i) = c(x^j) = c(y^{i,j}) = c(z^{i,j})$.*

Proof of the claim. Consider the $(1,i,j)$th coordinate. Four cases arise, according to whether x^i and x^j lie on the same or opposite side of the partition, and whether x^i and $y^{i,j}$ lie on the same or opposite side of the partition. We demonstrate the proof in the case where both x^j and $y^{i,j}$ appear on the opposite side of the partition than x^i. In that case, we have

$$-c(x^i) + c(z^{i,j}) = c(x^j) - c(y^{i,j}) \in \mathcal{Z}.$$

If $c(y^{i,j}) = 0$, then

$$c(z^{i,j}) = 0 \Longrightarrow 0 \geq -c(x^i) = c(x^j) \geq 0 \Longrightarrow c(x^i) = c(x^j) = 0.$$

If $c(y^{i,j}) = 1$, then

$$c(z^{i,j}) = 1 \Longrightarrow 0 \leq 1 - c(x^i) = c(x^j) - 1 \leq 0 \Longrightarrow c(x^i) = c(x^j) = 1.$$

Otherwise, $|c(x^j) - c(y^{i,j})| < 1$ and $c(x^j) - c(y^{i,j}) \in \mathcal{Z}$, so

$$c(x^j) - c(y^{i,j}) = 0 \Longrightarrow -c(x^i) + c(z^{i,j}) = 0 \Longrightarrow c(x^i) = c(z^{i,j}) = c(y^{i,j}) = c(x^j),$$

as claimed. The other cases are simpler and are left for the reader to verify.

The claim implies that, for all $1 \leq i < j \leq 2n$, we have $c(x^i) = c(x^j)$, so for all $i \in [2n]$ we get $c(x^i) = c$ for some number $c \geq 0$. If $c = 0$ then, by the claim again, for all i, j such that $1 \leq i < j \leq 2n$ we also have $c(y^{i,j}) = c(z^{i,j}) = 0$; But then all coefficients are zero, which is impossible. So $c > 0$.

Now, let $I = \{i : x^i \in S\}$, $J = \{j : x^j \in T\}$. We have shown in the beginning of the proof that, for any $1 \leq i < j \leq 2n$, we have $c(y^{i,j}) = c(z^{i,j})$ and $y^{i,j}$, $z^{i,j}$ are on opposite sides of the partition, so we have

$$0 = \sum_{s\in S} c(s) - \sum_{t\in T} c(t) = c|I| - c|J|,$$

implying $|I| = |J|$, and so (I,J) is a partition of $[2n]$ such that $|I| = |J| = n$.

Finally, consider the 0th coordinate. We have

$$\sum_{i\in I} cka_i = \sum_{i\in I} cx_0^i = \sum_{s\in S} c(s)s_0 = \sum_{t\in T} c(t)t_0 = \sum_{j\in J} cx_0^j = \sum_{j\in J} cka_j$$

which implies that $\sum_{i\in I} a_i = \sum_{j\in J} a_j$, so (I,J) is a balanced partition for (n,B). □

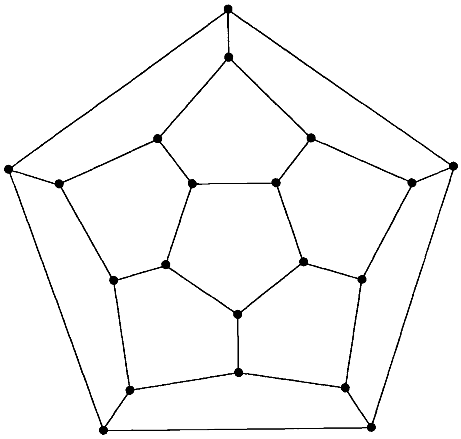


FIG. 4

7. Remaining questions. Computing the lower and upper bounds given in Corollary 2.2 and Corollary 3.7, we obtain $11 \leq r(3) \leq 24$. For simple polytopes we can do a little better.

PROPOSITION 7.1. $11 \leq r(3, SP) \leq 21$.

Proof. Let P be a simple 3-polytope, and let f_i be the number of i-faces of P $(i = 0, 1, 2)$. Suppose $f_0 \geq 21$. Let g_i be the number of vertices of P contained in its ith facet $(i = 1, \cdots, f_2)$. We claim that there exists a facet having $g_i \geq r(2) = 6$. Now $f_1 = 3f_0/2$ (P simple), so Euler's formula implies $f_0 = 2(f_2 - 2)$. Counting incidences of vertices and facets, we get $\sum_{i=1}^{f_2} g_i = 3f_0 = 6f_2 - 12$. Now $f_0 \geq 21$, so $f_2 \geq (3 \cdot 21 + 12)/6 > 12$. Thus, the average number of vertices on a facet is $(6f_2 - 12)/f_2 = 6 - 12/f_2 > 5$, so there must be a facet containing at least six vertices, proving the claim. But then the set of vertices of this facet has a Radon partition, by Observation 1.4, so the set $\text{ext}(P)$ has a Radon partition as well. □

It will be interesting to find tighter lower and upper bounds for $r(d)$ and $r(d, SP)$, or at least the exact values of $r(3)$ and $r(3, SP)$. Even more specifically, does the set of extreme points of any lattice 3-polytope realization of the dodecahedron (see Fig. 4) always admit a Radon partition?

Acknowledgements. I am grateful to Leslie E. Trotter, Jr., for many discussions that led me both to ask the question about the Radon number for the integer lattice and to provide a partial solution to it, and for his careful reading of the manuscript. I also thank Günter M. Ziegler and Joseph S. B. Mitchell for helpful remarks concerning the presentation of this paper. Finally, I am indebted to an anonymous referee for valuable remarks, for helping in simplifying the presentation in several places, and for pointing out to me that Lemma 4.6 holds under an hypothesis weaker than stated in the original version of the paper.

REFERENCES

[1] D. E. BELL, *A theorem concerning the integer lattice*,Stud. in Appl. Math., 56 (1977), pp. 187–188.

[2] A. BJÖRNER, M. LAS VERGNAS, B. STURMFELS, N. WHITE, AND G. M. ZIEGLER, *Oriented Matroids*, Cambridge University Press, Cambridge, to appear.

[3] A. BRØNDSTED, *An Introduction to Convex Polytopes*, Springer-Verlag, Berlin, New York, 1983.

[4] W. COOK, J. FONLUPT, AND A. SCHRIJVER, *An integer analogue of Caratheodory's theorem*, J. Combin. Theory Ser. B, 40 (1986), pp. 63–70.

[5] L. DANZER, B. GRÜNBAUM, AND V. KLEE, *Helly's theorem and its relatives*, Proc. Sympos. Pure Math., 7 (1963), pp. 101–180.

[6] J. P. DOIGNON, *Convexity in crystallographical lattices*, J. Geom., 3 (1973), pp. 71–85.

[7] J. ECKHOFF, *Der satz von Radon in konvexen produktstrukturen* I, Monatsh. Math., 72 (1968), pp. 303–314.

[8] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability*, W. H. Freeman, San Francisco, CA, 1979.

[9] P. M. GRUBER AND C. G. LEKKERKERKER, *Geometry of Numbers*, North–Holland, Amsterdam, 1987.

[10] B. GRÜNBAUM, *Convex Polytopes*, John Wiley, New York, 1967.

[11] P. C. HAMMER, *Extended topology: Caratheodory's theorem on convex hulls*, Rend. Circ. Math. Palermo (2), 14 (1965), pp. 34–42.

[12] D. C. KAY AND E. W. WOMBLE, *Axiomatic convexity theory and relationships between the Caratheodory, Helly and Radon numbers*, Pacific J. Math., 38 (1971), pp. 471–485.

[13] H. W. LENSTRA, JR., *Integer programming with a fixed number of variables*, Math. Oper. Res., 8 (1983), pp. 538–548.

[14] F. W. LEVI, *On Helly's theorem and the axioms of convexity*, J. Indian Math. Society, 15 (1951), pp. 65–76.

[15] H. E. SCARF, *An observation on the structure of production sets with indivisibilities*, Proc. Nat. Acad. Sci. U.S.A., 74 (1977), pp. 3637–3641.

[16] A. SCHRIJVER, *Theory of Linear and Integer Programming*, John Wiley, New York, 1986.

[17] G. SIERKSMA, *Relationships between Caratheodory, Helly, Radon and exchange numbers of convexity spaces*, Nieuw Arch. Wisk. (4), 3 (1977), pp. 115–132.

[18] H. TVERBERG, *A generalization of Radon's theorem*, J. London Math. Society, 41 (1966), pp. 123–128.

SIAM J. DISCRETE. MATH.
Vol. 4, No. 3, pp. 448–451, August 1991

013

A NOTE ON PLANAR GRAPHS AND CIRCLE ORDERS*

EDWARD R. SCHEINERMAN†

Abstract. A partially ordered set P is called a *circle order* if one can assign to each element $a \in P$ a circular disk in the plane C_a so that $a < b$ if and only if $C_a \subset C_b$. To a graph $G = (V, E)$ associate a poset $P(G)$ whose elements are the vertices and edges of G. $v < e$ in $P(G)$ exactly when $v \in V$, $e \in E$, and v is an endpoint of e. It is shown that G is planar if and only if $P(G)$ is a circle order.

Key words. planar graphs, circle (containment) orders

AMS(MOS) subject classifications. 05C10, 05C75, 06A10

The purpose of this paper is to provide a characterization of planar graphs by showing that natural partially ordered sets associated with planar graphs are circle orders.

A *circle order* is a partially ordered set P with the property that to each element $a \in P$ we can assign a circular disk (i.e., a circle together with its interior) C_a so that $a < b$ if and only if $C_a \subset C_b$. The mapping $a \mapsto C_a$ is called a *circle containment representation* of P. Note that if the centers and radii of a and b are $\mathbf{p}_a, \mathbf{p}_b$ and r_a, r_b, respectively, then

$$C_a \subset C_b \iff |\mathbf{p}_a - \mathbf{p}_b| < r_b - r_a.$$

Circle orders (and their relatives) have attracted a great deal of interest [2]–[5], [7]. One of the more nagging problems in this area is the following question.

THE QUESTION. *Is every finite three-dimensional poset a circle order?*

In [5] (see also [3]) it is shown that the infinite three-dimensional poset $\mathbf{Z}^3$ (triples of integers ordered coordinatewise) is *not* a circle order, yet if the word "circle" is replaced by "regular 10,000-gon" the answer to the question is "yes."

(The circle order concept extends naturally to higher dimensions yielding sphere orders. These partial orders arise as "causality" orders in physics (see [4]) and it is known [2] that there are posets that are the containment order of spheres in $\mathbf{R}^{k+1}$ but not of spheres in $\mathbf{R}^k$ for every positive integer k.)

One way to approach the question is to consider specific kinds of three-dimensional orders and determine whether or not they are circle orders. A theorem of Schnyder [6] provides a natural candidate class as follows.

Given a graph $G = (V, E)$ we can define the *vertex-edge incidence poset* $P(G)$ as follows. The elements of $P(G)$ are the vertices and edges of G, i.e., $P(G) = V \cup E$. In $P(G)$ we have $v < e$ exactly when $v \in V$, $e \in E$, and $v \in e$, i.e., v is an endpoint of e.

Perhaps the nicest result on the orders $P(G)$ is the following result due to Schnyder [6].

THEOREM 1. *Let G be a graph and let $P(G)$ be its vertex-edge incidence poset. The graph G is planar if and only if* $\dim P(G) \leq 3$.

Theorem 1, together with an affirmative answer to the question, would yield the implication *if G is planar, then $P(G)$ is a circle order.* This implication turns out to be

* Received by the editors April 2, 1990; accepted for publication July 25, 1990.

† Department of Mathematical Sciences, The Johns Hopkins University, Baltimore, Maryland 21218 (e-mail: ers@cs.jhu.edu or ers@jhunix.bitnet). This research was supported in part by Office of Naval Research contract N00014-85-K-0622.

an easy consequence of a result due to Thurston's [8] work in geometric topology (see Theorem 4 below). Moreover, we also prove here the opposite implication, namely, *if $P(G)$ is a circle order, then G is planar.* Thus our main result is the following characterization of planar graphs via circle orders.

THEOREM 2. *Let G be a graph and let $P(G)$ be its vertex-edge incidence poset. The graph G is planar if and only if $P(G)$ is a circle order.*

In our proof we find it more convenient to work with the dual poset $\hat{P}(G)$ of $P(G)$. (Recall that the *dual* $\hat{P}$ of P is a poset with the same elements as P such that $x <_P y \iff y <_{\hat{P}} x$.) We lose no generality thanks to the following observation.

LEMMA 3. *A poset P is a circle order if and only if its dual $\hat{P}$ is a circle order as well.*

Proof. Since duality is an involution, it is enough to show that if P is a circle order then so is its dual $\hat{P}$. Suppose P is a circle order and to each $a \in P$ we assign a circle $C(\mathbf{p}_a, r_a)$ with center $\mathbf{p}_a$ and radius r_a. Choose a number M greater than r_a (for all $a \in P$). Now assign $a \mapsto C(\mathbf{p}_a, M - r_a)$ which gives a circle containment representation for $\hat{P}$. □

Thus it is enough to prove that G is planar if and only if $\hat{P}(G)$ is a circle order.

Proof of Theorem 2. First, assume that $\hat{P}(G)$ is a circle order. To each vertex v we assign a circle C_v with center $\mathbf{p}_v$ and radius r_v and to each edge vw we have C_{vw} with center $\mathbf{p}_{vw}$ and radius r_{vw}. The inclusions we have are exactly those of the form $C_v \supset C_{vw}$.

We show that G is planar by finding an explicit embedding. Each vertex $v \in V(G)$ is embedded at the point $\mathbf{p}_v$, the center of its representing circle. Each edge vw is embedded as the "two-step" path consisting of the line segment from $\mathbf{p}_v$ to $\mathbf{p}_{vw}$ followed by the segment from $\mathbf{p}_{vw}$ to $\mathbf{p}_w$. We must show that these edges are noncrossing.

Suppose, for sake of contradiction, that $ab, cd \in E(G)$ and their embeddings cross. Without loss of generality, line segment $\mathbf{p}_a\mathbf{p}_{ab}$ intersects line segment $\mathbf{p}_c\mathbf{p}_{cd}$.

By the triangle-inequality we see that

$$|\mathbf{p}_a - \mathbf{p}_{ab}| + |\mathbf{p}_c - \mathbf{p}_{cd}| \geq |\mathbf{p}_a - \mathbf{p}_{cd}| + |\mathbf{p}_c - \mathbf{p}_{ab}|.$$

Add $r_{ab} + r_{cd}$ to both sides and use the facts

$$\begin{array}{lcll} r_a & > & |\mathbf{p}_a - \mathbf{p}_{ab}| + r_{ab} & \text{(because } C_a \supset C_{ab}\text{)}, \\ r_c & > & |\mathbf{p}_c - \mathbf{p}_{cd}| + r_{cd} & \text{(because } C_c \supset C_{cd}\text{)} \end{array}$$

to find that

$$r_a + r_c > (|\mathbf{p}_a - \mathbf{p}_{cd}| + r_{cd}) + (|\mathbf{p}_c - \mathbf{p}_{ab}| + r_{ab}).$$

It follows that either $r_a > |\mathbf{p}_a - \mathbf{p}_{cd}| + r_{cd}$ or $r_c > |\mathbf{p}_c - \mathbf{p}_{ab}| + r_{ab}$. The first contradicts $C_a \not\supset C_{cd}$ and the second contradicts $C_c \not\supset C_{ab}$. Thus edges ab and cd do not cross.

Note 1. It is possible, however, that edges such as ab and ac do cross if segments $\mathbf{p}_a\mathbf{p}_{ab}$ and $\mathbf{p}_c\mathbf{p}_{ac}$ intersect. However, crossings of edges emanating from the same vertex are easy to repair locally. See Fig. 1.

Note 2. The two-step paths cannot be readily replaced by straight line segments between the centers of the vertex circles. Figure 2 shows how the line segments $\mathbf{p}_a\mathbf{p}_b$ and $\mathbf{p}_c\mathbf{p}_d$ might intersect and how two-step paths avoid this collision.

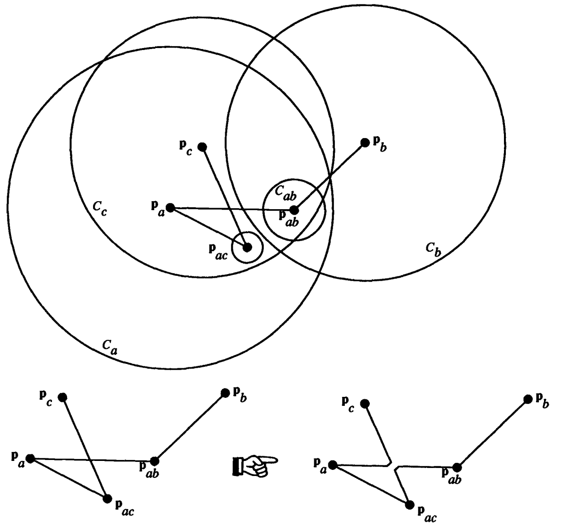


FIG. 1. *How line segments* $\mathbf{p}_a\mathbf{p}_{ab}$ *and* $\mathbf{p}_c\mathbf{p}_{ac}$ *might intersect and how to repair such an intersection.*

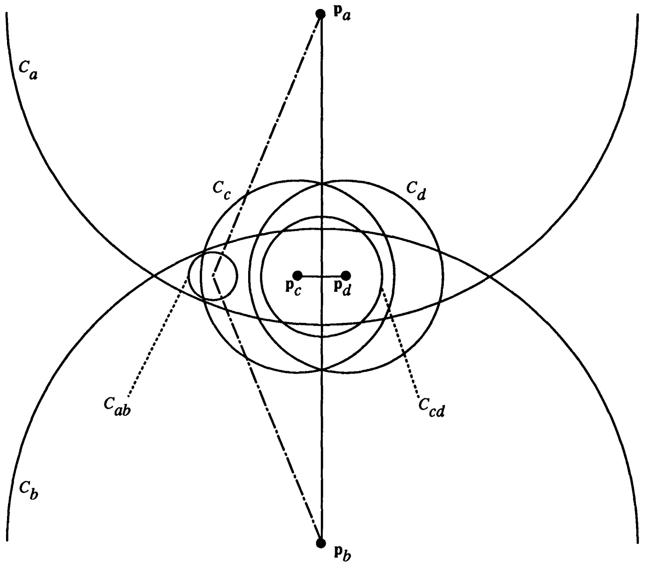


FIG. 2. *Why two-step paths are necessary.*

It therefore follows that if $\hat{P}(G)$ is a circle order, then G is planar.

The converse follows easily from the following result due to Thurston [8].

THEOREM 4. *Let G be a planar graph. We can assign to each $v \in V(G)$ a circular disk C_v so that $vw \in E(G)$ implies C_v and C_w intersect at a single point and $vw \notin E(G)$ implies that C_v and C_w are disjoint.*

If G is planar let the circles assigned to vertices of G be the circles whose existence is guaranteed in Theorem 4. If $vw \in E(G)$ let C_{vw} be centered at the unique intersection point of C_v and C_w with radius 0. Clearly, this is a circle representation of $\hat{P}(G)$. (For those unhappy with circles of radius 0, note that we can simply increase all radii by a fixed amount, say 1, and no containments are created or destroyed.) □

Combining Theorems 1 and 2 we obtain the following lovely result.

THEOREM 5. *Let G be a graph and let $P(G)$ be its vertex-edge incidence poset. The following statements are equivalent:*

- *G is a planar.*
- *$P(G)$ is a circle order.*
- $\dim P(G) \leq 3$.

REFERENCES

[1] G. BRIGHTWELL AND W. T. TROTTER, *The order dimension of convex polytopes*, preprint.

[2] G. BRIGHTWELL AND P. WINKLER, *Sphere orders*, Order, 6 (1989) pp. 323–343.

[3] G. H. HURLBERT, *A short proof that* $\mathbf{N}^3$ *is not a circle containment order*, Order, 5 (1988) pp. 235–237.

[4] D. MEYER, *Spherical containment and the Minkowski dimension of partial orders*, preprint.

[5] E. R. SCHEINERMAN AND J. C. WIERMAN, *On circle containment orders*, Order, 4 (1988) pp. 315–318.

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

[7] 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.

[8] W. THURSTON, *The Geometry and Topology of Three-Manifolds*, unpublished book.

SIAM J. DISC. MATH.
Vol. 4, No. 4, pp. 453-471, November 1991

EXTENDABILITY, DIMENSIONS, AND DIAGRAMS OF CYCLIC ORDERS*

PETER ALLES†, JAROSLAV NEŠETŘIL‡, AND SVATOPLUK POLJAK‡

Abstract. Several classes of cyclic orders arising from geometrical, algebraical, and combinatorial structures are introduced, and their extendability to total cyclic orders is studied. By analogy to Dushnik–Miller dimension for partial orders we define, for circular orders, intersection and product dimension that may differ up to a factor of two. A class of cyclic orders that allow a graphic representation similar to Hasse diagrams is also studied.

Key words. cyclic order, dimension, partial order, cycles

AMS(MOS) subject classifications. 06A10, 06A99, 05C99

1. Introduction.

1.1. A set X with a set $T \subset X^3$ of triples is called a cyclic order if and only if T is cyclic, asymmetric, and transitive. We can easily find examples for cyclic orders in various mathematical areas. The most well-known class of cyclic orders is derived from partial orders (we call them poset-generated cyclic orders). Other classes that we will study in more detail in §§ 2 and 3 are the class of arc orders generated by sets of intervals on a circle in the plane, the class of incidence orders that come from incidence systems, the class of cyclic orders generated by triangulated graphs, and the class of path orders derived from a set of paths in the plane with common initial and final point.

More cyclic orders can be obtained from categorical operations like subobject, product, and intersection. The most interesting class is the class of circular orders that are intersections of total cyclic orders (circles). In § 2 we show that this class is closed with respect to subobjects and products. We show by a counting argument that incidence orders in general are not circular. On the other hand, path orders, arc orders, and poset-generated cyclic orders are circular.

The decision whether a cyclic order is totally extendable is known to be NP-complete. However, for an infinite cyclic order, this property is determined by its finite cyclic suborders since we prove that a cyclic order is totally extendable (circular) if and only if each finite cyclic suborder is totally extendable (circular).

By analogy to Dushnik–Miller dimension for partial orders we introduce in § 5 for the class of circular orders the intersection and product dimension. It turns out that for a circular order the two dimensions may differ up to a factor of two. We also give several estimates for the dimension of certain circular orders.

There exist two characterizations of cyclic orders. The first is given by permutations on a set, the second, which is discussed in detail in § 4, characterizes a cyclic order by its maximal total cyclic suborders. A specification of the latter equivalent description of cyclic orders enables the graphical representation of certain cyclic orders (graphical cyclic orders) by means of oriented graphs (Hasse diagrams). We also study relations of the class of graphical cyclic orders to other classes.

* Received by the editors May 10, 1990; accepted for publication September 4, 1990.

† Digital Equipment GmbH, Geschäftsstelle Finanzwirtschaft, Halmstr. 25, 6000 Frankfurt–Niederrad, Germany.

‡ Department of Applied Mathematics, Charles University, Malostranske nam. 25, 11800 Prague, Czechoslovakia.

In this article we study both finite and infinite cyclic orders. When dealing with the cyclic orders arising from some geometrical representation, we restrict ourselves only to the finite case, though the results often allow extension to the countable case.

1.2. We start with the following definition.

DEFINITION. Let X be a set, and $T \subset X^3$ be a set of triples. Then

$$C=(X,T)=(X(C),T(C))$$

is called *cyclic order* if and only if the ternary relation T is

(C1) Cyclic, i.e., $(x, y, z) \in T$ implies $(y, z, x) \in T$;

(C2) Asymmetric, i.e., $(x, y, z) \in T$ implies $(z, y, x) \notin T$; and

(C3) Transitive, i.e., $(x, y, z), (x, z, w) \in T$ implies $(x, y, w) \in T$.

A cyclic order $C = (X, T)$ is called a *total cyclic order* or *circle* if and only if T is

(C4) Total, i.e., for each $x, y, z \in X, x \neq y \neq z \neq x$, either $(x, y, z) \in T$ or $(z, y, x) \in T$.

Note that for a cyclic order $C = (X, T)$, $(x, y, z) \in T$ implies $x \neq y \neq z \neq x$.

1.3. The notion of cyclic orders was first introduced by Huntington [13] in 1924 and then independently many times in [11], [32], and [19] with geometrical motivation, in [8] and [22] with algebraic motivation. Axioms for total cyclic orders appear in [12] and [5].

Despite these many sources there are only few results on cyclic orders. The most interesting results concern extensions of cyclic orders to total cyclic orders [8], [17] mentioned in § 2.6. This is also our main scheme here.

1.4. Now we give a collection of examples of cyclic orders.

1.4.1. In this paper, a partial order $P = (X, R)$ always means that the relation $R \subset X^2$ is asymmetric (hence antireflexive) and transitive.

A cyclic order $C = (X, T)$ is called *poset-generated*, indicated by C_P, if and only if there exists a partial order $P = (X, R)$ such that

$$T=\{(x,y,z),(y,z,x),(z,x,y) \mid (x,y),(y,z) \in R\}.$$

1.4.2. Let X be an at most countable set of arcs on a circle in the Euclidean plane. (An arc means here an open, semi-open, or closed interval on a circle such that its closure is not the whole circle. Such configurations appear in the analysis of phasing traffic signals [35]. The related intersection graphs were characterized by Tucker [37], [38].) Let three arcs form a triple if and only if they are pairwise disjoint and ordered clockwise on the circle. We call this cyclic order an *arc order*. Especially, if each two arcs in X are disjoint, then this yields a total cyclic order.

1.4.3. Let $I = (X, \mathscr{B})$ be an incidence system satisfying $|b| \geqq 3$ for each $b \in \mathscr{B}$ and $|b \cap b'| \leqq 1$ for each $b, b' \in \mathscr{B}, b \neq b'$. On each block $b \in \mathscr{B}$, $b = \{b_1, b_2, \cdots\}$, choose a cyclic orientation $b_1 <_b b_2 <_b \cdots <_b b_1$ on its elements. Then we get a cyclic order $C = (X, T)$, called *incidence order*, from I by

$$(x,y,z) \in T \quad \text{iff } x <_b y <_b z <_b x \text{ for some } b \in \mathscr{B}.$$

1.4.4. A graph $G = (X, E)$ is called *oriented* if and only if $E \subset X^2$ is asymmetric. A finite-oriented graph $G = (X, E)$ is called a *circuit* if and only if for $X = \{x_1, x_2, \cdots, x_n\}$, $E = \{(x_i, x_{i+1}) \mid i = 1, \cdots, n-1\} \cup \{(x_n, x_1)\}$. G' is called *chordless circuit in* G if and only if the circuit G' is an induced subgraph of G.

Now, let $G = (X, E)$ be a triangulated oriented graph, i.e., there are no chordless circuits of length greater than three in G, and let each edge $e \in E$ be in some directed triangle in G. Then G induces a cyclic order $C = (X, T)$ in the following way:

$$(x,y,z) \in T \quad \text{iff} \, (x,y),(y,z),(z,x) \in E.$$

1.4.5. Let α, β be two arbitrary but fixed points in the Euclidean plane E^2. A *path* x is a continuous and injective mapping $x : [0, 1] \to E^2$ with $x(0) = \alpha$, $x(1) = \beta$. Let X be a set of paths such that each pair of distinct paths intersect finitely many times. For each three paths $x, y, z \in X$, $x \neq y \neq z \neq x$, let S_{xyz} be a circle in the plane with center α such that x, y, z do not intersect on S_{xyz} or in its interior except in α.

Now we can derive a cyclic order $C = (X, T)$, called *path order*, from X in the following way:

$(x, y, z) \in T$ if and only if x, y, z are pairwise disjoint (except in α, β) and x, y, z in this order intersect S_{xyz} clockwise. See Fig. 1.

Example.

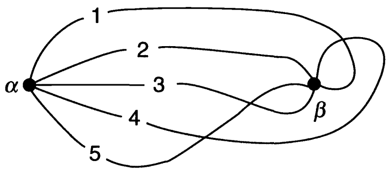


FIG. 1

$X = \{1, 2, 3, 4, 5\}$ and $T = \{(1, 2, 3), (2, 3, 1), (3, 1, 2), (1, 2, 5), (2, 5, 1), (5, 1, 2), (2, 3, 4), (3, 4, 2), (4, 2, 3)\}$.

1.5. An equivalent definition of cyclic orders can be given by a certain set of permutations. Let π be a permutation on a set X. Let us call π a 3-*cycle* (or 3-*permutation*) if it consists of exactly one cycle of length three leaving all other points fixed.

PROPOSITION. *Let S be a set of 3-cycles on X satisfying the following*:

(P1) *If $\pi \in S$, then $\pi^{-1} \notin S$.*

(P2) *If $\pi, \rho \in S$ and $\pi \circ \rho$ is a 3-cycle ($\circ$ denoting the usual composition of permutations), then $\pi \circ \rho \in S$.*

Then $C = (X, T)$ where $T = \{(x, y, z) \mid y = \pi(x), z = \pi(y)$ and $x = \pi(z)$ for some $\pi \in S\}$ is a cyclic order. Conversely, every cyclic order can be obtained in this way.

For a second characterization of cyclic orders see the theorem in § 4.5.

1.6. Let $C = (X, T)$ be a cyclic order, and let $X' \subset X$ be a proper subset of X. If the restriction $C|_{X'}$ of C to X' is a circle, then $C' = C|_{X'} = (X', T \cap X'^3)$ is called a *circle in* C. C' is called *maximal circle in* C if and only if it is a circle in C and there is no $X'' \subseteq X$ such that $X' \subset X''$ and $C|_{X''}$ is a circle in C. Using Zorn's lemma it is easy to see that each circle in a cyclic order is contained in some maximal circle (cf. [13]).

For a cyclic order C, let $\mathcal{M}(C)$ designate the *set of all maximal circles in* C. For a circle $C = (X, T)$ and $x, y \in X$, $x \neq y$, let $C(x, y) \subset X$ designate the following set:

$$C(x,y) := \{z \in X \mid (x,z,y) \in T\}.$$

Let $C = (X, T)$ and $C' = (X, T')$ be cyclic orders (on the same underlying set X) and $T \subseteq T'$. C' is called *total extension* of C if and only if C' is a circle.

1.7. We now give some more definitions that can be used to obtain new cyclic orders from already existing ones.

Let $C = (X, T)$ be a cyclic order and $X' \subset X$ be a proper subset of X. Then the restriction $C' := C|_{X'}$ of C to X' is called *cyclic suborder*, indicated by $C' \leqq C$.

Let $\mathcal{F} = \{C_i = (X_i, T_i) \mid i \in I\}$ be a family of cyclic orders. Then the (*direct*) *product* $\times\mathcal{F}$ of the family F is given by $X(\times\mathcal{F}) = \times_{i\in I} X_i$ (the Cartesian product), and $T(\times\mathcal{F}) = \times_{i\in I} T_i$, where $((x_i)_{i\in I'}(y_i)_{i\in I'}(z_i)_{i\in I}) \in \times_{i\in I} T_i$ if and only if $(x_i, y_i, z_i) \in T_i$ for each $i \in I$.

The homomorphisms (cf. § 2.4)

$$p_j : \underset{i\in I}{\times} C_i \rightarrow C_j, \qquad (x_i)_{i\in I} \mapsto x_j,$$

are called (*natural*) *projections*. The *intersection* $\cap\mathcal{F}$ of the family $\mathcal{F}$ is defined by

$$X(\cap\mathcal{F}) = \bigcap_{i\in I} X_i, \qquad T(\cap\mathcal{F}) = \bigcap_{i\in I} T_i.$$

By definition of cyclic suborder, product, and intersection of cyclic orders, the next theorem follows immediately.

THEOREM. *The class of cyclic orders is closed with respect to subobjects, products, and intersections.*

1.8. The most interesting class of cyclic orders that will be investigated in §§ 2 and 5 is the class of circular orders.

DEFINITION. A cyclic order is called *circular order* if and only if it is the intersection of a family of circles.

1.9. For convenience, we denote a finite circle C with $X(C) = \{x_1, \cdots, x_n\}$ simply by $x_1x_2\cdots x_n$ to indicate the order on C:

(*) $(x_i, x_j, x_k) \in T(C)$ iff either $i < j < k$ or $j < k < i$ or $k < i < j$.

Finally, if $\{t_1, t_2, \cdots, t_k\}$ is a set of sequences each of length at least three, we write $T = \langle t_1, t_2, \cdots, t_k\rangle$ to indicate that T is the set of all triples and all cyclic permutations of triples which can be derived from t_i according to rule (*). For example, $T = \langle 1345, 2345\rangle$ denotes the set of triples of the disk order of the example in § 1.4.5.

2. Circular orders.

2.1. In this section we begin the investigation of the class of circular orders (for the definition see § 1.8). We show that this class is closed with respect to subobjects and products and we prove that circles are the only subdirectly irreducible cyclic orders in the class of circular orders.

Then we reflect on the extendability of cyclic orders to total cyclic orders. Examples of cyclic orders that do not have any total extension are well known. We show the existence of an infinite series of incidence orders that are not totally extendable. We also prove that a cyclic order is totally extendable, respectively circular, if and only if each of its finite cyclic suborders is totally extendable, respectively circular.

2.2. THEOREM. *The class of circular orders coincides with the class of all subobjects of products of circles.*

Proof. Obviously, the intersection of a family of circles is isomorphic to a cyclic suborder of the product of this family (viz., the "diagonal" of the product). We prove the converse by defining a set of total extensions of the product of a family of circles representing the product as the intersection of these extensions.

Let $\mathscr{F} = \{C_i = (X_i, T_i) \mid i \in I\}$ be a family of circles, and let I be a linearly ordered index set. For $i \in I$, let $s_i \in X_i$ be arbitrary but fixed elements, and define

$$R_i := \{(y,z) \mid (s_i, y, z) \in T_i\} \cup \{(s_i, y) \mid y \in X_i - \{s_i\}\}.$$

First, for each $i \in I$, we define two lexicographic orderings $<_i$ and $<_{\underline{i}}$ of $|I|$-tuples. Let $x = (x_i)_{i \in I}$ and $y = (y_i)_{i \in I}$ be $|I|$-tuples. Then, for $i \in I$,

$$\begin{aligned}
x <_i y :\Leftrightarrow\ & \text{either } (x_i, y_i) \in R_i \\
& \text{or } (x_j, y_j) \in R_j \text{ for some } j > i \\
& \quad \text{and } x_k = y_k \text{ for all } i \leqq k < j \\
& \text{or } (x_j, y_j) \in R_j \text{ for some } j < i \\
& \quad \text{and } x_k = y_k \text{ for all } i \leqq k \text{ or } k < j, \\
x <_{\underline{i}} y :\Leftrightarrow\ & \text{either } (x_i, y_i) \in R_i \\
& \text{or } (y_j, x_j) \in R_j \text{ for some } j > i \\
& \quad \text{and } x_k = y_k \text{ for all } i \leqq k < j \\
& \text{or } (y_j, x_j) \in R_j \text{ for some } j < i \\
& \quad \text{and } x_k = y_k \text{ for all } i \leqq k \text{ or } k < j.
\end{aligned}$$

Now we define total extensions $D_k = (\times_{i \in I} X_i, U_k)$, $k \in \{j, \underline{j}\}$, $j \in I$, of $\times\mathscr{F}$ by

$$(x, y, z) \in U_k \quad \text{iff either } x <_k y <_k z \text{ or } y <_k z <_k x \text{ or } z <_k x <_k y$$

for $x, y, z \in \times_{i \in I} X_i$, $x \neq y \neq z \neq x$. For $(x, y, z) \in \times_{i \in I} T_i$, we have $x_k \neq y_k \neq z_k \neq x_k$ for each $k \in I$, and it is easy to see that $(x, y, z) \in U_j$, $U_{\underline{j}}$ for each $j \in I$. Suppose $(z, y, x) \notin \times_{i \in I} T_i$ for $x, y, z \in \times_{i \in I} X_i$ with $x \neq y \neq z \neq x$. Then either there exist coordinates $i_1, i_2 \in I$ such that $(x_{i_1}, y_{i_1}, z_{i_1}) \in T_{i_1}$, $(x_{i_2}, z_{i_2}, y_{i_2}) \in T_{i_2}$ or there is a coordinate $i \in I$ such that $|\{x_i, y_i, z_i\}| < 3$. In the first case, we have $(x, y, z), (x, z, y) \notin U_{i_1} \cap U_{i_2}$, in the second, $(x, y, z), (x, z, y) \notin U_i \cap U_{\underline{i}}$. Hence,

$$\mathop{\times}_{i \in I} T_i = \mathop{\times}_{j \in I} U_j \cap U_{\underline{j}} \quad \text{and} \quad \mathop{\times}_{i \in I} C_i = \mathop{\times}_{j \in I} D_j \cap D_{\underline{j}}. \qquad \square$$

2.3. Now it is routine to show the following theorem.

THEOREM. *Subobjects and products of circular orders are circular again.*

2.4. However, the minimum cardinality of a family of circles necessary to represent a given circular order as intersection of this family does in general not coincide with the minimum cardinality of a family of circles such that the circular order is isomorphic to a subobject of a product of this family. This feature will be discussed in more detail in § 5.

Next we prove why it is reasonable to express the "complexity" of a circular order by the minimum number of circles necessary to represent it. To this end, we first have to give some additional definitions.

Let $C = (X, T)$ and $C' = (X', T')$ be cyclic orders and $f: X \to X'$ be a mapping. Then f is called a *homomorphism*, indicated by $f: C \to C'$, if and only if for each $x, y, z \in X$, $(x, y, z) \in T$ implies $(fx, fy, fz) \in T'$.

If f is an injective homomorphism, then f is called an *embedding* if and only if $fC \leqq C'$. If f is a bijective homomorphism and $fT = T'$, then f is called an *isomorphism* and C, C' are said to be *isomorphic*, indicated as usual by $C \simeq C'$.

Remark. For a homomorphism $f: C \to C'$ and $x, y \in X(C)$, $x \neq y$, $fx = fy$ is possible only if x, y are not both in some triple of $T(C)$.

2.5. The cyclic order $C = (X, T)$ is called *subdirectly irreducible* in a category $\mathscr{C}$ of cyclic orders if and only if for each embedding $f: C \to \times_{i \in I} C_i$ into a product of cyclic orders from $\mathscr{C}$ such that all $p_i f: C \to C_i$ are onto, at least one of $p_i f$ is an isomorphism.

THEOREM. *Circles are the only subdirectly irreducible cyclic orders in the class of circular orders.*

Proof. Suppose $C = (X, T)$ to be a circle and the inclusion mapping (embedding) $f: C \twoheadrightarrow \times_{i \in I} C_i$ into a product of circles to be such that each $p_i f: C \twoheadrightarrow C_i$ is onto. Since C is cyclically totally ordered and all $p_i f$ are homomorphisms, each $p_i f$ has to be one to one and thereby $C \simeq C_i$ $(i \in I)$.

Conversely, suppose the circular order $C = (X, T)$ are not totally ordered. Then there is a family $\mathscr{F} = \{C_i = (X, T_i) \mid i \in I\}$ of circles with $C = \bigcap \mathscr{F}$ and an embedding $f: C \twoheadrightarrow \times \mathscr{F}$, $x \mapsto (x, x, \cdots)$, since $(x, y, z) \in T(\bigcap \mathscr{F})$ if and only if $(fx, fy, fz) \in T(\times \mathscr{F})$. By definition of the embedding f, each $p_i f$ is onto and even one to one, but obviously none of C_i is isomorphic to C. Hence C is not subdirectly irreducible. □

2.6. Not each cyclic order is circular. There even exist cyclic orders that do not have any total extension. The first example was given by Meggido [17]. Here we prove the existence of such examples by a counting argument. It indicates that a "random" cyclic order is without total extensions. This proof also reveals that incidence orders (for the definition see § 1.4.3) in general are not totally extendable.

Let $(X, \mathscr{M})$ be a finite, simple, and uniform set system, i.e., $3 \leqq |M| = |M'|$ and $|M \cap M'| \leqq 1$ for each $M, M' \in \mathscr{M}$, $M \neq M'$. On each set $M \in \mathscr{M}$ choose a cyclic orientation $<_M$ of its elements and define an incidence-order $C = (X, T)$ from $(X, \mathscr{M})$ as described in § 1.4.3.

PROPOSITION. *If* $|\mathscr{M}| > |X| \cdot \log |X|$, *then there exists a family of cyclic orientations* $(<_M)_{M \in \mathscr{M}}$ *such that the incidence order* C *fails to be totally extendable.*

Proof. For given $(X, \mathscr{M})$ there exist at least $2^{|\mathscr{M}|}$ different cyclic orders C (by the above construction) since $2^{|\mathscr{M}|}$ is a lower bound for the number of different orientations of the sets $M \in \mathscr{M}$. However, there exist only $(|X| - 1)!$ circles on X, and obviously two different incidence orders cannot have the same circle as total extension. The bound then follows by using Stirling's formula. □

For instance, for each Steiner triple system of order at least 19 there exist cyclic orientations of the blocks such that the derived cyclic order has no total extension.

2.7. It was proved in [8] that the recognition of cyclic orders having total extensions is NP-complete. For an infinite cyclic order this property is determined by its finite cyclic suborders as follows.

THEOREM.

(i) *A cyclic order is totally extendable if and only if each finite cyclic suborder is totally extendable.*

(ii) *A cyclic order is circular if and only if each finite cyclic suborder is circular.*

Proof. It is sufficient to prove the "if-part" for both statements. We use Rado's theorem (which is equivalent to axiom of choice).

RADO'S THEOREM [30, Thm. 7.1.4]. *Let* $G(V, V')$ *be a bipartite graph that is locally finite in* V. *For each finite set* $A \subset V$ *let there be defined a choice function* $f_A(a)$, $a \in A$, *associating with* a *the unique edge* $(a, f_A(a))$ *from it. Then there exists a choice function* $f_V(v)$ *defined for all of* V *with the property that for each* A *there exists a set* $B \supseteq A$ *such that* $f_A(a) = f_B(a)$, $a \in A$.

(i) Let $C = (X, T)$ be a cyclic order such that for each finite subset $Y \subset X$ there is a total cyclic order (Y, T_Y) with $T_Y \supseteq T|_Y$. We apply Rado's Theorem as follows.

Set $V := \binom{X}{3}$, the set of all unordered triples of X, and $V' := \{+1, -1\}$. Let $G(V, V')$ be the complete bipartite graph on V, V'. Let $<$ be some (fixed) linear order of X. For each finite $A \subset V$ define the choice function f_A as follows.

Set $Y := \bigcup A \subset X$ and for every $\{a, b, c\} \in A$, $a < b < c$, and let

$$f_A(\{a,b,c\}) = \begin{cases} +1 & \text{if}\,(a,b,c) \in T_Y, \\ -1 & \text{if}\,(a,c,b) \in T_Y. \end{cases}$$

Clearly, f_V determines a total extension $C' = (X, T')$ of C. (Define $(a, b, c) \in T'$ if and only if $f_V(\{a, b, c\}) = +1$ for $a < b < c$, and $(a, c, b) \in T'$ if and only if $f_V(\{a, b, c\}) = -1$ for $a < b < c$. Thus T' is complete and extends T.)

(ii) Let $C = (X, T)$ be a cyclic order such that each finite cyclic suborder is circular, i.e., for each finite $Y \subset X$, there is a family of circles $\{(Y, T_i) | i \in I_Y\}$ with $T|_Y = \bigcap_{i \in I_Y} T_i$. Let (a, b, c) be a triple which is not in T. Then, for each finite subset $Y \subset X$ with $\{a, b, c\} \subseteq Y$, there is a total extension of $T|_Y$ not containing (a, b, c). Applying Rado's Theorem, we obtain a total extension T_{abc} of T not containing (a, b, c). Thus C is circular. □

3. Path orders, arc orders, and poset-generated cyclic orders.

3.1. We will continue our study of certain subclasses of cyclic orders by the investigation of path orders. We show that path orders are circular and circular orders are in general not path orders. We also prove that arc orders are path orders and hence circular but not each path order is an arc order. Finally, we prove that finite poset-generated cyclic orders are path orders and investigate the relation of this class to other classes.

3.2. PROPOSITION. *Path orders are totally extendable.*

Proof. Let $C = (X, T)$ be a path order. We can assume that X is finite, since the statement then follows for infinite path orders by the theorem in § 2.7. By assumption (§ 1.4.5), for each three distinct elements x, y, z in X, there exists a circle S_{xyz} in the plane with center α such that x, y, z do not intersect on S_{xyz} or in its interior except in α. (In the sequel, by a point of intersection we always mean a point different from α and β; if two paths do not intersect in this sense, we call them disjoint.) Hence, since X is finite, there exists a circle S_X in the plane with center α such that no two paths in X intersect on S_X or in its interior. The points of intersection of paths from X with S_X define a total cyclic order $C_0 = (X, T_0)$ on X which extends the path order C. □

3.3. THEOREM. *Path orders are circular.*

Proof. Again, we may assume that the path order $C = (X, T)$ is finite. Let $C_0 = (X, T_0)$ be the total extension of C defined in Proposition 3.2. For points γ, δ, $\gamma \neq \delta$, on a path $x \in X$, let $\gamma <_x \delta$ indicate that a walk on x from α to δ has to pass γ. Let $x, y \in X$, $x \neq y$, be two paths that intersect. Let $\gamma \neq \alpha$ be the first point of intersection of x and y on x when walking on x from α to β, i.e., if $\delta \in x \cap y$, $\alpha \neq \delta \neq \gamma$, then $\gamma <_x \delta$.

If the finite face in the plane bounded by $x([0, x^{-1}(\gamma)])$ and $y([0, y^{-1}(\gamma)])$ contains β, then let $N_{xy} := C_0(y, x) \cup \{x\}$, otherwise let $N_{xy} := C_0(x, y) \cup \{x\}$.

Let $M_{xy} \subseteq N_{xy}$ be the smallest set satisfying

(i) $x \in M_{xy}$, and

(ii) If $z \in M_{xy}$ and there is a path $w \in N_{xy}$ that is disjoint with z and either $(z, w, y) \in T_0$ if $N_{xy} = C_0(x, y) \cup \{x\}$ or $(w, z, y) \in T_0$ if $N_{xy} = C_0(y, x) \cup \{x\}$, then $w \in M_{xy}$.

Since M_{xy} is the smallest set with properties (i), (ii), each $z \in M_{xy}$, $z \neq x$, is disjoint with x and, by definition of N_{xy}, intersects y.

Now define a total cyclic order $C_{xy} = (X, T_{xy})$ of C by

$(a, b, c) \in T_{xy}$ iff either $(a, c, b) \in T_0$ if $\{a, b, c\} \cap M_{xy} = \{b\}$ or $\{a, b, c\} - M_{xy} = \{b\}$ or $(a, b, c) \in T_0$ in all other cases.

Figure 2 illustrates the case where $N_{xy} = C_0(x, y) \cup \{x\}$.

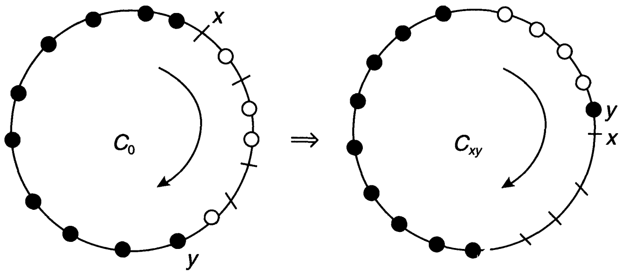


FIG. 2. *Members of M_{xy}*: ——; *members of $N_{xy} - M_{xy}$*: ○; *members of $X - N_{xy}$*: ●.

CLAIM 1. *C_{xy} extends C.*

Proof. We have to show that $T \subseteq T_{xy}$. Let $(a, b, c) \in T$, thus $(a, b, c) \in T_0$. Then not both x and y are contained in (a, b, c). If we check all possibilities that can arise for elements a, b, c, we may distinguish the following cases:

(i) If $\{a, b, c\} \cap M_{xy} = \emptyset$ or $\{a, b, c\} \subseteq M_{xy}$, then $(a, b, c) \in T_{xy}$.

(ii) $\{a, b, c\} \cap M_{xy} = \{a, c\}$ cannot occur.

(iii) In all other cases, we have $\{a, b, c\} - M_{xy} \subseteq X - N_{xy} - \{y\}$ and therefore $(a, b, c) \in T_{xy}$.

This proves Claim 1.

CLAIM 2. *$C = C_0 \cap \bigcap \{C_{xy} | x, y \in X$ are intersecting paths$\}$.*

Proof. Let $C' := C_0 \cap \bigcap \{C_{xy} | x, y \in X$ are intersecting paths$\}$. By Claim 1, it suffices to show that $T(C') \subseteq T$. Let $x, y, z \in X$, $x \neq y \neq z \neq x$, be paths with (x, y, z), $(z, y, x) \notin T$. Without loss of generality let $(x, y, z) \in T_0$ and let x and y intersect. If $N_{xy} = C_0(x, y) \cup \{x\}$, then $z \in X - N_{xy} - \{y\}$ and $(y, x, z) \in T_{xy}$. Otherwise we have either $z \notin M_{xy}$ whence $(y, x, z) \in T_{xy}$, or $z \in M_{xy}$ whence y and z intersect. Then $N_{yz} = C_0(y, z) \cup \{y\} (N_{yz} = C_0(z, y) \cup \{y\}$ cannot occur then), whence $x \in X - N_{yz} - \{z\}$ and $(z, y, x) \in T_{yz}$. Therefore in each case, we have $(x, y, z), (z, y, x) \notin T(C')$ proving $T(C') \subseteq T$.

The theorem is proved by Claim 2 and the theorem in § 2.7. □

3.4. Not each circular order is a path order: Let $C_1 = 12345$, $C_2 = 14235$, $C_3 = 14253$, and $C_4 = 14523$. Then $T(\bigcap_{i=1}^{4} C_i) = \langle 123, 145, 234 \rangle$, which is not representable as path order since the paths representing 1, 2, and 4 must be disjoint but neither 124 nor 142 belongs to $T(C)$.

3.5. Obviously, path orders are closed under taking subobjects. However, they are not closed with respect to products: Let $C = 123456$; then $C \times C$ contains, e.g., the triples $(1, 1)(3, 3)(4, 4)$; $(1, 1)(2, 5)(4, 6)$; and $(3, 3)(4, 4)(2, 5)$, with neither $(1, 1)(3, 3)(2, 5)$ nor $(1, 1)(2, 5)(3, 3)$ corresponding to the situation in § 3.4.

3.6. Next we show that arc orders are path orders. On the other hand, not each path order is an arc order. We also show that products of arc orders need not be arc orders.

THEOREM. *Arc orders are path orders.*

Proof. We first prove the theorem for an arc order $C = (X, T)$, where $X = \{x_i | i \in I\}$ is a set of closed arcs on the circle S in the plane. We describe the procedure for generating the related path-configuration in an informal way since the idea is simple and a formalized definition would be ugly. Let α be the center of S and let S' be a circle with center α and radius twice the radius of S. Let $X' = \{x'_i | i \in I\}$ be the central projection (with center α) of arcs X on S'. Let $x_i = [a_i, b_i] (i \in I)$ with $a_i \leqq b_i$ with respect to clockwise ordering on S; by analogy, $x'_i = [a'_i, b'_i]$ with $a'_i \leqq b'_i$ on S'. Finally, let β be a point in the infinite face of the plane defined by S'.

Then, for each $i \in I$ and $a_i < b_i$, let P_i be a path (in the sense of the definition in § 1.4.5) leading from α to β which consists of a straight line from α to a'_i, a curve (path) from a'_i to b_i, a straight line from b_i to b'_i and a curve from b'_i to β (if $a_i = b_i$, then let P_i consist of a straight line from α to a'_i and a curve from a'_i to β) such that $\{P_i \mid i \in I\}$ defines a path order and for $i \neq j$, P_i intersects P_j if and only if $x_i \cap x_j \supset \{\alpha, \beta\}$. Obviously, the path order induced by $\{P_i \mid i \in I\}$ is equivalent to the given arc order. We demonstrate the procedure for the arc order given in § 1.4.2 by Fig. 3.

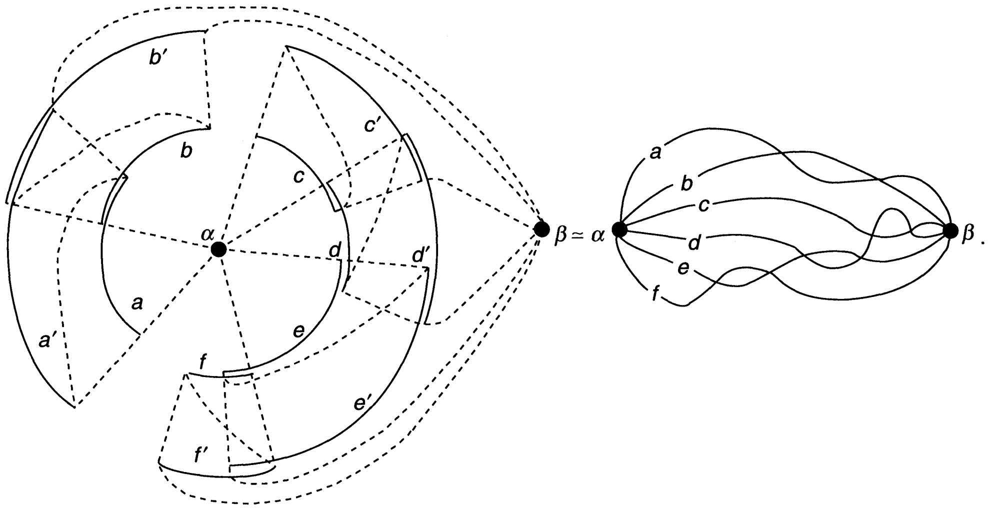
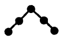


FIG. 3

If $X = \{x_i \mid i \in I\}$ is a set of arbitrary arcs, then we replace X by a set $\bar{X} = \{\bar{x}_i \mid i \in I\}$ of closed arcs that induce the same arc order. First, replace each point $\sigma \in S$, which is an endpoint of arcs $\{x_j \mid j \in J\}$ with $\varnothing \neq J \subset I$, by a closed interval of length 2^{-m}, where $m := \min \{j \in J\}$. (Since X is at most countable, we can assume without loss of generality that $I \subseteq \mathbb{N}$.) All other points on S remain unchanged. Now, for each $i \in I$, let $\bar{x}_i$ be the closure of x_i on the "extended" circle. It is easy to see, that for each $i, j \in I$, $x_i \cap x_j = \varnothing$ if and only if $\bar{x}_i \cap \bar{x}_j = \varnothing$. Hence $\bar{X}$ induces the same arc order as X and C is a path order according to the first part of this proof. □

3.7. On the other hand, not each path order can be represented as arc order since the poset-generated cyclic order derived from the partial order  (which is also a path order according to the theorem in § 3.9) cannot be represented as arc order.

3.8. Products of arc orders in general are not arc orders. For example, for a 4-circle C (i.e., a circle of length 4), $C \times C \times C$ cannot be represented as arc order. More generally, if C is a cyclic order in which the disjoint union of a 3-circle and a 4-circle is a cyclic suborder, then C is not an arc order.

3.9. The end of this chapter is dedicated to poset-generated cyclic orders (see § 1.4.1). We show that this class is a proper subclass in the class of path orders. Finally, we show the existence of arc orders that are not poset-generated and the converse.

THEOREM. *Finite poset-generated cyclic orders are path orders.*

Proof. Let $P = (X, R)$ be a partial order, and let $\mathscr{F} = \{P_i = (X, R_i) \mid i \in I\}$ be a family of chains with $P = \bigcap\mathscr{F}$. Represent $\mathscr{F}$ by a set of parallel and uniformly oriented directed lines L_i, $i \in I$, such that for $x, y \in X$, $x \neq y$, x proceeds y on L_i if and only if $(x', y) \in R_i (i \in I)$. Join all corresponding points on each two neighbouring lines. Join all points on the leftmost line with a new point α left of all lines and all points on the rightmost line with another point β right of all lines such that no two of these connections cross each other. Then the set of paths gained by this procedure induces a path order which coincides with the cyclic order C_P generated by P. □

3.10. Conversely, not each path order is induced by a poset since, if x, y, respectively y, z, are disjoint paths, x, z need not be disjoint. However, each path order with endpoints α, β lying on the boundary of the face surrounded by the paths and containing all paths is generated by a poset.

3.11. Poset-generated cyclic orders are obviously closed under taking subobjects but not with respect to products, which is shown by the example in § 3.5. The class of circular orders is the smallest class containing path orders and being closed with respect to products.

3.12. Not each poset-generated cyclic order is an arc order since the cyclic order derived from the disjoint union of chains of length four cannot be represented as arc order (cf. § 3.8). On the other hand, the arc order in Fig. 4 cannot be gained from a partial order.

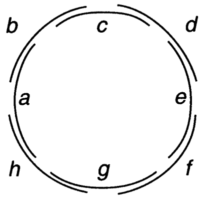


FIG. 4

4. Graphical cyclic orders.

4.1. Usually when working with an abstract structure it is convenient to have the possibility of drawing pictures to gain more insight. Unfortunately, there is no obvious way of drawing diagrams for a general cyclic order.

In this section we introduce a certain subclass of cyclic orders that can be represented by means of oriented graphs. Since the maximal circles in cyclic orders play an outstanding role in the process of transition from cyclic orders to oriented graphs, we first study their "behaviour" in cyclic orders. This analysis leads to a new characterization of cyclic orders by their maximal circles. A specification of this result then enables us to introduce the class of graphical cyclic orders which can be represented by a certain class of oriented graphs called Hasse diagrams (of cyclic orders). Triangulated oriented graphs having each edge in some directed triangle are Hasse diagrams.

At the end of this section we investigate the relation of this class to other classes of cyclic orders.

4.2. For definitions and notation used in the following, see § 1.6.

DEFINITION. Let C_1, C_2 be circles, $\{x, y\} \subseteq X(C_1) \cap X(C_2)$, $C_1(x, y) \neq \varnothing$, $C_2(y, x) \neq \varnothing$. Then

$$C \triangleq \{x\} C_1(x,y) \{y\} C_2(y,x)$$

denotes a circle with

$$X(C)=\{x,y\}\cup C_1(x,y)\cup C_2(y,x),$$

$$T(C)=T(C_1\,|\,\{x,y\}\cup C_1(x,y))\cup T(C_2\,|\,\{x,y\}\cup C_2(y,x))\cup\langle xzy\,|\,z\in C_1(x,y)\rangle$$

$$\cup\langle xyw\,|\,w\in C_2(y,x)\rangle\cup\langle xzw,ywz\,|\,z\in C_1(x,y),w\in C_2(y,x)\rangle.$$

4.3. Definition. For a cyclic order $C=(X,T)$ and $\mathcal{M}(C)=\{M_i\,|\,i\in I\}$ its set of all maximal circles, we define the following axiom of intersection.

(AI) For all $i\neq j$ and $x,y\in X(M_i)\cap X(M_j)$, $x\neq y$,
$M_i(x,y)\neq\varnothing$ iff $M_j(x,y)\neq\varnothing$,
$M_i(y,x)\neq\varnothing$ iff $M_j(y,x)\neq\varnothing$.
If $M_i(x,y)\neq\varnothing\neq M_i(y,x)$, then there are $k,l\in I$ with
$M_k\triangleq\{x\}M_i(x,y)\{y\}M_j(y,x)$,
$M_l\triangleq\{x\}M_j(x,y)\{y\}M_i(y,x)$.

4.4. Lemma. *Let $\mathcal{M}=\{M_i\,|\,i\in I\}$ be a family of circles satisfying* (AI). *Let no circle of $\mathcal{M}$ be contained in any other circle of $\mathcal{M}$. Then, for all $i\neq j$ and $x,y,z\in X(M_i)\cap X(M_j)$, $x\neq y\neq z\neq x$:*

(1) $M_i(x,y)\cap M_j(y,x)=\varnothing$, $M_i(y,x)\cap M_j(x,y)=\varnothing$.

(2) $(x,y,z)\in T(M_i)$ *if and only if* $(x,y,z)\in T(M_j)$.

4.5. Theorem. *For a cyclic order C, $\mathcal{M}(C)$ satisfies* (AI). *Conversely, let $\mathcal{M}=\{C_i=(X_i,T_i)\,|\,i\in I\}$ be a family of maximal circles satisfying* (AI). *Then there exists a unique cyclic order $C=(X,T)$ with $X=\bigcup_{i\in I}X_i$ and $\mathcal{M}(C)=\mathcal{M}$.*

Proof. $\Rightarrow$ Assume $M_i(x,y)\neq\varnothing$ and $M_j(x,y)=\varnothing$. Then there are $z\in M_i(x,y)$, $w\in M_j(y,x)$ with $xzy\in T(M_i)$, $xyw\in T(M_j)$. By (C1) and (C3), xzw, $ywz\in T$. Let $z'\in M_i(x,y)$, $z\neq z'$, and without loss of generality xzz', $xz'y$, $zz'y\in T(M_i)$. Then $xz'w$, ywz', $zz'w\in T$, i.e., $xzz'yw$ is a circle in C. Analogously, we get for $w'\in M_j(y,x)$, $w\neq w'$, that $xzyww'$ is a circle in C. Thus we get that there is a circle M_k in $\mathcal{M}(C)$ containing $\{x\}\cup M_i(x,y)\cup\{y\}\cup M_j(y,x)$. Since M_j is supposed to be maximal, this contradicts the assumption $M_j(x,y)=\varnothing$.

By the same argument, we see that $M_j(x,y)\neq\varnothing$ implies $M_i(x,y)\neq\varnothing$. By analogy, we prove that $M_i(y,x)\neq\varnothing$ if and only if $M_j(y,x)\neq\varnothing$.

Hence, if $M_i(x,y)\neq\varnothing\neq M_i(y,x)$, then there are circles M_k, $M_l\in\mathcal{M}(C)$ with $X(M_k)\supseteq\{x\}\cup M_i(x,y)\cup\{y\}\cup M_j(y,x)$, $X(M_1)\supseteq\{x\}\cup M_j(x,y)\cup\{y\}\cup M_i(y,x)$. Suppose $v\in X(M_k)-\{x\}\cup M_i(x,y)\cup\{y\}\cup M_j(y,x)$. Let $z\in M_i(y,x)$ be arbitrary. If possible, choose $u,w\in\{x\}\cup M_i(x,y)\cup\{y\}$ with $uvw\in T(M_k)$ such that $uwz\in T(M_i)$. Then uvz, $vwz\in T$, which contradicts the fact that M_i is maximal. Otherwise, we have $xyv\in T(M_k)$. Let $z\in M_j(x,y)$ be arbitrary. Then yvz, $zvx\in T$, which contradicts the fact that M_j is maximal. Therefore, $X(M_k)=\{x\}\cup M_i(x,y)\cup\{y\}\cup M_j(y,x)$. By analogy, we prove that $X(M_l)=\{x\}\cup M_j(x,y)\cup\{y\}\cup M_i(y,x)$. Obviously, M_k, M_l are "compatible" with M_i, M_j whence $T(M_k)$, $T(M_l)$ are as stated in the claim.

$\Leftarrow$ First we show that C is uniquely determined. Let $xyz\in T$. Then there exists a maximal circle $(Y,T\,|_Y)$ in C containing xyz. Since $\mathcal{M}(C)=\mathcal{M}$, there is some $i\in I$ such that $C_i=(Y,T\,|_Y)$, hence $xyz\in T_i$. Conversely, let $xyz\in T_i$ for some $i\in I$, then $xyz\in T$. Hence it remains to show that C exists. Define $T:=\bigcup_{i\in I}T_i$. Since each C_i is a cyclic order, (C1) holds for C. Obviously, (C2) follows from the second part of the lemma in § 4.4. Now let xyz, $xzw\in T$. Then there are $i,j\in I$ with $xyz\in T_i$, $xzw\in T_j$. If $i=j$, we are done. Otherwise we have $|C_i\cap C_j|\geqq 2$. Since $C_i(x,z)\neq\varnothing\neq C_j(x,z)$,

$C_j(z, x) \neq \emptyset \neq C_i(z, x)$ there is $k \in I$ with $C_k \triangleq \{x\} C_i(x, z) \{z\} C_j(z, x)$, hence $xyw \in T_k$ and $xyw \in T$. This proves (C3). □

4.6. For the definition of the class of graphical cyclic orders we first define the "axiom of intersection" for oriented graphs similar to the one in the definition in § 4.3 for maximal circles. (For definitions and notation concerning graphs we refer to § 1.4.4, respectively, [4].) Then we describe the transition from cyclic orders to oriented graphs and vice versa and mention some problems arising in this process. Finally, a characterization of oriented graphs that represent cyclic orders is given.

For an oriented graph G, let $\mathscr{M}(G)$ designate the *set of all chordless circuits in G*. For a circuit $G = (X, E)$ and $x, y, z \in X, x \neq y \neq z \neq x$, we write $x \xrightarrow{G} y \xrightarrow{G} z$ if and only if there is a directed path in G leading from x to y without traversing z and a directed path in G leading from y to z without traversing x. For a circuit $G = (X, E)$ and $x, y \in X, x \neq y$, let $G(x, y) \subset X$ designate the following set:

$$G(x,y) := \{z \in X \mid x \xrightarrow{G} z \xrightarrow{G} y\}.$$

4.7. DEFINITION. Let G_1, G_2 be chordless circuits, $x, y \in X(G_1) \cap X(G_2)$, $G_1(x, y) \neq \emptyset$, $G_2(y, x) \neq \emptyset$. Then

$$G \triangleq \{x\} G_1(x,y) \{y\} G_2(y,x)$$

denotes a chordless circuit with

$$X(G) = \{x,y\} \cup G_1(x,y) \cup G_2(y,x),$$
$$E(G) = E(G_1 \mid \{x,y\} \cup G_1(x,y)) \cup E(G_2 \mid \{x,y\} \cup G_2(y,x)).$$

4.8. DEFINITION. For an oriented graph $G = (X, E)$ and $\mathscr{M}(G) = \{M_i \mid i \in I\}$ its set of all chordless circuits, we define the following axiom of intersection:

(AI) For all $i \neq j$ and $x, y \in X(M_i) \cap X(M_j)$, $x \neq y$,
$M_i(x, y) \neq \emptyset$ iff $M_j(x, y) \neq \emptyset$,
$M_i(y, x) \neq \emptyset$ iff $M_j(y, x) \neq \emptyset$.
If $M_i(x, y) \neq \emptyset \neq M_i(y, x)$, then there are $k, l \in I$ with
$M_k \triangleq \{x\} M_i(x, y) \{y\} M_j(y, x)$,
$M_l \triangleq \{x\} M_j(x, y) \{y\} M_i(y, x)$.

4.9. LEMMA. *Let* $\mathscr{M} = \{M_i \mid i \in I\}$ *be a family of chordless circuits satisfying* (AI). *Then, for all* $i \neq j$ *and* $x, y, z \in X(M_i) \cap X(M_j)$, $x \neq y \neq z \neq x$:

(1) $M_i(x, y) \cap M_j(y, x) = \emptyset$, $M_i(y, x) \cap M_j(x, y) = \emptyset$.

(2) $x \xrightarrow{M_i} y \xrightarrow{M_j} z$ *if and only if* $x \xrightarrow{M_j} y \xrightarrow{M_j} z$.

4.10. DEFINITION. (i) For an oriented graph $G = (X, E)$ and $\mathscr{M}(G)$ its set of all chordless circuits, let $\mathscr{C}G = (X, T)$ be defined as follows. For each $x, y, z \in X, x \neq y \neq z \neq x$,

$$(x,y,z) \in T \quad \text{iff } x \xrightarrow{M} y \xrightarrow{M} z \text{ for some } M \in \mathscr{M}(G).$$

(ii) For a cyclic order $C = (X, T)$ and $\mathscr{M}(C)$ its set of all maximal circles, let $\mathscr{G}C = (X, E)$ be defined as follows. For each $x, y \in X, x \neq y$,

$(x, y) \in E$ iff $(x, y, z) \in T(M)$ for some $z \in X$, $M \in \mathscr{M}(C)$ such that $(x, u, y) \notin T(M)$ for each $u \in X$.

4.11. THEOREM. *For an oriented graph G,* $\mathscr{C}G$ *is a cyclic order if and only if the set of all chordless cycles* $\mathscr{M}(G)$ *satisfies* (AI).

Proof. Obviously, each chordless circuit of G is matched to a maximal circle of $\mathscr{C}G$. On the other hand, each maximal circuit of $\mathscr{C}G$ is induced by a chordless circuit of G. Hence, necessity follows from part one of the proof of the theorem in § 4.5.

Sufficiency. (C1) follows from the definition of the operation $\mathscr{C}$ and from the definition of $\cdot\!\to\!\cdot\!\to\!\cdot$. (C2) follows from the second part of the lemma in § 4.9. (C3) is proved by an argument similar to the "=" part of the proof of Theorem 4.5. □

4.12. PROPOSITION. *For a cyclic order C, $\mathscr{G}C$ is an oriented graph that in general does not satisfy* (AI).

Proof. We have to show that $\mathscr{G}C$ is an oriented graph. Suppose $(x, y), (y, x) \in E(\mathscr{G}C)$ for some $x, y \in X, x \neq y$. Then there are circles $M_i, M_j \in \mathscr{M}(C), M_i \neq M_j$, and elements $u, v \in X$ such that $(x, y, u) \in T(M_i)$, $(y, x, v) \in T(M_j)$ and $M_i(x, y) = \varnothing = M_j(y, x)$. Then, by (C3), $(x, v, u), (y, u, v) \in T$. Hence $xvyu$ is a circle in C. Thus there is a circle M_k in C containing $x, M_i(x, y), y, M_j(y, x)$, contradicting the fact that M_i, M_j are maximal. Hence, $\mathscr{G}C$ is an oriented graph.

To show that in general (AI) does not hold, let $C = (X, T)$ be the cyclic order on $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ with $T = \langle 1234, 3567, 1968 \rangle$ (Fig. 5). Then $\{1, 2, 3, 5, 6, 9\}$, $\{1, 8, 6, 7, 3, 4\}$ induce chordless circuits M, N in $\mathscr{G}C$ with $1 \xrightarrow{M} 3 \xrightarrow{M} 6$ and $1 \xrightarrow{N} 6 \xrightarrow{N} 3$. (AI) is obviously invalid, since $\{1\}N(1, 3)\{3\}M(3, 1)$ does not induce a chordless circuit in $\mathscr{G}C$. □

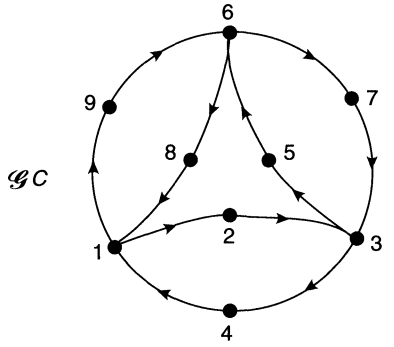


FIG. 5

4.13. DEFINITION. A cyclic order C is said to be *graphical* if and only if $C = \mathscr{C}\mathscr{G}C$, and an oriented graph G is called a *Hasse diagram* (of the cyclic order $\mathscr{C}G$) if and only if $G = \mathscr{G}\mathscr{C}G$ and $\mathscr{M}(G)$ satisfies (AI).

Obviously, if C is graphical, then $\mathscr{G}C$ satisfies (AI), and C is graphical if and only if $\mathscr{G}C$ is the Hasse diagram of C.

4.14. THEOREM. *An oriented graph G is a Hasse diagram if and only if $\mathscr{M}(G)$ satisfies* (AI) *and each edge of G is covered by a chordless circuit of G.*

Proof. Necessity is clear by the theorem in § 4.11.

Sufficiency. Since $G = \mathscr{G}\mathscr{C}G$ and by the definition of the operation $\mathscr{C}$, each edge of G belongs to a chordless circuit. (AI) follows from the fact, that $\mathscr{M}(\mathscr{C}G) = \mathscr{C}\mathscr{M}(G)$. □

The following facts are easily verified.

4.15. PROPOSITION. *Let $G = (X, E)$ be a triangulated oriented graph such that each edge $e \in E$ belongs to some directed triangle. Then G is a Hasse diagram.*

COROLLARY. *Each tournament with each edge in a directed triangle is a Hasse diagram.*

4.16. Unfortunately, there is no obvious characterization of graphical cyclic orders. For a cyclic order C to be graphical it must not contain a substructure like the one of the counterexample in the proposition of § 4.12. It is unclear, however, whether for graphical cyclic orders there exists a finite family of forbidden substructures.

4.17. PROPOSITION. *Let C, D be cyclic order, $C \leqq D$, and let D be graphical. Then C is also graphical.*

Graphical cyclic orders are not closed under products, which is demonstrated by the following example. Let $C_1 = 123$ and $C_2 = 123456$ be two circles that are obviously graphical. Each circle in $C_1 \times C_2$ has length three. Then $\mathscr{G}(C_1 \times C_2)$ contains, in particular, the (induced) subgraph in Fig. 6. Thus, $T(\mathscr{C}\mathscr{G}(C_1 \times C_2))$ contains the triples $(1, 1)$-$(2, 5)(3, 3)$ and $(1, 4)(2, 2)(3, 6)$, which cannot belong to $T(C_1 \times C_2)$. Hence $C_1 \times C_2 \neq \mathscr{C}\mathscr{G}(C_1 \times C_2)$ and $C_1 \times C_2$ is not graphical.

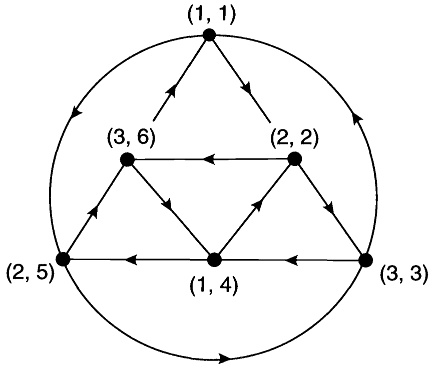


FIG. 6

4.18. Finally, we give a few remarks on the relation of the class of graphical orders to other classes:

—Circular orders are in general not graphical. Conversely, graphical cyclic orders are in general not circular.

—Arc orders are in general not graphical.

THEOREM. *Let P be a partial order of finite height. Then, C_P is graphical.*

Proof. Let $C = C_P$ be generated by $P = (X, R)$. The maximal circles of C are exactly the circuits generated by maximal chains in P. We have to show that $C = \mathscr{C}\mathscr{G}C$. Suppose there is a chordless circuit M in $\mathscr{G}C$ that is not induced by a maximal chain in P. Then M consists of "segments" of circuits that are induced by maximal chains. Hence, there exist $x, y, z \in X(M)$ with $x \overset{M}{\to} y \overset{M}{\to} z$, $(x, y), (z, y) \in R$ and $(x, z), (z, x) \notin R$. Furthermore, there exist chains $L_x = (X_x, R_x)$, $L_y = (X_y, R_y)$ in P with $X_x \cup X_y \subset X(M)$, $(u, x) \in R_x$ for all $u \in X_x$ and $(y, v) \in R_y$ for all $v \in X_y$. Since $(x, y) \in R$, there is a maximal chain $L = (X', R')$ that contains L_x and L_y: $X_x \cup X_y \subseteq X' \subset X(M)$ and $R_x \cup R_y \subset R'$. Since $z \notin X'$ and the definition of C, there exists a chord in M in contradiction to the definition of M. □

5. Dimension theory for circular orders.

5.1. In the theory of partial orders it is well known (cf., e.g., [33]) that for a poset P, the minimum number of chains whose intersection is P coincides with the minimum number of chains such that P is isomorphic to a subposet of the product of these chains. This number is called the (Dushnik-Miller) dimension of P, denoted by dim P. Here we similarly introduce two concepts of dimension for circular orders (in accordance with the theorems in §§ 2.2 and 2.5), which turn out not to coincide. However, both dimensions agree up to a factor two.

5.2. Definition. Let $C = (X, T)$ be a circular order. Then the *intersection dimension* and the *product dimension* of C are defined by

$$\operatorname{idim} C := \min \{ |I| \mid C = \bigcap_{i\in I} C_i, C_i \text{ circle } (i \in I)\},$$

$$\operatorname{pdim} C := \min \{ |I| \mid C \leqq \mathop{\times}_{i\in I} C_i, C_i \text{ circle } (i \in I)\}.$$

5.3. Theorem. *For a circular order C, the intersection dimension and the product dimension satisfy*

$$\operatorname{pdim} C \leqq \operatorname{idim} C \leqq 2 \cdot \operatorname{pdim} C.$$

Moreover, both equalities hold infinitely often.

Proof. Since each representation of C as intersection of circles provides an embedding into the product of these circles, we have pdim $C \leqq$ idim C for any circular order C. Moreover, for each $n \in \mathbb{N}$, there exists a circular order C_n with pdim $C_n = n =$ idim C_n; e.g., take a circle of length n and one isolated point (cf. § 5.8). On the other hand, by the theorem in § 2.2, each product of n circles can be represented as the intersection of $2n$ circles, proving idim $C \leqq 2 \cdot$ pdim C for any circular order C. Again, for each $n \in \mathbb{N}$, there exists a circular order C'_n with idim $C'_n = 2n = 2 \cdot$ pdim C'_n; e.g., take a product of n circles of length $2n$ (cf. the proposition in § 5.9). □

5.4. The rest of this section is dedicated to the estimation of intersection dimension of circular orders.

Proposition. *For a circular order $C = (X, T)$ which is not a circle,*

$$\operatorname{idim} C \leqq 2 \cdot \left(\binom{|X|}{3} - \frac{|T|}{3} \right).$$

Proof. The expression on the right side is an upper bound for the cardinality of a family $\mathscr{F}$ of total extensions of C such that for each $x, y, z \in X$, $x \neq y \neq z \neq x$, with $(x, y, z), (z, y, x) \notin T$, there are circles $C_1, C_2 \in \mathscr{F}$ with $(x, y, z) \in T(C_1)$ and $(z, y, x) \in T(C_2)$. □

Corollary. *For a circular order C,* idim $C \leqq \frac{1}{3} \cdot |X(C)|^3$.

But we do not even know whether there exists a class K of circular orders and a constant $\kappa > 1$ such that idim $C > \kappa \cdot |X(C)|$ for each $C \in K$.

5.5. Clearly, the class of circular orders is hereditary with respect to both dimensions, i.e., for $C_1 \leqq C_2$, idim $C_1 \leqq$ idim C_2 and pdim $C_1 \leqq$ pdim C_2.

5.6. Proposition. *Let C_1, C_2 be circular orders. Then*

$$\operatorname{idim} C_1 \times C_2 \leqq 4 \cdot \operatorname{idim} C_1 \cdot \operatorname{idim} C_2.$$

Proof. This follows from the theorems in §§ 2.2 and 2.3. □

5.7. Next we give some concrete values for the intersection dimension of finite cyclic orders.

Proposition. *Let $C = (X, T)$ be a finite cyclic order, $\mathscr{M}(C) = \{M, N\}$ be its set of maximal circles, $X(M) \cap X(N) \neq \emptyset$, $X(C) = X(M) \cup X(N)$, and either $M(x, y) = N(x, y)$ or $M(y, x) = N(y, x)$ for each $x, y \in X(M) \cap X(N)$, $x \neq y$, if $|M \cap N| > 1$. Then C is circular and*

$$\operatorname{idim} C = |M| + |N| - 2 \cdot |M \cap N|.$$

Proof. Without loss of generality let $M = x_1 x_2 \cdots x_m$ (cf. 1.9), $N = y_1 y_2 \cdots y_n$ and $M \cap N = x_1 x_2 \cdots x_k = y_1 y_2 \cdots y_k$ with $k < m, n$. This means that we assume $x_i =$

y_i, $i = 1, \cdots, k$ and $T = \langle (x_\alpha, x_\beta, x_\gamma) \,|\, 1 \leqq \alpha < \beta < \gamma \leqq m \rangle \cup \langle (y_\alpha, y_\beta, y_\gamma) \,|\, 1 \leqq \alpha < \beta < \gamma \leqq n \rangle$. Since particularly, T does not contain triples of the form

$$x_{\alpha-1}x_\alpha y_\beta, \quad x_{\alpha-1}y_\beta x_\alpha, \quad y_{\beta-1}y_\beta x_\alpha, \quad y_{\beta-1}x_\alpha x_\beta,$$

$$x_m x_1 y_\beta, \quad x_m y_\beta x_1, \quad y_n y_1 x_\alpha, \quad y_n x_\alpha y_1$$

for $k < \alpha \leqq m$, $k < \beta \leqq n$, but

$$x_\alpha x_\beta x_\gamma, \qquad x_\alpha x_\beta y_\delta \in T$$

for $1 \leqq \alpha < \beta \leqq k < \gamma \leqq m$, $k < \delta \leqq n$, we have

$$\text{idim } C \geqq m + n - 2k.$$

(For each of the above triples that are not in T there has to be a circle containing it that is a total extension of C. This proves the lower bound.)

On the other hand, we have the following "encoding." Let $C_i = (X, T_i)$, $i = 1, 2, \cdots, m + n - 2k$, be given by

$$C_1 = x_1 x_2 \cdots x_k x_{k+1} \cdots x_m y_{k+1} y_{k+2} \cdots y_n,$$

$$C_2 = x_1 x_2 \cdots x_k y_{k+1} x_{k+1} x_{k+2} \cdots x_m y_{k+2} y_{k+3} \cdots y_n,$$

$$C_3 = x_1 x_2 \cdots x_k y_{k+1} y_{k+2} x_{k+1} x_{k+2} \cdots x_m y_{k+3} y_{k+4} \cdots y_n,$$

$$\vdots$$

$$C_{n-k} = x_1 x_2 \cdots x_k y_{k+1} y_{k+2} \cdots y_{n-1} x_{k+1} x_{k+2} \cdots x_m y_n,$$

$$C_{n-k+1} = y_1 y_2 \cdots y_k y_{k+1} \cdots y_n x_{k+1} x_{k+2} \cdots x_m,$$

$$C_{n-k+2} = y_1 y_2 \cdots y_k x_{k+1} y_{k+1} y_{k+2} \cdots y_n x_{k+2} x_{k+3} \cdots x_m,$$

$$C_{n-k+3} = y_1 y_2 \cdots y_k x_{k+1} x_{k+2} y_{k+1} y_{k+2} \cdots y_n x_{k+3} x_{k+4} \cdots x_m,$$

$$\vdots$$

$$C_{n-k+m-k} = y_1 y_2 \cdots y_k x_{k+1} x_{k+2} \cdots x_{m-1} y_{k+1} y_{k+2} \cdots y_n x_m.$$

Obviously, $T(C) \subseteq T(\cap C_i)$. Let $(x, y, z), (z, y, x) \notin T(C)$. Then (x, y, z) is one of the following triples:

(a) $xyz = x_\alpha x_\beta y_\gamma$ where $k < \beta < m$, $\alpha < \beta$ and $k < \gamma \leqq n$:

$$xyz \in T_{n-k+\beta-k+1} \quad \text{and} \quad zyx \in T_{n-k+\beta-k}.$$

(b) $xyz = x_\alpha x_m y_\gamma$ where $\alpha < m$ and $k < \gamma \leqq n$:

$$xyz \in T_1 \quad \text{and} \quad zyx \in T_{n-k+1}.$$

(c) $xyz = y_\alpha y_\beta x_\gamma$ where $k < \beta < n$, $\alpha < \beta$ and $k < \gamma \leqq m$:

$$xyz \in T_{\beta-k+1} \quad \text{and} \quad zyx \in T_{\beta-k}.$$

(d) $xyz = y_\alpha y_n x_\gamma$ where $\alpha < n$ and $k < \gamma \leqq m$:

$$xyz \in T_{n-k+1} \quad \text{and} \quad zyx \in T_1.$$

Hence, in each case, $(x, y, z), (z, y, x) \notin T(\cap_{i=1}^{m+n-2k} C_i)$, showing that

$$\text{idim } C \leqq m + n - 2k. \qquad \square$$

5.8. The proof of the next proposition is analogous to that of § 5.7 and therefore is omitted.

PROPOSITION. *Let the finite circular order $C_1 = (X_1, T)$ be a cyclic suborder of $C_2 = (X_1 \dot{\cup} X_2, T)$, i.e., C_2 arises from C_1 by adding isolated points. Then C_2 is also circular, and*

$$\operatorname{idim} C_2 = \operatorname{idim}(C_1 \dot{\cup} \{x\}).$$

In particular, if C_1 is a circle, then

$$\operatorname{idim} C_2 = |C_1|.$$

5.9. PROPOSITION. *Let $\mathscr{F} = \{C_i = (X_i, T_i) \mid i = 1, 2, \cdots, n\}$ be a family of circles. Then $\times\mathscr{F}$ is circular and*

$$\min_{1 \leqq i \leqq n} \{|X_i|, 2n\} \leqq \operatorname{idim} \times\mathscr{F} \leqq 2n.$$

Proof. The upper bound follows from the theorem in § 2.2.

Let $m := \min_{1 \leqq i \leqq n}\{|X_i|, 2n\}$ and $C_i = x_{i1}x_{i2}\cdots x_{ik_i}$ with $k_i \geqq m$, $i \in I = \{1, 2, \cdots, n\}$. For $C = (x_{i1})_{i \in I}(x_{i2})_{i \in I}\cdots(x_{im})_{i \in I}$ a circle of length m in $\times\mathscr{F}$ and

$$x = (x_{11}, x_{32}, x_{53}, \cdots, x_{(2p-1)p}, x_{(2p-1)(p+1)}, \cdots, x_{(2p-1)n}),$$

where $m \in \{2p - 1, 2p\}$, we have $\operatorname{idim} \times\mathscr{F} \geqq m$ according to the proposition in § 5.8 since x is isolated in $C \dot{\cup} \{x\}$. □

Remark. Let $C = xyz$ be a circle of length 3 and $n \geqq 2$. Then $C^n \simeq 3^{n-1} \cdot C$ (i.e., the disjoint union of 3^{n-1} copies of C) and $\operatorname{idim} C^n = 3$. However, we do not know the intersection dimension of the nth power of a circle of length $m > 3$.

5.10. We note that the disjoint union of a finite number of finite circles is circular. Moreover, we have the following proposition.

PROPOSITION. *Let $C = (X, T)$ be a cyclic order with $X = \dot{\bigcup}_{i \in I} X_i$ and $T = \dot{\bigcup}_{i \in I} T_i$, such that each $C_i = (X_i, T_i)$ is a finite circle in C, $|I| > 1$ and finite. Then C is circular and*

$$\operatorname{idim} C = \max_{i \in I} |C_i|.$$

Proof. Let I be a linearly ordered index set, $m := \max_{i \in I} |C_i|$ and $C_i = x^i_1 x^i_2 \cdots x^i_{n_i}$ $(i \in I)$. Define total extensions $D_j = (X, U_j)$, $j = 1, 2, \cdots, m$, in the following way.

Let $(x, y, z) = (x^r_u, x^s_v, x^t_w) \in U_j$ if and only if

```
either $(x,y,z) \in T_i$ for some $i \in I$
or if $j=1$, either $r=s\neq t$, $u<v$ or $r\neq s=t$, $v<w$ or $r>s>t$,
   if $j>1$, either $r=s\neq t$ and
                 if $j<n_r$, either $j\leqq u<v$
                             or $u<v<j$ or $u\geqq j$, $v<j$,
                 if $j\geqq n_r$, either $u=n_r$, $v<n_r$
                             or $u<v<n_r$
             or $r\neq s=t$ and
                 if $j<n_s$, either $j\leqq v<w$
                             or $v<w<j$ or $v\geqq j$, $w<j$,
                 if $j\geqq n_s$, either $v=n_s$, $w<n_s$
                             or $v<w<n_s$
             or $r<s<t$.
```

Then we claim that $\bigcap_{j=1}^m D_j = (X, \bigcap_{j=1}^m U_j) = (X, T) = C$. To see that, let $x, y, z \in X$, $x \neq y \neq z \neq x$, $x \in X_r$, $y \in X_s$, $z \in X_t$. Then

(a) $r = s = t$: $(x, y, z) \in T$ if and only if $(x, y, z) \in U_j$, $j = 1, 2, \cdots, m$.

(b) $r = s \neq t$: Then $(x, y, z) \notin T$ and, since each cyclic permutation of each $C_i (i \in I)$ is contained in a least one $D_j (j \in \{1, 2, \cdots, m\})$, there are indices $j_1, j_2 \in \{1, 2, \cdots, m\}$ such that $(x, y, z) \in U_{j_1}$ and $(z, y, x) \in U_{j_2}$.

(c) $r \neq s = t$: analogous (b).

(d) $r \neq s \neq t$: Then $(x, y, z) \notin T$ and there is some $j \in \{2, 3, \cdots, m\}$ such that $(x, y, z), (z, y, x) \notin U_1 \cap U_j$.

Hence, we have idim $C \leqq m$.

Let $C_i = x_1^i x_2^i \cdots x_m^i$ be a circle in C with maximum number of elements, and let x be any element of any other circle. Since all triples (x, x_j^i, x_{j+1}^i) $(j = 1, 2, \cdots, m-1)$, (x, x_m^i, x_1^i) are not in T, it is idim $C \geqq m$ according to the proposition in § 5.8. □

5.11. Proposition. *Let $C = (X, T)$ be a finite path-order, and let*

$$n := |\{\{x,y\} \subset X \mid x \textit{ and } y \textit{ are intersecting paths}\}| + 1.$$

Then, idim $C \leqq n$.

Proof. Apply the theorem in § 3.3. □

5.12. For the rest of this section we would like to give some estimates for poset-generated cyclic orders.

Let $P = (X, R)$ be a poset. Then $J(P)$ designates the set of all *incomparable* pairs in P:

$$J(P) := \{\{x,y\} \mid (x,y),(y,x) \notin R\}.$$

Proposition. *Let P be a finite poset which is not a chain and let the cyclic order C be generated by P, $C = C_P$. Put $n := |P| \geqq 2$. Then,*

$$\text{idim } C_P \leqq 2 \cdot |J(P)| \leqq n(n-1).$$

Proof. For each $x, y, z \in X = X(P) = X(C_P)$, $x \neq y \neq z \neq x$, $(x, y, z) \notin T(C_P)$ if and only if either $(z, y, x) \in T(C_P)$ or two of the elements are incomparable in P. For each $\{x, y\} \in J(P)$, there exist linear extensions $P_{xy} = (X, R_{xy})$ and $P_{yx} = (X, R_{yx})$ of P with $(x, y) \in R_{xy}$ and $(y, z) \in R_{xy}$ for each $(x, z) \in R$, respectively, $(y, x) \in R_{yx}$ and $(z, y) \in R_{yx}$ for each $(z, x) \in R$. Then obviously,

$$C_P = \bigcap_{\{x,y\} \in J(P)} C_{P_{xy}} \cap C_{P_{yx}}. \qquad \square$$

If C_P has more than three elements, then for $n = |P|$,

$$\text{idim } C_P \leqq n(n-1) - 6.$$

5.13. An immediate consequence of the proposition in § 5.8 is the following proposition.

Proposition. *Let the cyclic order $C = C_P$ be generated by a finite poset P. Let P' be a chain in P and let $x \in X(P) - X(P')$ satisfy either* $(x, \sup P') \notin R(P)$ *if* $(\inf P', x) \in R(P)$ *or* $(\inf P', x) \notin R(P)$ *if* $(x, \sup P') \in R(P)$. *Then,*

$$\text{idim } C_P \geqq |P'|.$$

5.14. By § 5.13, there exist for each $n \geqq 3$ posets P_n with $|X(P_n)| = n + 1$, dim $P_n = 2$ and idim $C_{P_n} = n$. Even for each $k \geqq 2$, $n \geqq 3$, there exist posets $P_{n,k}$ with

$$\dim P_{n,k} = k \text{ and idim } C_{P_{n,k}} > k(n-1);$$

e.g., put $P_{n,k} = \mathbf{n}^k + \mathbf{1}$ where $\mathbf{n}$ designates the chain of length n. Since for each $n \in \mathbb{N}$ there exist posets P_n of height two and dim $P_n = n$, idim $C_{P_n} = 2$ (since then $T(C_{P_n}) = \varnothing$) (e.g., take the poset P_n determined by the family of all 1-element and $(n-1)$-element subsets of an n-element set ordered by inclusion), dim P is in general not bounded by idim C_P. But we conjecture that there is a constant κ such that $\dim P \leqq \kappa \cdot \operatorname{idim} C_P$ for each poset P with the property that each maximal chain in P has length at least three.

REFERENCES

[1] P. Alles, *Total extendability and circularity of cyclic orders are not finitely axiomatizable*, Preprint-Nr. 981, Fachbereich Mathematik, Technische Hochschule, Darmstadt, April 1986.

[2] P. Alles, J. Nešetřil, and S. Poljak, *Extendability, dimensions and diagrams of cyclic orders*, Preprint-Nr. 944, Fachbereich Mathematik, Technische Hochschule, Darmstadt, October 1985.

[3] J. L. Bell and A. B. Slomson, *Models and Ultraproducts*, Amsterdam, London, 1969.

[4] B. Bollobás, *Graph Theory*, Springer-Verlag, New York, 1979.

[5] E. Čech, *Point Sets*, Academia Prague, 1969.

[6] I. Chajda and V. Novák, *On extensions of cyclic orders*, Časopis Pěst. Mat. 110 (1985), pp. 116–121.

[7] B. Dushnik and E. W. Miller, *Partially ordered sets*, Amer. J. Math., 63 (1941), pp. 600–610.

[8] Z. Galil and N. Meggido, *Cyclic Ordering is* NP-*complete*, Theoret. Comput. Sci., 5 (1977), pp. 179–182.

[9] E. Harzheim, *Ein Endlichkeitssatz über die Dimension teilweise geordneter Mengen*, Math. Nachr., 46 (1970), pp. 183–188.

[10] H. Herrlich and G. E. Strecker, *Category Theory*, Boston, 1973.

[11] A. Heyting, *Axiomatic Projective Geometry*, New York, 1963.

[12] E. V. Huntington, *A set of independent postulates for cyclic order*, Proc. Nat. Acad. Sci. U.S.A., 2 (1916), pp. 630–631.

[13] ———, *Set of completely independent postulates for cyclic order*, Proc. Nat. Acad. Sci. U.S.A., 10 (1924), pp. 74–78.

[14] E. V. Huntington and J. R. Kline, *Sets of independent postulates for betweenness*, Trans. Amer. Math. Soc., 18 (1917), pp. 301–325.

[15] L. Lovász, J. Nešetřil, and A. Pultr, *On a product dimension of graphs*, J. Combin. Theory Ser. B, 29 (1980), pp. 47–67.

[16] W. Magnus, A. Karrass, and D. Solitar, *Combinatorial Group Theory*, New York, 1966.

[17] N. Meggido, *Partial and complete cyclic orders*, Bull. Amer. Math. Soc., 82 (1976), pp. 274–276.

[18] H. Meschkowski, *Grundlagen der euklidischen Geometrie*, Mannheim, 1966.

[19] G. Müller, *Lineare und zyklische Ordnung*, Praxis Math., 16 (1974), pp. 261–269.

[20] J. Nešetřil and A. Pultr, *A Dushnik-Miller Type Dimension of Graphs and its Complexity*, Lecture Notes in Computer Science, Vol. 56, Springer-Verlag, Berlin, New York, 1977, pp. 482–493.

[21] J. Nešetřil and V. Rödl, *A simple proof of the Galvin-Ramsey property of the class of all finite graphs and a dimension of a graph*, Discrete Math. 23 (1978), pp. 49–55.

[22] V. Novák, *Cyclically ordered sets*, Czech. Math. J., 32 (1982), pp. 460–473.

[23] ———, *Cuts in cyclically ordered sets*, Czech. Math. J., 34 (1984), pp. 322–333.

[24] ———, *On some minimal problem*, Arch. Math., (1984), pp. 95–100.

[25] ———, *Operations on cyclically ordered sets*, Arch. Math., (1984), pp. 133–140.

[26] V. Novák and M. Novotný, *On determination of a cyclic order*, Czech. Math. J., 33 (1983), pp. 555–563.

[27] ———, *Dimension theory for cyclically and cocyclically ordered sets*, Czech. Math. J., 33 (1983), pp. 647–653.

[28] ———, *On a power of cyclically ordered sets*, Časopis Pěst. Mat., 109 (1984), pp. 421–424.

[29] ———, *Universal cyclically ordered sets*, Czech. Math. J., 35 (1985), pp. 158–161.

[30] O. Ore, *Theory of Graphs*, Providence, 1962.

[31] R. Rado, *Axiomatic treatment of rank in infinite sets*, Canad. J. Math., 1 (1949), pp. 337–343.

[32] L. Rédei, *Begründung der euklidischen und nichteuklidischen Geometrien*, Budapest, 1965.

[33] I. Rival, ed., *Ordered Sets*, Reidel, Dordrecht, 1982.

[34] G. Sabidussi, *Subdirect representations of graphs*, Colloq. Math. Soc. János Bolyai, 10 (1975), pp. 1199–1226.

[35] K. E. Stoffers, *Scheduling of traffic lights—a new approach*, Transportation Research, 2 (1968), pp. 199–234.

[36] E. Szpilrajn, *Sur l'extension de l'ordre partiel*, Fund. Math., 16 (1930), pp. 386–389.

[37] A. Tucker, *Characterizing circular-arc graphs*, Bull. Amer. Math. Soc., 76 (1970), pp. 1257–1260.

[38] ———, *Matrix characterizations of circular-arc graphs*, Pacific J. Math., 39 (1971), pp. 535–545.

[39] P. Alles, *Erweiterungen, Diagramme und Dimension zyklischer Ordnungen*, Technische Hochschule, Darmstadt, Ph.D. thesis, June 1986.

[40] M. Hall, Jr., *Combinatorial Theory*, Blaisdell, New York, 1967.

SIAM J. DISC. MATH.
Vol. 4, No. 4, pp. 472-480, November 1991

FAULT TOLERANT SORTING NETWORKS*

SHAY ASSAF† AND ELI UPFAL‡

Abstract. A general technique for enhancing the reliability of sorting networks and other comparator based networks is presented. The technique converts any network that uses unreliable comparators to a fault tolerant network that produces the correct output with overwhelming probability, even if each comparator is faulty with some probability smaller than $\frac{1}{2}$, independent of the other comparators. The depth of the fault tolerant network is only a constant times the depth of the original network; the width of the network is increased by a logarithmic factor.

Key words. sorting network, fault tolerant computing

AMS(MOS) subject classifications. 68P10, 68R05, 68Q25

1. Introduction. One of the major problems in large scale systems is the inevitable presence of fault elements. The larger the system, the larger the probability that some fraction of the system will fail to operate correctly. One way to overcome this difficulty is to design algorithms that perform correctly in the presence of a significant number of faulty components.

This paper presents a general technique for enhancing the reliability of networks built of unreliable comparators. We demonstrate the novel technique through the construction of fault tolerant sorting networks. We comment on applications of our technique to other comparator based networks, such as merging and selection networks, in the last section.

A comparator is a 2-input, 2-output device capable of sorting two elements. A sorting network consists of a collection of registers that pass through a number of levels of comparators. Registers store input elements we wish to sort and comparators connect pairs of registers. When two registers pass through a comparator, the two values entering via the registers exit in sorted order. No more than one comparator is connected to any register per level; the number of levels is the depth of the network. A network is a sorting network if any set of elements entering the first level of the network exits in sorted order in the last level. An asymptotically optimal $O(\log N)$[1] depth network was first given by Ajtai, Komlos, and Szemeredi [AKS83].

Yao and Yao [YY85] were the first to study sorting networks with stochastic faults. They assume a weak model of faults in which faulty comparators directly output the inputs without comparing them. A comparator in that model never changes the order of a pair that is already sorted. A deterministic version of this fault model was studied by Rudolph [Rud85] and Schimmer and Starke [SS88].

In the current work, we consider a stronger fault model in which a faulty comparator outputs the two input values in an arbitrary order (or even outputs one of the input values in both outputs). Such faulty comparators can destroy the correct order among a list that is already sorted. Our goal is to construct a fault tolerant network for the strong model of faults that computes the correct sorted list with high probability, even if each comparator is faulty, with some fixed probability smaller than $\frac{1}{2}$, independent of the

* Received by the editors March 28, 1990; accepted for publication (in revised form) August 10, 1990.

† Department of Applied Mathematics, The Weizmann Institute, Rehovot, Israel.

‡ IBM Almaden Research Center, San Jose, California, and Department of Applied Mathematics, The Weizmann Institute of Science, Rehovot, Israel. Work at the Weizmann Institute was supported in part by a Bat-Sheva de Rothschild Award and by a Revson Career Development Award.

[1] All logarithms in this paper are taken to base 2.

other comparators. In other words, we require the fault tolerant sorting network to compute, with high probability, the correct output even if a constant fraction of its comparators, chosen randomly, are faulty.

Sorting networks usually use N registers, one for each input. The following argument shows that no N register network can achieve our goal. Let X_1 denote the register that outputs the smallest element in the list. Clearly X_1 is compared at least once in the network; let $\tilde{X}$ denote the last register to which X_1 is compared. If there are no more than N registers in the network, either X_1 or $\tilde{X}$ does not store the minimum value when they are compared. If that comparator is faulty, the network does not output the correct order. Thus the failure probability of a fault tolerant network that sorts N elements with N registers cannot be smaller than the failure probability of each individual comparator.

To obtain a smaller failure probability we allow the network to use more than N registers. The network has N marked registers that receive the input elements and output the sorted list; the other registers are for internal use through the intermediate computation. All registers store only copies of the input values. To use the extra registers, the network can copy values from one register to another. We measure the width of a network by the maximum number of registers used by the network in one level.

The problem of noisy transmission of data has been extensively studied and can be handled efficiently by error correction codes [MS77]. In the current work, we concentrate on faulty comparators and assume no faults in the copying and transmission of values in the network.

Our main result is a general technique for converting any sorting network to a fault tolerant sorting network. Given a depth $O(d)$ network for sorting N elements, our technique constructs a sorting network of depth $O(d)$ and width $O(N \log N)$ that computes the correct output with probability $1 - 1/N$, even if each comparator is faulty with some fixed probability smaller than $\frac{1}{2}$, independent of the other comparators.

The study of constructing reliable systems from unreliable components goes back to the work of von Neumann [VN56]. See also Pippenger [Pip85] for recent results. In these works, the computing power of Boolean circuits is used in order to improve the reliability of circuits with noisy gates. Our new result shows that using comparators alone, we can achieve similar results in the context of comparator based networks. The crux of our method is an expander based component that uses comparators and replaces the "counting component" used in the construction of reliable Boolean circuits with noisy gates. While the comparator based component cannot count, we show that it is powerful enough for enhancing the reliability of the sorting network.

2.1. The fault tolerant sorting network: overview of the network. The network has two main parts. The first part simulates the comparators of the original sorting network. Through this part, each register X_i of the N registers of the original network is simulated by a set $\bar{X}_i = \{X_i^1, \cdots, X_i^m\}$ of $m = \log N$ registers.

We say that a register X_i^ℓ stores the correct value at a given stage of the computation if it stores the same value as the register X_i of the original network at the corresponding stage of the original network. For the correctness of the computation we require that at each step of the computation all but a fixed fraction of each set $\bar{X}_i$ store the correct value. When register X_i is compared with X_j in the original network, our network compares each X_i^ℓ with X_j^ℓ, $\ell = 1, \cdots, m$. Since a fraction of the comparators might be faulty, each comparison might increase the fraction of wrong values. The heart of our simulation is a constant depth *majority-expander* component that reinforces the majority in each set $\bar{X}_i$ after simulating each comparison of the original network.

The second part of the network generates the output. The input to this part are N sets, each with $m = \log N$ values. With high probability, each set contains only a small fraction of values that are not the correct value for that set. The goal of this part of the network is to reduce the m values of each set to one correct value. This goal is achieved by successive use of a *majority-preserver* component.

2.2. Basic components. The majority-expander and the majority-preserver components are built of two basic components: the *enforced comparators* and the *stochastic halver*.

DEFINITION. A *$\tilde{p}$-enforced comparator* is a construction with two inputs and two outputs that computes the maximum and minimum of the two inputs with error probability bounded by $\tilde{p}$.

THEOREM 1. *For every $0 < \tilde{p} < p < \frac{1}{2}$ there is a construction of depth and width $3(2/(3-6p))^{1/(\log 23/18)} \log 2/\tilde{p}$ that is a $\tilde{p}$-enforced comparator even if each of its comparators is faulty with some probability bounded by $p < \frac{1}{2}$.*

Proof. We compute the maximum of X and Y by iterative use of the operator $\max[\max[X, Y], \max[X, Y]]$.

Assume that we have a construction M_i that has depth D_i, width W_i, and computes the maximum of two elements with failure probability bounded by p_i. Plugging M_i into the above operator (see Fig. 1) gives a component M_{i+1} that has depth $2D_i$ and width $2W_i$. Noting that M_{i+1} fails if and only if at least two of its three M_i components fail, we get that it computes the maximum of two elements with failure probability

$$(1) \qquad p_{i+1} \leqq \binom{3}{2}(1-p_i)p_i^2 + p_i^3 = 3p_i^2 - 2p_i^3.$$

The failure probability of the basic comparators, used as M_0, is bounded by p. Thus $p_0 = p$ and $D_0 = W_0 = 1$. Let $\varepsilon_i = \frac{1}{2} - p_i$. By plugging $p_i = \frac{1}{2} - \varepsilon_i$ into (1) we get $\varepsilon_{i+1} = \varepsilon_i(\frac{3}{2} - 2\varepsilon_i^2)$. If $p_i \geqq \frac{1}{6}$, then $\varepsilon_i \leqq \frac{2}{6}$, and $\varepsilon_{i+1} \geqq \varepsilon_i(\frac{3}{2} - 2(\frac{2}{6})^2) \geqq 23/18\varepsilon_i$. Thus if $p_0 \geqq \frac{1}{6}$, after $t_1 = 1/\log 23/18 \log 1/(3(1/2-p))$ iterations, $\varepsilon_{t_1} \geqq \frac{2}{6}$ or $p_{t_1} \leqq \frac{1}{6}$.

If $p_i \leqq \frac{1}{6}$ then $p_{j+t_1} \leqq 3^{2^j - 1}(\frac{1}{6})^{2^j} \leqq (\frac{1}{2})^{2^j}$. Let $t_2 = \log\log(2/\tilde{p})$ then $p_{t_1+t_2} \leqq \tilde{p}/2$.

Using a similar construction, we can compute the minimum of X and Y with probability $\tilde{p}/2$. Thus we get a construction that computes the maximum and the minimum of X and Y with error probability $\tilde{p}$. The depth and width of the whole construction is bounded by

$$2^{t_1+t_2+1} = 2\left(\frac{2}{3-6p}\right)^{1/(\log 23/18)} \log\frac{2}{\tilde{p}}.$$

To use this construction we need $2^{t_1+t_2}$ copies of each of the two inputs. These copies can be constructed in depth $t_1 + t_2$. □

The construction of the majority-expander is based on explicit construction of expander graphs. let $G = (A, B, E)$ denote a bipartite graph, and let $\Gamma(X)$ denote the set

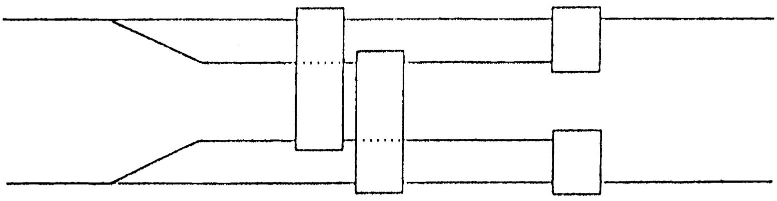


FIG. 1. *A p_{i+1}-enforced comparator made up of p_i-enforced comparators.*

of neighbors of a set of vertices X, i.e.,

$$\Gamma(X) = \{y \mid (x, y) \in E \text{ for some } x \in X\}.$$

DEFINITION. $G = (A, B, E)$ is an (α, β, m, d)-*expander* if $|A| = |B| = m$, the degree of every vertex in G is d, and for every set of vertices X such that $|X| \leqq \alpha m$, $|\Gamma(X)| \geqq \beta |X|$.

LEMMA 2. *For any $\alpha < 1$ and β such that $\alpha\beta < 1$ there is an explicit construction of an (α, β, m, d)-expander with $d \leqq (8\beta(1 - \alpha))/(1 - \alpha\beta)$.*

Proof. The construction in [LPS86] gives a d-regular bipartite graph with second largest eigenvalue bounded by $2\sqrt{d}$ for any $d = p + 1$, p prime. By [Alo86], if the second eigenvalue is bounded by $2\sqrt{d}$ the expansion of each set of size bounded by αm is at least $\beta \geqq d/((d - 4)\alpha + 4)$. Thus, $d = (4\beta(1 - \alpha))/(1 - \alpha\beta)$ is sufficient for expansion β. To guarantee the existence of d of the form $p + 1$, p prime, it suffices by Bertrand's postulate [HW75, p. 343] to take twice this value. □

DEFINITION. A set $\bar{X}$ ε-*represents* a value $\mathscr{V}$, if at least $(1 - \varepsilon)|\bar{X}|$ registers in $\bar{X}$ store the value $\mathscr{V}$.

DEFINITION. An $(\ell, \eta, \delta, \tilde{p})$-*stochastic halver* is a construction with 2ℓ inputs and two sets of ℓ outputs. If the input set $\theta_1 + \theta_2$-represents a value $\mathscr{V}$, $\theta_1, \theta_2 < \frac{1}{2}$, such that no more than $2\theta_1\ell$ input values are smaller than $\mathscr{V}$ and no more than $2\theta_2\ell$ input values are larger than $\mathscr{V}$, then, with probability $1 - \tilde{p}$, the upper output set has no more than $\eta\ell$ values larger than $\mathscr{V}$ and no more than $(2\theta_1 + \delta)\ell$ values smaller than $\mathscr{V}$, and the lower output set has no more than $\eta\ell$ values smaller than $\mathscr{V}$ and no more than $(2\theta_2 + \delta)\ell$ values larger than $\mathscr{V}$.

THEOREM 3. *For every fixed $0 < \eta < 1$, and for every $0 \leqq i \leqq \log m$, there is an explicit construction with depth $c_1((2^{i+5})/\eta^2 + \log 4ed/\eta^2)$ and width $c_2 m/2^i$, where c_1 and c_2 depend on η and p but not on m or i, that is, a $(m/2^i, \eta, \eta, 2^{-4m})$-stochastic halver even if each of its comparators is faulty with some fixed probability smaller than $\frac{1}{2}$.*

Proof. The construction is similar to the halver component in [AKS83]; only the analysis is different, taking into account the faulty comparators. Let $\bar{X} = \{X_1, \cdots, X_\ell\}$ and $\bar{Y} = \{Y_1, \cdots, Y_\ell\}$ denote two sets of $\ell = m/2^i$ registers each. The input is given in the 2ℓ registers $\bar{X} \cup \bar{Y}$. The lower output set is $\bar{Y}$ and the upper output set is $\bar{X}$. We connect the registers in $\bar{X}$ to the registers of $\bar{Y}$ by an $(\eta, (1 - \eta + \eta^2/2)/(\eta - \eta^2/4), \ell, d)$-expander. Each edge of the expander corresponds to an $(\eta^2/4ed \times 2^{-2^{i+5}/\eta^2})$-enforced comparison between $X \in \bar{X}$ to $Y \in \bar{Y}$ moving the larger value to Y and the smaller value to X. To schedule the comparisons, we color the edges of the expander with d colors. The comparisons are executed according to the color order; thus, each register is involved in no more than one comparison at a time. The depth and width of each enforced comparator is, according to Theorem 1, bounded by $3(2/(3 - 6p))^{1/(\log 23/18)}((2^{i+5})/\eta^2 + \log(8ed/\eta^2))$. The construction contains d phases and each phase executes $m/2^i$ enforced comparators in parallel. Thus we get a network of depth $c_1((2^{i+5})/\eta^2 + \log(8ed/\eta^2))$ and width $c_2 m/2^i$, where c_1 and c_2 depend on η and p but not on m or i.

Let $E(X)$ and $E(Y)$ denote the sets of registers in $\bar{X}$ and $\bar{Y}$, respectively, that are adjacent to faulty enforced comparators. The probability that more than $\eta^2\ell/4$ out of the $d\ell$ enforced comparators are faulty is bounded by

$$\sum_{k > \eta^2\ell/4} \binom{d\frac{m}{2^i}}{k} \left(\frac{\eta^2}{4ed2^{2^{i+5}/\eta^2}}\right)^k \leqq 2\left(\frac{4ed}{\eta^2}\right)^{\frac{\eta^2 m}{2^{i+2}}} \left(\frac{\eta^2}{4ed2^{2^{i+5}/\eta^2}}\right)^{\frac{\eta^2 m}{2^{i+2}}} \leqq 2^{-4m}.$$

Thus with probability $1 - 2^{-4m}$, $E(X)$ and $E(Y)$ are not larger than $\eta^2\ell/4$. A faulty enforced comparator may output one of the input values in its two outputs; thus, it can change the total number of values that are larger or smaller than $\mathcal{V}$. The input set has no more than $2\theta_1\ell$ values smaller than $\mathcal{V}$, and no more than $2\theta_2\ell$ values larger than $\mathcal{V}$. Thus, if there are no more than $\eta^2\ell/4$ faulty enforced comparators, the upper output set $\bar{X}$ has no more than $(2\theta_1 + \eta^2/4)\ell \leqq (2\theta_1 + \eta)\ell$ values smaller than $\mathcal{V}$, and the lower output set $\bar{Y}$ has no more $(2\theta_2 + \eta^2/4)\ell \leqq (2\theta_2 + \eta)\ell$ values larger than $\mathcal{V}$.

Let $H(X)$ denote the registers in $\bar{X}$ storing, at the end of the execution of this construction, values that are larger than $\mathcal{V}$. If $x \notin E(X)$, then the value of x does not increase while executing this construction; similarly, the value of $y \notin E(Y)$ does not decrease. If $x \in H(X) - E(X)$ and $y \notin E(Y)$ is a neighbor of x, then the value of y at the end of the execution must be larger than the value of x, and thus must be larger than $\mathcal{V}$.

If $|H(X)| > \eta\ell$, $|E(X)| \leqq \eta^2\ell/4$, and $|E(Y)| \leqq \eta^2\ell/4$, then the total number of registers that at the end of the execution of this component store elements that are larger than $\mathcal{V}$ is at least

$$|H(X)| + |\Gamma(H(X)-E(X))| - |E(Y)| > \eta\ell + \frac{1-\eta+\eta^2/2}{\eta-\eta^2/4}\left(\eta - \frac{\eta^2}{4}\right) \times \ell - \frac{\eta^2}{4}\ell = \left(1 + \frac{\eta^2}{4}\right)\ell > \left(2\theta_2 + \frac{\eta^2}{4}\right)\ell.$$

But the input set had no more than $2\theta_2\ell$ values larger than $\mathcal{V}$, and $\eta^2\ell/4$ faulty enforced comparators can add no more than another $\eta^2\ell/4$ values larger than $\mathcal{V}$. Thus $|H(X)|$ must be smaller than $\eta\ell$. By a similar argument, $\bar{Y}$ has no more than $\eta\ell$ values that are smaller than $\mathcal{V}$. □

The fault tolerant network uses $m = \log N$ registers to simulate each register of the original sorting network. The invariant kept by the fault tolerant construction is that at least $(1 - \varepsilon)m$ of the m registers always store the correct value of the original network. The following construction enables the network to enforce the majority among the m registers in constant depth.

DEFINITION. An $(\delta, \varepsilon, m, \tilde{p})$-*majority expander* is a construction with m inputs and m outputs. If more than $(1 - \delta)m$ of the inputs have the same value, then with probability $1 - \tilde{p}$, this value ends up in at least $(1 - \varepsilon)m$ of the output registers (i.e., if the input set δ-represents a value $\mathcal{V}$, then with probability $1 - \tilde{p}$ the output set ε-represents $\mathcal{V}$).

In what follows, the $O(f)$ notation means that the expression is bounded by cf where c is a constant that is independent of m and i but might depend on ε.

THEOREM 4. *For every fixed $\varepsilon > 0$ there exists a construction of width $O(m)$ and depth $O(1)$ that is an $(\frac{1}{5}, \varepsilon, m, 2^{-4m+1})$-majority expander, even if each of its comparators is faulty with some fixed probability smaller than $\frac{1}{2}$.*

Proof. Let $\bar{X} = \{X_1, \cdots, X_m\}$ denote the m input registers, and let $\bar{Y} = \{Y_1, \cdots, Y_m\}$ denote an additional set of m registers. Let $\mathcal{V}$ denote the value that appears in more than $4m/5$ inputs. The majority expander network has four stages (see Fig. 2). Stage 1 copies the values of $\bar{X}$ (the input values) to $\bar{Y}$. At the end of stage 1, no more than $2m/5$ registers in $\bar{X} \cup \bar{Y}$ store values smaller than $\mathcal{V}$, and no more than $2m/5$ registers in $\bar{X} \cup \bar{Y}$ store values larger than $\mathcal{V}$. In stage 2, the set $\bar{X}$ is connected to $\bar{Y}$ by an $(m, \varepsilon/4, \delta, 2^{-4m})$-stochastic halver, where $\delta = \min[\varepsilon/4, \frac{1}{10}]$, moving the larger values to $\bar{Y}$ and the smaller values to $\bar{X}$. All but $\varepsilon m/4$ of the m largest values are moved to $\bar{Y}$, and at the end of stage 2 $\bar{X}$ has no more than $\varepsilon m/4$ values larger than $\mathcal{V}$ and no

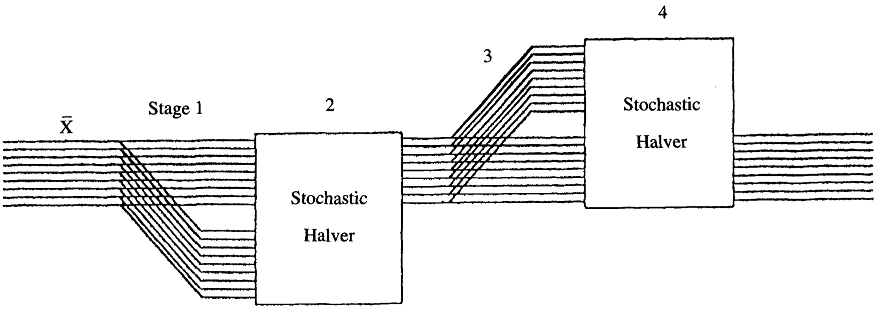


FIG. 2. *Majority expander.*

more than $m/2$ values smaller than $\mathcal{V}$. Stage 3 copies again the values of $\bar{X}$ to $\bar{Y}$ (erasing the previous values of $\bar{Y}$). At the end of this stage $\bar{X} \cup \bar{Y}$ has no more than $2\varepsilon m/4$ values larger than $\mathcal{V}$ and no more than m values smaller than $\mathcal{V}$. In stage 4, $\bar{Y}$ is connected to $\bar{X}$ with by another $(m, \varepsilon/4, \delta, 2^{-4m})$-stochastic halver, moving the larger values to $\bar{X}$ and the smaller values to $\bar{Y}$. After the execution of this stage, $\bar{X}$ ends up with no more than $3\varepsilon m/4$ values larger than $\mathcal{V}$, and no more than $\varepsilon m/4$ values smaller than $\mathcal{V}$. Thus, the output $\bar{X}$ has at least $(1-\varepsilon)m$ copies of the majority value $\mathcal{V}$ with probability $1 - 2^{-4m+1}$. □

Using the enforced comparator and the majority expander we can now construct the fault tolerant comparator that simulates the perfect comparators of the original network.

DEFINITION. An $(\varepsilon, m, \tilde{p})$-*fault-tolerant* comparator is a construction with two sets of m inputs, and two sets of m outputs. If the two input sets ε-represent the values $\mathcal{V}_1$ and $\mathcal{V}_2$, respectively, then, with probability $1-\tilde{p}$, the upper output set ε-represents $\min[\mathcal{V}_1, \mathcal{V}_2]$ and the lower output set ε-represents $\max[\mathcal{V}_1, \mathcal{V}_2]$.

THEOREM 5. *There exists a construction of depth $O(1)$ and width $O(m)$ that is a $(\frac{1}{15}, m, 2^{-4m+3})$-fault-tolerant comparator even if each of the comparators in the construction is faulty with some fixed probability smaller than $\frac{1}{2}$.*

Proof. (See Fig. 3.) Let $\bar{X} = \{X_1, \cdots, X_m\}$ and $\bar{Y} = \{Y_1, \cdots, Y_m\}$ be the two sets of input registers, let $\bar{X}$ be the upper output set, and $\bar{Y}$ the lower output set. We use

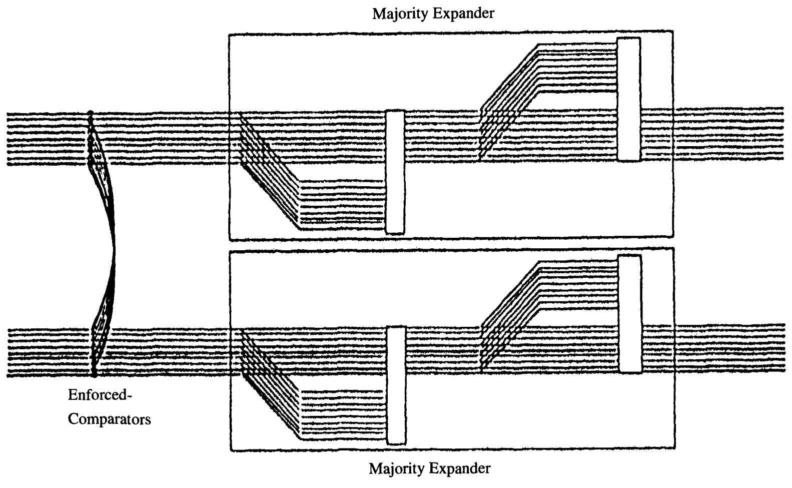


FIG. 3. *A fault tolerant comparator that simulates a simple comparison.*

2^{-67}-enforced comparators to compare each X_ℓ to Y_ℓ, $\ell = 1, \cdots, m$, moving the larger values to the Y registers and the smaller values to the X registers.

A register X_ℓ might have an incorrect value after the comparison for three reasons:

(a) X_ℓ had an incorrect value before the comparison.

(b) Y_ℓ had an incorrect value before the comparison.

(c) The comparison between X_ℓ and Y_ℓ was faulty.

The probability that more than $m/15$ out of the m comparisons are faulty is bounded by

$$\sum_{k > m/15} \binom{m}{k} \left(\frac{1}{2^{67}} \right)^k \leqq 2^{-4m}.$$

Thus with probability $1 - 2^{-4m}$ at the end of this stage both $\bar{X}$ and $\bar{Y}$ $\frac{3}{15}$-represent the correct values. Using the bounded depth $(\frac{1}{5}, \frac{1}{15}, m, 2^{-4m+1})$-majority expander of Theorem 4, we reduce the number of incorrect values in each of these sets to $m/15$. □

The following construction is used in the last phase of the network, where one correct value is extracted from $m = \log N$ registers, among which at least $(1 - \varepsilon)m$ of the values are correct.

DEFINITION. An $(\ell, \varepsilon, \tilde{p})$*-majority preserver* is a network with ℓ inputs and $\ell/2$ outputs. If at least $(1 - \varepsilon)\ell$ of the input values are equal, then this value appears in at least $(1 - \varepsilon)\ell/2$ of the output registers.

THEOREM 6. *For every* $\varepsilon < \frac{1}{5}$, *and* $0 \leqq i \leqq \log m$ *there is an explicit construction of a network of depth* $O(2^i)$ *and width* $O(m)$ *that is an* $(m/2^i, \varepsilon, 2^{-3m})$*-majority preserver, even if each comparator in the construction is faulty with some fixed probability smaller than* $\frac{1}{2}$.

Proof. Let $\ell = m/2^i$, $\bar{X} = \{X_1, \cdots, X_{\ell/2}\}$ and $\bar{Y} = \{Y_1, \cdots, Y_{\ell/2}\}$. Assume that the ℓ input registers are $\bar{X} \cup \bar{Y}$, and the $\ell/2$ output registers are $\bar{X}$. Denote by $\mathcal{V}$ the majority value that appears in at least $(1 - \varepsilon)\ell$ input registers. The construction has three stages (see Fig. 4). Stage 1 connects $\bar{X}$ to $\bar{Y}$ by an $(m/2^i, \varepsilon/4, \varepsilon/4, 2^{-4m})$-stochastic halver, moving the smaller values to $\bar{Y}$. Since $\bar{X} \cup \bar{Y}$ have no more than $\varepsilon\ell$ values smaller than $\mathcal{V}$, $\bar{X}$ has no more than $\varepsilon\ell/8$ values smaller than $\mathcal{V}$ at the end of this stage. Stage 2 copies the values of $\bar{X}$ to $\bar{Y}$. At the end of stage 2, $\bar{X} \cup \bar{Y}$ has no more than $2\varepsilon\ell/8$ values smaller than $\mathcal{V}$, and no more than $(2 + \frac{1}{4})\varepsilon\ell$ values larger than $\mathcal{V}$. In stage 3, $\bar{Y}$ is connected to $\bar{X}$ by an $(m/2^i, \varepsilon/4, \varepsilon/4, 2^{-4m})$-stochastic halver, moving the smaller values to X. At the end of this stage, $\bar{X}$ has no more than $3\varepsilon\ell/8$ values smaller than $\mathcal{V}$ and no more than $\varepsilon\ell/8$ values larger than $\mathcal{V}$. □

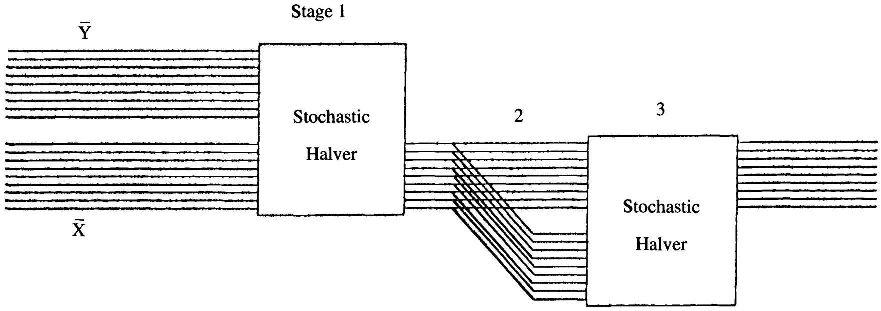


FIG. 4. *Majority preserver.*

2.3. The complete network. The first part of the fault tolerant network simulates the comparators of the original sorting network. Let $X_1, \cdots, X_N$ denote the N input registers of the original network. The first log log N levels generate $m = \log N$ copies of each input register. Using the log N copies of each original register, we can simulate the comparators of the original network by $(\frac{1}{15}, m, 2^{-4m+3})$-fault-tolerant comparators. The original sorting network has no more than N^2 comparators; thus, with probability $1 - 2^{-2m+3}$ the outputs of every fault tolerant comparator $\frac{1}{15}$-represents the value of the corresponding register at the corresponding level of the original sorting network. In particular, at the end of this part of the fault tolerant sorting network, each set $\bar{X}$, $\frac{1}{15}$-represents the value that register X outputs in the original network.

The second part of the network reduces each set $\bar{X}_i$ having at least $(1 - \frac{1}{15})m$ correct copies to one register X_i storing one correct value. We extract the correct copy by log log N stages of majority preservers (see Fig. 5). The input to stage i, $i = 0, \cdots, \log\log N - 1$, is a set of $m/2^i$ registers with at least $(1 - \frac{1}{5})m/2^i$ correct values. Stage i uses an $(m/2^i, \frac{1}{5}, 2^{-3m})$-majority preserver. The depth of the majority preserver is $O(2^i)$, its width is $O(m)$, and with probability $1 - 2^{-3m}$ it produces a set of $m/2^{i+1}$ registers with at least $(1 - \frac{1}{5})m/2^{i+1}$ registers storing the correct values. After $\log m = \log\log N$ nonfaulty iterations, we are left with one register storing the correct value.

The depth of the second part of the network is bounded by

$$O\left(\sum_{i=1}^{\log\log N} 2^i\right) = O(\log N).$$

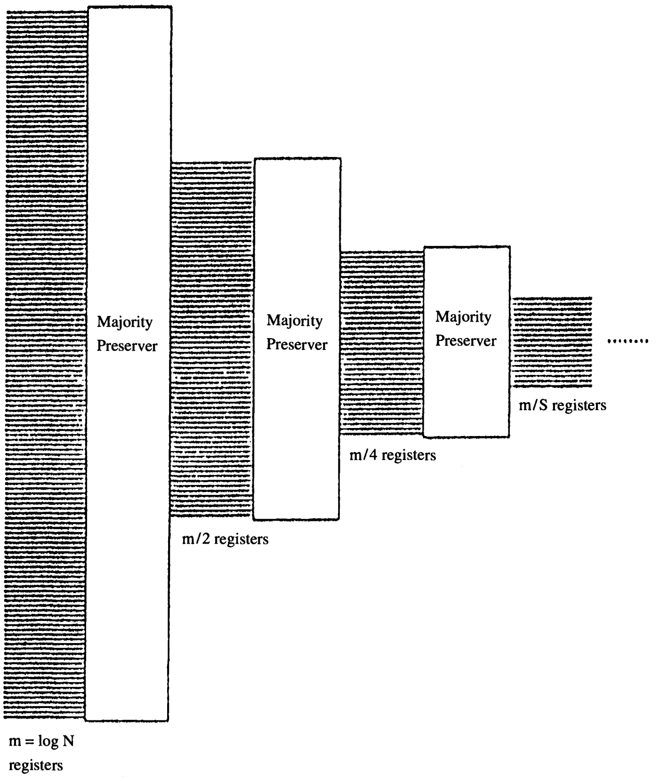


FIG. 5. *Extracting one correct value from a set that ε represents it.*

The width of the second part is $O(mN) = O(N \log N)$, and with probability $1 - N(\log\log N)2^{-3 \log N}$ it produces the N correct outputs. Since the depth of any network that sorts N elements is at least $\log N$, we have proved Theorem 7.

THEOREM 7. *Given a sorting network with depth d and width N, there is a fault tolerant sorting network of depth $O(d)$ and width $O(N \log N)$ that with probability $1 - 1/N$ sorts correctly every list of N elements, even if each comparison in the network is faulty with some fixed probability smaller than $\frac{1}{2}$.*

3. Remarks and open problems.

As mentioned before, our technique is applicable to any comparator based network. Thus we can construct fault tolerant networks of depth $O(\log N)$ and width $O(N \log N)$ for merging two sorted lists of size N each, or for finding the maximum of N elements. The failure probability in both cases is smaller than $1/N$ when each comparator is faulty with some fixed probability smaller than $\frac{1}{2}$, independent of the other comparators.

A comparator network for finding the maximum of N elements uses $O(N)$ comparators. Thus our fault tolerant version of that network uses $O(N \log N)$ comparators. This bound matches the $\Omega(N \log N)$ lower bound proven by Yao and Yao [YY85] for even a weaker model of faults. The optimality of the fault tolerant sorting network and the fault tolerant merging network, in terms of width and total number of comparators, is still open. We showed in the Introduction that some redundancy is essential for small failure probability, but we do not know if $\Omega(\log N)$ redundancy is necessary.

REFERENCES

[AKS83] M. AJTAI, J. KOMLOS, AND E. SZEMEREDI, *An $o(n \log n)$ sorting network*, Combinatorica, 3 (1983), pp. 1–19.

[Alo86] N. ALON, *Eigenvalues and expanders*, Combinatorica, 6 (1986), pp. 83–96.

[Bat68] E. B. BATCHER, *Sorting networks and their applications*, in Proc. AFIPS Spring Joint Computer Conference, Montvale, New Jersey, June 1968, pp. 307–314.

[HW75] G. H. HARDY AND E. M. WRIGHT, *An Introduction to the Theory of Numbers*, Oxford University Press, London, 1975.

[LPS86] A. LUBOTZKY, R. PHILLIPS, AND P. SARNAK, *Ramanugan conjecture and explicit construction of expanders*, in 18th Annual Symposium on Theory of Computing, Berkeley, California, May 1986, pp. 240–246.

[MS56] E. F. MORE AND C. SHANNON, *Reliable circuits using less reliable relays* i-ii, J. Franklin Inst., 262 (1956), pp. 191–208.

[MS77] F. J. MACWILLIAMS AND N. J. A. SLOANE, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1977.

[Pip85] N. PIPPENGER, *On networks of noisy gates*, in 26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, October 1985, pp. 30–38.

[Rud85] L. RUDOLPH, *A robust sorting network*, IEEE Trans. Comput., C-34 (1985), pp. 326–335.

[SS88] M. SCHIMMER AND C. STARK, *A correction network for N-sorter*, in VLSI Algorithm and Architecture, 3rd Aegean Workshop on Computing, AWOC, Corfu, Greece, June 1988, pp. 444–455.

[VN56] J. VON NEUMANN, *Probabilistic logics and the synthesis of reliable organisms from unreliable components*, in Automata Studies, Princeton University Press, Princeton, New Jersey, 1956, pp. 43–98.

[YY85] A. C. YAO AND F. F. YAO, *On fault-tolerant networks for sorting*, SIAM J. Comput., 14 (1985), pp. 120–128.

SIAM J. DISC. MATH.
Vol. 4, No. 4, pp. 481-501, November 1991

003

DETERMINISTIC DECOMPOSITION OF RECURSIVE GRAPH CLASSES*

RICHARD B. BORIE†, R. GARY PARKER‡, AND CRAIG A. TOVEY§

Abstract. The popular class of series-parallel graphs can be built recursively from single edges by combining smaller components via connections only at a fixed pair of vertices called *terminals*. This recursive construction property with a limited number of terminals is essential to the linear time solution of problems on these graphs. A second useful property of these graphs is that *decomposition is deterministic* with respect to the series-parallel rules. This implies that the parse-tree of decomposition (which is required by the algorithms) can be determined in a straightforward manner by repeatedly applying the decomposition rules. Subject to retaining these properties, we examine how far the series-parallel graphs can be generalized. Corollaries of our results yield the deterministic decomposition of the series-parallel and Halin graph classes.

Key words. series-parallel graph, decomposition, recursive graph class, dynamic programming, linear-time algorithm

AMS(MOS) subject classifications. 05C70, 68R10, 90C39

1. Recursive graph classes.

1.1. Introduction. Recently, much effort has focused on the recognition of certain recursively constructed graph classes—trees, series-parallel graphs, Halin graphs, partial k-trees, bandwidth k-graphs—and the development of efficient algorithms for $\mathcal{NP}$-hard problems when instances are restricted to graphs belonging to these classes.

The common feature of each recursive graph class is that any sufficiently large member is composed from smaller members of the same class, joined by merging distinguished vertices known as *terminals*. Many structures that occur in such diverse areas as telecommunications networks, VLSI design, and software systems are hierarchical or modular, and might therefore be modeled by recursively constructed graphs.

Fast algorithms on these recursive graph classes are typically based on dynamic programming, so that a solution to a large member can be determined directly from solutions to the smaller members which constitute it, using a recurrence relation specific to the problem. If the number of terminals is restricted to some fixed value k, the recurrence relation can be evaluated efficiently. This in turn leads to an efficient algorithm, assuming a *decomposition tree* for any graph in the class can be found quickly. For example, a decomposition tree for a series-parallel graph can be found easily by repeatedly applying the series-parallel operations in any arbitrary fashion.

Our aim in this paper is to capture and extend the structure of this *deterministic decomposition* property exhibited by the series-parallel class. We introduce a simple 3-parameter notation for families of composition rules, broad enough to describe any operation that joins k-terminal graphs together by merging terminals. For example, the series-parallel graphs will correspond to the triple $[2, 1, 3]$. Our principal result is the identification of those families of composition rules that enjoy the deterministic decomposition property. In particular, we will show that a family of composition rules $[k, u, r]$ produces a recursive graph class with deterministic decomposition if and

* Received by the editors June 13, 1988; accepted for publication (in revised form) October 26, 1990.

† Department of Computer Science, Box 870290, University of Alabama, Tuscaloosa, Alabama 35487-0290. The work of this author was supported by a National Science Foundation Graduate Fellowship, School of Information and Computer Science, Georgia Institute of Technology.

‡ School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332.

§ School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332-0205. The work of this author was supported by a Presidential Young Investigator Award from the National Science Foundation (ECS-8451032) and a matching grant from Digital Equipment Corporation.

only if $[k, u, r]$ has one of the following forms: $[2, r-2, r]$, $[3, r-3, r]$, $[k, 0, k]$, or $[k, 1, k+1]$. Even when graph classes are generated from rules involving two or more triples, we are able to show that a sufficient condition for deterministic decomposition is that each such triple have one of these forms; thus far, however, a converse necessary condition eludes us for this case.

1.2. Recursively constructed graph classes. The recursive graph classes mentioned above (trees, series-parallel graphs, Halin graphs, etc.) can be described by similar definitions. In order to discuss these classes using a common framework, we first introduce the following terminology. A *k-terminal graph* $G = (V, T, E)$ has a vertex set V, an edge set E, and a (possibly ordered) set of *terminals* $T = \{t_1, \cdots, t_{|T|}\} \subseteq V$, where $|T| \leqq k$. A *recursively constructed class* $C(B, R)$ in some universal set U is specified by base elements $B \subseteq U$ and a finite rule set $R = \{f_1, \cdots, f_n\}$ where each $f_i : U^{m_i} \rightarrow U$ is a recursive composition operation with arity m_i; C is then the closure of B in U by $f_1, \cdots, f_n$. Typically, for some k, U is the set of k-terminal graphs and B is a set of connected k-terminal graphs (V, T, E) with $V = T$. But each such base graph is trivially composed of individual edges, so it is reasonable and convenient to use $C(R)$ to denote $C(B, R)$ where B only contains K_2.

A *decomposition tree* of a k-terminal graph $G \in C(B, R)$ is a rooted tree with vertex labels g and f such that

- $g_v = G$ if v is the root,
- $f_v \in R$ if v is an interior node,
- $g_v = f_v(g_{v_1}, \cdots, g_{v_m})$ if interior node v has children $v_1, \cdots, v_m$, and
- $g_v \in B$ if v is a leaf.

In previous research as well as the efforts reported here, permitted composition operations take the same general form. For $1 \leqq i \leqq m$, let $G_i = (V_i, T_i, E_i) \in U$, such that $V_1, \cdots, V_m$ are mutually disjoint vertex sets. Let $G = (V, T, E) \in U$ as well. A *valid vertex mapping* is a (total surjective) function $f : \bigcup_{1 \leqq i \leqq m} V_i \rightarrow V$ such that

- vertices from the same G_i remain distinct: $v_1 \in V_i, v_2 \in V_i, f(v_1) = f(v_2) \Rightarrow v_1 = v_2$;
- only (but not necessarily all) terminals map to terminals: $v \in V_i, f(v) \in T \Rightarrow v \in T_i$;
- only terminals can merge: $v_1 \in V_{i_1}, v_2 \in V_{i_2}, i_1 \neq i_2, f(v_1) = f(v_2) \Rightarrow v_1 \in T_{i_1}$, $v_2 \in T_{i_2}$ (in this case vertices v_1 and v_2 are said to be *identified*); and
- edges are preserved: $(\exists i)(\{v_1, v_2\} \in E_i) \Leftrightarrow \{f(v_1), f(v_2)\} \in E$.

If f is a valid vertex mapping, then the corresponding m-ary composition operation (also denoted by f) is written $f(G_1, \cdots, G_m) = G$.

With composition operations now precisely defined, note that the resulting graph G is determined entirely (up to isomorphism) by which terminals from each T_i are identified and by which vertices are in T. However, it is quite cumbersome to define a composition operation by writing the corresponding valid vertex mapping. Instead, it is customary to let $f(v) = v$ for $v \notin \bigcup_{1 \leqq i \leqq m} T_i$; to list which vertices from the various T_i are identified by f; and, if $v_1, \cdots, v_a$ are merged together, to denote the $f(v_i)$ either by one of the v_i or by a new vertex name. Hence, in the remainder of the paper, we will employ this less formal notation. It is also convenient to write $V = f(\bigcup_{1 \leqq i \leqq m} V_i) = \bigcup_{1 \leqq i \leqq m} V_i$ and $E = f(\bigcup_{1 \leqq i \leqq m} E_i) = \bigcup_{1 \leqq i \leqq m} E_i$, where these union operators consider identified vertices to be identical.

To illustrate, let U be the set of 2-terminal graphs. Then the class of *series-parallel graphs* is $C(\{s, p, j\})$, where each of s, p, and j is a binary operation that produces a graph $G = (V_1 \cup V_2, \{t_1, t_2\}, E_1 \cup E_2)$ when given operands $G_i = (V_i, \{t_{i1}, t_{i2}\}, E_i)$,

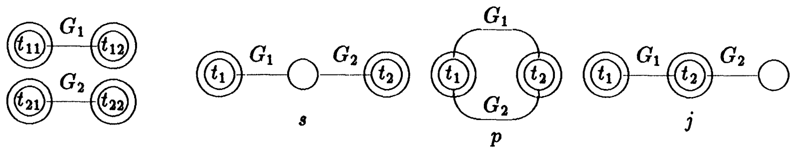


FIG. 1. *Series, parallel, and jackknife operations.*

for $i = 1, 2$. These operations are defined as follows (see also Fig. 1; in all figures, doubly circled vertices denote terminals).

- The *series* operation s identifies t_{12} with t_{21} (which then loses its terminal status), and assigns $t_1 = t_{11}$ and $t_2 = t_{22}$.
- The *parallel* operation p identifies t_{11} with t_{21} and t_{12} with t_{22}, and assigns $t_1 = t_{11} = t_{21}$ and $t_2 = t_{12} = t_{22}$.
- The *jackknife* operation j identifies t_{12} with t_{21}, and assigns $t_1 = t_{11}$ and $t_2 = t_{12} = t_{21}$ (t_{22} loses its terminal status).

The study of series-parallel graphs can be traced at least back to Duffin [6]. However, Takamizawa, Nishizeki, and Saito [16] appear to have initiated the development of linear time algorithms for otherwise hard graph problems when instances are restricted to such graphs. More recently, Wimer et al. [18]–[20], [7], Liu and Geldmacher [8], Wald and Colbourn [17], Bern, Lawler, and Wong [5], Richey, Parker, and Rardin [10], [12], [13], and others have shown the existence of thousands of such algorithms. Furthermore, the decomposition tree of a series-parallel graph can be created in linear time [8], [9], which in turn leads to linear time algorithms for most problems that are polynomial-time solvable by dynamic programming when instances are restricted to series-parallel graphs.

The partial k-trees (see Arnborg, Corneil, and Proskurowski [1]–[4]) provide a generalization of the series-parallel graphs. Indeed, the class of partial 1-trees is coincident with the class of trees, and the class of partial 2-trees is coincident with the class of series-parallel graphs, while Halin graphs are contained in the class of partial 3-trees. Wimer and Hedetniemi [18], [19] show that the class of partial k-trees can be defined as a $(k+1)$-terminal recursive graph class, and Arnborg, Corneil, and Proskurowski [2], [3] show that this class can be recognized in polynomial time by a bottom-up method. The partial k-trees can also be recognized in $O(n^3)$ time using the graph minor results of Robertson and Seymour [14], [15], but the proof of this fact is nonconstructive and the recognition algorithms have proven difficult to find.

1.3. Efficient algorithms on recursive graph classes. Given a graph (V, E), the minimum vertex cover problem seeks a smallest subset $VC \subseteq V$ such that $VC \cap \{i,j\} \neq \emptyset$ for every edge $\{i,j\} \in E$. While this problem is well known to be $\mathcal{NP}$-hard, in general, it is easy to see how it can be efficiently solved on the recursive class of series-parallel graphs. Consider a series-parallel graph $G = (V, T, E)$ and for each $S \subseteq T$ define a property $P_S(G)$ to be the cardinality of a minimum vertex cover VC of G such that $S = VC \cap T$.

If G is the base graph K_2, then

- $P_\emptyset(G) = \infty$ (no such cover exists),
- $P_{t_1}(G) = 1$,
- $P_{t_2}(G) = 1$, and
- $P_{t_1t_2}(G) = 2$.

To develop appropriate recurrence relations for a dynamic programming solution, we start by constructing multiplication tables for each of the composition operations.

When G_1 and G_2 are composed by an operation f to form G, the multiplication table for operation f shows which of the possible pairs of sets $S_1 \subseteq T_1$ and $S_2 \subseteq T_2$ are compatible. In addition, the table for f shows the value of the corresponding $S \subseteq T$ for each such compatible set pair. Figure 2 gives the multiplication tables for operations s, p, and j.

It is now straightforward to construct the recurrence relations directly from the multiplication tables. The formulas simply compute the optimal values from among the compositions of the compatible pairs.

Thus, if $G = s(G_1, G_2)$ then

- $P_{\varnothing}(G) = \min\{P_{\varnothing}(G_1) + P_{\varnothing}(G_2),\ P_{t_{12}}(G_1) + P_{t_{21}}(G_2) - 1\}$,
- $P_{t_1}(G) = \min\{P_{t_{11}}(G_1) + P_{\varnothing}(G_2),\ P_{t_{11},t_{12}}(G_1) + P_{t_{21}}(G_2) - 1\}$,
- $P_{t_2}(G) = \min\{P_{\varnothing}(G_1) + P_{t_{22}}(G_2),\ P_{t_{12}}(G_1) + P_{t_{21},t_{22}}(G_2) - 1\}$, and
- $P_{t_1t_2}(G) = \min\{P_{t_{11}}(G_1) + P_{t_{22}}(G_2),\ P_{t_{11},t_{12}}(G_1) + P_{t_{21},t_{22}}(G_2) - 1\}$.

If $G = p(G_1, G_2)$ then

- $P_{\varnothing}(G) = P_{\varnothing}(G_1) + P_{\varnothing}(G_2)$,
- $P_{t_1}(G) = P_{t_{11}}(G_1) + P_{t_{21}}(G_2) - 1$,
- $P_{t_2}(G) = P_{t_{12}}(G_1) + P_{t_{22}}(G_2) - 1$, and
- $P_{t_1t_2}(G) = P_{t_{11},t_{12}}(G_1) + P_{t_{21},t_{22}}(G_2) - 2$.

If $G = j(G_1, G_2)$ then

- $P_{\varnothing}(G) = \min\{P_{\varnothing}(G_1) + P_{\varnothing}(G_2),\ P_{\varnothing}(G_1) + P_{t_{22}}(G_2)\}$,
- $P_{t_1}(G) = \min\{P_{t_{11}}(G_1) + P_{\varnothing}(G_2),\ P_{t_{11}}(G_1) + P_{t_{22}}(G_2)\}$,
- $P_{t_2}(G) = \min\{P_{t_{12}}(G_1) + P_{t_{21}}(G_2) - 1,\ P_{t_{12}}(G_1) + P_{t_{21},t_{22}}(G_2) - 1\}$, and
- $P_{t_1t_2}(G) = \min\{P_{t_{11},t_{12}}(G_1) + P_{t_{21}}(G_2) - 1,\ P_{t_{11},t_{12}}(G_1) + P_{t_{21},t_{22}}(G_2) - 1\}$.

Figure 3 shows a decomposition tree for a given series-parallel graph, while Fig. 4 shows the dynamic programming solution for the minimum vertex cover problem on the stated graph, where each 4-tuple has the form $(P_{\varnothing},\ P_{t_1},\ P_{t_2},\ P_{t_1t_2})$. The optimal solution value is simply the minimum value in the 4-tuple associated with the root node of the decomposition tree (in this case 4).

This is the standard dynamic programming notion that underlies linear time algorithms on series-parallel graphs. The time required is linear because there is only a constant amount of information to be computed for each node of the decomposition tree, and the size of this decomposition tree is linear in the cardinality of the edge set of G. Furthermore, it should be obvious that a similar method would work for a class of k-terminal graphs, for any fixed value of k, once the decomposition tree for a graph in the class is found.

As a quick illustration, consider the 3-terminal operation $G = f(G_1, G_2, G_3)$ illustrated in Fig. 5. The recurrence formulas for the minimum vertex cover problem for f can be

s	$\emptyset$	t_{21}	t_{22}	$t_{21}t_{22}$
$\emptyset$	$\emptyset$	-	t_2	-
t_{11}	t_1	-	t_1t_2	-
t_{12}	-	$\emptyset$	-	t_2
$t_{11}t_{12}$	-	t_1	-	t_1t_2

p	$\emptyset$	t_{21}	t_{22}	$t_{21}t_{22}$
$\emptyset$	$\emptyset$	-	-	-
t_{11}	-	t_1	-	-
t_{12}	-	-	t_2	-
$t_{11}t_{12}$	-	-	-	t_1t_2

j	$\emptyset$	t_{21}	t_{22}	$t_{21}t_{22}$
$\emptyset$	$\emptyset$	-	$\emptyset$	-
t_{11}	t_1	-	t_1	-
t_{12}	-	t_2	-	t_2
$t_{11}t_{12}$	-	t_1t_2	-	t_1t_2

FIG. 2. *Multiplication tables for s, p, and j for vertex cover.*

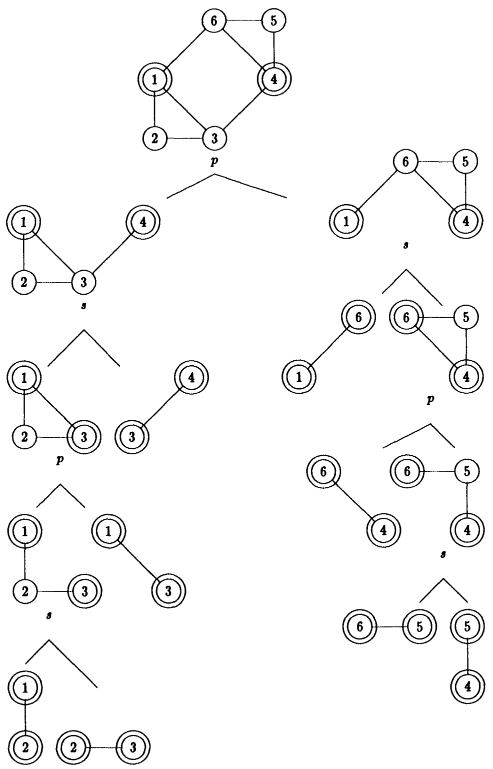


FIG. 3. *A decomposition tree for a series-parallel graph.*

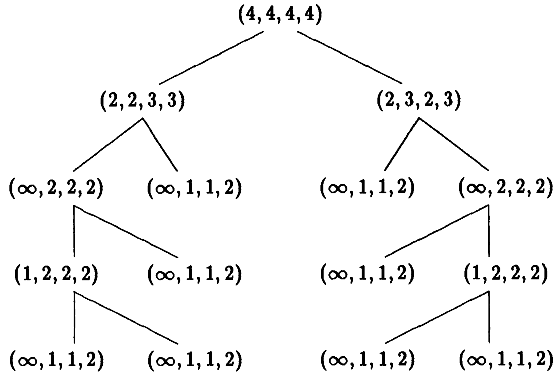


FIG. 4. *Dynamic programming on a series-parallel graph.*

stated as follows:

$$P_{\emptyset}(G) = \min\,\{P_{\emptyset}(G_1)+P_{\emptyset}(G_2)+P_{\emptyset}(G_3), P_8(G_1)+P_8(G_2)+P_8(G_3)-2\},$$

$$P_1(G) = \min\,\{P_1(G_1)+P_1(G_2)+P_{\emptyset}(G_3)-1, P_{1,8}(G_1)+P_{1,8}(G_2)+P_8(G_3)-3\},$$

$$P_3(G) = \min\,\{P_{\emptyset}(G_1)+P_3(G_2)+P_3(G_3)-1, P_8(G_1)+P_{3,8}(G_2)+P_{3,8}(G_3)-3\},$$

$$P_6(G)=\min\{P_6(G_1)+P_{\varnothing}(G_2)+P_6(G_3)-1, P_{6,8}(G_1)+P_8(G_2)+P_{6,8}(G_3)-3\},$$

$$P_{1,3}(G)=\min\{P_1(G_1)+P_{1,3}(G_2)+P_3(G_3)-2, P_{1,8}(G_1)+P_{1,3,8}(G_2)+P_{3,8}(G_3)-4\},$$

$$P_{1,6}(G)=\min\{P_{1,6}(G_1)+P_1(G_2)+P_6(G_3)-2, P_{1,6,8}(G_1)+P_{1,8}(G_2)+P_{6,8}(G_3)-4\},$$

$$P_{3,6}(G)=\min\{P_6(G_1)+P_3(G_2)+P_{3,6}(G_3)-2, P_{6,8}(G_1)+P_{3,8}(G_2)+P_{3,6,8}(G_3)-4\},$$

and

$$P_{1,3,6}(G)$$
$$=\min\{P_{1,6}(G_1)+P_{1,3}(G_2)+P_{3,6}(G_3)-3, P_{1,6,8}(G_1)+P_{1,3,8}(G_2)+P_{3,6,8}(G_3)-5\}.$$

Thus, if the values for G_1, G_2, G_3 have already been evaluated, then the values for G can be computed, as shown in Fig. 6. Here the cardinality of a minimum vertex cover is $P_{3,6}(G) = 4$ with a corresponding cover given by $\{2, 3, 5, 6\}$.

Note that it suffices to maintain the values of 2^k properties in order to solve the minimum vertex cover problem, given the decomposition tree for a recursively constructed k-terminal graph. Thus, if the number of terminals were unbounded, the amount of information required could grow exponentially, or even faster for some problems.

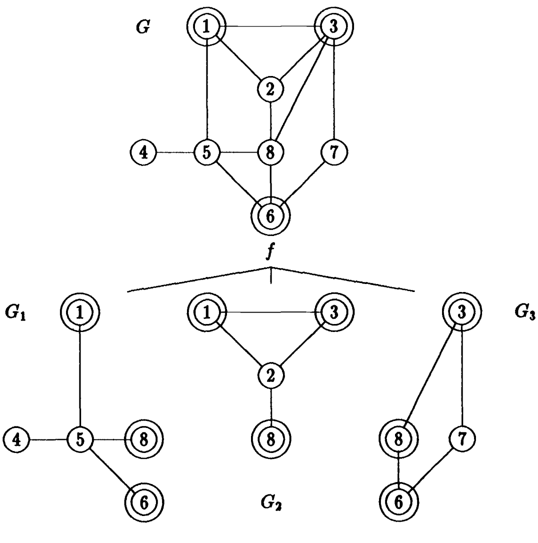


FIG. 5. *A 3-terminal operation $G = f(G_1, G_2, G_3)$.*

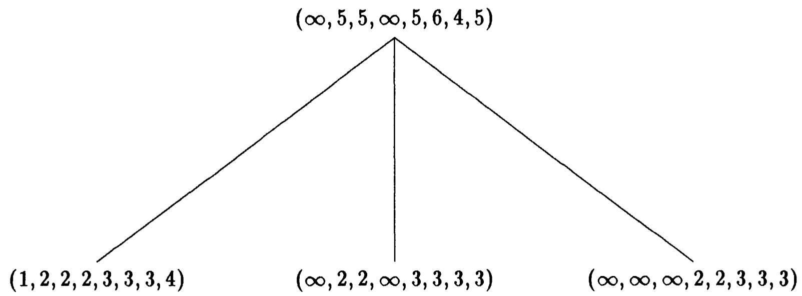


FIG. 6. *Dynamic programming on a 3-terminal graph.*

Of course, many other $\mathcal{NP}$-hard problems also have efficient dynamic programming solutions when restricted to k-terminal graph classes, once a decomposition tree for the graph is known. This provides motivation for generalizing to a recursively constructed graph class that is as large as possible, but which still possesses an efficient algorithm for recognition and decomposition.

2. Graph decomposition.

2.1. Recursive (k, u, r)-operations. It should be clear by now that the following condition is essential to the polynomial (often linear) time performance of dynamic programming algorithms [5], [20].

> *Finite terminal set*: The number of terminals among graphs in a recursive graph class must be bounded by some fixed value k.

This section develops additional natural properties that a set of decomposition rules should satisfy in order to allow efficient dynamic programming algorithms, while the remainder of the paper will derive conditions under which a set of rules satisfies those properties.

The initial problem is to develop a very broad class of composition operations. The terminology should be general enough to describe any operation that joins k-terminal graphs together by identification of terminals. The remaining problem then will be to determine which operations behave well. Accordingly, let a c-ary *recursive* (k, u, r)-*operation* be a function $f(G_1, \cdots, G_c) = G = (V, T, E)$ on k-terminal graphs that satisfies the following conditions:

- $|T| \leqq k$ and $|T_i| \leqq k$ for each $G_i = (V_i, T_i, E_i)$,
- $V = \bigcup_{i=1}^{c} V_i$,
- $E = \bigcup_{i=1}^{c} E_i$,
- $T \subseteq \bigcup_{i=1}^{c} T_i$,
- $|\bigcup_{i=1}^{c} T_i| - |T| \leqq u < r$,
- $|\bigcup_{i=1}^{c} T_i| \leqq r$,
- $|T_i| - |T| \leqq s = k + u - r \geqq 0$ for each i, and
- $V_{i_1} \cap V_{i_2} = T_{i_1} \cap T_{i_2}$ for each $i_1 \neq i_2$.

Conceptually, a (k, u, r)-operation joins k-terminal graphs G_i at their common terminals, producing a k-terminal graph G where there are up to r vertices in G that were terminals in the constituent G_i, and where up to u of these become undistinguished in the resultant G. It follows that r need never exceed $k + u$. None of the constituent G_i can have more than $s = k + u - r \geqq 0$ more terminals than G has. For any $0 \leqq k \leqq r$, $0 \leqq u < r$, $k + u \geqq r$, we let $[k, u, r]$ denote the family of all (k, u, r)-operations. Hence the series and parallel operations are both in $[2, 1, 3]$ (in fact the parallel operation is also in $[2, 0, 3]$).

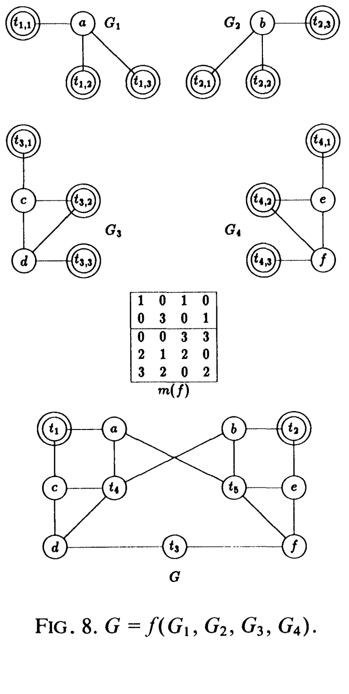
Now, the recursive operation f can be represented by a matrix $m(f)$ with $d \leqq r$ rows and c columns, such that $0 \leqq m_{i,j}(f) \leqq |T_j|$ for $1 \leqq i \leqq d$, $1 \leqq j \leqq c$. The nonzero elements $m_{i,j}(f)$ of the ith row indicate which terminals of each G_j are identified to create the ith new vertex of G. (If an element has value 0, then no terminal from G_j is used in the creation of this ith vertex.) The first $|T|$ rows of $m(f)$ indicate the ordered set of terminals of G, and the remaining rows indicate new vertices which result by identifying terminals from the T_j but which become undistinguished (nonterminals of G). If we let $t(f) = |T|$, then f is completely specified as $f = (m, t)$. Figure 7 illustrates the matrices for the series, parallel, and jackknife operations, while Fig. 8 shows $m(f)$ for a 4-ary (3, 3, 5)-operation $G = f(G_1, G_2, G_3, G_4)$.

$$\underset{m(s)}{\begin{array}{|cc|}\hline 1 & 0 \\ 0 & 2 \\ \hline 2 & 1 \\ \hline\end{array}} \qquad \underset{m(p)}{\begin{array}{|cc|}\hline 1 & 1 \\ 2 & 2 \\ \hline\end{array}} \qquad \underset{m(j)}{\begin{array}{|cc|}\hline 1 & 0 \\ 2 & 1 \\ \hline 0 & 2 \\ \hline\end{array}}$$

FIG. 7. *Matrices for series, parallel, and jackknife.*

Of greater interest, we can show that the Halin graphs are 3-terminal graphs which can be formed using binary (3, 0, 3)- and (3, 1, 4)-operations. Recall that a Halin graph is a planar graph whose edge set can be partitioned into a spanning tree with no degree-2 vertices, and a cycle on the leaves of this spanning tree. The graph in Fig. 9(a) is Halin with terminals denoting the root, leftmost leaf, and rightmost leaf.

Now let a *hypoHalin graph* be a planar graph whose edge set can be partitioned into a spanning tree and a path from its leftmost leaf to its rightmost leaf, that passes through the other leaves of this spanning tree (see Fig. 9(b)). If H is a Halin graph with terminals $\{t_1, t_2, t_3\}$, then H can be decomposed by a (3, 0, 3)-operation into a K_2 with edge $e = \{t_2, t_3\}$ and a hypoHalin graph $H - e$. The claim is that any nontrivial hypoHalin graph is decomposable into two smaller hypoHalin graphs. To see this, we can let l be that leaf in $H - e$ which is closest to t_3 such that it can be reached from t_2 by a path that passes through neither other leaves nor t_1, as in Fig. 9(b). (Alternatively, we could choose l as that leaf which is closest to t_2 such that it can be reached from t_3 by a path that passes through neither other leaves nor t_1.) Then $H - e$ is decomposed by a (3, 1, 4)-operation into two graphs with terminals $\{t_1, t_2, l\}$ and $\{t_1, l, t_3\}$. Finally, (3, 1, 4)-operations can be used to eliminate degree-1 vertices, thus producing two smaller hypoHalin graphs. The preceding argument, together with Theorem 7, will be used to show that the Halin graphs are efficiently recognizable by a natural top-down decomposition scheme.



$$\underset{m(f)}{\begin{array}{|cccc|}\hline 1 & 0 & 1 & 0 \\ 0 & 3 & 0 & 1 \\ \hline 0 & 0 & 3 & 3 \\ 2 & 1 & 2 & 0 \\ 3 & 2 & 0 & 2 \\ \hline\end{array}}$$

FIG. 8. $G = f(G_1, G_2, G_3, G_4)$.

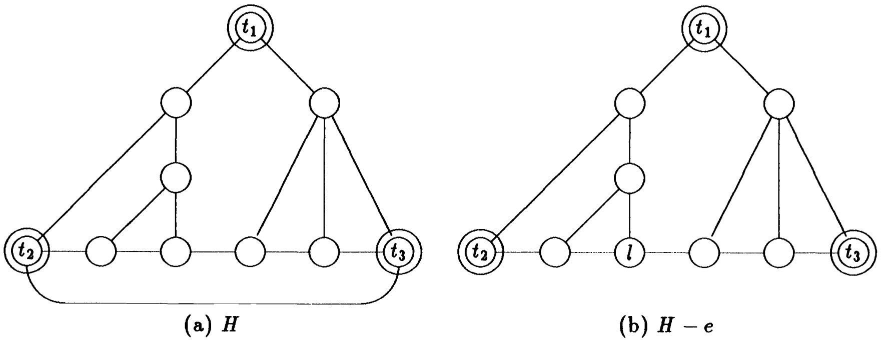


FIG. 9. *Halin and hypoHalin graphs.*

2.2. Deterministic decomposition. A finite terminal set is not by itself sufficient; to apply an algorithm to a graph, the decomposition tree for the graph must be known. That is, given a graph G, it must be easy to determine whether or not G is in the class, and if so to exhibit its decomposition tree. Therefore, a second useful property of a set of decomposition rules can be stated informally as follows.

> *Deterministic decomposition*: Membership in a recursive graph class can be determined by simply applying the recursive operations to decompose the candidate graph, until no remaining subgraph can be decomposed further.

To see why this property is important, we can examine Figs. 10 and 11, which illustrate the undesirable possibility of *nondeterministic* decomposition. Both operation

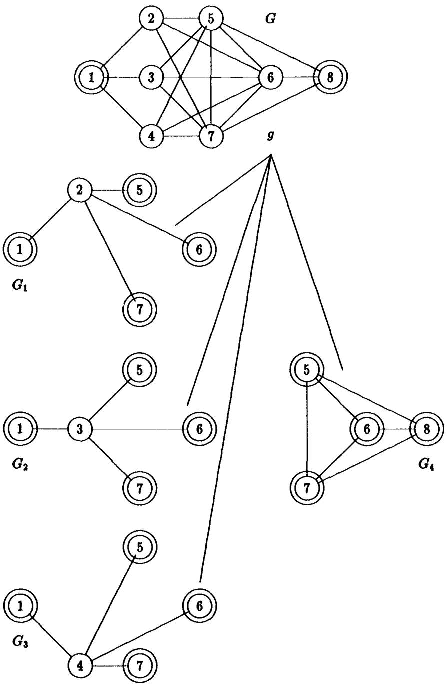


FIG. 10. *A* (4, 3, 5)*-operation g.*

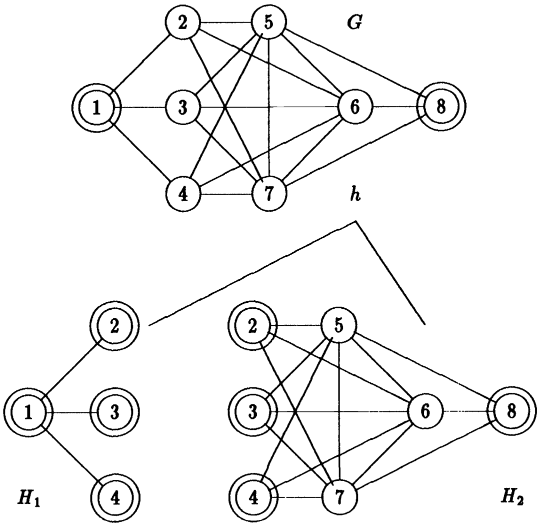


FIG. 11. *A* (4, 3, 5)*-operation h.*

g in Fig. 10 and operation h in Fig. 11 are (4, 3, 5)-operations; in fact, they decompose the same graph G. Each of G_1, G_2, G_3, G_4 decomposes easily into base graphs at the next step by additional (4, 3, 5)-operations. However, H_2 does not so decompose (that would require a (4, 3, 7)-operation), so the class containing G decomposes in a *nondeterministic* fashion.

When creating a decomposition tree, each step involves splitting a graph $G = (V, T, E)$ with $|T| \leqq k$ into several subgraphs by some (k, u, r)-operation. These subgraphs are entirely determined by the selection of up to u undistinguished nonterminal vertices, which then become terminals. We call the set D of size $|D| = d \leqq r$, i.e., the original terminals plus the coerced terminals, a *disconnecting set*, because its removal from the graph would leave the nonterminals of the various subgraphs in distinct connected components. More formally, denote the components of $G - D$ by $H_1, \cdots, H_m$. The (k, u, r)-operation is only applicable if each of the values $|\Gamma_G(H_i) \cap D| \leqq k$ (where Γ represents the neighborhood set). The operation decomposes G into $G_1, \cdots, G_m$, and possibly some components that are merely edges between terminals, where each G_i is obtainable from H_i by adding the terminals $\Gamma_G(H_i) \cap D$ and the edges from G which connect them to H_i.

As shown earlier, it is possible to encounter alternative choices for a disconnecting set; in fact, there may be a choice between one set of size d and another of size d'. Moreover, it seems entirely too restrictive to completely disallow such graph classes. It could be that whichever selection is made, a decomposition tree is still guaranteed to be found if the graph is a member of the class. The intent of deterministic decomposition, therefore, is not that the decomposition tree for each graph in the class must be unique, but rather that *some* decomposition tree will be found for such a graph through a top-down decomposition procedure without backtracking.

To motivate a precise definition, first observe that a graph's membership in a recursive class C depends on its specification of terminals. That is, there could be some graph $(V', T', E') \in C$ and vertices $x \in T'$, $y \notin T'$ such that $(V', T' - \{x\} \cup \{y\}, E') \notin C$. For example, the graph in Fig. 12(a) is series-parallel, but would *not* be if w and y were its terminals.

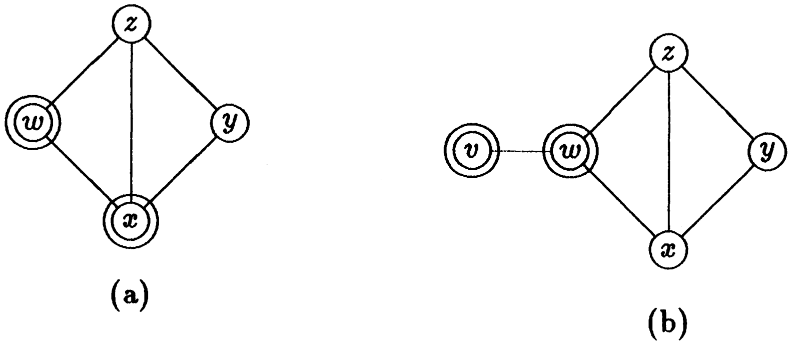


FIG. 12. *Two series-parallel graphs.*

Furthermore, it might be best to refrain from making decisions during decomposition, such as whether to include x or y in the set of terminals, if such a decision could be postponed until later when additional information might be available. This leads to a critical observation: operations that allow vertices to be added into the terminal set arbitrarily during decomposition lead to nondeterministic decomposition. To see this, suppose f is an operation such that some $x \in T_i$ but $x \notin T$, and x is not involved in any identification. Let $G_i = (V', T', E')$ where $T' = T_i$, and select $G_1, \cdots, G_{i-1}, G_{i+1}, \cdots, G_m \in C$ so that $G = f(G_1, \cdots, G_m) \in C$. But then $G = f(G_1, \cdots, G_{i-1}, H, G_{i+1}, G_m)$, where $H = (V', T' - \{x\} \cup \{y\}, E') \notin C$, so arbitrarily choosing H rather than G_i cannot lead to a decomposition tree.

The point can be made again with the class of series-parallel graphs, where the jackknife operation arbitrarily selects a vertex to become a new terminal. In Fig. 12(b), if x or z is chosen, the graph will decompose completely; but if y is chosen, one of the subgraphs is not series-parallel.

These outcomes indicate that there should be some minimality requirement on the size of the set D employed to split the graph, so that decisions regarding which vertices are to become terminals can be delayed until later during decomposition. Fortunately, we can develop the decomposition procedure to make certain intelligent decisions in order to eliminate some obviously inferior selections. Before precisely defining deterministic decomposition, it is useful to introduce the following terminology. Given a k-terminal recursive graph class $C(B, R)$, we say that a graph G is *prime* if there exist no $f \in R$ and k-terminal graphs $G_1, \cdots, G_m$ such that $G = f(G_1, \cdots, G_m)$. A prime $G \notin B$ (usually, this means that G is not a mere edge) is a *nontrivial* prime.

The above arguments motivate the following definition. A recursive class $C(B, R)$ satisfies *deterministic decomposition* if, for any decomposition of a graph $G \in C$ that selects only minimal disconnecting sets until each remaining element is prime, it is guaranteed that each such prime is trivial. (The minimality condition assures that no disconnecting set D will be chosen if there is any proper subset $D' \subseteq D$ that is also a disconnecting set, because otherwise the selection of D would contradict the intuition that the vertices of $D - D'$ should not be coerced into terminals prematurely.) Therefore the method used to create a decomposition tree can be described by the top-down procedure of Fig. 13. A graph G, when decomposed in this manner using the rule set of a deterministically decomposable class $C(R)$, leads to only K_2 subgraphs if and only if $G \in C(R)$.

During decomposition of a graph with n vertices, there are $O(n^u)$ possible disconnecting sets to examine for each (k, u, r)-operation in R. The deterministic decomposition

```
Let $D$ be a tree with one vertex $v$ such that $g_v = G$.
While $D$ has a leaf $v$ such that $g_v \notin B$ do
    If $g_v$ is prime then
        Report failure and abort.
    Else if $g_v = f(G_1, \ldots, G_m)$ uses a minimal disconnecting set then
        Let $f_v = f$.
        Create children $v_1, \ldots, v_c$ of $v$ with each $g_{v_i} = G_i$.
Report that $D$ is a decomposition tree for $G$.
```

FIG. 13. *Top-down decomposition procedure.*

property thus guarantees a polynomial time recognition algorithm. Moreover, the arity of a (k, u, r)-operation can be bounded by the constant 2^r, since the number of subsets of the disconnecting set is $2^d \leqq 2^r$; if more than one component receives exactly the same terminals during decomposition, temporarily leave these together as a single component, then perform parallel-type $(k, 0, k)$-operations as necessary to complete the decomposition step. In this manner we can guarantee that the multiplication tables used in the dynamic program will have bounded dimension, and hence each evaluation takes constant time. Therefore many $\mathcal{NP}$-hard problems are solvable in polynomial time with dynamic programming algorithms when instances are restricted to graph classes that satisfy deterministic decomposition. The problem is to determine what sets of operations yield recursive classes that are deterministically decomposable.

The following sections establish succinct conditions on a set of decomposition rules in order to satisfy the finite terminal and deterministic decomposition properties. These results provide some insight in response to the question of how far series-parallel graphs can be extended while preserving their desirable properties. For simplicity, we consider only recursive graph classes that are built from the single base graph K_2.

3. Operations that yield nondeterministic decomposition.

Our main theorems will distinguish between "good" and "bad" $[k, u, r]$-families. In particular, a graph class $C([k, u, r])$ satisfies deterministic decomposition if and only if $[k, u, r]$ has one of the "good" forms $[2, r-2, r]$, $[3, r-3, r]$, $[k, 0, k]$, or $[k, 1, k+1]$. The tree diagram in Fig. 14 illustrates the different cases to be considered in the proof of these theorems. Most "bad" cases arise from $s = k + u - r > 0$, i.e., from allowing a constituent graph to have more terminals than the composed graph.

We first introduce some notation. Given vertex sets U and W, let UW denote the edge set $\{\{u, w\} : u \in U, w \in W, u \neq w\}$. Thus UU denotes a clique on vertices U; in relevant figures to follow, we will often depict such a clique by merely encircling all the vertices of U. The following sequence of lemmas establishes that certain operations are "bad" by exhibiting graphs whose decomposition is nondeterministic with respect to those operations.

LEMMA 1. *$C([k, u, r])$ does not satisfy deterministic decomposition if $s = k + u - r > 0$, $3 \leqq k < r$, and $r \leqq 2k - 2$.*

Proof. Consider a graph $G = (V, T, E)$ defined by

- $V = T \cup A \cup B$,
- $T = \{t_1, \cdots, t_{r-u}\}$,
- $A = \{a_1, \cdots, a_{s+1}\}$,
- $B = \{b_1, \cdots, b_{\min(u,k-1)}\}$, and
- $E = \{t_1\}A \cup AB \cup BB \cup B(T - \{t_1\})$.

Figure 15 illustrates G for the case $k = 5$, $u = 4$, $r = 7$.

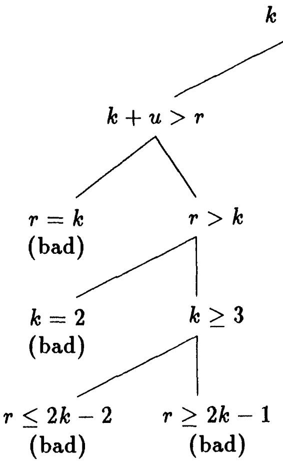


FIG. 14. *Outline of proof of theorems.*

Now, one minimal disconnecting set of G is $T \cup B$, which has cardinality $r - u + \min(u, k - 1) \leqq r$. It splits G into some edges and $s + 1$ stars, $G_i = (B \cup \{t_1, a_i\}, B \cup \{t_1\}, \{a_i\}(B \cup \{t_1\}))$, with $1 + \min(u, k - 1) \leqq k$ terminals apiece. Each star G_i then decomposes into edges at the next step, using a $(k, 1, k + 1)$-operation, which is in the family $[k, u, r]$.

But another minimal disconnecting set is $T \cup A$, which has cardinality $k + 1 \leqq r$. It leaves a subgraph $H = (V - \{t_1\}, A \cup T - \{t_1\}, E - \{t_1\}A)$, which has k terminals. Observe that $|V - \{t_1\}| = k + \min(u, k - 1) = \min(u + k, 2k - 1) \geqq r + 1$, so that no (k, u, r)-operation could completely decompose H into edges at the next step. Suppose without loss of generality that $b_i \notin D$, where D is the disconnecting set used at the next step. Then there exists a component containing b_i and at least $k + 1$ terminals. But this contradicts the bound $|T_i| \leqq k$ for (k, u, r)-operations. Thus no such disconnecting set D can exist, H is prime, and there is nondeterministic decomposition. □

LEMMA 2. *$C([k, u, r])$ does not satisfy deterministic decomposition if $s = k + u - r > 0$, $3 \leqq k < r$, and $r \geqq 2k - 1$.*

Proof. Consider a graph $G = (V, T, E)$ defined by

- $V = T \cup A \cup B$,
- $T = \{t_1, \cdots, t_{r-u}\}$,
- $A = \{a_1, \cdots, a_{s+1}\}$,

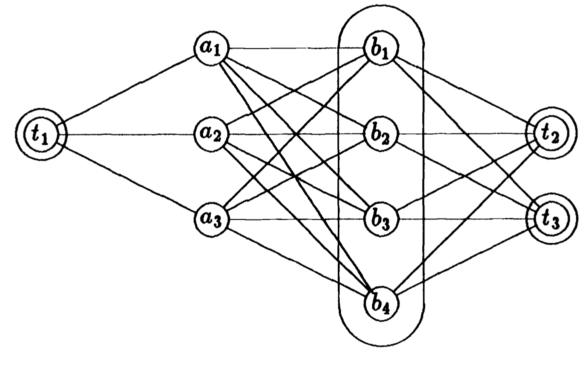


FIG. 15. *Graph for Lemma 1.*

- $B = \{b_1, \cdots, b_u\}$, and
- $E = \{t_1\}A \cup \{\{a_i, b_l\} : (l - i) \bmod u \leqq k - 2\} \cup BB \cup B(T - \{t_1\})$.

Figure 16 illustrates G for the case $k = 5$, $u = 6$, $r = 9$.

One minimal disconnecting set of G is $T \cup B$, which has cardinality r and which splits G into some edges and $s + 1$ stars G_i, each with a central nonterminal a_i adjacent to k terminals. Each star G_i easily decomposes into edges at the next step using a $(k, 1, k + 1)$-operation, which is in $[k, u, r]$.

Another minimal disconnecting set is $T \cup A$, which has cardinality $k + 1 \leqq r$. It leaves a subgraph $H = (V - \{t_1\}, A \cup T - \{t_1\}, E - \{t_1\}A)$, which has k terminals. Observe that $|V - \{t_1\}| = u + k \geqq r + 1$, so H cannot be completely decomposed into edges at the next step with any (k, u, r)-operation. Also note that G has been constructed such that each $A' \subset A$ satisfies $|\Gamma_H(A')| \geqq |A'| + 1$. Let D be the disconnecting set used at the next step, choose any $b_i \notin D$, and let A' be the set of vertices in A that are *not* included in the component containing b_i. So there exists a component containing b_i and at least $k + 1$ terminals $(T - \{t_1\}) \cup (A - A') \cup \Gamma_H(A')$, which contradicts the bound $|T_i| \leqq k$ for (k, u, r)-operations. Therefore no such disconnecting set D can exist, H is prime, and there is nondeterministic decomposition. □

LEMMA 3. *$C([2, r - 1, r])$ does not satisfy deterministic decomposition if $r \geqq 3$.*

Proof. Consider a graph $G = (V, \{t\}, E)$ defined by

- $V = A \cup \{t, b, c\}$,
- $A = \{a_1, \cdots, a_{r-1}\}$, and
- $E = \{t\}A \cup AA \cup \{\{a_1, b\}, \{a_2, b\}, \{b, c\}\}$.

Figure 17 illustrates G for the case $r = 5$.

One minimal disconnecting set of G is $\{t\} \cup A$, which leads to complete decomposition using (2, 1, 3)- and (2, 1, 2)-operations, each of which is in the family $[2, r - 1, r]$.

An alternative minimal disconnecting set of G is $\{t, b\}$, which leaves a subgraph $H = (V - \{c\}, \{t, b\}, E - \{\{b, c\}\})$. Observe that $|V - \{c\}| = r + 1$, so the disconnecting set D used at the next step cannot completely decompose H into edges. If $\{a_1, a_2\} \not\subset D$, then some component will contain at least three terminals t, b, a_i, where

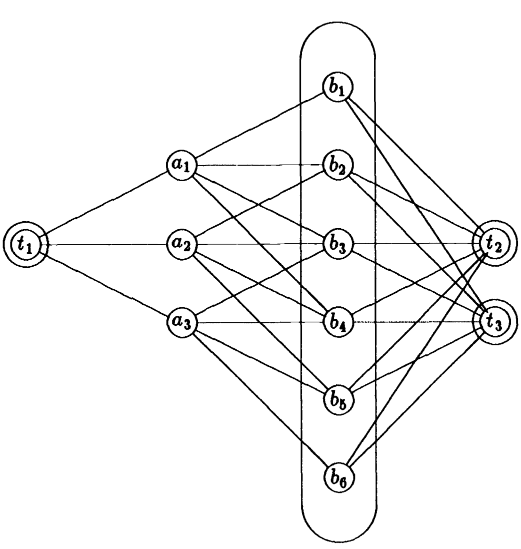


FIG. 16. *Graph for Lemma* 2.

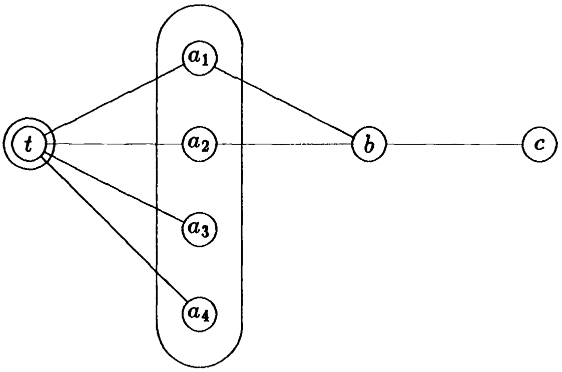


FIG. 17. *Graph for Lemma* 3.

$a_i \in D$. But if $\{a_1, a_2\} \subset D$, then some component will contain at least three terminals t, a_1, a_2. Each of these cases contradicts the bound of $|T_i| \leqq k = 2$ for $(2, r-1, r)$-operations, so no such disconnecting set D can exist, H is prime, and there is nondeterministic decomposition. □

LEMMA 4. *$C([k, u, k])$ does not satisfy deterministic decomposition if $u > 0$.*

Proof. Consider a graph $G = (V, T, E)$ defined by

- $V = T \cup A \cup \{b, c\}$,
- $T = \{t_1, \cdots, t_{k-u}\}$,
- $A = \{a_1, \cdots, a_u\}$, and
- $E = TA \cup AA \cup A\{b\} \cup \{b, c\}$.

Figure 18 illustrates G for the case $k = 9$, $u = 5$.

One minimal disconnecting set of G is $T \cup A$, which has cardinality k. It splits G into some edges and a graph $(A \cup \{b, c\}, A, \{b\}(A \cup \{c\}))$, which easily decomposes into edges at the next step by choosing the disconnecting set $A \cup \{b\}$. This disconnecting set yields a $(u, 1, u+1)$-operation, which is in the family $[k, u, k]$ because $1 \leqq u < k$.

An alternative minimal disconnecting set is $T \cup \{b\}$, which has cardinality $k - u + 1 \leqq k$ and which leaves a subgraph $H = (V - \{c\}, T \cup \{b\}, E - \{\{b, c\}\})$. Observe

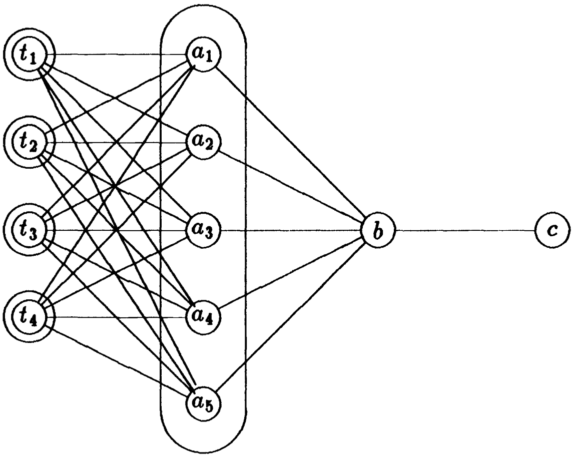


FIG. 18. *Graph for Lemma* 4.

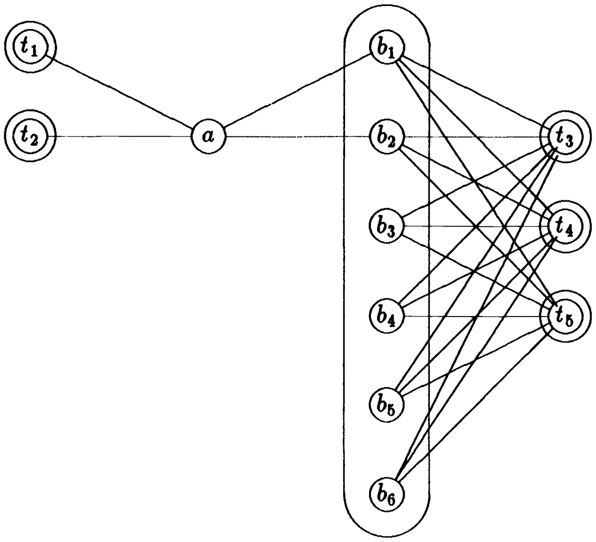


FIG. 19. *Graph for Lemma* 5.

that $|V - \{c\}| = k + 1$, so H cannot be completely decomposed into edges at the next step using any (k, u, k)-operation. In fact, no matter what sequence of disconnecting sets is used to decompose H, the operation that finally decomposes H into edges must have the form $(k', u', k + 1)$, which is not in $[k, u, k]$. Thus H does not decompose completely and there is nondeterministic decomposition. □

LEMMA 5. $C([k, r - k, r])$ *does not satisfy deterministic decomposition if* $4 \leqq k \leqq r - 2$.

Proof. Consider a graph $G = (V, T, E)$ defined by

- $V = T \cup \{a\} \cup B$,
- $T = \{t_1, \cdots, t_k\}$,
- $B = \{b_1, \cdots, b_{r-k+2}\}$, and
- $E = \{\{t_1, a\}, \{t_2, a\}, \{a, b_1\}, \{a, b_2\}\} \cup BB \cup B(T - \{t_1, t_2\})$.

Figure 19 illustrates G for the case $k = 5$, $r = 9$.

One minimal disconnecting set of G is $T \cup \{b_1, b_2\}$, which has cardinality $k + 2 \leqq r$. It splits G into some edges and two larger components, both of which decompose into edges at the next step, one by a $(k, r - k, r)$-operation, and the other by a $(4, 1, 5)$-operation which is in $[k, r - k, r]$ because $r \geqq 6$.

A second minimal disconnecting set is $T \cup \{a\}$; this choice leaves a subgraph $H = (V - \{t_1, t_2\}, T - \{t_1, t_2\} \cup \{a\}, E - \{\{t_1, a\}, \{t_2, a\}\})$, which has $k - 1$ terminals. Suppose D is the disconnecting set used at the next step, and let $|D| = d$. Observe that $|V - \{t_0, t_1\}| = r + 1$, so D cannot completely decompose H into edges using a $(k, r - k, r)$-operation. If $\{b_1, b_2\} \not\subset D$, then $d \geqq k$, the operation corresponding to D is a $(d, d - k + 1, d)$-operation, and $s = d + (d - k + 1) - d \geqq 1$. Otherwise $\{b_1, b_2\} \subset D$, $d \geqq k + 1$, the operation applied is a $(d - 1, d - k + 1, d)$-operation, and $s = (d - 1) + (d - k + 1) - d \geqq 1$. But any $(k, r - k, r)$-operation must have $s \leqq k + (r - k) - r = 0$, so neither of these two possibilities for D yields an operation in $[k, r - k, r]$. Hence no disconnecting set D can be employed, H is prime, and there is nondeterministic decomposition. □

4. Deterministically decomposable graph classes. Lemmas 1–5 indicate certain undesirable families of rules that can lead to nondeterministic decomposition. Our main result is that the exclusion of these rules is sufficient to guarantee deterministic decomposition. To establish this, some terminology and a technical lemma are needed.

We call a vertex subset S a *separating set* between two other vertex subsets U and W if every path from each $u \in U$ to each $w \in W$ must pass through some $s \in S$. As an example, $\{a\}$ is a separating set for $\{t_1, t_2\}$ and $\{b_1, b_2\}$ in the graph of Fig. 19. We also define the *path function* $P_G(X)$, where X is a subset of the nonterminals of a graph G, to be the maximum number of vertex-disjoint paths from not necessarily distinct vertices in X to distinct terminals of G. For instance, if G is the graph of Fig. 19, then $P_G(\{b_1, b_2\}) = 4$. Now Lemma 6 establishes a useful result about "good" recursive operations.

LEMMA 6. *If f is a $(k, r-k, r)$-operation such that either $k \leqq 3$ or $r \leqq k+1$, $G = f(\cdots, H, \cdots)$, and X is any subset of the nonterminals of H, then $P_G(X) = P_H(X)$.*

Proof. Obviously $P_G(X) \leqq P_H(X)$, because the disjoint paths used for G can be followed outward from X until distinct terminals of H are reached. Suppose $P_G(X) < P_H(X)$, and let D be the disconnecting set used by operation f. Then in the decomposition tree, H must have a sibling $H' = (V', T', E')$ with a (nonempty) separating set $S \subseteq V'$ between $T' \cap (D - T)$ and $T' \cap T$, such that $|S| < |T' \cap (D-T)|$ and $|S| < |T' \cap T|$. That is, S restricts the number of paths in G between vertices in X and terminals in $T' \cap T$, but the number of paths in H between X and $T' \cap (D - T)$ is not restricted by S because such paths do not pass through S.

Consider $D = T \cup \{b_1, b_2\}$ in Fig. 19. Let H be the component of G with vertices $B \cup \{t_3, t_4, t_5\}$, and let X be the nonterminals $\{b_3, \cdots, b_6\}$ of H. Then $S = \{a\}$ is the separating set in H', the other component.

Therefore

$$k \geqq |T'| \geqq |T' \cap (D-T)| + |T' \cap T| \geqq 2+2=4$$

and

$$u=r-k \geqq |T' \cap (D-T)| \geqq 2.$$

Thus if $k \leqq 3$ or $r \leqq k+1$, then $P_G(X) = P_H(X)$. □

The following theorem now establishes when decomposition is deterministic.

THEOREM 7. *The recursive class $C(R)$ satisfies deterministic decomposition if each family of operations in R has one of the following forms:*[1]

- $[2, r-2, r]$,
- $[3, r-3, r]$,
- $[k, 0, k]$, *or*
- $[k, 1, k+1]$.

Proof. Suppose each operation in R has one of these "good" forms, and let G_0 be a minimum size graph in $C(R)$ among those that decompose nondeterministically. Because $G_0 \in C(R)$, it possesses a decomposition tree DT. Also, by choice of G_0, there exists some disconnecting set D of G_0 and corresponding operation $h \in R$ such that $G_0 = h(\cdots, G_0', \cdots)$, where $G_0' \notin C(R)$.

It suffices to exhibit a decomposition tree DT' for G_0', and thereby reach a contradiction; we essentially just restrict the decomposition tree DT of G_0 to the subgraph G_0' of G_0. Each component graph at a node of DT' will be a subgraph of a component encountered in DT, where the components in DT' are possibly finer because the presence in DT' of terminals from D earlier than in DT can force additional splitting. Choose any nonterminal w of G_0'; we find the path in DT' from the root to the node at which w

[1] This statement resembles the well-known Church-Rosser theorem, which in the domain of lambda calculus asserts that if an expression has an equivalent normal form, it can be found by successive leftmost reductions.

becomes a terminal. Since such a path can be found for each nonterminal w using the same construction DT', this DT' completely decomposes G'_0.

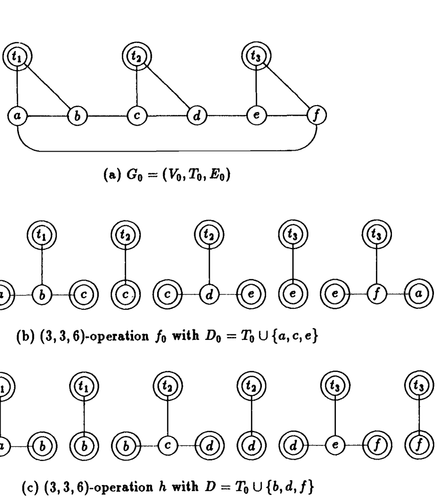
Consider the path in DT from the root to the node at which w becomes a terminal, say $G_i = f_i(\cdots, G_{i+1}, \cdots)$ for $0 \leqq i \leqq m-1$, where w is a nonterminal of G_{m-1} but a terminal of G_m. We next construct a corresponding path in DT', say $G'_i = f'_i(\cdots, G'_{i+1}, \cdots)$ for $0 \leqq i \leqq m-1$, such that w is a terminal in G'_m. Denote each $G_i = (V_i, T_i, E_i)$ and $G'_i = (V'_i, T'_i, E'_i)$. Suppose each $f_i = (k_i, r_i - k_i, r_i)$ employs a disconnecting set $D_i \supseteq T_i$; observe that each $k_i \geqq k_{i+1}$. Let $f'_i = (k'_i, r'_i - k'_i, r'_i)$ have disconnecting set $D'_i = (D_i \cup D) \cap V'_i$, where $D'_i \supseteq T'_i$. Hence $T'_i = (T_i \cup D) \cap V'_i$.

An example should aid in understanding the above notation. The graph $G_0 = (V_0, T_0, E_0)$ in the class $C([3, 3, 6])$, shown in Fig. 20(a), has two possible decompositions. One $(3, 3, 6)$-operation f_0 employs the disconnecting set $D_0 = T_0 \cup \{a, c, e\}$ to decompose G_0 into the components shown in Fig. 20(b). This operation f_0 is the first step in forming a decomposition tree DT for G_0; each of the nonprime components resulting from f_0 is easily decomposed into base graphs with a $(3, 1, 4)$-operation (which is in $[3, 3, 6]$) at the next level of DT. Another $(3, 3, 6)$-operation h utilizes the disconnecting set $D = T_0 \cup \{b, d, f\}$ to decompose G_0 into the components shown in Fig. 20(c). Let $G'_0 = (V'_0, T'_0, E'_0)$ be the component resulting from h in which $V'_0 = \{t_2, b, c, d\}$ and $T'_0 = \{t_2, b, d\}$, and select $w = c$. Consider the path in DT from the root to the operation that makes c into a terminal; the only operation on this path is f_0. The construction of a decomposition tree DT' for G'_0 thus begins with an operation f'_0 which uses a disconnecting set $D'_0 = (D_0 \cup D) \cap V'_0 = T'_0 \cup \{c\}$; obviously f'_0 is a $(3, 1, 4)$-operation (which again is in $[3, 3, 6]$).

It remains to show that each $f'_i \in R$ and that each f'_i either can be applied to decompose the corresponding G'_i or is essentially a null operation (i.e., $G'_i = G'_{i+1}$). For conciseness, we adopt the conventions that $U_i = T'_i - T_i$ (the portion of $D - T_i$ that is found in G'_i), $W_i = T_i - T'_i$ (the terminals of G_i that are separated from w by D), and S_i is a minimum cardinality separating set between U_i and W_i in G_i (see Fig. 21).

To verify that f'_i is in R, it suffices to show that $k'_i \leqq k_i$ and $r'_i - k'_i \leqq r_i - k_i$. But $k'_i = |T'_i|$, $k_i = |T_i|$, $r'_i = |D'_i|$, and $r_i = |D_i|$, so

$$r'_i - k'_i = |D'_i - T'_i| = |(D_i - T_i) \cap V'_i| \leqq |D_i - T_i| = r_i - k_i.$$



(a) $G_0 = (V_0, T_0, E_0)$

(b) (3, 3, 6)-operation f_0 with $D_0 = T_0 \cup \{a, c, e\}$

(c) (3, 3, 6)-operation h with $D = T_0 \cup \{b, d, f\}$

FIG. 20. *Two decompositions of a graph* $G_0 \in C([3, 3, 6])$.

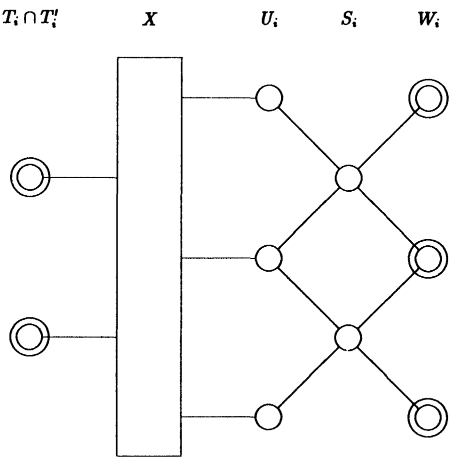


FIG. 21. *Illustration of U_i, S_i, W_i in graph G_i.*

Select the smallest l such that $k'_l > k_l$; the applicability of operation h to G_0 guarantees that any such $l \geqq 1$. If X denotes the set of nonterminals of G'_l, then $P_{G'_l}(X) > P_{G_l}(X)$ and thus $|U_l| > |S_l| \geqq 1$. Hence $r_0 - k_0 = |D - T_0| \geqq |U_l| \geqq 2$; referring to the allowable operations, we see that $k_0 \leqq 3$. But $k_0 \geqq k'_0 \geqq |U_l| + 1 \geqq 3$, so $k_0 = 3$, $|U_l| = 2$, and $|S_l| = 1$. But $|T'_l \cap T_0| \leqq 1$, for otherwise $k'_0 \geqq |U_l| + 2 \geqq 4$, which would deny the validity of h. Similarly at most one terminal from T_0 is reachable by a path from S_l without passing through U_l, or else h would again be invalid. Subject to the minimality requirement on the selection of disconnecting set D and the determination that $|S_l| = 1$, there is no place for a third terminal of T_0, so $k_0 \leqq 2$, a contradiction. Therefore $k'_l \leqq k_l$, as desired, which (with $r'_l - k'_l \leqq r_l - k_l$) implies $f'_l \in R$.

To verify that each f'_i either can be applied to decompose G'_i or is essentially a null operation (i.e., $G'_i = G'_{i+1}$), it suffices to show that each $k'_{i+1} \leqq k'_i$. Choose the smallest i such that f'_i is not applicable because G'_{i+1} has more terminals than its parent G'_i; i.e., $k'_{i+1} > k'_i$. Hence $P_{G'_{i+1}}(X) > P_{G'_i}(X)$, where X here denotes the set of nonterminals of G'_{i+1}. But every operation in R satisfies the conditions of Lemma 6, so $P_{G_i}(X) = P_{G_{i+1}}(X)$ by operation $f_i \in R$. Furthermore, $P_{G'_i}(X) = P_{G_i}(X)$ by choice of i, which when combined with the above gives $P_{G'_{i+1}}(X) > P_{G_{i+1}}(X)$.

Next observe that if the removal of some vertices from U_i could lead to a greater corresponding reduction in the cardinality of S_i, then S_i would not be a minimum separating set, because the elements removed from S_i could be replaced by those removed from U_i. This observation can be expressed as

$$|S_i| - |S_{i+1}| \leqq |U_i| - |U_{i+1}|.$$

Therefore

$$P_{G'_{i+1}}(X) - P_{G_{i+1}}(X) = |U_{i+1}| - |S_{i+1}| \leqq |U_i| - |S_i| = P_{G'_i}(X) - P_{G_i}(X) = 0,$$

which implies $P_{G'_{i+1}}(X) \leqq P_{G_{i+1}}(X)$, a contradiction, so each $k'_{i+1} \leqq k'_i$ and the proof is complete. □

As an example, the recursive class $C([2, 7, 9] \cup [3, 5, 8] \cup [5, 1, 6] \cup [7, 0, 7])$ satisfies deterministic decomposition. Combining Theorem 7 with the preceding lemmas yields the following result.

THEOREM 8. *The recursive class $C([k, u, r])$ satisfies deterministic decomposition if and only if $[k, u, r]$ has one of the following forms*:

- $[2, r-2, r]$,
- $[3, r-3, r]$,
- $[k, 0, k]$, *or*
- $[k, 1, k+1]$.

Proof. One direction is a consequence of Theorem 7. The other direction follows from Lemmas 1–5. □

Finally, a pair of results mentioned earlier can now be seen to follow easily from the theorems above.

COROLLARY 9. *The biconnected series-parallel graphs* (*series and parallel operations, but no jackknife*) *can be specified as* $C([2, 1, 3])$, *and thus satisfy deterministic decomposition*.

COROLLARY 10. *The Halin graphs are contained in* $C([3, 1, 4])$, *and thus satisfy deterministic decomposition*.

5. Conclusions.

Our results extend the popular class of series-parallel graphs while maintaining the property of deterministic decomposition. The following are some natural directions for further investigation.

- Identification of other existing or newly formed graph classes which are generalized by the (k, u, r)-operations.
- Examination of recursive classes for which B contains base graphs other than K_2.
- A complete characterization of sets of operation families that yield deterministic decomposition; that is, the discovery of a proof or a counterexample for the converse of Theorem 7. The technical difficulty lies in the complex interactions between several "bad" operations, or between "bad" and "good" operations, in the same rule set. While intuition might suggest that such a rule set is unlikely to lead to deterministic decomposition, it is possible that the ill effects of a "bad" operation could always be compensated for by the presence of another operation.

Acknowledgment. The authors acknowledge the very helpful comments of an anonymous referee, correcting several errors in an earlier version of this paper.

REFERENCES

[1] S. ARNBORG, *Efficient algorithms for combinatorial problems on graphs with bounded decomposability*, 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. Algebraic Discrete Meth., 8 (1987), pp. 277-284.

[3] S. ARNBORG AND A. PROSKUROWSKI, *Characterization and recognition of partial k-trees*, TRITA-NA-8402, Royal Institute of Technology, Sweden, 1984.

[4] ———, *Linear time algorithms for* NP-*hard problems on graphs embedded in k-trees*, TRITA-NA-8404, Royal Institute of Technology, Sweden, 1984.

[5] M. W. BERN, E. L. LAWLER, AND A. L. WONG, *Linear time computation of optimal subgraphs of decomposable graphs*, J. Algorithms, 8 (1987), pp. 216-235.

[6] R. J. DUFFIN, *Topology of series-parallel networks*, J. Math. Anal. Appl., 10 (1965), pp. 303-318.

[7] E. HARE, S. HEDETNIEMI, R. LASKAR, K. PETERS, AND T. WIMER, *Linear-time computability of combinatorial problems on generalized series-parallel graphs*, Discrete Algorithms and Complexity, 15 (1987), pp. 437-457.

[8] P. C. LIU AND R. C. GELDMACHER, *An $O(\max(m, n))$ algorithm for finding a subgraph homeomorphic to K_4*, in Proc. 11th Southeastern Conference on Combinatorics, Graph Theory, and Computing, Utilitas Math., Winnipeg, Canada, 1980, pp. 597-609.

[9] R. L. RARDIN, R. G. PARKER, AND D. K. WAGNER, *Definitions, properties, and algorithms for detecting*

series-parallel graphs, Tech. Report, Department of Industrial Engineering, Purdue University, West Lafayette, IN, 1982.

[10] M. B. RICHEY, *Combinatorial optimization on series-parallel graphs: algorithms and complexity*, Ph.D. thesis, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 1985.

[11] M. B. RICHEY AND R. G. PARKER, *On multiple Steiner subgraph problems*, Networks, 16 (1986), pp. 423–438.

[12] ———, *Minimum-maximal matching in series-parallel graphs*, European J. Oper. Res., 33 (1987), pp. 98–105.

[13] M. B. RICHEY, R. G. PARKER, AND R. L. RARDIN, *An efficiently solvable case of the minimum weight equivalent subgraph problem*, Networks, 15 (1985), pp. 217–228.

[14] N. ROBERTSON AND P. SEYMOUR, *Disjoint paths—a survey*, SIAM J. Algebraic Discrete Meth., 6 (1985), pp. 300–305.

[15] ———, *Graph minors—a survey*, Surveys in Combinatorics, Cambridge University Press, Cambridge, UK, 1985, pp. 153–171.

[16] K. TAKAMIZAWA, T. NISHIZEKI, AND N. SAITO, *Linear-time computability of combinatorial problems on series-parallel graphs*, J. Assoc. Comput. Mach., 29 (1982), pp. 623–641.

[17] J. A. WALD AND C. J. COLBOURN, *Steiner trees, partial 2-trees, and minimum* IFI *networks*, Networks, 13 (1983), pp. 159–167.

[18] T. V. WIMER, *Linear algorithms on k-terminal graphs*, Ph.D. thesis, Report No. URI-030, Clemson University, Clemson, SC, 1987.

[19] T. V. WIMER AND S. T. HEDETNIEMI, *K-terminal recursive families of graphs*, in Proc. 250th Anniversary Conference on Graph Theory, Fort Wayne, IN, 1986.

[20] T. V. WIMER, S. T. HEDETNIEMI, AND R. LASKAR, *A methodology for constructing linear graph algorithms*, Congr. Numer., 50 (1985), pp. 43–60.

SIAM J. DISC. MATH.
Vol. 4, No. 4, pp. 502–510, November 1991

004

THE DETECTION OF CHEATERS IN THRESHOLD SCHEMES*

E. F. BRICKELL† AND D. R. STINSON‡

Abstract. Informally, a (t, w)-*threshold scheme* is a way of distributing partial information (*shadows*) to w participants, so that any t of them can easily calculate a *key* (or *secret*), but no subset of fewer than t participants can determine the key. Presented in this paper is an unconditionally secure threshold scheme in which any cheating participant can be detected and identified with high probability by any honest participant, even if the cheater is in coalition with other participants. Also given is a construction that will detect with high probability a dealer who distributes inconsistent shadows (shares) to the honest participants. The scheme is not perfect; a set of $t - 1$ participants can rule out at most $1 + \binom{w-t+1}{t-1}$ possible keys, given the information they have. In this scheme, the key will be an element of $\mathrm{GF}(q)$ for some prime power q. Hence q can be chosen large enough so that the amount of information obtained by any $t - 1$ participants is negligible.

Key words. threshold scheme, secret sharing scheme

AMS(MOS) subject classifications. 94A60, 05B25

1. Introduction. Informally, a (t, w)-*threshold scheme* is a way of distributing partial information (*shadows*) to w participants, so that any t of them can easily calculate a *key* (or *secret*), but no subset of fewer than t participants can determine the key. Threshold schemes are also known as *secret sharing schemes*. A *perfect* threshold scheme is one in which no subset of fewer than t participants can determine any partial information regarding the key.

Threshold schemes were first described independently by Blakley [2] and Shamir [10] in 1979. Since then, many constructions have been given for threshold schemes. More recently, various researchers have considered the problem of guarding against the presence of cheaters in threshold schemes. It is conceivable that any subset of the participants may attempt to *cheat*, that is, to deceive any of the other participants by lying about the shadows they possess. There is also the possibility that the person distributing the shadows (the *dealer*) may attempt to cheat. The dealer might distribute an inconsistent set of shadows, so that the key cannot be determined correctly, or so that different subsets of t participants would calculate different keys from the shadows they possess. If this is done without the knowledge or cooperation of any of the participants, we refer to this form of cheating as *disruption*. However, if this cheating is done in cooperation with one or more of the participants, we call it *collusion*.

A threshold scheme is said to be *unconditionally secure* (against cheating) if the probability of successful cheating is limited to a specified probability even if the cheaters are assumed to have infinite computational resources. Under the assumption that the dealer is honest, several constructions have been given for threshold schemes that are unconditionally secure against cheating [3], [8], [11], [12]. We now briefly summarize the properties of these threshold schemes.

As far as we are aware, the first researchers to address the problem of cheaters in threshold schemes were McEliece and Sarwate in [8]. They use an error-correcting code

* Received by the editors September 12, 1988; accepted for publication (in revised form) October 26, 1990. A preliminary version of this paper appeared in Advances in Cryptology, CRYPTO 88 Proceedings, Lecture Notes in Computer Science, Vol. 403, Springer-Verlag, New York, Berlin, 1990, pp. 564–577.

† Sandia National Laboratories, Albuquerque, New Mexico 87185. This research was supported by U.S. Department of Energy contract DE-AC04-76DP00789.

‡ Computer Science and Engineering, University of Nebraska, Lincoln, Nebraska 68588. This research was supported by Natural Sciences and Engineering Research Council grant A9287 and by the Center for Communication and Information Science at the University of Nebraska.

to construct a threshold scheme in which any group of $t + 2e$ participants which includes at most e cheaters can correctly calculate the key.

Tompa and Woll [12] proceed as follows. The dealer specifies a subset K_0 of the set of possible keys K. A key will be accepted as authentic only if it is an element of K_0. If a set of t participants calculate the key to be an element of $K \backslash K_0$, then they realize that one of them is cheating. The probability of successful cheating is at most $1 - t|K_0|/|K|$, even if $t - 1$ participants conspire to cheat another participant. However, even though participants can detect when cheating has occurred, they cannot determine who is cheating.

The construction of Simmons [11] is more general in that it can be applied to most existing threshold schemes. This method detects cheating only if at least $t + 1$ participants exchange their shadows. Define a set S of at least t shadows to be *consistent* if all t-subsets of S determine the same key. Then a key is accepted as authentic only if there is a consistent subset of at least $t + 1$ shadows that determine it. If $t + e$ participants exchange shadows and there are at most $e - 1$ cheaters among them, then they possess a consistent subset of at least $t + 1$ shadows. Unfortunately, the only known method to determine the existence of a consistent set of $t + 1$ shadows is an exhaustive search.

Finally, Chaum [3] has suggested the following approach. For *each* bit b to be communicated to the ith participant, the dealer chooses $2w - 2$ large random numbers r_{j0} and r_{j1} $(1 \leqq j \leqq w, j \neq i)$. For each j, r_{j0} and r_{j1} are given to participant j. The dealer gives to the ith participant the bit b and all r_{jb} $(1 \leqq j \leqq w, j \neq i)$. Then r_{jb} is used to authenticate the bit b (as 0 or 1, respectively) to participant j. This procedure is used for every bit communicated to each participant.

In the schemes discussed above, it is assumed that the dealer is honest. Also, the Tompa and Woll scheme and the Simmons construction require that the participants be able to simultaneously release their shadows in order to ensure that no participant is able to obtain partial information about the shadows of the other participants before releasing his own shadow. Simultaneous release of shadows is *not* required in the Chaum scheme.

Verifiable secret sharing schemes were introduced by Chor et al. in [5]. These are threshold schemes that provide protection against dealer disruption and collusion. Other such schemes have been presented by Benaloh [1], Goldreich, Micali, and Wigderson [7], and Feldman [6]. These schemes provide *computational security* only, since they rely on computational assumptions regarding certain encryption schemes.

Chaum, Crepeau, and Damgard [4] use threshold schemes as a building block in unconditionally secure multiparty protocols. They tolerate both dealer disruption and collusion, but require that less than one third of the participants cheat. Under these assumptions, they describe a scheme that is unconditionally secure and which allows the key to be determined correctly by the honest participants.

The threshold scheme we present provides unconditional security and gives the honest participants the ability to *identify* cheaters, assuming the dealer is honest. Also, we do *not* require that the participants simultaneously release their shadows. The properties of our construction can be summarized as follows.

1. The key is an element of GF(q), and each shadow is a t-dimensional vector over GF(q) (q will be some large prime power).
2. Any participant who attempts to cheat will be identified by any honest participant with probability $1 - 1/(q - 1)$.
3. Even if there is only one honest participant and the remaining $w - 1$ participants form a coalition in order to deceive him, their probability of cheating successfully is only $(w - t + 1)/(q - 1)$.

4. The scheme is nearly perfect. A group of $t-1$ participants can eliminate at most $1+\binom{w-t+1}{t-1}$ possible keys and can obtain no other partial information about the key. If q is large, this will cause no difficulty in practice.
5. The scheme can also protect against dealer disruption, by using a "cut-and-choose" technique similar to that of [4].

Note that to obtain a desired level of security, it may be necessary to choose q to be quite large. If the number of desired secrets, say r, is small compared to q, then we can define a mapping ϕ from GF(q) to the set of secrets in such a way that for each secret s the cardinality of the inverse image of s, $|\phi^{-1}(s)|$, is approximately equal to q/r.

Independently and simultaneously, Rabin and Ben-Or [9] developed a perfect threshold scheme that has very similar properties to ours. Their scheme is also unconditionally secure, protects against dealer disruption, and allows the honest participants to identify cheaters. Their scheme requires each participant to receive $3w-2$ elements of GF(q) from the dealer (in private); while in our scheme, each participant receives only $w+2t-3$ elements of GF(q). However, the dealer requires less computational power in their scheme. They also extend their construction to protect against dealer collusion when the total number of cheaters is less than $w/2$.

2. The construction. Our construction is a modification of Blakley's threshold scheme [2], which we now review briefly. Suppose the participants are denoted A_i, $1 \leqq i \leqq w$, and the dealer is denoted by D. Let V be a t-dimensional vector space over GF(q), where q is some large prime power. First, D fixes a line L in V. This line is made known to all the participants. There are q possible keys, namely, the q points on L. If D wants to distribute shadows corresponding to a key p, he first constructs a random $(t-1)$-dimensional subspace H that meets L in a point. Then he constructs the hyperplane $H_p = H + p$. (Note that $H_p \cap L = p$.) Finally, he picks w random points on H_p, denoted s_i $(1 \leqq i \leqq w)$, such that the points in the set $\{p\} \cup \{s_i\colon 1 \leqq i \leqq w\}$ are in general position (that is, no j of them lie on a flat of dimension $j-2$, if $j \leqq t$). The point s_i is the shadow that D gives to A_i (see Fig. 1).

Any t participants can uniquely determine the hyperplane H_p and then obtain p by calculating $H_p \cap L = p$. However, a subset of t' $(< t)$ participants know only that H_p contains the flat F of dimension $t'-1$ generated by the shadows they possess. For any

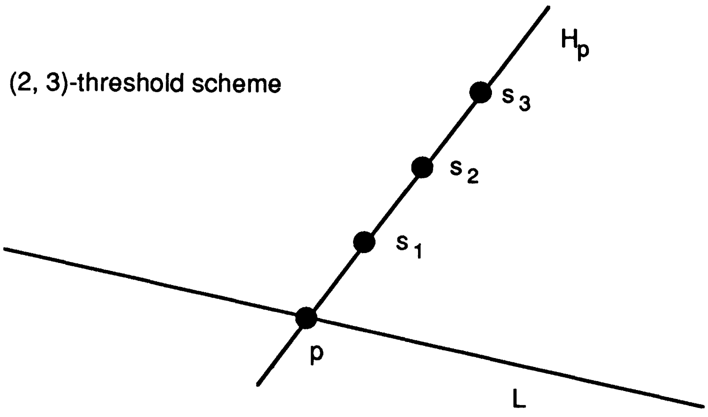


FIG. 1

p' on L, there is a hyperplane H_p containing F and p'. Hence they have no information as to the point p. Thus the scheme is indeed a (t, w)-threshold scheme.

To guard against cheating, we modify the threshold scheme. D will distribute extra information to the participants, along with the shadows. For ease of exposition we first discuss the case $t = 2$. In this case H is a 1-dimensional subspace, and the hyperplane H_p is a line. D constructs w random 1-dimensional subspaces, denoted H_i $(1 \leqq i \leqq w)$, each of which is distinct from H. We do *not* require that the subspaces H_i $(1 \leqq i \leqq w)$ be distinct. D gives to each A_j the $w - 1$ parallel lines $H_{ji} = H_j + s_i$, $1 \leqq i \leqq w$, $i \neq j$. These lines H_{ji} are called *supershadows*. Note that H_{ji} is given only to A_j. See Fig. 2.

We must first show that knowledge of the supershadows does not enable any one participant to determine the key. Let us consider A_1. He knows that $s_2 \in H_{12}$. This *does* give him some partial information, namely, that the key $p \neq H_{12} \cap L$. For if $p = H_{12} \cap L$, then $p = s_2$, which is not allowed. Similarly, A_1 knows that $p \neq H_{1i} \cap L$, for any i, $2 \leqq i \leqq w$. Also, $p \neq H_{11} \cap L$, where H_{11} denotes the line through s_1 parallel to the H_{1i}'s. For this would require that $H_p = H_{11}$, but $s_2 \notin H_{11}$. Thus A_1 has ruled out w possibilities for p. However, the key, p, could be any point p_0 on L other than these w points, since the line p_0s_1 will intersect each H_{1i} in a point not on L. Each of these $q - w$ possibilities for p is equally likely to occur.

Hence each participant can rule out w possibilities for the key and knows that the key is equally likely to be one of the $q - w$ remaining possibilities. Thus the scheme is no longer perfect. However, if q is large relative to w, this will cause no difficulty in practice. (A variation of this scheme, described in § 4, allows only one possible value to be ruled out for the key in the case $t = 2$.)

Next, we consider the possibility that certain participants will cheat, by lying as to what shadows they possess. In the worst case, $w - 1$ participants, say A_i $(2 \leqq i \leqq w)$ will form a coalition in order to try to convince A_1 that the key is some value $p' \neq p$. We will assume that $w \geqq 3$, so that the coalition can determine the line H_p and the key p before attempting to deceive A_1. Note that they can also calculate s_1, since $s_1 = H_p \cap H_{21}$, for example.

Suppose A_2 tells A_1 that his shadow is some point s_2' rather than s_2. A_2 will not choose s_2' to be any point on L, or any point on the line through s_1 parallel to L, since A_1 would then realize that A_2 is lying. Also, A_2 will not choose s_2' to be a point on H_p,

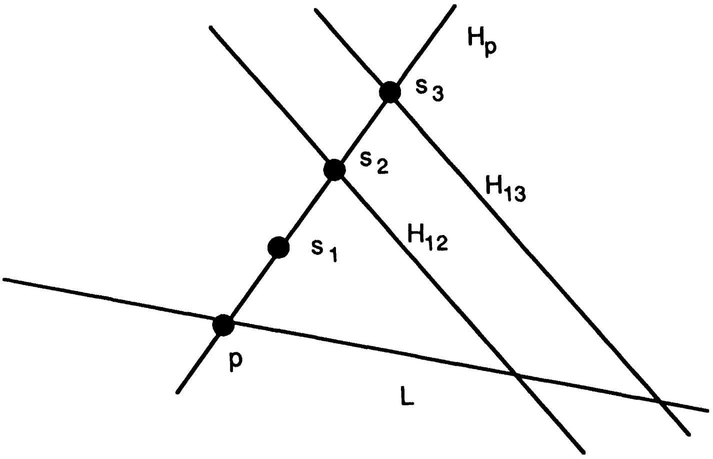


FIG. 2

since this would not deceive A_1 as to the value of p. Hence he will choose s'_2 to be one of the remaining $q^2 - 3q + 2$ points. For any such choice of s'_2, there is a unique line H'_{12} joining s'_2 and s_2. A_1 will be deceived if and only if $H'_{12} = H_{12}$. Since $H_{12} \neq H_p$, there are $q - 1$ possibilities for H_{12}, all equally likely. Each of these $q - 1$ lines through s_2 contains $q - 2$ of the $q^2 - 3q + 2$ points mentioned above. Thus the chance that A_2 deceives A_1 is $1/(q-1)$. See Fig. 3.

If all the other A_i $(2 \leqq i \leqq w)$ try independently to deceive A_1 in a similar fashion, the probability that at least one of them succeeds is $1 - (1 - (1/(q-1)))^{w-1} \leqq (w-1)/(q-1)$. Their best strategy is to conspire; if they ensure that no two of the lines $s'_i s_i$ are parallel, then A_1 will be deceived by one of them with probability equal to $(w-1)/(q-1)$. This will be a negligible quantity if q is large compared to w.

If $w = 2$, then the analysis is slightly different. Suppose A_2 attempts to deceive A_1. If A_2 can obtain the value of s_1, then the arguments proceed as before, and A_2 can deceive A_1 with probability $1/(q-1)$. (This could happen if A_1 reveals s_1 to A_2 before A_2 reveals s_2 to A_1, for example.) If A_2 cannot obtain the value of s_1, then his probability of deceiving A_1 is decreased to $1/q$, since he might choose s'_2 to be a point on H_p.

Let us now consider the general case $t \geqq 3$. Recall that H is a $(t-1)$-dimensional subspace and H_p is a hyperplane. D constructs w random $(t-1)$-dimensional subspaces, denoted H_i $(1 \leqq i \leqq w)$. We require that the intersection of H with $j - 1$ of these H_i's is a subspace of dimension $t - j$, if $j \leqq t$. (In the case $t = 2$ this condition reduces to the previous requirement that the H_i's $(1 \leqq i \leqq w)$ be distinct from H.) The $w - 1$ supershadows D gives to each A_j are the parallel hyperplanes $H_{ji} = H_j + s_i$, $1 \leqq i \leqq w$, $i \neq j$.

One way to select the H_i's is as follows. First, choose w subspaces of H, denoted K_i $(1 \leqq i \leqq w)$, each of dimension $t - 2$, in general position. Then select w points not in H, denoted q_i $(1 \leqq i \leqq w)$. These points need not be distinct. Finally, define H_i to be the subspace spanned by K_i and q_i $(1 \leqq i \leqq w)$.

First, we show that knowledge of the supershadows does not enable any $t - 1$ participants to determine the key. Suppose that participants A_i, $1 \leqq i \leqq t - 1$, attempt to determine the key. They know that H_p contains F, the $(t-2)$-dimensional flat generated by $s_1, \cdots, s_{t-1}$. They know also that a shadow s_j $(t \leqq j \leqq w)$ occurs on the line L_j which is the intersection of the H_{ij}, $1 \leqq i \leqq t-1$. (Since L_j meets H_p in a point, it has dimension

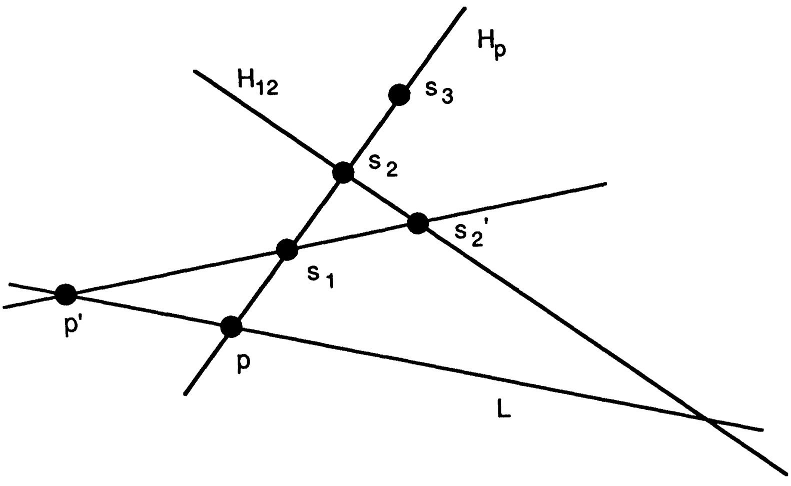


FIG. 3

one and is indeed a line.) Note that any two of these lines L_j are parallel, since the hyperplanes H_{ij} are parallel (for any fixed i).

We claim that for any j, $t \leqq j \leqq w$, L_j and F generate the whole n-dimensional space (consequently, $L_j \cap F = \emptyset$). This is seen as follows. Suppose L_j and F are contained in some hyperplane H', for some j, $t \leqq j \leqq w$. Since $s_j \in L_j$, $s_1, \cdots, s_{t-1} \in F$, and since the s_i's are in general position, we have that $H' = H_p$. Then $L_j \subseteq H_p \cap H_{1j} \cap H_{2j} \cap \cdots \cap H_{(t-1)j}$. It follows that $H \cap H_1 \cap H_2 \cap \cdots \cap H_{(t-1)}$ has dimension at least one, which is ruled out by the way in which the hyperplanes H_i were chosen.

Next, we observe that $F \cap L = \emptyset$. It is impossible that $L \subseteq F$ since $H_p \cap L = \{p\}$ and $F \subseteq H_p$. Also, F and L cannot intersect in a point, for this point would have to be p, which would contradict the requirement that the shadows are in general position with respect to p.

It is now easy to verify that there is a unique point p' on L such that the hyperplane determined by F and p' is parallel to each L_j, $t \leqq j \leqq w$. Then the key $p \neq p'$. For if $p = p'$, then $H_p \cap L_j = \emptyset$; but $s_j \in H_p \cap L_j$, a contradiction. This enables the participants A_i $(1 \leqq i \leqq t-1)$ to rule out one possible value for the key.

There are in fact other points that can be ruled out as possible values for the key. We saw earlier that when $t = 2$, the $w - 1$ points $L \cap L_j$ $(t \leqq j \leqq w)$ can also be eliminated as possible values for p. In general, the number of possible keys that can be ruled out (other than the point p') is $\binom{w-t+1}{t-1}$.

We can see this as follows. Let $j_1, \cdots, j_{t-1}$ be distinct integers such that $t \leqq j_i \leqq w$ $(1 \leqq i \leqq t-1)$, and let U be the flat spanned by the L_{j_i} $(1 \leqq i \leqq t-1)$. Since the lines L_j are all parallel, U has dimension at most $t - 1$. The flat T spanned by the points s_{j_i} $(1 \leqq i \leqq t-1)$ has dimension $t - 2$, and is contained in $U \cap H_p$. As well, $L_j \cap H_p = \{s_j\}$, for any j, $t \leqq j \leqq w$. It follows that the dimension of U is exactly $t - 1$ and $T = U \cap H_p$.

Next, we observe that it is impossible that $L \subseteq U$. Since $L \cap H_p = \{p\}$, this would force $p \in T$. But then the $t - 1$ shadows $s_{j_1}, \cdots, s_{j_{t-1}}$, and p would then be contained in the flat T having dimension at most $t - 2$. Hence either $L \cap U$ is empty, or $L \cap U$ is a point, say r. In the latter case, r cannot be the key, since (as before) the $t - 1$ shadows $s_{j_1}, \cdots, s_{j_{t-1}}$ and r would then be contained in the flat T. Hence, it is possible that $t - 1$ participants can rule out as many as $1 + \binom{w-t+1}{t-1}$ possible values for the key.

Example. Suppose we have a $(3, 5)$-threshold scheme over GF(q), for some large prime q. Suppose L is the line $(\beta, 0, 0)$ $(\beta \in \mathrm{GF}(q))$, $s_1 = (1, 1, 2)$ and $s_2 = (1, 1, 6)$. Thus F is the line $(1, 1, \beta)$ $(\beta \in \mathrm{GF}(q))$. Suppose also that L_3 is the line $(1 + \alpha, 3 - \alpha, 2)$ $(\alpha \in \mathrm{GF}(q))$, L_4 is the line $(1 + \alpha, -\alpha, 1)$, and L_5 is the line $(8 + \alpha, -\alpha, 3)$. (These three lines are parallel, having direction vector $(1, -1, 0)$.) A_1 and A_2 would analyze the situation as follows. Suppose the key is $p = (x_0, 0, 0)$. Then, H_p is the plane $x + y(x_0 - 1) = x_0$. This plane intersects L_3, L_4, and L_5 if and only if $x_0 \neq 2$. Thus $(2, 0, 0)$ is ruled out as the key. Three other points can also be ruled out. For example, L_3 and L_4 generate the plane U having equation $x + y - 3z = -2$. U meets L in the point $(-2, 0, 0)$. If -2 were the key, then H_p would have equation $x - 3y = -2$. Hence it would follow that $s_3 = (\frac{5}{2}, \frac{3}{2}, 2)$ and $s_4 = (\frac{1}{4}, \frac{3}{4}, 1)$ (all arithmetic being done in GF(q)). Then s_3, s_4, and p are all collinear, a contradiction. In a similar manner, -4 is ruled out by consideration of L_3 and L_5, and $-\frac{5}{2}$ is eliminated by consideration of L_4 and L_5.

The last topic we examine in this section is the probability of successful cheating. Suppose $w - 1$ participants, say A_i $(2 \leqq i \leqq w)$ form a coalition in order to try to convince A_1 that the key is some value $p' \neq p$. Their best strategy is to leave $t - 2$ of their shadows unchanged, and lie about the remaining $w - t + 1$ shadows. The probability that A_1 will not detect that any particular shadow is a forgery is $1/(q - 1)$, as in the $t = 2$ case. The chance that A_1 is fooled by at least one of the $w - t + 1$ altered shadows is at most $(w - t + 1)/(q - 1)$.

3. A cut-and-choose procedure to eliminate dealer disruption. We can eliminate the possibility of dealer disruption by using a *cut-and-choose* procedure, as in [4] and [1]. Let K be some security parameter (say $K = 50$). Suppose H_p is the hyperplane $a \cdot x = c$. The following protocol will be repeated K times.

1. D generates a random nonsingular matrix M and a random t-tuple b. D then computes $s'_i = s_i M^T + b$ and gives s'_i to A_i in private, $1 \leqq i \leqq w$ (the superscript "T" denotes transpose). So, the s'_i are obtained from the s_i by a random affine transformation.
2. Depending on a flip f of a three-sided coin, D performs (a), (b), or (c).

(a) If $f = 1$, then D reveals M and b, and each A_i verifies that $s'_i = s_i M^T + b$.

(b) If $f = 2$, then D computes $a' = aM^{-1}$ and $c' = c + a' \cdot b$, and reveals a' and c'. Then each A_i verifies that $a' \cdot s'_i = c'$.

(c) If $f = 3$, then D reveals all values s'_i, $1 \leqq i \leqq w$. Then any A_i can verify that no t-subset of $\{s'_i : 1 \leqq i \leqq w\}$ is on a flat of dimension at most $t - 2$.

If the dealer can answer all three challenges (a), (b), and (c), then it must be the case that $c = a \cdot s_i$, $1 \leqq i \leqq w$ (that is, the shadows all lie on a hyperplane) and that the shadows s_i are in general position. If the dealer attempts to cheat, he can answer at most two of the three challenges in any given round of the protocol. Hence the probability of the dealer fooling any given set of t honest participants after K rounds in $(\frac{2}{3})^K$.

It is also easy to see that no useful information is revealed to the participants by this protocol. If operation 2(a) is performed in any round of the protocol, then the participants learn only the affine transformation used in that round. This is of no use in determining the key. If 2(b) is performed, then the participants obtain the hyperplane $a' \cdot x = c'$. This tells them nothing about H_p, since any hyperplane can be mapped to any other hyperplane by means of an affine transformation.

Finally, let us suppose that operation 2(c) is performed, and a set of $t - 1$ participants, say A_i $(1 \leqq i \leqq t - 1)$ reveal to each other all their shadows. We will show that this information cannot disqualify any of the possible secrets that could not have been disqualified already. Define F to be the flat of dimension of $t - 2$ generated by the s_i $(1 \leqq i \leqq t - 1)$. Let r be a "guess" as to the value of the secret, and define H_r to be the hyperplane generated by F and r. (r should not be one of the values that the $t - 1$ participants can eliminate as a possible value of the secret using only their shadows and supershadows.)

As noted earlier, these $t - 1$ participants can compute, for each value j such that $t \leqq j \leqq w$, a line L_j such that $s_j \in L_j$. Any two of these lines L_j are parallel. They can then compute $r_j = L_j \cap H_r$, $t \leqq j \leqq w$. We will show that the possible list of shadows $(s_1, \cdots, s_{t-1}, r_t, \cdots, r_w)$ is compatible with the guess that r is the secret. That is, we prove that if the dealer is honest, then there must be an affine transformation which maps the ordered list of points $(s_1, \cdots, s_{t-1}, r_t, \cdots, r_w)$ to $(s'_1, \cdots, s'_{t-1}, s'_t, \cdots, s'_w)$.

Let f_1 denote an affine transformation which maps the ordered list of points $(s_1, \cdots, s_{t-1}, r_t)$ to $(s_1, \cdots, s_{t-1}, s_t)$. We claim that $f_1(r_j) = s_j$, for $t + 1 \leqq j \leqq w$. This is seen as follows. Let $t + 1 \leqq j \leqq w$. Note that $r_j = L_j \cap H_r$ and $s_j = L_j \cap H_p$. Since $r_j \in H_r$, we can write $r_j = \alpha_1 s_1 + \cdots \alpha_{t-1} s_{t-1} + \alpha_t r_t$, where $\sum_{1 \leqq i \leqq t} \alpha_i = 1$. Then $f_1(r_j) = \alpha_1 s_1 + \cdots + \alpha_{t-1} s_{t-1} + \alpha_t s_t = z$, say. We want to show that $z = s_j$. Note that $z - r_j = \alpha_t(s_t - r_t)$. Now r_t and s_t are both in the line L_t, r_j and s_j are both in the line L_j, and L_t and L_j are parallel. Therefore, it follows that $\alpha_t(s_t - r_t) = \beta_j(s_j - r_j)$ for some β_j, and hence $z \in L_j$. Furthermore, f maps H_r to H_p. Since $r_j \in H_r$, we must have $z \in H_p$. Then $z = L_j \cap H_p = s_j$, as desired.

We already know that the affine transformation chosen by D in step 1 of the protocol maps the ordered list of points $(s_1, \cdots, s_w)$ to $(s'_1, \cdots, s'_w)$. The composition of these two affine transformations is again an affine transformation, and it maps the ordered list of points $(s_1, \cdots, s_{t-1}, r_t, \cdots, r_w)$ to $(s'_1, \cdots, s'_{t-1}, s'_t, \cdots, s'_w)$.

It is possible to prove that this protocol is a perfect zero knowledge protocol and that it remains perfect zero knowledge even for a group of $t - 1$ participants. More precisely, it is possible to simulate executions of this protocol such that the simulated conversations among the dealer and $t - 1$ of the participants have an identical probability distribution to the actual conversations. From the analysis given above, such a simulator is straightforward to construct using standard techniques such as those used for the perfect zero knowledge protocol for graph isomorphism given in [7].

Note that we require the existence of a *broadcast channel* in step 2 of this protocol. This is a channel in which it is guaranteed that every participant receives the *same* information from the dealer (i.e. the values of M and b in 2(a); or a' and c' in 2(b); or the s'_i's in 2(c)). If a broadcast channel is not used, then the dealer could attempt to cheat during this protocol by giving different information to different participants.

We also observe that we can still obtain some protection against dealer disruption even if we never use operation 2(c). For a set of t participants can cooperate to check that the values s'_i they have received in any round of the protocol do not lie on a flat of dimension of dimension less than $t - 2$. This gives them no information as to the values of each others' shadows s_i, since any ordered list $(s_1, \cdots, s_1)$ can be mapped to $(s'_1, \cdots, s'_t)$ by an affine transformation.

We can also do a cut-and-choose procedure on the supershadows. Here the object is to convince each participant A_i that $s_j \in H_{ij}$, $i \neq j$, without revealing s_j. Suppose the hyperplane H_{ij} is given by the equation $a_i \cdot x = b_{ij}$, $1 \leqq i, j \leqq w$, $i \neq j$. The following protocol will be repeated K times.

1. For $1 \leqq j \leqq w$, D generates a random t-tuple s'_j, and gives s'_j to A_j. D then computes $b'_{ij} = a_i \cdot s'_j$ and gives b'_{ij} to A_i, $1 \leqq i, j \leqq w$, $i \neq j$.
2. Depending on a coin flip f, D performs (a) or (b).
   (a) If $f =$ "heads," then D reveals all s'_j, $1 \leqq j \leqq w$, and each A_i verifies that $b'_{ij} = a_i \cdot s'_j$.
   (b) If $f =$ "tails," then D reveals all $s_j + s'_j$, $1 \leqq j \leqq w$, and each A_i verifies that $a_i \cdot (s_j + s'_j) = b_{ij} + b'_{ij}$, $1 \leqq j \leqq w$.

The analysis of dealer disruption is similar to the previous situation. If the dealer can answer *both* challenges (a) and (b) in any given round of the protocol, then it must be the case that $a_i \cdot s_j = b_{ij}$, $1 \leqq i, j \leqq w$, $i \neq j$. That is, the shadow s_j lies on the hyperplane $a_i \cdot x = b_{ij}$. As before, the probability of the dealer fooling any t honest participants in all K rounds is 2^{-K}.

Next, we consider whether any information about the shadows is released by this protocol. As before, if operation 2(a) is performed in any round of the protocol, then clearly no information about the shadow is released. If operation 2(b) is done, then A_i learns all values $s_j + s'_j$, but this tells him nothing about any s_j.

Finally, observe that we require a broadcast channel in step 2, as in the previous protocol.

Although the protocol protects against dealer disruption, we cannot guard against collusion of the dealer and any participant. Suppose D colludes with participant A_1. D can tell A_1 all the supershadows H_{i1} and all the shadows s_i, $2 \leqq i \leqq w$. No collusion can be detected in the cut-and-choose procedure, since A_1 never reveals any information. Then suppose a group of t participants including A_1, say $\{A_i\colon 1 \leqq i \leqq t\}$, attempt to determine the key. A_1 can compute the intersection L_1 of the $t - 1$ hyperplanes H_{i1}, $2 \leqq i \leqq t$. Note that L_1 is a line. If A_1 claims that his shadow is any point on L_1 other than s_1, then the other $t - 1$ participants will not detect that he is cheating, and they will calculate an incorrect key. In this way, A_1 can make the other $t - 1$ participants believe the key is any value he desires.

4. Remarks. There are many variations of this threshold scheme. For example, the threshold scheme could be implemented in a projective space rather than in an affine space. In the case $t = 2$, less partial information is revealed in a projective setting. D would fix a line L in a projective plane P. As before, the key p would be a point on L. D also picks a random line H intersecting L in p and distributes points on $H \setminus \{p\}$ as the shadows. Supershadows are obtained as follows. For each participant A_i, D picks a point $q_i \in L \setminus \{p\}$ (these points need not be distinct). The supershadow H_{ij} is the line $s_j q_i$. With supershadows defined in this way, each participant A_i can only rule out the point q_i as the key (note that A_i can compute q_i as the intersection of any two of the supershadows he possesses).

Another question is the amount of computation required. The dealer must verify certain conditions, including that the shadows are in general position. This is not difficult for small t and w, but could require a lot of time if t and w are large. Is there an implementation of our scheme that is still computationally efficient for large t and w?

Yet another issue is the amount of (secret) information that needs to be communicated in the form of shadows and supershadows. We ask if a scheme can be constructed that requires less information to be distributed.

Finally, we ask if it is possible to construct a threshold scheme that provides unconditional security against collusion of the dealer and one or more participants.

Acknowledgments. This research was discussed with David Chaum and his colleagues at Centre for Mathematics and Computer Science (C.W.I.) Amsterdam. We thank them for their helpful observations and comments. We also thank Marijke De Soete for useful comments.

REFERENCES

[1] J. C. BENALOH, *Secret sharing homomorphisms: keeping shares of a secret secret*, Advances in Cryptology—CRYPTO 86 Proceedings, Lecture Notes in Computer Science, Vol. 263, Springer-Verlag, Berlin, 1987, pp. 251-260.

[2] G. R. BLAKLEY, *Safeguarding cryptographic keys*, Proc. National Computer Conference, AFIPS Conference Proceedings, 48 (1979), pp. 313-317.

[3] D. CHAUM, personal communication, 1988.

[4] D. CHAUM, C. CREPEAU, AND I. DAMGARD, *Multiparty unconditionally secure protocols*, Proceedings of the 20th ACM Symposium on the Theory of Computing, Chicago, IL, 1988, pp. 11-19.

[5] B. CHOR, S. GOLDWASSER, S. MICALI, AND B. AWERBUCH, *Verifiable secret sharing and achieving simultaneity in the presence of faults*, Proc. 26th IEEE Symposium on Foundations of Computer Science, Portland, OR, 1985, pp. 383-395.

[6] P. FELDMAN, *A practical scheme for noninteractive verifiable secret sharing*, Proc. 28th IEEE Symposium on Foundations of Computer Science, Los Angeles, CA, 1986, pp. 427-437.

[7] O. GOLDREICH, S. MICALI, AND A. WIGDERSON, *Proofs that yield nothing but their validity and a methodology of cryptographic protocol design*, Proc. 27th IEEE Symposium on Foundations of Computer Science, Toronto, 1986, pp. 174-187.

[8] R. J. MCELIECE AND D. V. SARWATE, *On sharing secrets and Reed-Soloman codes*, Comm. ACM, 24 (1981), pp. 583-584.

[9] T. RABIN AND M. BEN-OR, *Verifiable secret sharing and multiparty protocols with honest majority*, Proc. 21st ACM Symposium on Theory of Computing, 1989, pp. 73-85.

[10] A. SHAMIR, *How to share a secret*, Comm. ACM, 22 (1979), pp. 612-613.

[11] G. SIMMONS, *Robust shared secret schemes or "how to be sure you have the right answer even though you don't know the question,"* Congr. Numer., 68 (1989), pp. 215-248.

[12] M. TOMPA AND H. WOLL, *How to share a secret with cheaters*, J. Cryptol., 1 (1988), pp. 133-138.

SIAM J. DISC. MATH.
Vol. 4, No. 4, pp. 511-527, November 1991

ON THE EXISTENCE OF HAMILTONIAN CIRCUITS IN FAULTY HYPERCUBES*

MEE YEE CHAN† AND SHIANG-JEN LEE†

Abstract. The problem of finding Hamiltonian circuits in faulty hypercubes is explored. There are many different Hamiltonian circuits in a nonfaulty hypercube. The question of interest here is the following: if a certain number of links are removed from the hypercube, will a Hamiltonian circuit still exist? In partial answer to this question are the following results. First, it is shown that for any n-cube ($n \geqq 3$) with $\leqq 2n - 5$ link faults in which each node is incident to at least two nonfaulty links, there exists a Hamiltonian circuit consisting of only nonfaulty links. Since as will be shown, there exists an n-cube with $2n - 4$ faulty links, in which each node is incident to at least two nonfaulty links, for which there is no Hamiltonian circuit, this result is optimal. Second, it is shown that the problem of determining whether an n-cube with an arbitrary number of link faults has a Hamiltonian circuit is NP-complete.

Key words. hypercubes, Hamiltonian circuits, embeddings, fault-tolerance, NP-complete

AMS(MOS) subject classifications. 05C10, 05C38, 68M10, 68M15, 68R05, 68P10

1. Introduction. In this paper, we explore the problem of finding Hamiltonian circuits in faulty hypercubes. There are many different Hamiltonian circuits in a nonfaulty hypercube. The question of interest here is the following: if a certain number of links is removed from the hypercube, will a Hamiltonian circuit still exist?

The motivation behind this question rests with the recent work on embedding networks such as rings, grids, and trees in hypercubes [BCLR], [BI], [BMS], [C1], [C2], [C3], [HJ], [G], [SS], [W]. Such embeddings demonstrate the ability of the hypercube parallel computer architecture to simulate a wide range of other topologies. In a binary hypercube of dimension n, or n-cube, there are 2^n nodes and $n2^{n-1}$ links. With so many nodes and links, fault tolerance and fault tolerant embeddings are issues. Many researchers have gone to great lengths to show the robustness and fault tolerance of the hypercube, focusing on the hypercube's ability to route and reconfigure itself despite faults [B], [CL1], [CL2], [HLN1], [HLN2], [PM]. The problem addressed in this paper is related to the fault-tolerant embedding of rings in hypercubes.

In partial answer to our question, we have the following results. First, we show that for any n-cube ($n \geqq 3$) with $\leqq 2n - 5$ link faults in which each node is incident to at least two nonfaulty links, there exists a Hamiltonian circuit consisting of only nonfaulty links. Since, as we will see, there exists an n-cube with $2n - 4$ faulty links, in which each node is incident to at least two nonfaulty links, for which there is no Hamiltonian circuit, this result is optimal. Second, we show that the problem of determining whether an n-cube with an arbitrary number of link faults has a Hamiltonian circuit is NP-complete. These two results are presented in §§ 2 and 3, respectively.

A binary n-cube can be viewed as an undirected graph of 2^n nodes, each node labelled with a unique n-bit string. There is a link between two nodes if and only if their labellings differ in exactly one bit position. Two nodes whose labels disagree in exactly one bit position d are said to be *neighbors across dimension* d, and the link between them is said to be *on dimension* d, where dimension 1 corresponds to the leftmost bit. For convenience in the rest of the presentation, we will refer to the neighbor of node u across dimension d as $N(u, d)$ and the link on dimension d incident with u as $L(u, d)$.

* Received by the editors October 10, 1989; accepted for publication (in revised form) September 4, 1990.

† Computer Science Program, The University of Texas at Dallas, Richardson, Texas 75083.

Links will also be denoted by strings of length n over $\{0, 1, *\}$ with exactly one *. For example, *0100 denotes the link between nodes 00100 and 10100 in a 5-cube. We also use strings of length n over $\{0, 1, *\}$ to denote subcubes of an n-cube; a string with exactly m *'s describes a m-cube. For example, *01** denotes the 3-cube involving the eight nodes 00100, 00101, 00110, 00111, 10100, 10101, 10110, and 10111 of a 5-cube.

2. Hypercubes with a limited number of faults. In order to have a Hamiltonian circuit, each node would have to be incident to at least two nonfaulty links. We begin with a lemma that says that any n-cube with $\leqq n - 2$ link faults has a Hamiltonian circuit. This sets the style of proof for later lemmas that ultimately guarantee the existence of a Hamiltonian circuit in an n-cube with $\leqq 2n - 5$ faults, where each node is incident to at least two nonfaulty links.

LEMMA 1. *In an n-cube with* $\leqq n - 2$ *link faults, for any node S, there exists a fault-avoiding Hamiltonian path P from S to* $N(S, 1)$.

Proof. The proof is by induction on n.

Induction Basis. For $n < 3$, there are no faults. Consider $n = 3$. One of the paths depicted in Fig. 1 will avoid a single link fault.

Induction Step. As the induction hypothesis, we assume that the lemma is true for $n < k$. To prove that the lemma is true for $n = k$, first let F denote the set of faults in the k-cube and let $S = s_1 s_2 \cdots s_k$. There exists a dimension i on which not all of the faults in F agree. (By the way, faults *0100 and 1*000 do not agree on dimensions 1, 2, and 3.) This means that subcubes $* \cdots * s_i * \cdots *$ and $* \cdots * \bar{s}_i * \cdots *$ each have $\leqq k - 3$ faults.

Case I. $i = 1$.

With $|F| \leqq k - 2$, there exists a dimension $j \neq 1$ such that $L(N(S, j), 1)$ is nonfaulty. By the induction hypothesis, there exists a fault-avoiding Hamiltonian path P' from S to $N(S, j)$ in $s_1 ** \cdots *$ and a fault-avoiding Hamiltonian path P'' from $N(N(S, j), 1)$ to $N(S, 1)$ in $\bar{s}_1 ** \cdots *$. The path P consists of P', $L(N(S, j), 1)$ and P''.

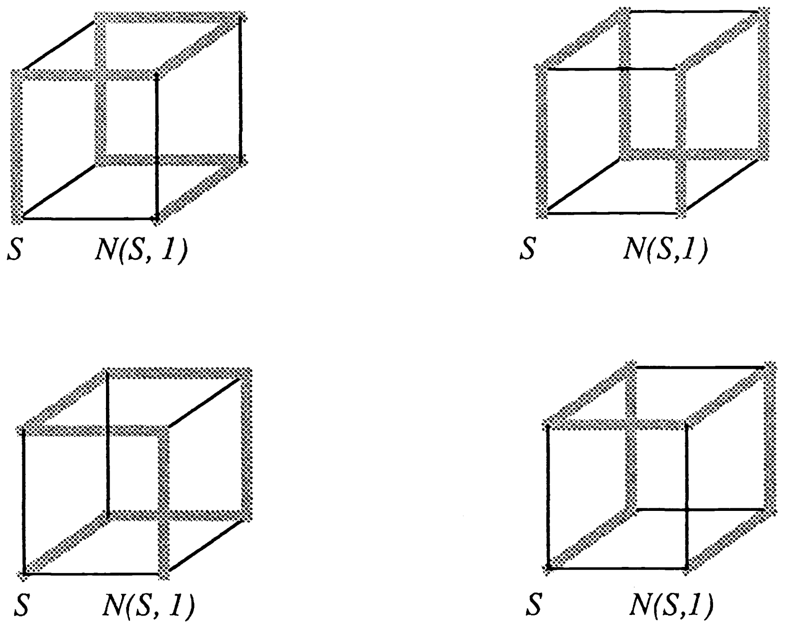


FIG. 1

Case II. $i \neq 1$.

Suppose, without loss of generality, that $i = 2$. By the induction hypothesis, there exists a fault-avoiding Hamiltonian path P' from S to $N(S, 1)$ in $*s_2** \cdots *$. Let the sequence of nodes in path P' be $S \equiv u_1, u_2, \cdots, u_{2^{k-1}} \equiv N(S, 1)$. Since $|F| \leqq k - 2$, there exists a node u_i in P' such that $L(u_i, 2)$ and $L(u_{i+1}, 2)$ are both nonfaulty. By the induction hypothesis, there exists a fault-avoiding Hamiltonian path P'' from $N(u_i, 2)$ to $N(u_{i+1}, 2)$ in $*\bar{s}_2** \cdots *$. The path P consists of the segment of P' from S to u_i, $L(u_i, 2)$, path P'' from $N(u_i, 2)$ to $N(u_{i+1}, 2)$, $L(u_{i+1}, 2)$, and the segment of P' from u_{i+1} to $N(S, 1)$.

The next three lemmas concern faulty 4-cubes and 5-cubes. They contribute to the induction basis for our inductive proof of the existence of a Hamiltonian circuit in a hypercube with up to $2n - 5$ faulty links. Interestingly, as the size of the hypercube increases, the result becomes easier to prove. The most complicated case is the 5-cube; this is our reason for separating out the 5-cube case. Both Lemmas 3 and 4 contribute to the argument about the existence of a Hamiltonian circuit in a 5-cube with up to 5 faulty links, each node incident to at least two nonfaulty links.

LEMMA 2. *In a* 4-*cube with* $\leqq 3$ *link faults where each node is incident to at least two nonfaulty links, for any node* S, *there exists a fault-avoiding Hamiltonian path* P *from* S *to* $N(S, 1)$.

Proof. The proof is divided into two major cases. Let $S = s_1s_2s_3s_4$.

Case I. There exists a dimension i such that removing all links on i will result in two 3-cubes with $\leqq 1$ fault each.

(a) $i = 1$.

If there exists a dimension $j \neq 1$ such that $L(N(S, j), 1)$ is nonfaulty, then the path P consists of the fault-avoiding Hamiltonian path from S to $N(S, j)$ in s_1***, $L(N(S, j), 1)$ and the fault-avoiding Hamiltonian path from $N(N(S, j), 1)$ to $N(S, 1)$ in $\bar{s}_1***$. Otherwise, the three faults must be $*\bar{s}_2s_3s_4$, $*s_2\bar{s}_3s_4$ and $*s_2s_3\bar{s}_4$, in which case, the path shown in Fig. 2 will avoid the faulty links.

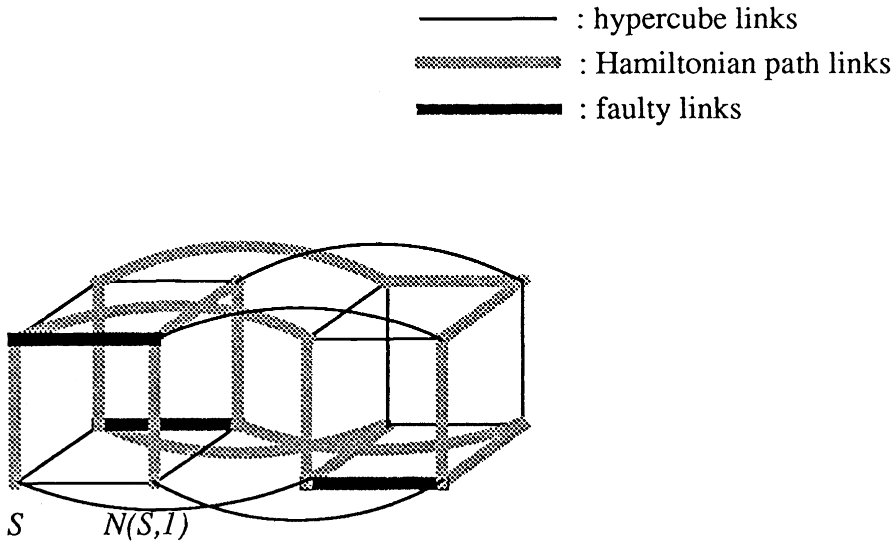


FIG. 2

(b) $i \neq 1$.

Suppose, without loss of generality, $i = 2$. There exists a fault-avoiding Hamiltonian path P' from S to $N(S, 1)$ in $*s_2**$. Let the sequence of nodes in P' be $S \equiv u_1, u_2, \cdots, u_7, u_8 \equiv N(S, 1)$. Since there are $\leqq 3$ faults, there exists a u_i in P' such that $L(u_i, 2)$ and $L(u_{i+1}, 2)$ are both nonfaulty. The path P consists of the segment of P' from S to u_i, $L(u_i, 2)$, the fault-avoiding Hamiltonian path from $N(u_i, 2)$ to $N(u_{i+1}, 2)$ in $*\bar{s}_2**$, $L(u_{i+1}, 2)$, and the segment of P' from u_{i+1} to $N(S, 1)$.

Case II. Otherwise.

The faulty 4-cube must be isomorphic to the one shown in Fig. 3. With the help of the fault-avoiding Hamiltonian paths and circuits depicted in Fig. 4 for this faulty 4-cube, the reader can easily verify that, for each pair of adjacent nodes in the hypercube, there exists a fault-avoiding Hamiltonian path starting at one of the two nodes and ending at the other. If, in the isomorphism, S is mapped to node A and $N(S, 1)$ is mapped to node B, we would be particularly interested in the fault-avoiding Hamiltonian path between A and B, where A and B are the two end nodes of one of the nonfaulty links of the Hamiltonian circuit described in Fig. 4(a), 4(b), or 4(c), or the faulty links described in Fig. 4(d), 4(e), or 4(f).

LEMMA 3. *In a 5-cube with $\leqq 5$ link faults where each node is incident to at least three nonfaulty links, for any node S, there exists a fault-avoiding Hamiltonian path P from S to $N(S, 1)$.*

Proof. In a 5-cube satisfying the hypothesis, it is not difficult to see that there is a dimension i whose removal splits the 5-cube into two 4-cubes, each with at most 3 faults. From here, the proof is divided into two major cases. Let $S = s_1s_2s_3s_4s_5$.

Case I. There exists such a dimension $i \neq 1$.

Since each node is incident to at least three nonfaulty links in the 5-cube, within both 4-cubes, each node will be incident to at least two nonfaulty links. Suppose, without loss of generality, $i = 2$. There exists a fault-avoiding Hamiltonian path P' from S to $N(S, 1)$ in $*s_2***$. Let the sequence of nodes in P' be $S \equiv u_1, u_2, \cdots, u_{15}, u_{16} \equiv N(S, 1)$. Since there are $\leqq 5$ faults, there exists a u_i in P' such that $L(u_i, 2)$ and $L(u_{i+1}, 2)$ are both nonfaulty. The path P consists of the segment of P' from S to u_i, $L(u_i, 2)$, the fault-avoiding Hamiltonian path from $N(u_i, 2)$ to $N(u_{i+1}, 2)$ in $*\bar{s}_2***$, $L(u_{i+1}, 2)$, and the segment of P' from u_{i+1} to $N(S, 1)$.

Case II. Otherwise.

Removing the links on dimension 1 must result in two 4-cubes with $\leqq 3$ faults each. Again, within each 4-cube, each node will be incident to at least two nonfaulty links. If there exists a dimension $j \neq 1$ such that $L(N(S, j), 1)$ is nonfaulty, then the path P consists of the fault-avoiding Hamiltonian path from S to $N(S, j)$ in s_1****, $L(N(S, j), 1)$ and the fault-avoiding Hamiltonian path from $N(N(S, j), 1)$ to $N(S, 1)$ in $\bar{s}_1****$. Otherwise, the set of faults must be either $\{*\bar{s}_2s_3s_4s_5, *s_2\bar{s}_3s_4s_5, *s_2s_3\bar{s}_4s_5,$

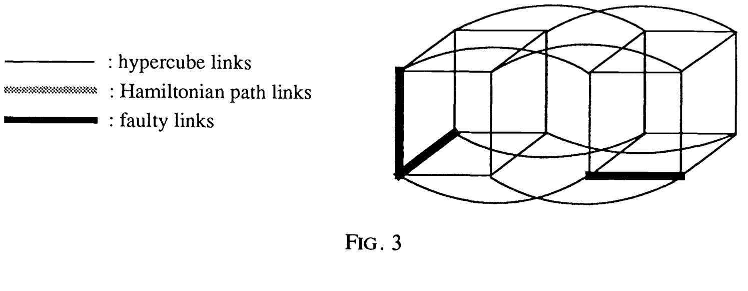
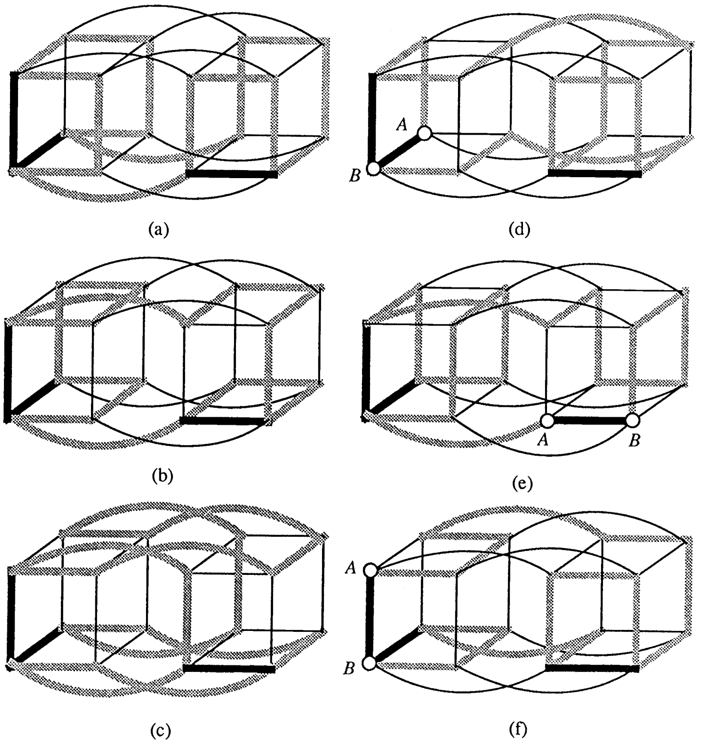


FIG. 3

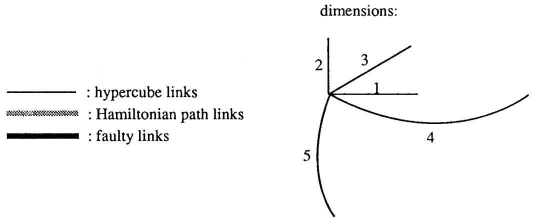


(a) (d)

(b) (e)

(c) (f)

FIG. 4



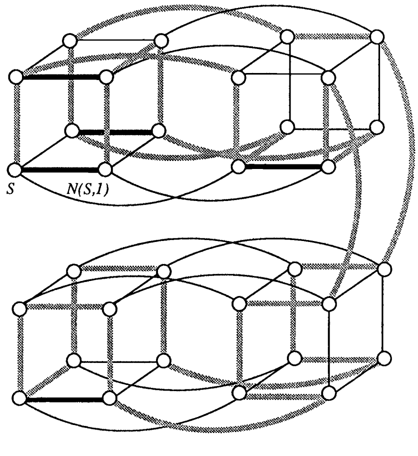


FIG. 5

$^*s_2s_3s_4\bar{s}_5\}$, or $\{^*\bar{s}_2s_3s_4s_5, {}^*s_2\bar{s}_3s_4s_5, {}^*s_2s_3\bar{s}_4s_5, {}^*s_2s_3s_4\bar{s}_5, {}^*s_2s_3s_4s_5\}$. The path shown in Fig. 5 will avoid these faults.

LEMMA 4. *In a 5-cube with $\leqq 5$ link faults where each node is incident to at least two nonfaulty links, for any node S, there exists a fault-avoiding Hamiltonian path P from S to $N(S, 1)$.*

Proof. If each node is incident to at least three nonfaulty links, we can apply Lemma 3. So without loss of generality, suppose some node is incident to three faulty links. The proof is divided into two major cases.

Case I. There exists a dimension i such that removing all links on i will result in two 4-cubes with $\leqq 3$ faults each, and within each 4-cube, every node is incident to at least two nonfaulty links.

(a) $i = 1$.

Since there will be $\leqq 3$ link faults on any one dimension, there exists a dimension $j \neq 1$ such that $L(N(S, j), 1)$ is nonfaulty. Thus, the path P consists of the fault-avoiding Hamiltonian path from S to $N(S, j)$ in $s_1{}^{****}$, $L(N(S, j), 1)$ and the fault-avoiding Hamiltonian path from $N(N(S, j), 1)$ to $N(S, 1)$ in $\bar{s}_1{}^{****}$.

(b) $i \neq 1$.

Suppose, without loss of generality, $i = 2$. There exists a fault-avoiding Hamiltonian path P' from S to $N(S, 1)$ in $^*s_2{}^{***}$. Let the sequence of nodes in P' be $S \equiv u_1, u_2, \cdots, u_{15}, u_{16} \equiv N(S, 1)$. Since there are $\leqq 5$ faults, there exists a u_i in P' such that $L(u_i, 2)$ and $L(u_{i+1}, 2)$ are both nonfaulty. The path P consists of the segment of P' from S to u_i, $L(u_i, 2)$, the fault-avoiding Hamiltonian path from $N(u_i, 2)$ to $N(u_{i+1}, 2)$ in $^*\bar{s}_2{}^{***}$, $L(u_{i+1}, 2)$, and the segment of P' from u_{i+1} to $N(S, 1)$.

Case II. Otherwise.

Without loss of generality, suppose that the node incident to three faulty links is 00000, and these three faulty links are 00*00, 000*0, and 0000*. In order not to fall into Case I, the remaining two faulty links must not have a 1 or a * in dimension 3, 4, or 5. At the same time, since each node is incident to at least two nonfaulty links, the two remaining faulty links must not be incident to 00000. Thus, the two remaining faulty links must be 1*000 and *1000, and so, the faulty 5-cube must be isomorphic to the one shown in Fig. 6. Figure 7 depicts fault-avoiding Hamiltonian paths and circuits within the top half of this faulty 5-cube. Note that the bottom half is a 4-cube with no faults. With the help of Fig. 7, the reader can verify that, for each pair of adjacent nodes in the hypercube, there exists a fault-avoiding Hamiltonian path starting at one of the two nodes and ending at the other. If, in the isomorphism, S is mapped to node A and $N(S, 1)$ is mapped to node B, we would be particularly interested in the fault-avoiding Hamiltonian path between adjacent pair A and B, where A and B are the two end nodes of one of the nonfaulty links of the Hamiltonian circuit described in Fig. 7(a), 7(b), or 7(c), or the faulty links described in Fig. 7(d), 7(e), 7(f), or 7(g). For example, with A and B as marked in Fig. 6, Fig. 7(a) shows a fault-avoiding Hamiltonian path P' from A to B in the top 4-cube. A Hamiltonian path P'' exists from X' to Y' in the bottom 4-cube. The path consisting of the link from A to X, the link from X to X', path P'', the link from Y' to Y, and the segment of P' from Y to B is a fault-avoiding Hamiltonian path from A to B in the faulty 5-cube.

Finally, we have the proof of our first theorem.

THEOREM 1. *In an n-cube with $\leqq 2n - 5$ link faults, where each node is incident to at least two nonfaulty links and $n \geqq 3$, for any node S, there exists a fault-avoiding Hamiltonian path P from S to $N(S, 1)$.*

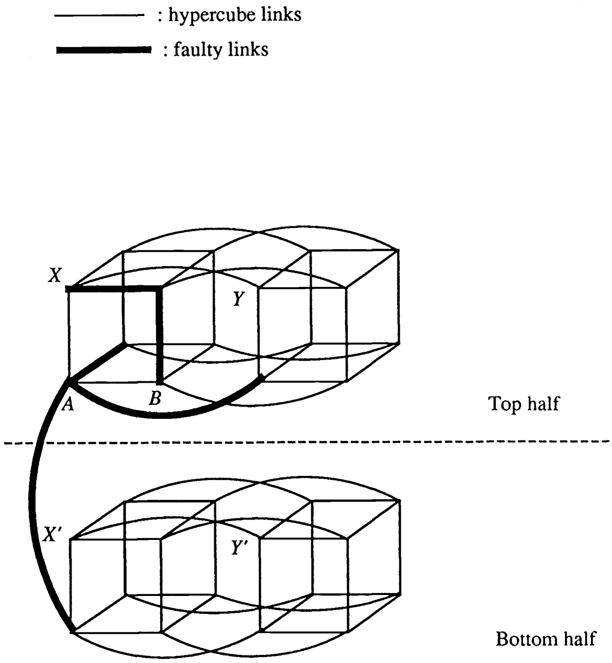
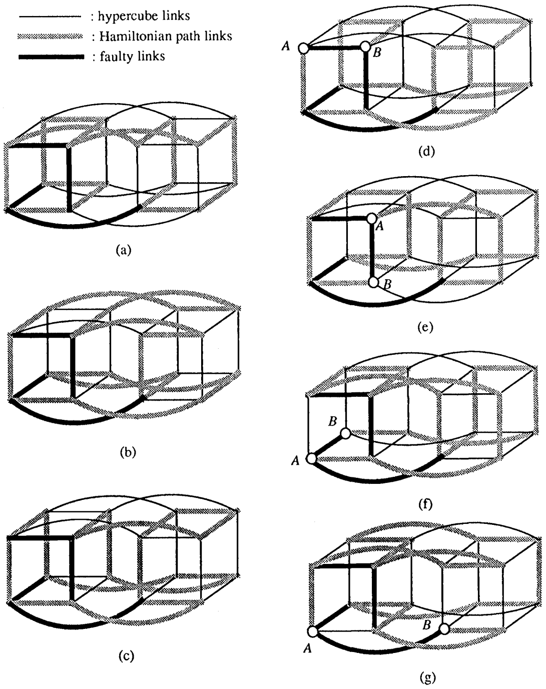


FIG. 6



(a)

(b)

(c)

(d)

(e)

(f)

(g)

FIG. 7

Proof. The proof is by induction on n.

[Induction Basis] Lemmas 1, 2, and 4 take care of the 3-cube, 4-cube, and 5-cube, respectively.

[Induction Step] As the induction hypothesis, we assume that the lemma is true for $n < k$. The proof for $n = k > 5$ is divided into two major cases. Let $S = s_1 s_2 \cdots s_k$, and assume, for convenience, exactly $2k - 5$ link faults.

Case I. Every node is incident to at least three nonfaulty links.

With $2k - 5$ faults in the k-cube $(k > 5)$, there must be at least two faults in the same dimension. Hence, there must be a dimension i such that removing all links on i will result in two $(k - 1)$-cubes with $\leqq 2k - 7 = 2(k - 1) - 5$ faults each. Since each node is incident to at least three nonfaulty links, within each $(k - 1)$-cube each node will be incident to at least two nonfaulty links.

(a) There is such an i where $i \neq 1$.

Suppose, without loss of generality, $i = 2$. There exists a fault-avoiding Hamiltonian path P' from S to $N(S, 1)$ in $*s_2** \cdots *$. Let the sequence of nodes in P' be $S \equiv u_1, u_2, \cdots, u_{2^{k-1}} \equiv N(S, 1)$. Since there are $\leqq 2k - 7 < 2^{k-2}$ faults, there exists a u_i in P' such that $L(u_i, 2)$ and $L(u_{i+1}, 2)$ are both nonfaulty. The path P consists of the segment of P' from S to u_i, $L(u_i, 2)$, the fault-avoiding Hamiltonian path from $N(u_i, 2)$ to $N(u_{i+1}, 2)$ in $*\bar{s}_2** \cdots *$, $L(u_{i+1}, 2)$, and the segment of P' from u_{i+1} to $N(S, 1)$.

(b) Otherwise.

For $k > 5$, there will exist a dimension $j \neq 1$ such that $L(N(S, j), 1)$ is nonfaulty. (In order not to fall into Case I(a) and not to have such a dimension j, the set of faults must be in $\{*\bar{s}_2 s_3 s_4 \cdots s_k, *s_2 \bar{s}_3 s_4 \cdots s_k, \cdots, *s_2 s_3 s_4 \cdots \bar{s}_k, *s_2 s_3 s_4 \cdots s_k\}$; for $k > 5$, there are fewer than $2k - 5$ faults.) Thus, the path P consists of the fault-avoiding Hamiltonian path from S to $N(S, j)$ in $s_1** \cdots *$, $L(N(S, j), 1)$ and the fault-avoiding Hamiltonian path from $N(N(S, j), 1)$ to $N(S, 1)$ in $\bar{s}_1** \cdots *$.

Case II. Some node X is incident to $k - 2$ faulty links.

Let X be incident to faulty links on dimensions $d_1, d_2, \cdots, d_{k-2}$. For $k > 5$, there will exist a dimension $i \in \{d_1, d_2, \cdots, d_{k-2}\}$ such that removing the links on dimension i must result in two $(k - 1)$-cubes with $\leqq 2k - 7$ faults each; hence, within each $(k - 1)$-cube, every node will be incident to at least two nonfaulty links. (In order not to have such an i, the faulty links not adjacent to X must be confined to the 2-cube depicted in Fig. 8, i.e., leaving e_3 and e_4 to accommodate $k - 3 > 2$ faults.)

(a) $i = 1$.

Since there will be $\leqq k - 2$ link faults on any one dimension, there exists a dimension $j \neq 1$ such that $L(N(S, j), 1)$ is nonfaulty. Thus, the path P consists of the fault-avoiding Hamiltonian path from S to $N(S, j)$ in $s_1** \cdots *$, $L(N(S, j), 1)$ and the fault-avoiding Hamiltonian path from $N(N(S, j), 1)$ to $N(S, 1)$ in $\bar{s}_1** \cdots *$.

(b) $i \neq 1$.

Suppose, without loss of generality, $i = 2$. There exists a fault-avoiding Hamiltonian path P' from S to $N(S, 1)$ in $*s_2** \cdots *$. Let the sequence of nodes in P' be $S \equiv u_1, u_2, \cdots, u_{2^{k-1}} \equiv N(S, 1)$. Since there are $\leqq 2k - 7 < 2^{k-2}$ faults, there exists a u_i in P' such that $L(u_i, 2)$ and $L(u_{i+1}, 2)$ are both nonfaulty. The path P consists of the segment of P' from S to u_i, $L(u_i, 2)$, the fault-avoiding Hamiltonian path from $N(u_i, 2)$ to $N(u_{i+1}, 2)$ in $*\bar{s}_2** \cdots *$, $L(u_{i+1}, 2)$, and the segment of P' from u_{i+1} to $N(S, 1)$.

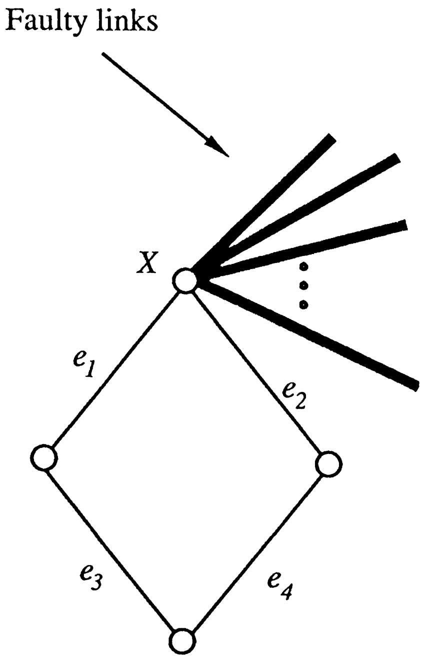


FIG. 8

This result is optimal since there exists an n-cube with $2n - 4$ faults, each node incident to at least two nonfaulty links, for which no Hamiltonian circuit exists. Simply consider the $2n - 4$ faulty links: $00{*}00\cdots 00$, $000{*}0\cdots 00$, $\cdots$, $00000\cdots 0{*}$, $11{*}00\cdots 00$, $110{*}0\cdots 00$, $\cdots$, $11000\cdots 0{*}$. Note that, in this scenario, nodes $0000\cdots 00$ and $1100\cdots 00$ each have exactly two nonfaulty links incident to them. These four links, ${*}000\cdots 00$, $0{*}00\cdots 00$, ${*}100\cdots 00$, and $1{*}00\cdots 00$ form a cycle by themselves, making a Hamiltonian circuit impossible for $n \geqq 3$. We can, in fact, prove the more general statement embodied in Lemma 5.

LEMMA 5. *For $2 \leqq m < n$, there exists an n-cube with $2^{m-1}(n - m)$ faulty links, each node incident to at least m nonfaulty links, for which no Hamiltonian circuit exists.*

Proof. The proof is by construction.

First of all, consider an m-cube Q_m. Let *O-nodes* be the nodes that have an odd number of 1's in their m-bit binary strings and *E-nodes* be the nodes that have an even number of 1's. There are exactly 2^{m-1} O-nodes and 2^{m-1} E-nodes in Q_m. Note that all of the m neighbors of an O-node are E-nodes, and all of the m neighbors of an E-node are O-nodes in Q_m.

Select an m-subcube $\bar{Q}_m$ in the n-cube Q_n. For each O-node in $\bar{Q}_m$, remove all $n - m$ adjacent links which are not in $\bar{Q}_m$ so that it is adjacent to E-nodes only. Every one of the $2^{m-1}(n - m)$ removed links is distinct. View these removed links as faulty links. So each node in Q_n is incident to at least m nonfaulty links, and there are $2^{m-1}(n - m)$ faulty links.

Suppose there exists a Hamiltonian circuit HC in Q_n. Each O-node in $\bar{Q}_m$ in HC must be between two E-nodes in $\bar{Q}_m$. There are 2^{m-1} such O-nodes. We need at least $2^{m-1} + 1$ E-nodes in order to do the job. But there are only 2^{m-1} E-nodes in $\bar{Q}_m$. So, there is no Hamiltonian circuit in Q_n.

3. Hypercubes with an arbitrary number of faults. In this section, we prove the NP-completeness of the following problem.

Hamiltonian circuit in faulty hypercube (HCFH).
INSTANCE: A k-cube Q with faulty links.
QUESTION: Does Q contain a Hamiltonian circuit comprised of only nonfaulty links?

To this end, we make use of 3-satisfiability.

3-Satisfiability (3SAT).
INSTANCE: Collection $C=\{c_1, c_2, \cdots, c_m\}$ of clauses on a finite set $U=\{u_1, u_2, \cdots, u_n\}$ of variables such that $|c_i|=3$ for $1 \leqq i \leqq m$.
QUESTION: Is there a truth assignment for U that satisfies all the clauses in C?

THEOREM 2. *Hamiltonian circuit in faulty hypercube is* NP-*complete.*

Proof. It can be seen that HCFH $\in$ NP, since a nondeterministic algorithm can guess an ordering of the vertices and check in polynomial time that all the required links are nonfaulty links of the hypercube.

We describe a transformation from 3SAT to HCFH. Let $U=\{u_1, u_2, \cdots, u_n\}$ and $C=\{c_1, c_2, \cdots, c_m\}$ be an instance of 3SAT. Without loss of generality, we assume that, if a variable u_j appears in a clause, it appears either as u_j or $\bar{u}_j$ but not both, and moreover, $n=2^{\lceil \log_2 n \rceil}$ and $m=2^{\lceil \log_2 (m+2) \rceil}-2$ (these assumptions can be made valid by increasing the size of the problem by a constant factor). We will be concerned with a k-cube, where $k=6+\lceil \log_2 n \rceil+\lceil \log_2 (m+2) \rceil$, with nonfaulty and faulty links as described in the next few paragraphs. We first define three kinds of components: the *V-square* component, the *C-square* component, and the *triangle* component.

The V-square component is the faulty 4-cube shown in Fig. 9(a) with 18 nonfaulty and 14 faulty links. External nonfaulty links connecting this component to the rest of the k-cube will be incident to only 4 possible entry/exit points: nodes A, E, a, e. Figure 9(b) shows the only three possible configurations for a Hamiltonian circuit's passage through a V-square component. The reader may readily verify that the V-square component has the property that

(i) entering the component at node A implies an exit at node a, and vice versa, and

(ii) entering the component at node E implies an exit at node e, and vice versa.

The C-square component is simply the faulty 4-cube shown in Fig. 13(a) with 15 nonfaulty and 17 faulty links. The entry/exit points for this component will be nodes A, E, a, e, G, H.

The triangle component is the faulty 4-cube shown in Fig. 10(a) with 17 nonfaulty and 15 faulty links. External nonfaulty links connecting this component to the rest of the k-cube will be incident to only 3 possible entry/exit points: nodes A, D, E. Fig. 10(b) shows the only two possible configurations for a Hamiltonian circuit's passage through a triangle component. The reader may verify that the triangle component has the property that

(i) entering the component at node D implies an exit at either A or E,
(ii) entering the component at node A implies an exit at D,
(iii) entering the component at node E implies an exit at D, and
(iv) all of the nodes in the component must be visited in one pass through the triangle.

We view the k-cube as a $4(m+2)$ row by n column mesh of 4-cube components following the pattern prescribed in Fig. 11. V-squares will correspond to the variables of

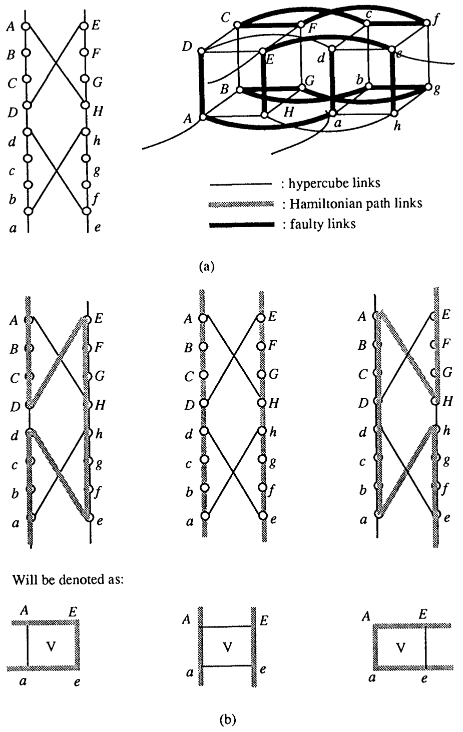


FIG. 9. V-*square component.*

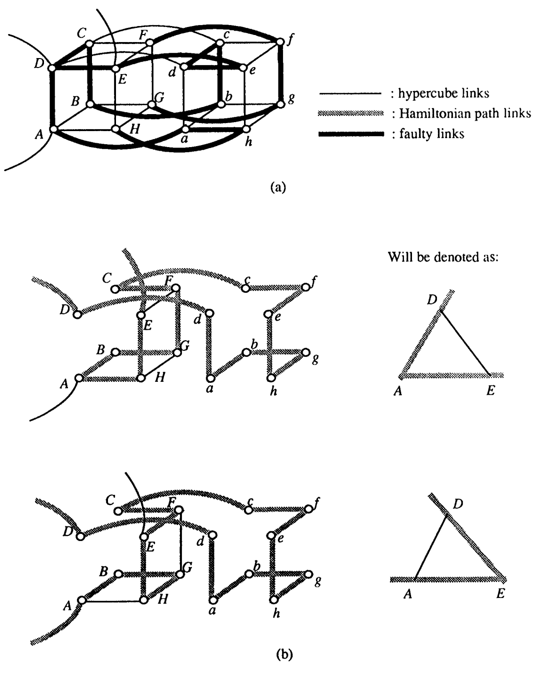


FIG. 10. *Triangle component.*

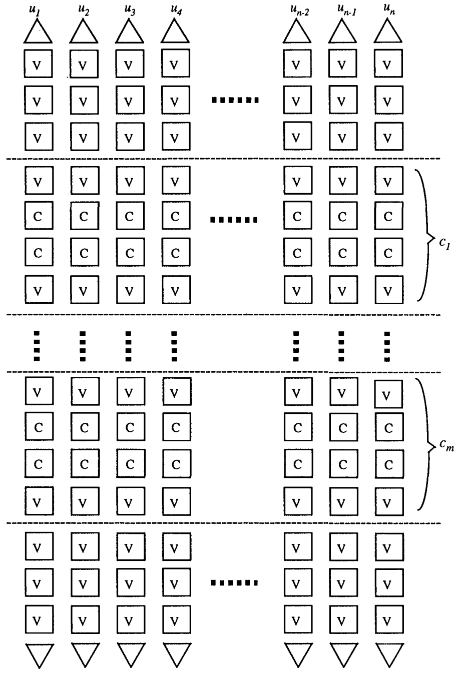


FIG. 11

the 3SAT instance, while C-squares will correspond to the clauses. More specifically, the V-squares of column j are related to variable u_j, and entering/exiting these squares from the A-side corresponds to a *true* setting of the variable while entering/exiting from the E-side corresponds to a *false* setting. The ith block of C-squares will correspond to clause c_i. Triangles help to assign truth values to variables.

To complete the description of the faulty k-cube, we need to specify the nonfaulty links between the 4-cubes in the mesh. There are three kinds of nonfaulty links between 4-cubes: *V-links*, *C-links*, and *VC-links*. V-links are connections between V-squares and V-squares, and between triangles and V-squares; they follow the pattern shown in Fig. 12. C-links are connections between C-squares within a block, and they follow the ring-like pattern shown in Fig. 13(b). Finally, VC-links are the connections between V-squares and C-squares; Fig. 14 explains (there are no VC-links in the jth column of block i if neither u_j nor $\bar{u}_j$ belongs to c_i). The reader may verify that all of these links are indeed hypercube links; i.e., the graph described is indeed a subgraph of a nonfaulty k-cube. All of the rest of the links between 4-cubes are considered faulty.

It is easy to see that the construction of this faulty k-cube can be accomplished in polynomial time. All that remains is to show that C is satisfiable if and only if this k-cube Q has a Hamiltonian circuit comprised of only nonfaulty links.

Suppose there exists a Hamiltonian circuit P in Q. In our construction of Q from the 3SAT instance, each literal of clause c_i introduces a pair of VC-links that "hook" onto the two C-squares for c_i. Examine the C-square in Fig. 13(a). The fact that the links (A, B), (h, a), (d, e), and (E, F) must be included in Hamiltonian circuit P forces that:

(i) the link (A, a) and VC-link entering/exiting from node A cannot coexist in P;

(ii) the link (A, a) and the entering/exiting link from node a cannot coexist in P;

(iii) P must either include both the VC-link entering/exiting from node A and the entering/exiting link from node a, or neither;

(iv) the link (E, e) and the VC-link entering/exiting from node E cannot coexist in P;

(v) the link (E, e) and the entering/exiting link from node e cannot coexist in P, and

(vi) P must either include both the VC-link entering/exiting from node E and the entering/exiting link from node e, or neither.

So by Fig. 13(b), for the two C-squares of a literal for c_i, P must either include both the VC-links that are hooked onto them, or neither. Looking down a particular column j of the mesh, we further observe that, in order for P to indeed be a Hamiltonian circuit, entering/exiting the top triangle at node A (analogously, node E) implies that all V-squares in the column will be entered/exited *only* at node A or node a (node E or node e), and the bottom triangle will also be entered/exited at node A (node E). Figure 15 illustrates the two possible configurations for a Hamiltonian circuit's passage through column j. Thus, our procedure for obtaining a satisfying truth assignment $t : U \rightarrow \{T, F\}$ for C from P is to simply look at the first row of triangles. If P includes the path from D to E to A in the triangle at the top of column j, then $t(u_j) = T$; otherwise, $t(u_j) = F$. To see that this truth assignment satisfies each of the clause $c_i \in C$, consider the ring of C-squares for c_i. The very fact that P enters/exits this ring via a pair of VC-links joining 4-cubes in some column j means that the truth assignment for u_j causes c_i to be satisfied. Thus, t is a satisfying truth assignment for C.

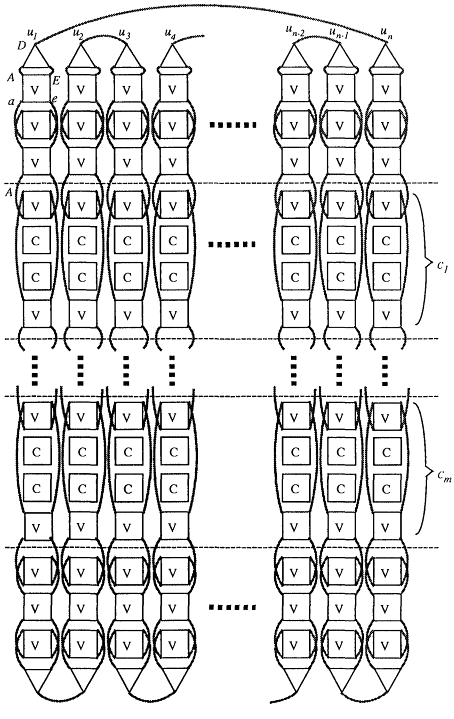


FIG. 12. *The* V-*links*.

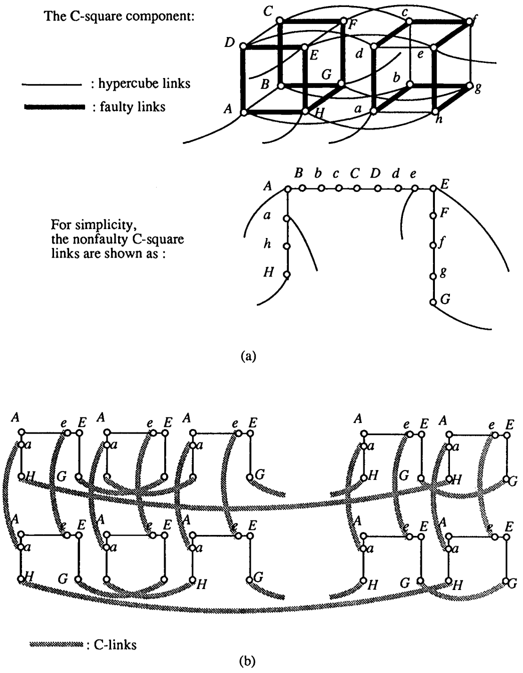


FIG. 13. *The* C-*square component and* C-*links*.

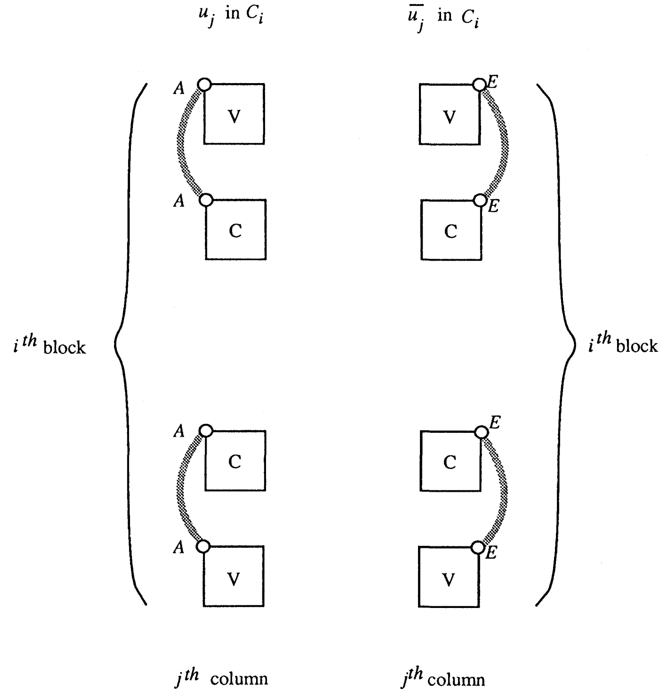


FIG. 14. VC-*links*.

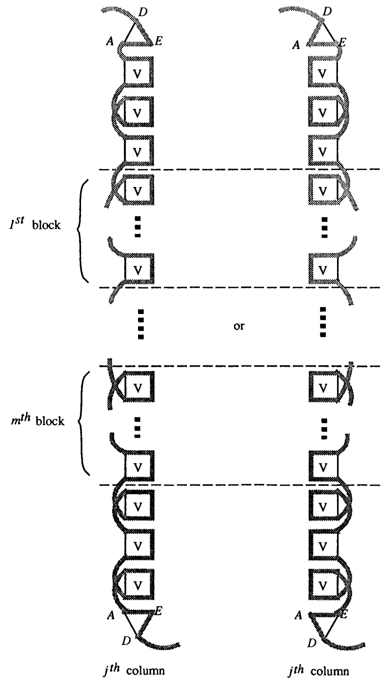


FIG. 15

Conversely, suppose that $t : U \to \{T, F\}$ is a satisfying truth assignment for C. We first describe a simple circuit P which visits all of the nodes of triangles and V-squares. If $t(u_j) = T$, then P will include:

(i) the path from node A to node a within each of the V-squares in column j of the mesh,

(ii) the path from node D to A within each of the two triangles in column j, and

(iii) the nonfaulty links going from node A to node a between triangles/V-squares and V-squares in column j.

If $t(u_j) = F$, then P will include

(i) the path from node E to node e within each of the V-squares in column j of the mesh,

(ii) the path from node D to E within each of the two triangles in column j, and

(iii) the nonfaulty links going from node E to node e between triangles/V-squares and V-squares in column j.

P also includes all of the nonfaulty links from node D to node D between triangles. To make our circuit include all of the nodes of C-squares as well, and thus arrive at a Hamiltonian circuit, we modify P. For each clause $c_i \in C$, there will be a literal, involving, say, variable u_j, in clause c_i that is true under the truth assignment t. If the literal is u_j, Fig. 16 illustrates the modification that will include the nodes of the C-squares for clause c_i in the circuit. Let C-square$(i, j, 1)$ and C-square$(i, j, 2)$ be the two C-squares corresponding to u_j and c_i. The sequence of the nodes in the C-squares traversed by P is:
the path from A to G in C-square$(i, j, 1)$,
followed by the path from G to H in C-square$(i, j + 1, 1)$,
followed by the path from H to G in C-square$(i, j + 2, 1)$, $\cdots$
followed by the path from G to H in C-square$(i, n, 1)$,

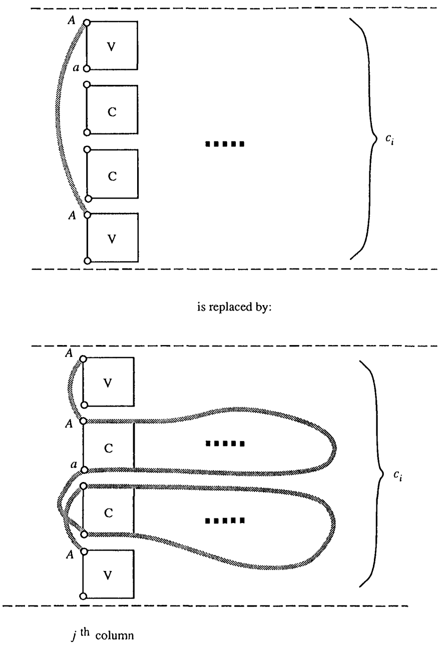


FIG. 16

followed by the path from H to G in C-square$(i, 1, 1)$,
followed by the path from G to H in C-square$(i, 2, 1)$, $\cdots$
followed by the path from G to H in C-square$(i, j-1, 1)$,
followed by H, h, a in C-square$(i, j, 1)$,
followed by a, h, H in C-square$(i, j, 2)$,
followed by the path from H to G in C-square$(i, j-1, 2)$, $\cdots$
followed by the path from G to H in C-square$(i, 1, 2)$,
followed by the path from H to G in C-square$(i, n, 2)$,
followed by the path from G to H in C-square$(i, n-1, 2)$, $\cdots$
followed by the path from H to G in C-square$(i, j+1, 2)$,
followed by the path from G to A in C-square$(i, j, 2)$.
A similar modification will do the trick if the literal is $\bar{u}_j$. Such modifications will give the desired Hamiltonian circuit.

4. Concluding remarks. In summary, we showed that there exists a Hamiltonian circuit consisting of only nonfaulty links in an n-cube with $\leqq 2n-5$ link faults, in which each node is incident to at least two nonfaulty links. Since there exists an n-cube with $2n-4$ faulty links, in which each node is incident to at least two nonfaulty links, and for which there is no Hamiltonian circuit, this result is optimal. In addition, we found that $2^{m-1}(n-m)$ faulty links are enough to destroy the existence of a Hamiltonian circuit in an n-cube, in which each node is incident to at least m nonfaulty links, $2 \leqq m < n$. The problem of determining the minimum number of faulty links in an n-cube, where each node is incident to at least m nonfaulty links, so that no Hamiltonian circuit exists is still open.

We also showed that the problem of determining whether an n-cube with an arbitrary number of link faults has a Hamiltonian circuit to be NP-complete. This result immediately implies that the problem of determining whether an n-cube with an arbitrary number of link faults has a torus which includes all of the nodes is NP-complete since a Hamiltonian circuit is a special case of torus. This result also implies that the problem of determining the largest ring in an n-cube with an arbitrary number of link faults is NP-hard.

REFERENCES

[B] P. BANERJEE, *Reconfiguring a hypercube multiprocessor in the presence of faults*, Proc. of the Conference on Hypercubes, Concurrent Computers and Applications, Monterey, CA, 1989.

[BMS] S. BETTAYEB, Z. MILLER, AND I. H. SUDBOROUGH, *Embedding grids into hypercubes*, Proceedings of the 3rd Aegean Workshop on Computing, Patras, Greece, 1988.

[BCLR] S. BHATT, F. CHUNG, T. LEIGHTON, AND A. ROSENBERG, *Optimal simulations of tree machines*, Proc. of IEEE Foundations of Computer Science, Toronto, Ontario, 1986.

[BI] S. N. BHATT AND I. C. F. IPSEN, *How to embed trees in hypercubes*, Technical Report YALEU/DCS/RR-443, Yale University, December 1985.

[C1] M. Y. CHAN, *Dilation-2 embeddings of grids into hypercubes*, Proc. of International Conference on Parallel Processing, St. Charles, IL, 1988.

[C2] ———, *Embedding of d-dimensional grids into optimal hypercubes*, Proc. of Association for Computing Machinery Symposium on Parallel Algorithms and Architectures, Sante Fe, NM, 1989.

[C3] ———, *Embedding of grids into optimal hypercubes*, SIAM J. Comput., 20 (1991), pp. 834–864.

[CL1] M. Y. CHAN AND S-J. LEE, *Distributed fault-tolerant embeddings of rings in hypercubes*, J. Parallel and Distrib. Comput., to appear.

[CL2] ———, *Fault-tolerant embeddings of complete binary trees and rings in hypercubes*, Technical Report UTDCS-17-89, University of Texas at Dallas, August 1989.

[G] D. S. GREENBERG, *Minimum expansion embeddings of meshes in hypercubes*, Technical Report YALEU/DCS/TR-535, Yale University, August 1987.

[HLN1] J. HASTAD, T. LEIGHTON, AND M. NEWMAN, *Reconfiguring a hypercube in the presence of faults*, Proc. of Association for Computing Machinery Symposium on Theory of Computing, New York, 1987, pp. 274–284.

[HLN2] ———, *Fast computation using faulty hypercubes*, Proc. of Association for Computing Machinery Symposium on Theory of Computing, Seattle, WA, 1989.

[HJ] C-T. HO AND S. L. JOHNSSON, *Embedding meshes in boolean cubes by graph decomposition*, Technical Report YALEU/DCS/TR-689, Yale University, March 1989.

[PM] F. J. PROVOST AND R. MELHEM, *Distributed fault tolerant embedding of binary trees and rings in hypercubes*, Proc. of International Workshop on Defect and Fault Tolerance in VLSI Systems, Springfield, MA, 1988.

[SS] Y. SAAD AND M. H. SCHULTZ, *Topological properties of hypercubes*, Research Report YALEU/DCS/RR-389, June 1985.

[W] A. WU, *Embedding of tree networks into hypercubes*, J. Parallel and Distrib. Comput. 2, (1985), pp. 238–249.

SIAM J. DISC. MATH.
Vol. 4, No. 4, pp. 528-534, November 1991

006

COMPACT CYLINDRICAL CHROMATIC SCHEDULING*

D. DE WERRA† AND PH. SOLOT†‡

Abstract. An edge coloring model is described for dealing with a special type of cyclic scheduling problem: each edge e of a graph has an integral weight $p_e \geqq 0$. An interval cyclic edge T-coloring is an assignment of p_e cyclically consecutive colors in $\{1, \cdots, T\}$ to each edge e such that no two adjacent edges share a common color and, for each bundle (collection of edges adjacent to a same node) or triangle A, all colors used on edges of A are cyclically consecutive. Let $T(p, G) = \max_A \{\Sigma p_e : e \in A : A \text{ is a bundle or a triangle}\}$. A graph G is called *ice-perfect* if, for any assignment p of values p_e to the edges, there exists an interval cyclic edge $T(p, G)$-coloring. We show that a graph is ice-perfect if and only if it is a triangle or a bipartite outerplanar graph. Applications to scheduling in flexible manufacturing systems are mentioned.

Key words. scheduling, edge coloring, production, flexible manufacturing systems, open shop, cylindrical scheduling

AMS(MOS) subject classifications. 05C15, 90B35

1. Introduction. In production it is often the case that one must manufacture a collection of items on processors in a cyclic way: there is a so-called production cycle that is repeated continuously.

Consider an open shop scheduling problem: a collection of processors $P_1, \cdots, P_m$ is given with a set of jobs $J_1, \cdots, J_n$. Each job J_j consists of a set of tasks $T_{1j}, T_{2j}, \cdots, T_{mj}$ with processing times $p_{1j}, p_{2j}, \cdots, p_{mj}$. Task T_{ij} of J_j has to be processed on P_i; no two tasks T_{ij}, T_{kj} of the same job J_j can be processed at the same time. Furthermore, no processor can work on two tasks at a time. We also assume that the tasks $T_{1j}, \cdots, T_{mj}$ can be processed in any order. No preemptions are allowed when processing a task.

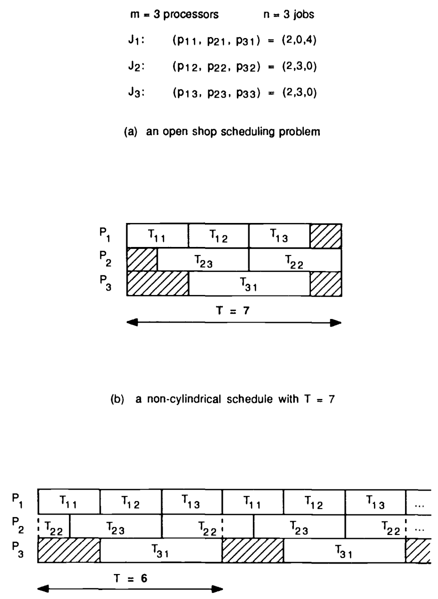
Let T be the total processing time of the jobs (in an open shop); if the production cycle is repeated, tasks that are being processed at the beginning and at the end of the production cycle can be considered as nonpreempted; they are simply continued at the beginning of the next cycle. Figure 1(a) shows an example of an open-shop scheduling problem; it is not difficult to see that the minimum value of the total processing time T is 7. The corresponding schedule is shown in Fig. 1(b). Now let us assume that the production schedule is repeated; in such a case we have a processing time $T = 6$ for each production cycle. Such a *cylindrical schedule* is shown in Fig. 1(c).

In some cases, an additional requirement must be taken into account: there should be no idle time between processing of the various tasks of the same job. The reason for that may be that there is not much storage space in an automated workshop; the various tasks forming a job correspond to manufacturing a batch of identical parts that all must go through a specified set of machines. Once a batch is introduced into the workshop, we would like to have it processed without interruptions. Similarly, it is desired that each processor works with as few interruptions as possible. In a cyclic problem, there will be at most one idle period in each cycle. The main reason is to have a continuous-time interval available for performing maintenance in each cycle. Such circumstances arise, for instance, in a flexible manufacturing system.

* Received by the editors June 13, 1989; accepted for publication (in revised form) September 4, 1990.

† Département de Mathématiques, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland.

‡ The work of this author was supported by the Swiss National Scientific Research Council under grant 2.160-0.86 and by the Computer Integrated Manufacturing project of the Swiss Federal Institute of Technology.



m = 3 processors n = 3 jobs

J_1: $(p_{11}, p_{21}, p_{31}) = (2,0,4)$

J_2: $(p_{12}, p_{22}, p_{32}) = (2,3,0)$

J_3: $(p_{13}, p_{23}, p_{33}) = (2,3,0)$

(a) an open shop scheduling problem

(b) a non-cylindrical schedule with T = 7

(c) a cylindrical schedule with T = 6

FIG. 1. *An open-shop cylindrical scheduling problem.*

Schedules satisfying these requirements are called *compact*. From now on, all schedules must be compact, unless otherwise mentioned.

When the production cycle is repeated, we call such a problem a *nonpreemptive open-shop cylindrical scheduling problem* (NOSCS).

Let us associate with a NOSCS a graph-theoretical model as follows: each job J_j and each processor P_i corresponds to a node. Each task T_{ij} is represented by an edge $[P_i, J_j]$. Then, there exists a production cycle (i.e., a cylindrical schedule) with total processing time (or length) T if and only if we can assign to each edge $[P_i, J_j]$ a set of p_{ij} cyclically consecutive integers in $\{1, \cdots, T\}$ such that no two adjacent edges share a common integer and furthermore all integers assigned to an (inclusionwise) maximal set of mutually adjacent edges are also cyclically consecutive in $\{1, \cdots, T\}$. Figure 2 shows such a model for the example in Fig. 1(c).

Clearly, a lower bound on the length T of a cycle is given by

$$T(p) = \max \{\max_j \Sigma_i p_{ij}, \max_i \Sigma_j p_{ij}\}.$$

We will examine here when a cylindrical schedule in $T(p)$ time units does exist. More precisely, we will characterize graphs for which, given any assignment of values p_{ij} to its edges $[P_i, J_j]$, a cylindrical schedule in $T(p)$ time units can be found.

The graphs used here will have no loops; if, in addition, they have no multiple edges, they are *simple* graphs. An *induced subgraph* $G_{X'} = (X', E')$ of $G = (X, E)$ is obtained by taking a subset X' of the node set X of G and by introducing into E' all edges of G with both endpoints in X'. A *partial* graph $H = (X, F)$ of $G = (X, E)$ is obtained from G by possibly removing some edges from E. A *partial subgraph* $H_{X'} = (X', F')$ is a partial graph of $G_{X'}$.

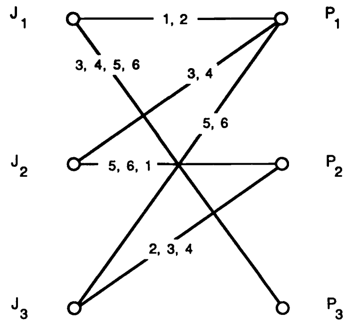


FIG. 2. *A graph-theoretical representation of the schedule of Fig.* 1(*c*).

A *classical edge k-coloring* of G is an assignment of one color in $\{1, \cdots, k\}$ to each edge in such a way that no two adjacent edges have a color in common. The smallest k for which G has a classical edge k-coloring is the *chromatic index* of G; it is denoted by $q(G)$. Moreover, we denote by $\Delta(G)$ the maximum cardinality of a set of mutually adjacent edges.

A graph G is called *line-perfect* [7] if, for every partial graph H, the trivial inequality $q(H) \geqq \Delta(H)$ is satisfied with equality.

All graph-theoretical terms not defined here can be found in [2] and [5].

2. Chromatic scheduling. In this section we will develop a graph-theoretical model that will allow us to study a variation of open-shop problems (see [1] and [3] for related problems and basic definitions). Let us assume that we have a collection of jobs $J_1, \cdots, J_n$ with processing times $p_1, \cdots, p_n$; a set of renewable resources $R_1, \cdots, R_p$ is given. One unit of each R_k is available at any moment and each job requires at most two specific resources during its processing. Since a resource cannot be shared by two or more jobs at the same time, no two jobs using the same resource can be processed simultaneously.

We can associate a graph G with this data as follows: each resource R_k corresponds to a node of G and each job J_j using resources R_u, R_v is an edge $J_j = [R_u, R_v]$ with weight p_j. Note that by introducing fictitious resources if needed, we may assume that each job uses exactly two resources.

We want to find a cylindrical schedule in T time units; this corresponds to an assignment of p_j cyclically consecutive integers in $\{1, \cdots, T\}$ to each edge associated with J_j. These integers will be called colors; they correspond to the time periods during which J_j will be processed during the cycle. No two adjacent edges can share a color due to the resource availability constraint.

This assignment must satisfy an additional requirement; we need to introduce some notation before formulating it.

A bundle $B(x)$ in a graph G is the collection of edges adjacent to node x. We denote by $\mathscr{A}(G)$ the set of all triangles and bundles in G.

The assignment of colors to the edges of G must be such that the colors associated to the edges of any set A in $\mathscr{A}(G)$ must form a set of cyclically consecutive colors in $\{1, \cdots, T\}$. If A is a bundle $B(R_k)$, it means that resource R_k must be used continuously (without interruptions) during each cycle of the cylindrical schedule. Note that the graphs G in this section are not necessarily bipartite, so $\mathscr{A}(G)$ may contain triangles in addition to the bundles of all nodes.

Such a coloring of the edges of G in T colors will be called an *interval cyclic edge T-coloring* (or, shortly, *ice T-coloring*).

Let p_e be the weight assigned to each edge e of G and define

$$(2.1)\qquad p(A) = \Sigma(p_e : e \in A) \text{ for each } A \in \mathscr{A}(G),$$

$$(2.2)\qquad T(p, G) = \max\{p(A) : A \in \mathscr{A}(G)\}.$$

Denote by $X'(G, p)$ the smallest T for which G (with weights p_e on its edges e) has an ice T-coloring. Clearly we have

$$(2.3)\qquad X'(G,p) \geqq T(p, G).$$

In the next section, we will characterize graphs for which (2.3) is an equality whatever integral weights are given to their edges.

Observe that, if $p_e = 0$ is given to edge e, this corresponds to removing edge e from G. So, when requiring that (2.3) holds for any assignment of nonnegative integers p_e to the edges e, we will in particular require it to hold for every partial subgraph of G (i.e., every graph obtained by removing some edges and some nodes from G).

Similar problems concerning noncylindrical schedules have been examined in [9]. Upper bounds on the total completion time of cylindrical and noncylindrical schedules have been derived from graph-theoretical models in [10].

3. Ice-perfect graphs. We first give the following two definitions.

A graph G is *ice-perfect* if, for any choice of integral weights p_e for the edges e, we have

$$(3.1)\qquad X'(G,p) = T(p, G).$$

If (3.1) holds for any choice of weights p_e in $N = \{0, 1, 2, 3, 4\}$, we will say that G is *Nice-perfect*. Observe that an ice-perfect graph is trivially Nice-perfect. It can also be easily seen that a Nice-perfect graph is line-perfect.

We observe that when we are considering ice T-colorings, we may assume without loss of generality that the graphs are simple; in terms of scheduling, all jobs involving the same pair of resources are merged into one new job. If (2.3) holds as an equality before merging the parallel edges, it will also hold after merging and conversely. This assumption will simplify the graph-theoretical statements and the proofs.

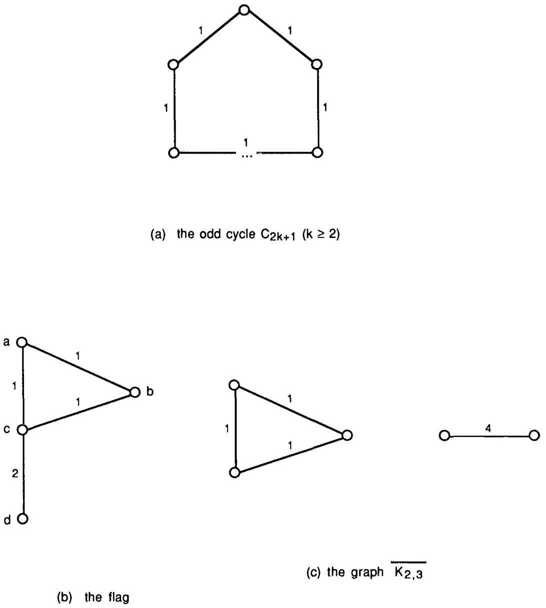
We will now examine some examples of graphs; for the odd cycle C_{2k+1} ($k \geqq 2$) with $p_e = 1$ for each edge (see Fig. 3(a)), we have $T(p, G) = 2$ and $X'(G, p) = 3$, so C_{2k+1} is not ice-perfect (it is not even line-perfect). For the flag in Fig. 3(b), the values p_e shown give $T(p, G) = 4$. We may without loss of generality color $[c, d]$ with $\{1, 2\}$, $[a, c]$ with $\{3\}$, and $[b, c]$ with $\{4\}$. Then there is no color that can be used for $[a, b]$: such a color should be cyclically adjacent to both 3 and 4 and different from both. So the flag is not Nice-perfect.

The graph $\overline{K_{2,3}}$ (i.e., a triangle and an isolated edge) is not Nice-perfect as can be seen from Fig. 3(c).

Observe that a triangle is, however, ice-perfect. A *mouth* $M(x, y)$ consists of three node-disjoint chains of the same parity (with length at least two) that link nodes x and y. Figure 4(a) shows an odd mouth and Fig. 4(b) an even mouth with weights p_e in N.

For this odd mouth we have $T(p, G) = 6$; without loss of generality we may color $[x, a_1]$ with $\{1, 2\}$, $[x, b_1]$ with $\{3, 4\}$, $[x, c_1]$ with $\{5, 6\}$; then $[a_{2k}, y]$ must get one color in $\{1, 2\}$, $[b_{2r}, y]$ one color in $\{3, 4\}$, and $[c_{2s}, y]$ one color in $\{5, 6\}$. This cannot give an ice 6-coloring (colors cannot be cyclically consecutive at y). So an odd mouth is not Nice-perfect.

For the even mouth in Fig. 4(b) we have $T(p, G) = 5$. Without loss of generality we color $[x, a_1]$, $[x, b_1]$, and $[x, c_1]$ with $\{1, 2\}$, $\{3\}$, and $\{4, 5\}$, respectively; then



(a) the odd cycle C_{2k+1} ($k \geq 2$)

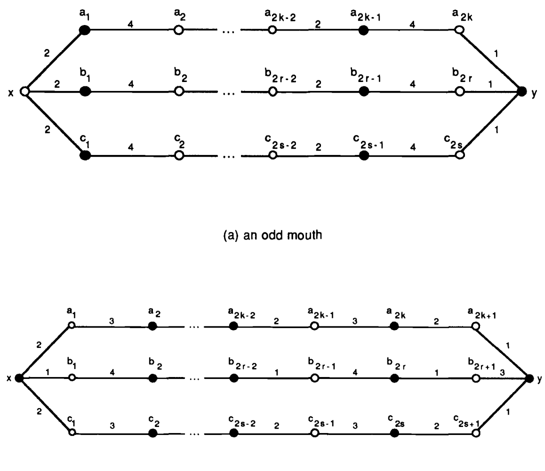
(b) the flag

(c) the graph $\overline{K_{2,3}}$

FIG. 3. *Some graphs that are not ice-perfect.*

$[a_{2k}, a_{2k+1}]$, $[b_{2r}, b_{2r+1}]$, and $[c_{2s}, c_{2s+1}]$ get also $\{1, 2\}$, $\{3\}$, and $\{4, 5\}$, respectively. Now $[b_{2r+1}, y]$ may receive either $\{4, 5, 1\}$ or $\{5, 1, 2\}$ because of the cyclic consecutivity requirement on $[b_{2r+1}, y]$. In any case, colors 1 and 5 will be given to $[b_{2r+1}, y]$. But $[a_{2k+1}, y]$, which could have $\{3\}$ or $\{5\}$, must then get $\{3\}$. So $[c_{2s+1}, y]$, which could have $\{3\}$ or $\{1\}$, has no available color. So an even mouth is not Nice-perfect.

In a graph G, a *cutnode* is a node whose removal increases the number of connected components of G. A graph is *nonseparable* if it is connected (nontrivial) and it has no



(a) an odd mouth

(b) an even mouth

FIG. 4. *More nonice-perfect graphs.*

cutnode. A *block* in G is an inclusionwise maximal nonseparable induced subgraph of G. We will now characterize the blocks of an ice-perfect graph. We need some additional definitions.

A graph is *planar* if it can be embedded in the plane in such a way that edges do not intersect. A graph is *outerplanar* if it can be embedded in such a way that all its nodes lie on the same face (usually chosen as the exterior face).

It is known that a graph is outerplanar if and only if it has no subgraph homeomorphic to K_4 or $K_{2,3}$ except $K_4 - e$ [6]. It follows that a bipartite graph is outerplanar if and only if it contains no mouth.

It is not difficult to see that any embedding on the plane of a bipartite outerplanar graph can be obtained in the following way.

Consider an elementary even cycle C_0 containing an edge $e_1 = [x, y]$; a *handle* $H(e_1)$ is an odd chain (of length at least three) linking x and y (the intermediate nodes of $H(e_1)$ are new nodes). Note that $C_0 \cup H(e_1)$ is still bipartite (and planar). We may now repeat this operation by choosing an edge $e_2 \neq e_1$ in $G_1 = C_0 \cup H(e_1)$ and introducing a handle $H(e_2)$ to obtain a graph G_2. A graph that can be obtained by starting from an even cycle C_0 and by repeatedly introducing handles $H(e_1), \cdots, H(e_q)$ on distinct edges $e_1, \cdots, e_q$ is called a *bipartite edge cactus* (or *bec graph*).

PROPOSITION 3.1. *Let G be a bec graph and let p be an assignment of nonnegative integral values to the edges of G; then, for any $k \geqq T(p, G)$, the graph G has an ice k-coloring.*

Proof. We will give an algorithm which produces an ice k-coloring in G.

Assume G is obtained from an initial cycle C_0 by choosing successively an edge e_1, introducing a handle $H(e_1)$, and an edge e_2 with a handle $H(e_2)$ and so on.

Let $k \geqq T(p, G)$. Let $\{1, \cdots, k\}$ be the set of colors. We traverse the cycle C_0 by giving each edge e alternately p_e cyclically consecutive colors ending with k and p_e cyclically consecutive colors starting with 1 ($\equiv k + 1$).

At this stage, observe that each edge $e = [x, y]$ in C_0 with colors $l, l + 1, \cdots, r$ satisfies the following:

(3.2) Each one of the sets of colors $S(x), S(y)$ on edges adjacent to x and to y, respectively, forms a cyclic interval in $\{1, \cdots, k\}$. Both $S(x)$ and $S(y)$ have l as an endpoint or both have r as an endpoint.

Now assume that edge $e_1 = [x_1, y_1]$ has received colors $l_1, \cdots, r_1$. According to (3.2), we may assume that r_1 is an endpoint of $S(x_1)$ and of $S(y_1)$ (the other case is symmetric). Starting at x_1, we color the edges e of $H(e_1)$ by giving them alternately p_e cyclically consecutive colors starting with $r_1 + 1$ and p_e cyclically consecutive colors ending with r_1. Such colors can always be found in $\{1, \cdots, k\}$ since $k \geqq T(p, G)$. We observe that (3.2) still holds for each edge different from the already used e_1.

By repeating this construction for each new e_i and for its handle $H(e_i)$ we get an ice k-coloring of G. □

We will consider that a block may consist of a single edge (it is also a graph with two nodes which cannot be disconnected by removing less than two nodes).

We may now state the following theorem.

THEOREM 3.2. *For a graph G, the following statements are equivalent:*

(a) *G is ice-perfect;*

(b) *G is Nice-perfect;*

(c) *G is a bipartite outerplanar graph or a triangle.*

Proof.

(a) ⇒ (b). The proof is trivial.

(b) ⇒ (c). Let G be a Nice-perfect graph. We will characterize the blocks of G. So, let B be a maximal two-connected induced subgraph of G. If B is a triangle, we have necessarily $G = B$ since the flag and $\overline{K_{2,3}}$ are not Nice-perfect and we are done. If B is an edge, there is nothing to prove. So, assume B is a nontrivial block. It is bipartite since C_{2k+1} is not Nice-perfect (for $k \geqq 2$).

Using the same procedure as in the construction of bec graphs we define a collection $e_1, \cdots, e_p$ of distinct edges of B with handles $H(e_1), \cdots, H(e_p)$. This shows that B can be reconstructed in the reverse order (by introducing handles $H(e_p), H(e_{p-1}), \cdots, H(e_1)$); hence B is a bec graph and G is bipartite and outerplanar.

(c) ⇒ (a). If G is a triangle, then it is ice-perfect. Assume now that G is bipartite and outerplanar; each block is outerplanar. Let $k = T(p, G)$ for an assignment p of nonnegative integer weights p_e to the edges of G. According to Proposition 3.1, each block of G has an ice k-coloring. We can then number the blocks $B_1, \cdots, B_s$ in such a way that each B_i ($i \geqq 2$) is linked by a single cutnode to the graph generated by the blocks B_j with $j < i$. Take B_1 with its ice k-coloring; then B_2 has also an ice k-coloring. By possibly permuting cyclically the colors $1, \cdots, k$ in B_2 we may get an ice k-coloring of B_2 such that all edges adjacent to the cutnode x linking B_1 and B_2 have different colors. This is possible since $k \geqq T(p, G)$. We thus obtain an ice k-coloring of $B_1 \cup B_2$. By repeating this construction for $B_3, B_4, \cdots, B_s$, we will obtain an ice k-coloring of G. □

4. Applications to scheduling. Production in cycles occurs frequently in industrial manufacturing processes. Such a situation will, as mentioned earlier, happen in flexible manufacturing systems: in a single time shift, several production cycles may be performed in a row. This may be an efficient way of working with a production system that is adequately set up for manufacturing a given product mix; the appropriate tools have been loaded in the tool magazines of the NC machines. Minimizing the length of the production cycle will be an objective of crucial importance.

The above model can provide a way of handling such a problem: it can be expressed as a cylindrical open-shop scheduling problem in some cases.

REFERENCES

[1] K. R. Baker, *Introduction to Sequencing and Scheduling*, John Wiley, New York, 1974.

[2] C. Berge, *Graphes*, Gauthier-Villars, Paris, 1983.

[3] J. Blazewicz, W. Cellary, R. Slowinski, and J. Weglarz, *Scheduling under Resource Constraints: Deterministic Models*, Baltzer AG, Basel, 1986.

[4] M. Cochand, D. de Werra, and R. Slowinski, *Preemptive scheduling with staircase and piecewise linear resource availability*, Zeitschrift für Operations Research, 33 (1989), pp. 297–313.

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

[6] F. Harary, *Graph Theory*, Addison-Wesley, Reading, MA, 1969.

[7] D. de Werra, *On line-perfect graphs*, Math. Programming, 15 (1978), pp. 236–238.

[8] ———, *On the two-phase method for preemptive scheduling*, European J. Oper. Res., 37 (1988), pp. 227–235.

[9] ———, *Almost nonpreemptive schedules*, Ann. Oper. Res., 26 (1990), pp. 243–256.

[10] D. de Werra and A. Hertz, *Consecutive colorings of graphs*, Zeitschrift für Operations Research, 32 (1988), pp. 1–8.

SIAM J. DISC. MATH.
Vol. 4, No. 4, pp. 535–549, November 1991

007

BOUND SMOOTHING UNDER CHIRALITY CONSTRAINTS*

ANDREAS W. M. DRESS† AND TIMOTHY F. HAVEL‡

Abstract. Procedures for determining the feasibility of lower and upper bounds on Euclidean distances of fixed dimension play a central role in the analysis of many kinds of scientific data. Shown in this paper is how results from graph optimization theory can be used to solve the feasibility problem in one dimension, subject to the condition that the order of the points along the real line is known. The solution is used to derive a PSPACE, $O(n^3 \cdot n!)$-time sequential algorithm for finding one-dimensional representations subject to arbitrary distance (and order) constraints. The wider applicability of these results in measurement theory is discussed, in particular, Roy's elegant proofs of the classical representation theorems for interval orders and semiorders, and they are used to obtain a new representation theorem for a ternary relation called *ε-collinearity*.

Key words. distance geometry, molecular conformation, preference relations

AMS(MOS) subject classifications. 51K05, 05C20, 92A40, 90A06

1. Introduction. The motivation for this work originally came from the geometric theory of molecular conformation. It is therefore useful, and may perhaps enhance this article's interest, to introduce some of the basic terminology and ideas of this field (for a complete account, the reader is referred to [4]).

The *conformation* of a molecule is the set of its possible spatial structures, or *conformers*. Most of the information that is available on the conformations of molecules can be formulated in terms of lower and upper bounds on their interatomic distances [4], [6], [14]. If A denotes the set of atoms in the molecule, the lower and upper bounds constitute functions $\lambda, v : A \times A \rightarrow R$, which are

(B1) Nonnegative: $\lambda, v \geqq 0$;

(B2) Symmetric: $\lambda(a,b) = \lambda(b,a), v(a,b) = v(b,a)$ for all $a, b \in A$;

(B3) Compatible: $\lambda \leqq v$ as functions;

and which

(B4) Vanish on the diagonal: $\lambda(a,a) = v(a,a) = 0$ for all $a \in A$.

In addition, the bounds must satisfy certain consistency relations, for example, $\lambda(a, b) \leqq v(a, c) + v(c, b)$ for all $a, b, c \in A$, which follow from Menger's intrinsic characterizations of the Euclidean metric (cf. [1]).

Another important form of chemical information concerns the orientation or *chirality* of rigid tetrahedra of atoms in the molecule: If $\pi : A \rightarrow R^3$ is a function that assigns to each atom $a \in A$ its Cartesian coordinates $\pi(a)$ in some possible conformer of the molecule, for a given indexing $A = \{a_1, \cdots, a_n\}$ of its atoms the chirality of each quadruple is given by

$$(1) \qquad \chi_\pi(a_{i_1}, \cdots, a_{i_4}) := \text{sign det} \begin{pmatrix} 1 & 1 & 1 & 1 \\ \pi(a_{i_1}) & \pi(a_{i_2}) & \pi(a_{i_3}) & \pi(a_{i_4}) \end{pmatrix},$$

* Received by the editors May 23, 1988; accepted for publication (in revised form) October 26, 1990. This research was supported by a NATO Scientific Affairs Division collaborative research grant (86/0701), and the Research Institute of Scripps Clinic.

† Fakultät für Mathematik, Universität Bielefeld, P.O. Box 8640, D-4800 Bielefeld 1, Germany.

‡ Institute of Science and Technology, University of Michigan, 2200 Bonisteel Boulevard, Ann Arbor, Michigan 48109. The research of this author was partially supported by National Institutes of Health grant GM-37708.

where $1 \leqq i_1 < \cdots < i_4 \leqq \#A$ ($\#A$ = cardinality of A). If we extend this to a function $\chi_\pi : A^4 \to \{0, \pm 1\}$ by antisymmetry, it has been shown that this function determines (and is determined by) the oriented matroid structure associated with the affine point configuration $\pi(A)$ [7]. In practice, only a rather small number of quadruples in the set of all quadruples will have a fixed chirality in all of the molecule's possible conformers, and hence the available chirality information typically takes the form of a function $\tilde{\chi} : A^4 \to 2^{\{0,\pm 1\}}$, which assigns to each ordered quadruple the *set* of its possible chiralities.

We call the triple of functions $(\lambda, v, \tilde{\chi})$ the *distance geometry description* of the molecule. The conformation of the molecule it defines is the set

$$(2)\qquad \Pi(\lambda, v, \tilde{\chi}) := \{\pi : A \to R^3 \mid \lambda(a,b) \leqq \|\pi(a) - \pi(b)\| \leqq v(a,b) \text{ and } \chi_\pi(a,b,c,d) \in \tilde{\chi}(a,b,c,d)\,\forall a,b,c,d \in A\}.$$

The first and foremost question that arises in analyzing such a description of a molecule is whether or not the given chemical information is geometrically feasible, in that $\Pi(\lambda, v, \tilde{\chi}) \neq \varnothing$. We call this the *fundamental problem of distance geometry*. There exist two main approaches towards answering this question, both of which yield useful supplementary information:

(I) *Coordinatization*. Find the Cartesian coordinates $\pi \in \Pi(\lambda, v, \tilde{\chi})$ of a feasible spatial structure.

(II) *Bound smoothing*. For all $a, b \in A$, determine the following extrema:

$$(3)\qquad \lambda_\Pi(a,b) := \inf(\|\pi(a) - \pi(b)\| \mid \pi \in \Pi(\lambda, v, \tilde{\chi})),$$

$$(4)\qquad v_\Pi(a,b) := \sup(\|\pi(a) - \pi(b)\| \mid \pi \in \Pi(\lambda, v, \tilde{\chi})).$$

Although the problem of determining the feasibility of a given distance geometry description is at least NP-hard in all dimensions [22], Tarski's work on quantifier elimination shows that it is decidable [24]. To date, however, no general algorithm for this problem is known which is efficient enough to be useful in any but the most trivial of cases. The coordinatization approach has proved most useful as a means of deriving sufficient conditions for feasibility, whereas the bound smoothing approach is most useful for the purpose of deriving necessary conditions.

The difficulty of the three-dimensional problem has prompted us to study the one-dimensional case first, where the analogue of chirality is just the order of pairs of points along the real line. It turns out that the one-dimensional problem also has a number of interesting applications, particularly to the theory of measurement in the social sciences, cf. [19], [20]. In this field an important goal is to be able to *represent* empirically observed relations by means of relations among a set of points along the real line. These latter relations are typically statements concerning the order of or distances between pairs of points, and include for example:

1. *The interval order problem*. Given a binary relation $\mathscr{R} \subseteq A \times A$, do there exist two functions $\vartheta, \psi : A \to R$ such that $\vartheta(a) < \psi(a)$ and $a\mathscr{R}b \Leftrightarrow \psi(b) < \vartheta(a)$, for all $a, b \in A$?
2. *The semiorder problem*. Given a binary relation $\mathscr{R} \subseteq A \times A$, does there exist a function $\varphi : A \to R$ such that $a\mathscr{R}b \Leftrightarrow \varphi(b) + 1 < \varphi(a)$, for all $a, b \in A$?

Thus these representation problems can be viewed as special, one-dimensional cases of the fundamental problem of distance geometry.

Our purpose in this paper is to present a simple and elegant solution to the one-dimensional version of the fundamental problem in the case that the order of the points along the real line is known, so that $\tilde{\chi} = \chi_\pi$ for some (now scalar-valued) function $\pi : A \to R$. From this, it is possible to solve both the coordinatization and bound smoothing problems ((I) and (II) above) for any distance geometry description $(\lambda, v, \tilde{\chi})$ by enumeration of all linear orders that are consistent with the given chirality constraints $\tilde{\chi}$. We also discuss the potentially widespread (and in part already known; cf. [20], [21]) applications that this result has to measurement theory, and give as one example a characterization of a ternary relation we call *ε-collinearity*.

2. A simple theorem. We begin with an almost trivial proof of a result that is well known to graph theorists (cf. [26]). In the following A denotes a set, and $\mathscr{D}_A^\omega$ denotes the complete digraph $\mathscr{D}_A$ on A with arc weights $\omega : A \times A \twoheadrightarrow R$. In addition, relative to ω we define the set

$$(5) \qquad \Pi_\omega := \{\pi \in R^A \mid \pi(b) - \pi(a) \leqq \omega(a,b)\},$$

and for any subset $\Pi \subseteq R^A$ we define the function $\omega_\Pi : A^2 \twoheadrightarrow R \cup \{\pm\infty\}$:

$$(6) \qquad (a,b) \mapsto \omega_\Pi(a,b) := \sup(\pi(b) - \pi(a) \mid \pi \in \Pi).$$

We call any $\pi \in \Pi_\omega$ a *realization* of ω, and $\hat{\omega} := \omega_{\Pi_\omega}$ the *limits* associated with ω. It is easily seen that the limits associated with $\hat{\omega}$ are just $\hat{\omega}$, i.e., $\omega_{\Pi_{\hat{\omega}}} = \hat{\omega}$.

THEOREM 1. *The set Π_ω is not empty if and only if $\mathscr{D}_A^\omega$ contains no closed directed cycles of negative total weight. Under these circumstances the associated limits for a, $b \in A$ are given by*

$$(7) \qquad \hat{\omega}(a,b) = \bar{\omega}(a,b) := \inf(\omega(a,a_1) + \cdots + \omega(a_n,b) \mid a_1, \cdots, a_n \in A),$$

i.e., by the lengths of the shortest paths in $\mathscr{D}_A^\omega$.

Proof. If for a map $\omega : A \times A \twoheadrightarrow R$ there exists some $\pi : A \twoheadrightarrow R$ with $\pi(b) - \pi(a) \leqq \omega(a, b)$ for all $a, b \in A$, then obviously for any cyclic sequence $a_0, a_1, \cdots, a_n = a_0$ we must have

$$\begin{aligned} &\omega(a_0,a_1) + \omega(a_1,a_2) + \cdots + \omega(a_{n-1},a_0) \\ (8) \qquad &\geqq (\pi(a_1) - \pi(a_0)) + (\pi(a_2) - \pi(a_1)) + \cdots + (\pi(a_0) - \pi(a_{n-1})) \\ &= 0. \end{aligned}$$

Vice versa, if this condition holds for any $a_0, a_1, \cdots, a_n = a_0 \in A$, then $-\omega(a_n, a_1) \leqq \omega(a_1, a_2) + \cdots + \omega(a_{n-1}, a_n)$ for all $a_1, \cdots, a_n \in A$. Hence for any fixed $x \in A$ the map $\pi_x : A \twoheadrightarrow R$ defined by

$$(9) \quad \pi_x(a) := \inf(\omega(x,a_1) + \omega(a_1,a_2) + \cdots + \omega(a_{n-1},a_n) + \omega(a_n,a) \mid a_1, \cdots, a_n \in A)$$

for every $a \in A$ is well defined and satisfies $-\omega(a, x) \leqq \pi_x(a) \leqq \omega(x, a)$. Moreover, for any fixed $a \in A$ it satisfies

$$\begin{aligned} &\pi_x(b) := \inf(\omega(x,a_1) + \cdots + \omega(a_n,b) \mid a_1, \cdots, a_n \in A) \\ (10) \qquad &\leqq \inf(\omega(x,a_1) + \cdots + \omega(a_{n-1},a_n) + \omega(a_n,b) \mid a_1, \cdots, a_n \in A \text{ and } a_n = a) \\ &= \pi_x(a) + \omega(a,b) \end{aligned}$$

and therefore $\pi_x(b) - \pi_x(a) \leqq \omega(a, b)$ for all $a, b \in A$, as desired.

To show finally that $\hat{\omega}(a, b)$ is indeed equal to the length $\bar{\omega}(a, b)$ of the shortest directed path connecting a to b in $\mathscr{D}_A^\omega$ (or, more precisely, the infimum over all such paths), we observe that if $\Pi_\omega \neq \emptyset$ and therefore $\pi_a \in \Pi_\omega$ for all $a \in A$ we have

$$\begin{aligned} \hat{\omega}(a,b) &= \sup\,(\pi(b)-\pi(a) \mid \pi \in \Pi_\omega) \\ &\leqq \inf\,(\omega(a,a_1)+\cdots+\omega(a_n,b) \mid a_1,\cdots,a_n \in A) \\ &= \bar{\omega}(a,b) = \pi_a(b) = \pi_a(b)-\pi_a(a) \qquad (11) \\ &\leqq \sup\,(\pi(b)-\pi(a) \mid \pi \in \Pi_\omega) \\ &= \hat{\omega}(a,b) \end{aligned}$$

for all a, $b \in A$. If $\Pi_\omega = \emptyset$, on the other hand, we have simply $\hat{\omega}(a, b) = \bar{\omega}(a, b) = -\infty$ for all a, $b \in A$. □

Remark 1.A. If in addition to the map $\omega : A^2 \to R$ we are given two maps η, ζ : $A \to R$ together with a realization $\pi \in \Pi_\omega$ such that $\eta \leqq \pi \leqq \zeta$, we necessarily have

$$(12) \qquad \eta(b)-\zeta(a) \leqq \pi(b)-\pi(a) \leqq \bar{\omega}(a,b).$$

That this condition is also sufficient for the existence of such a realization follows easily from Theorem 1, since if $A' := A \mathbin{\dot{\cup}} \{*\}$ and we define $\omega' : A' \times A' \to R$ by

$$(13) \qquad \omega'(a,b) := \begin{cases} \omega(a,b) & \text{if } a,b \in A, \\ \zeta(b) & \text{if } a=* \text{ and } b \in A, \\ -\eta(a) & \text{if } b=* \text{ and } a \in A, \\ 0 & \text{if } a=b=*, \end{cases}$$

then $\Pi_{\omega'} \neq \emptyset$ if and only if there exists no closed directed cycle of negative total ω'-weight in A', that is, if and only if we have

$$(14) \qquad \begin{aligned} 0 &\leqq \omega'(*,a)+\omega'(a,a_1)+\cdots+\omega'(a_n,b)+\omega'(b,*) \\ &= \zeta(a)+\omega(a,a_1)+\cdots+\omega(a_n,b)-\eta(b). \end{aligned}$$

This is equivalent to having $\eta(b) - \zeta(a) \leqq \bar{\omega}(a, b)$ for all a, $b \in A$, while $\pi' : A \cup \{*\} \to R$ is in $\Pi_{\omega'}$ if and only if $\pi : A \to R$: $a \mapsto \pi'(a) - \pi'(*)$ is in Π_ω and satisfies $\eta \leqq \pi \leqq \zeta$.

Remark 1.B. It follows in particular that for every subset $B \subseteq A$ every map π_B : $B \to R$ with

$$(15) \qquad \pi_B(b_2)-\pi_B(b_1) \leqq \bar{\omega}(b_1,b_2)$$

for all b_1, $b_2 \in B$ can be extended to a map $\pi \in \Pi_\omega$. Indeed, an extension $\pi : A \to R$ of π_B is in Π_ω if and only if

$$(16) \qquad \eta_B(a) := \sup_{b\in B}\,(\pi_B(b)-\bar{\omega}(a,b)) \leqq \pi(a) \leqq \zeta_B(a) := \inf_{b \in B}\,(\pi_B(b)+\bar{\omega}(b,a))$$

and $\pi(a_2) - \pi(a_1) \leqq \bar{\omega}(a_1, a_2)$ for all a, a_1, $a_2 \in A\backslash B$. Moreover, for all a_1, $a_2 \in A\backslash B$ we have

$$(17) \qquad \begin{aligned} &\eta_B(a_2)-\zeta_B(a_1) \\ &= \sup_{b_1,b_2 \in B}\,(\pi_B(b_2)-\bar{\omega}(a_2,b_2)-\pi_B(b_1)-\bar{\omega}(b_1,a_1)) \\ &\leqq \sup_{b_1,b_2 \in B}\,(\bar{\omega}(b_1,b_2)-\bar{\omega}(a_2,b_2)-\bar{\omega}(b_1,a_1)) \\ &\leqq \sup_{b_1,b_2 \in B}\,(\bar{\omega}(b_1,a_1)+\bar{\omega}(a_1,a_2)+\bar{\omega}(a_2,b_2)-\bar{\omega}(a_2,b_2)-\bar{\omega}(b_1,a_1)) \\ &= \bar{\omega}(a_1,a_2). \end{aligned}$$

Remark 1.C. If A is finite, Theorem 1 remains true even if ω is allowed to take on the value $+\infty$ occasionally; indeed, any time this happens the value $+\infty$ may be replaced by the sum $-\Sigma_{a,b\in A} \min(0, \omega(a, b))$ (or any larger value) without creating any closed directed cycles of negative total weight (unless such cycles existed before). This is not true if A is infinite, as shown by the example $A := Q$, $\omega(a, b) := \infty$ if $a < b$, $\omega(a, b) := -1$ if $a > b$ and $\omega(a, b) := 0$ if $a = b$. In other words, if A is infinite and $\omega : A \times A \twoheadrightarrow R \cup \{\infty\}$ satisfies $\omega(a, a) \geqq 0$ for all (or just for one) $a \in A$, then this does not necessarily imply $\bar{\omega}(a, b) > -\infty$ for all $a, b \in A$, and hence it does not guarantee the existence of some map $\pi : A \twoheadrightarrow R$ with $\pi(b) - \pi(a) \leqq \omega(a, b)$ for all $a, b \in A$.

3. Some corollaries. It seems that Roy was the first to apply Theorem 1 in the context of measurement theory [20], [21], where it enabled him to obtain new combinatorial characterizations of interval orders and semiorders. It has also been used by Doignon [5] to characterize semiorders with multiple thresholds, and by the present authors to obtain a finite "forbidden subgraph" characterization of strong double semiorders [9], as proposed in Cozzens and Roberts [3]. It seems to us, in fact, that Theorem 1 provides a very natural and powerful basis for every task dealing with the numerical representation of relational systems describing inexact ordinal and metrical information.

To support this contention, let us summarize existing results which show how Theorem 1 can be used to obtain streamlined proofs of the equivalence of all the known characterizations of interval orders and semiorders. An *interval order* is a binary relation $\mathscr{R} \subseteq A \times A$ on a finite set A such that

(S1) For all $a \in A$: $\neg a\mathscr{R}a$.

(S2) For all $a, b, c, d \in A$: $(a\mathscr{R}b$ and $c\mathscr{R}d) \Rightarrow (a\mathscr{R}d$ or $c\mathscr{R}b)$.

A *semiorder* is an interval order $\mathscr{R}$ that satisfies the following additional axiom:

(S3) For all $a, b, c, d \in A$: $(a\mathscr{R}b$ and $b\mathscr{R}c) \Rightarrow (a\mathscr{R}d$ or $d\mathscr{R}c)$.

THEOREM 2 (cf. Fishburn [12]). *If $\mathscr{R}$ is a binary relation on a finite set A, then the following statements are equivalent*:

(i) *$\mathscr{R}$ satisfies the axiom* (S2) *above.*

(ii) *There exist two functions $\vartheta, \psi : A \twoheadrightarrow R$ such that for all $a, b \in A$, $a\mathscr{R}b \Leftrightarrow \psi(b) < \vartheta(a)$.*

(iii) *There exist two functions $\vartheta, \psi : A \twoheadrightarrow R$ such that for all $a, b \in A$, $a\mathscr{R}b \Leftrightarrow \psi(b) < \vartheta(a)$ as well as $\neg a\mathscr{R}b \Leftrightarrow \vartheta(a) < \psi(b)$.*

(iv) *The digraph $\tilde{\mathscr{R}}_A = (A \times \{\pm 1\}, \tilde{\mathscr{R}})$ whose arcs are given by*

$$\tilde{\mathscr{R}} := \{[(a,-),(b,+)] \mid a\mathscr{R}b\} \cup \{[(b,+),(a,-)] \mid \neg a\mathscr{R}b\} \tag{18}$$

contains no closed directed cycles.

In particular, $\mathscr{R}$ is an interval order, i.e., it satisfies (S1) *in addition to* (S2), *if and only if there exist two functions $\vartheta, \psi : A \twoheadrightarrow R$ such that $\vartheta(a) < \psi(a)$ for all $a \in A$ and $\psi(b) < \vartheta(a) \Leftrightarrow a\mathscr{R}b$ for all $a, b \in A$, if and only if the digraph $\tilde{\mathscr{R}}'_A := (A \times \{\pm 1\}, \tilde{\mathscr{R}} \cup \{[(a, +), (a, -)] \mid a \in A\})$ contains no closed directed cycles.*

Proof. We shall prove these statements in the sequence (iv) $\Rightarrow$ (iii) $\Rightarrow$ (ii) $\Rightarrow$ (i) $\Rightarrow$ (iv).

(iv) $\Rightarrow$ (iii) Let us assign weights to the arcs of the complete digraph $\tilde{\mathscr{D}}_A$ on $A \times \{\pm 1\}$ by the rule

$$\omega(x,y) := \begin{cases} -1 & \text{if} [x,y] \in \tilde{\mathscr{R}}, \\ \infty & \text{otherwise.} \end{cases} \tag{19}$$

Then $\tilde{\mathscr{D}}_A$ contains no ω-negative cycles if and only if $\tilde{\mathscr{R}}_A$ contains no directed cycles at all. In this case, by Theorem 1 there exists a mapping $\pi : A \times \{\pm 1\} \to R$ such that

$$(20)\qquad \begin{aligned} &\pi(b,+)-\pi(a,-)\leqq\omega((a,-),(b,+))=-1 \quad \text{if } a\mathscr{R}b,\\ &\pi(a,-)-\pi(b,+)\leqq\omega((b,+),(a,-))=-1 \quad \text{if } \neg a\mathscr{R}b. \end{aligned}$$

Thus if we define $\vartheta(a) := \pi(a, -)$ and $\psi(a) := \pi(a, +)$ for all $a \in A$, we have $\psi(b) < \vartheta(a) \Leftrightarrow a\mathscr{R}b$ as well as $\psi(b) > \vartheta(a) \Leftrightarrow \neg a\mathscr{R}b$, as desired.

(iii) $\Rightarrow$ (ii) is trivial.

(ii) $\Rightarrow$ (i) Suppose we have $a, b, c, d \in A$ with $a\mathscr{R}b$ and $c\mathscr{R}d$, so that $\psi(b) < \vartheta(a)$ as well as $\psi(d) < \vartheta(c)$. Hence $\neg a\mathscr{R}d \Rightarrow \vartheta(a) \leqq \psi(d) \Rightarrow \psi(b) < \vartheta(a) \leqq \psi(d) < \vartheta(c) \Rightarrow c\mathscr{R}b$, as desired.

(i) $\Rightarrow$ (iv) We prove this by seeking a contradiction to the assumption that we have found a minimal counterexample. Since the digraph $\tilde{\mathscr{R}}_A$ is bipartite, any such cycle therein is of the form

$$(21)\qquad [(x,+),(a,-),(b,+),(c,-),(d,+),\cdots,(x,+)]$$

where $a\mathscr{R}b$, and its length obviously exceeds two. But then $a\mathscr{R}b$, $c\mathscr{R}d$, and $\neg c\mathscr{R}b$ together with (S2) implies $a\mathscr{R}d$, so we can omit $(b, +)$ and $(c, -)$ to get a shorter cycle $[(x, +), (a, -), (d, +), \cdots, (x, +)]$! □

We next prove the famous Scott–Suppes theorem on semiorders [23], together with later characterizations due to Roberts [17] and Roy and Vincke [21].

THEOREM 3. *Let $\mathscr{R} \subseteq A \times A$ be a binary relation on a finite set A. Then the following are equivalent*:

(i) *$\mathscr{R}$ is a semiorder.*

(ii) *There exists a function $\varphi : A \to R$ such that $\varphi(a) > \varphi(b) + 1 \Leftrightarrow a\mathscr{R}b$.*

(iii) *$\mathscr{R}$ is antisymmetric and there exists a linear order "$\prec$" on A which is compatible with $\mathscr{R}$, in the sense that*

$$(22)\qquad a\mathscr{R}b \Rightarrow a \succ b, \quad \textit{and}$$

$$(23)\qquad (a \succ b \succ c \textit{ and } a\mathscr{I}c) \Rightarrow (a\mathscr{I}b \textit{ and } b\mathscr{I}c),$$

where $\mathscr{I} = \mathscr{I}_{\mathscr{R}}$ is the symmetric complement of $\mathscr{R}$, given by $a\mathscr{I}b \Leftrightarrow (\neg a\mathscr{R}b$ and $\neg b\mathscr{R}a)$.

(iv) *The digraph on A whose arcs are given by $\mathscr{R} \cup \mathscr{I}$ has more arcs of $\mathscr{I}$ than $\mathscr{R}$ in every one of its closed directed cycles.*

Note that "$\prec$" and $\mathscr{I}$ together determine $\mathscr{R}$ via $a\mathscr{R}b \Leftrightarrow a \succ b$ *and* $\neg a\mathscr{I}b$.

Proof. We shall prove this in the order (i) $\Rightarrow$ (iv) $\Rightarrow$ (ii) $\Rightarrow$ (iii) $\Rightarrow$ (i).

(i) $\Rightarrow$ (iv) To prove this result, we make use of the following, purely combinatorial lemma.

LEMMA 3.A. *If $\mathscr{R} \subseteq A \times A$ and $\mathscr{I} := \{[a, b] \in A \times A \mid [a, b], [b, a] \notin \mathscr{R}\}$, then a minimal cycle $[a_0, a_1, \cdots, a_n = a_0] \in A^{n+1}$ with $[a_{i-1}, a_i] \in \mathscr{R} \cup \mathscr{I}$ for all $i = 1, \cdots, n$ and $\#\{i \in \{1, \cdots, n\} \mid [a_{i-1}, a_i] \in \mathscr{R}\} \geqq \#\{i \in \{1, \cdots, n\} \mid [a_{i-1}, a_i] \in \mathscr{I}\}$ necessarily has length $n \leqq 4$.*

Proof. We may assume without loss of generality that either $n \geqq 5$ and that $[a_0, a_1]$, $[a_1, a_2] \in \mathscr{R}$, $[a_2, a_3] \in \mathscr{I}$, or $[a_0, a_1]$, $[a_2, a_3] \in \mathscr{R}$, $[a_1, a_2] \in \mathscr{I}$, or $[a_{i-1}, a_i] \in \mathscr{R}$ for all $i = 1, \cdots, n$. But all these cases are impossible, since if $[a_0, a_3] \in \mathscr{R}$ the cycle $[a_0, a_3, \cdots, a_{n-1}, a_n = a_0]$ would be shorter and still would not have more $\mathscr{I}$-arcs than $\mathscr{R}$-arcs, while if $[a_0, a_3] \notin \mathscr{R}$ we have either $[a_3, a_0] \in \mathscr{R}$ or $[a_3, a_0] \in \mathscr{I}$, so $[a_0, a_1, a_2, a_3, a_4 = a_0]$ would be a shorter cycle, again with at most as many $\mathscr{I}$-arcs as $\mathscr{R}$-arcs. □

In view of this lemma it is enough to show that if $\mathscr{R} \subseteq A \times A$ is a semiorder, then any cycle $[a_0, a_1, \cdots, a_n = a_0] \in A^{n+1}$ of length $n \leqq 4$ contains more $\mathscr{I}$-arcs than $\mathscr{R}$-

arcs (cf. Fig. 1). Indeed for $n = 1$ this holds by axiom (S1); for $n = 2$ it holds since $a_0\mathcal{R}a_1$ and $a_1\mathcal{R}a_2$ would imply $a_0\mathcal{R}a_0$ or $a_1\mathcal{R}a_1$ by (S2), contradicting (S1); for $n = 3$ it holds since $a_0\mathcal{R}a_1$ and $a_1\mathcal{R}a_2$ implies the impossibility $a_0\mathcal{R}a_2$ by (S2) and (S1); finally, it holds for $n = 4$ since $a_0\mathcal{R}a_1$ and $a_1\mathcal{R}a_2$ implies $a_0\mathcal{R}a_3$ or $a_3\mathcal{R}a_2$ by (S3), while $a_0\mathcal{R}a_1$ and $a_2\mathcal{R}a_3$ implies $a_0\mathcal{R}a_3$ or $a_2\mathcal{R}a_1$, this time by (S2).

(iv) $\Rightarrow$ (ii) From our definition of a forbidden cycle, it is obvious that $a\mathcal{R}b$ implies $\neg b\mathcal{R}a$. We now define a weighting of the complete digraph on A as follows:

$$(24) \qquad \omega(a,b) := \begin{cases} -1-\varepsilon & \text{if } a\mathcal{R}b, \\ 1 & \text{if } a\mathcal{I}b \text{ and } a \neq b, \\ 0 & \text{if } a = b, \\ \infty & \text{if } b\mathcal{R}a \end{cases}$$

for all a, $b \in A$. If the digraph $(A, \mathcal{R} \cup \mathcal{I})$ contains no forbidden cycles, then for sufficiently small $\varepsilon > 0$ the above weighted digraph contains no negative cycles, from which it follows by Theorem 1 that there exists a function $\pi : A \rightarrow R$ with $\pi(b) - \pi(a) \leqq \omega(a, b)$ for all a, $b \in A$. By our choice of weights $\omega(a, b)$, this implies that $\pi(a) > \pi(b) + 1$ if and only if $a\mathcal{R}b$, so (ii) holds with $\pi := \varphi$, as desired.

(ii) $\Rightarrow$ (iii) We choose our linear order on A in the obvious way: $a \prec b \Leftrightarrow \varphi(a) < \varphi(b)$. Then $a\mathcal{R}b \Rightarrow \varphi(a) > \varphi(b) + 1 \Rightarrow a \succ b$ is clear. In addition, $a \succ b \succ c$ and $a\mathcal{I}c$ implies $\varphi(a) > \varphi(b) > \varphi(c)$ and $\varphi(a) - \varphi(c) \leqq 1$, so that $0 < \varphi(a) - \varphi(b) < \varphi(a) - \varphi(c) \leqq 1$ and $\varphi(b) - \varphi(c) < \varphi(a) - \varphi(c) \leqq 1$, i.e., $a\mathcal{I}b$ and $b\mathcal{I}c$, as desired.

(iii) $\Rightarrow$ (i) Since $a\mathcal{R}b \Rightarrow a \succ b \Rightarrow a \neq b$, (S1) is fulfilled. To verify (S2), assume $a, b, c, d \in A$ and $a\mathcal{R}b$ as well as $c\mathcal{R}d$. If $b = d$ there is nothing to prove; otherwise, by symmetry, we may assume $b \prec d$. Now $c\mathcal{R}d$ implies $c \succ d \succ b$ by (22), and therefore $\neg c\mathcal{I}b$ by (23) and $c\mathcal{R}d$. Hence either $c\mathcal{R}b$ or $b\mathcal{R}c$. But $b\mathcal{R}c$ cannot hold by (22), since $c \succ d \succ b$, so we get $c\mathcal{R}b$, as desired.

To verify (S3), assume a, b, c, $d \in A$ and $a\mathcal{R}b$ as well as $b\mathcal{R}c$. Again, if $b = d$ there is nothing to prove. If $b \prec d$, then $b\mathcal{R}c$ implies $d \succ b \succ c$ by (22) and therefore

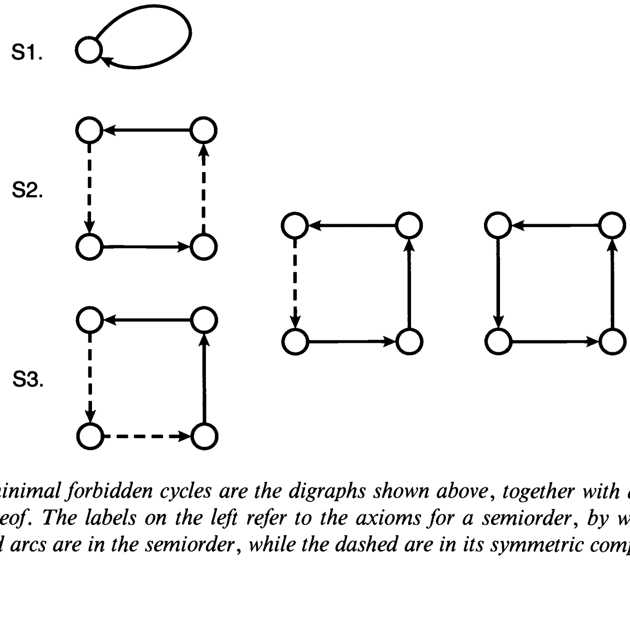


FIG. 1. *The minimal forbidden cycles are the digraphs shown above, together with all digraphs obtained by contraction thereof. The labels on the left refer to the axioms for a semiorder, by which these cycles are excluded. The solid arcs are in the semiorder, while the dashed are in its symmetric complement.*

$\neg c\mathscr{I}d$ by (23) and $b\mathscr{R}c$. Hence either $c\mathscr{R}d$ or $d\mathscr{R}c$. But $c\mathscr{R}d$ would contradict $d \succ b \succ c$ by (22), so we get $d\mathscr{R}c$, as desired. Similarly, $d \prec b$ implies $a\mathscr{R}d$. □

We remark that, according to [17], the ternary relation $\mathscr{B}_{(\mathscr{I},\prec)}$ defined by

$$(25) \qquad \mathscr{B}_{(\mathscr{I},\prec)} := \{[a,b,c] \in A^3 \mid a \prec b \prec c \text{ or } a\mathscr{I}b \text{ or } b\mathscr{I}c\}$$

satisfies the axioms T1–T7 of "ε-betweenness," which was considered there as part of a systematic study of "tolerance geometries." These axioms constitute a natural relaxation of Tarski's betweenness axioms [25]. It has also been shown by Roberts [18] that for any such $\mathscr{B} \subseteq A^3$ satisfying T1–T7, there exists a semiorder $\mathscr{R} \subseteq A^2$ and a linear order "$\prec$" on A, compatible with $\mathscr{R}$, such that $\mathscr{B} = \mathscr{B}_{(\mathscr{I},\prec)}$, where

$$(26) \qquad \mathscr{I} = \mathscr{I}_{\mathscr{R}} = \{\{a,b\} \subseteq A \mid \neg a\mathscr{R}b \text{ and } \neg b\mathscr{R}a\}$$

as above. Hence the equivalence (iii) $\Leftrightarrow$ (ii) established in Theorem 3 above can be viewed as a simple proof of yet another result of Roberts [18], which states that $\mathscr{B} \subseteq A^3$ satisfies the axioms T1–T7 of ε-betweenness if and only if there exists a map $\pi : A \twoheadrightarrow R$ and some $\varepsilon > 0$, say $\varepsilon = 1$, such that

$$(27) \quad \mathscr{B} = \{[a,b,c] \in A^3 \mid |\pi(a) - \pi(b)| + |\pi(b) - \pi(c)| \leqq |\pi(a) - \pi(c)| + 2\varepsilon\}.$$

Finally, we observe that Theorem 1 provides us with a simple solution to the fundamental problem of distance geometry in one dimension, provided that the chirality information is complete so that the order of the points along the real line is known.

THEOREM 4. *Let* λ, $v : A^2 \twoheadrightarrow R$ *be two symmetric functions which satisfy* (B1)–(B3), *and let* $\chi : A^2 \twoheadrightarrow \{-1, 0, +1\}$ *be a function with* $\chi(a, b) = -\chi(b, a)$ *and* $\chi(a, b) = 0 \Rightarrow v(a, b) = 0$, *for all* a, $b \in A$ (*as in* § 1). *Also suppose that the binary relation* "$\prec$" *on A defined by*

$$(28) \qquad \begin{aligned} &a \preceq b \quad \textit{and} \quad a \not\succeq b \quad \textit{if}\, \chi(a,b) = 1, \\ &a \preceq b \quad \textit{and} \quad a \succeq b \quad \textit{if}\, \chi(a,b) = 0, \\ &a \not\preceq b \quad \textit{and} \quad a \succeq b \quad \textit{if}\, \chi(a,b) = -1, \end{aligned}$$

for all a, $b \in A$ *is a weak order.*[1] *Then we have* $\Pi(\lambda, v, \chi) = \Pi_\omega$, *where*

$$(29) \qquad \omega(a,b) := \begin{cases} v(a,b) & \textit{if}\, a \not\succeq b, \\ -\lambda(a,b) & \textit{if}\, a \succeq b. \end{cases}$$

Proof. Under the given conditions, for all $\pi \in R^A$ and a, $b \in A$, we have

$$(30) \qquad \pi(b) - \pi(a) = \begin{cases} |\pi(b) - \pi(a)| \leqq v(a,b) & \text{if}\, a \not\succeq b, \\ -|\pi(b) - \pi(a)| \leqq -\lambda(a,b) & \text{if}\, a \succeq b, \end{cases}$$

if and only if $\lambda(a, b) \leqq |\pi(b) - \pi(a)| \leqq v(a, b)$, so that $\pi \in \Pi_\omega \Leftrightarrow \pi \in \Pi(\lambda, v, \tilde{\chi})$, as desired. □

4. Algorithms. Theorem 4 enables us to solve the fundamental problem of distance geometry in the one-dimensional case, by the enumeration of all weak orders that are compatible with a given chirality function $\tilde{\chi}$. In addition, if the given distance geometry description is feasible, they enable us to calculate the associated Euclidean limits, and to

[1] A *weak* order is one that is transitive and strongly complete, but for which $a \preceq b \preceq a$ is possible even when $a \neq b$.

compute a one-dimensional representation $\pi : A \twoheadrightarrow R$. In this section we shall give a formal statement of an algorithm that does this reasonably efficiently (although of course not in polynomial time), and discuss its wider complexity-theoretic implications. For simplicity, we shall restrict ourselves to the case in which an *embedding* (injective representation) is sought, because then it is sufficient to consider only chirality constraints of the form $\tilde{\chi}(a, b) \subseteq \{\pm 1\}$ for all $a, b \in A$ $(a \neq b)$. Hence the only weak orders needed are linear orders, which can be represented by permutations. We also assume that the atoms A have been ordered as $a_1, \cdots, a_n$, and refer to them by their indices.

The enumeration of permutations is most simply done by means of a recursive procedure. If n is a global variable giving the number of points, this procedure uses three global n by n arrays: The first of these, B, holds the given lower and upper distance bounds in its lower and upper triangles, respectively, i.e., $B[j, i] = \lambda(a_i, a_j)$ and $B[i, j] = v(a_i, a_j)$ for all $1 \leqq i < j \leqq n$. The second, X, is an antisymmetric array which contains the chirality constraints, i.e., $X[i, j] = 1 \Leftrightarrow \pi(a_j) - \pi(a_i) > 0$ in any admissible representation $\pi : A \twoheadrightarrow R$, or $X[i, j] = 0$ if the order of $\pi(a_i)$ and $\pi(a_j)$ is unconstrained. The third, S, will hold the smoothed lower and upper Euclidean limits in its lower and upper triangles when the procedure is completed. The following formal procedure fills in the details (wherein comments are delimited by "#" and newlines).

```
# Main procedure: all arguments are global to subsequent routines.
procedure Embed 1($n$,$B$,$S$,$X$) # The name is historical in origin.
    # Initialize $P$ (the initial permutation), $S$ (the array to contain the
    # smoothed bounds) and $T$ (the arc weights for the initial permutation).
    if $X[2, 1] = 1$ then $P$ := array([2, 1]) else $P$ := array([1, 2]) endif;
    for $i := 1$ to $n - 1$ do for $j := i + 1$ to $n$ do
            $S[i, j] := 0$,        $S[j, i] := \infty$;
    enddo enddo;
    $T$ := array($1\cdots2$, $1\cdots2$);
    $T[1, 1] := 0$, $T[2, 2] := 0$, $T[P[1], P[2]] := B[1, 2]$, $T[P[2], P[1]] := -B[2, 1]$;
    # Real_Work is TRUE if bounds and chiralities are feasible.
    if not Real_Work( $P$,$T$ )
    then   print( 'Infeasible constraints.' );
    else   print( 'Smoothed bounds:' ), print( $S$ );
    endif;
end.
```

At the kth level of the recursion we have a permutation P of $[1, \cdots, k \leqq n]$ together with an array of shortest path lengths T, which were computed at the previous level; initially, $k = 2$, $P = [2, 1]$ if $X[2, 1] = 1$ or $[1, 2]$ otherwise, $T[1, 1] := T[2, 2] := 0$ and $T[P[1], P[2]] := B[1, 2]$, $T[P[2], P[1]] := -B[2, 1]$ as above. (By starting with $k := 2$ and P initialized this way we avoid the redundant computation of the inversions of permutations.) To generate all possible permutations of $[1, \cdots, k + 1]$ that contain P as a subsequence, we simply insert $k + 1$ between the hth and $(h + 1)$th members of P for all $h = 0, \cdots, k$ (where insertion between 0 and 1 means prepending and insertion between k and $k + 1$ means appending). If this insertion is not compatible with the chirality constraints in X, we try to insert $k + 1$ at the next position in the sequence.

The new arc weights $W[i, j]$ are then equal to those in T for $1 \leqq i, j \leqq k$, whereas the arc weights $W[P[i], k + 1]$ and $W[k + 1, P[i]]$ will be equal to the corresponding upper bounds or to the negative lower bounds, depending on whether $P[i]$ comes before or after $k + 1$ in the new permutation. The arc weights for each compatible permutation are then tested for the presence of negative cycles; since the digraph changes only by the addition of a $(k+1)$th node at each level of the recursion, this is most efficiently done by updating the old matrix of shortest path lengths T, rather than computing it from scratch (the details of how the corresponding procedure *Update* does this may be found, for example, in [10]). The bounds and chiralities are feasible, of course, if any one of the permutations of $[1, \cdots, k]$ can be completed to give the order of the points along the real line in a representation. Altogether, we get:

```
# Here we do the real work of recursively enumerating all $X$-compatible permutations.
function Real_Work( P,T )
    $k$ := size of( $P$);
    if $k = n$ #End of recursion: output representation for the given permutation $P$.
    then print( 'Representation for', $P$ ), print( New_Representation( $n,T$ ) );
        New_Limits( $P,T$ );
        return( TRUE );
    endif;
    # Otherwise, try to insert the $(k+1)$th point at each possible position in $P$.
    feasible := FALSE, $W$ := array( $1 \cdots k+1$, $1 \cdots k+1$);
    for $h := 0$ to $k$
    do  for $i := 1$ to $k$ do for $j := 1$ to $k$ do $W[i,j] := T[i,j]$;
        # Set up new arc weights for the new permutation
        # while checking its compatibility with $X$.
        for $i := 1$ to $h$
        do  if $X[h,i] = 1$ then next $h$;           # skip rest of $h$-loop
            $W[P[i], k+1] := B[P[i], k+1]$;
            $W[k+1, P[i]] := -B[k+1, P[i]]$;
        enddo;
        for $i := h+1$ to $k$
        do  if $X[i,h] = 1$ then next $h$;           # skip rest of $h$-loop
            $W[P[i], k+1] := -B[k+1, P[i]]$;
            $W[k+1, P[i]] := B[P[i], k+1]$;
        enddo;
        # Update shortest paths in digraph on $\{1, \cdots, k\}$ by paths
        # through new node $k+1$ (this can be done in $O(k^2)$ time; a
        # FALSE return indicates that a negative cycle was found).
        if Update( $k+1,W$ )
        then  $Q$ := array([$P[1 \cdots h]$, $k+1$, $P[h+1 \cdots k]$]);
              # $Q$ is the next permutation (which becomes a new copy
              # of $P$ inside the following recursive call to Real_Work).
              if Real_Work( $Q,W$ ) then
                            feasible := TRUE;
        endif   endif;
    enddo;
    return( feasible );
end.
```

The recursion ends when $k = n$. At this point, we compute a representation $\pi : A \rightarrow R$ of the bounds, and revise our current estimate of the Euclidean limits in S. The representation is computed by putting down each new point $\pi(a_i)$ at some random position within the range implied by the current limits (shortest path lengths) and the previously chosen points $\{\pi(a_j) \mid j = 1, \cdots, i-1\}$. This works because Remark 1.B guarantees us that every possibility within this range can be extended to a complete representation.

```
# This function returns a representation $R$ between the limits for this permutation.
function $New\_Representation(\, n,T\,)$
    $R :=$ array$(1\cdots n)$, $R[1] := 0$;
    for $i := 2$ to $n$
    do  $lo\_lim := \max\,(R[i] - T[i,j] \mid j = 1\cdots i-1)$;
        $up\_lim := \min\,(R[i] + T[j,i] \mid j = 1\cdots i-1)$;
        # $Random$ returns a random number between its arguments.
        $R[i] := Random(lo\_lim,\, up\_lim)$;
    enddo;
    return$(\,R\,)$;
end.
```

Finally, since the upper and lower limits on the distances over all representations in which the order of the points on the real line is that specified by the complete permutation P are given by the directed path lengths relative to W and their negatives, respectively, for each pair of indices $P[i]$, $P[j]$, the revised estimate of the Euclidean upper limits is obtained by taking the larger of the previous estimate in S and the shortest path between them relative to W, while the revised estimate of the Euclidean lower limits is obtained by taking the smaller of the previous estimate and the negative of shortest path between them, i.e.,

```
# This procedure returns the new estimate of the Euclidean limits updated
# by the limits $T$ for the permutation $P$.
procedure $New\_Limits(\, P,T\,)$
    for $i := 1$ to $n$ do for $j := i + 1$ to $n$
    do  if $P[i] < P[j]$
        then  $S[P[i],P[j]] := \max\,(S[P[i],P[j]],\ T[P[i],P[j]])$;
              $S[P[j],P[i]] := \min\,(S[P[j],P[i]],\ -T[P[j],P[i]])$;
        else  $S[P[j],P[i]] := \max\,(S[P[j],P[i]],\ T[P[i],P[j]])$;
              $S[P[i],P[j]] := \min\,(S[P[i],P[j]],\ -T[P[j],P[i]])$;
        endif;
    enddo;
end.
```

This algorithm for computing the one-dimensional Euclidean limits extends our previous work [8] on computing the limits implied by the triangle inequality alone.[2]

Since matrices of shortest path lengths among $\#A = n$ nodes can be updated for the addition of a new node in time $O(n^2)$ [10], the above algorithm establishes that in one dimension the fundamental problem can be solved in time $O(n^3 \cdot n!)$.[3] Recent work in computational semialgebraic geometry [13], [15] has shown that the complexity of deciding the feasibility of any system of polynomial inequalities $p_1 \geqq 0, \cdots, p_m \geqq 0$ is $s^{O(v^2)}$, where v is the number of variables and $s = \sum_{k=1}^{m} \deg\,(p_k)$ is the sum of the total degrees. Assuming that we have no nontrivial chirality constraints, clearly the fundamental problem in all dimensions is equivalent to such a polynomial system with $p_k(\pi) := D_{ij}(\pi) - \lambda^2(a_i, a_j)$ or $p_k(\pi) := v^2(a_i, a_j) - D_{ij}(\pi)$ for $k = 1, \cdots, m = 2\binom{n}{2}$, where D_{ij} denotes the squared distance of $a_i, a_j \in A$ and $\pi \in (R^d)^A$ gives the coordinates of the atoms (cf. (2)). In one dimension the number of variables is $v = nd = n$, and

[2] The program itself has been implemented in the MAPLE symbolic programming language [2] running on a Sun 3/160, and is available upon request from Timothy Havel.

[3] Since only rational arithmetic is needed by the algorithm, no special model of computation over the reals is needed.

since the squared distances all have degree two, $s = 2m = 4\cdot\binom{n}{2} = 2n(n-1)$, so the complexity of the general algorithms applied to these systems of polynomials is $n^{O(n^2)}$. Because $n!$ is asymptotically proportional to $\sqrt{n}e^{-n}n^n$ by Sterling's approximation, the above algorithm improves upon the general algorithms in terms of asymptotic (as well as absolute) time.

The example $\lambda(a_i, a_j) = 1$ and $v(a_i, a_j) = n - 2$ for all $a_i, a_j \in A$ with no chirality constraints shows that these worst-case time bounds are actually tight, so that our algorithm cannot be expected to *reliably* solve problems with n greater than about 10. In many cases, however, much larger problems may be within reach. We point out that the average performance of the algorithm could be substantially improved upon by using the method of generating topological sorting arrangements described in [16] to find all X-compatible permutations, in which case the amount of time it required would only be proportional to the number of linear orders that are compatible with the chirality information encoded by $\tilde{\chi}$. If we restrict the input to those functions $\tilde{\chi}$ for which the associated digraph on A with arc set $\{[a, b] \in A^2 \mid \tilde{\chi}(a, b) = \{1\}\}$ contains a directed path through all of A as a subgraph (so that there exists at most one compatible linear order), this modified algorithm actually runs in polynomial-time. More general improvements in the algorithm's average performance can be obtained by deriving chirality information from the available distance information. For example, for all $a \neq b \neq c \neq a$ in A one knows a priori that

$$v(a,c) \leqq \lambda(a,b)+\lambda(b,c) \Rightarrow \chi(a,b)\cdot\chi(b,c) \neq 1, \tag{31}$$

so that any permutation containing $[a, b]$ need not be extended in any way that gives $[b, c]$. Similar preclusions also exist on four and more points, but it is doubtful that the time it may take to check them would be compensated for by the time they saved.

5. ε-Collinearity. Since the ε-betweenness relation that (25) defines can be studied in arbitrary metric spaces, Robert's characterization provides us with conditions under which ε-betweenness in metric spaces can be translated to ε-betweenness for points on a line. We now show that a similar relation in metric spaces, which we call *ε-collinearity*, is sufficient to guarantee that the metric space as a whole is one-dimensional to within ε.

Hence let A be a finite set and let $\rho : A \times A \to R$ be a function such that

(i) $\rho(a, a) = 0$ for all $a \in A$.

(ii) $\rho(a, b) = \rho(b, a) > 0$ for all $a, b \in A$ with $a \neq b$.

Clearly, all metrics on A are examples of such functions (which are sometimes called *semimetrics*). Given an $\varepsilon > 0$, we define a triple $a, b, c \in A$ to be ε-collinear (with respect to ρ) if

$$|\rho(a,b)+\rho(a,c)+\rho(b,c)-2\max(\rho(a,b),\rho(a,c),\rho(b,c))| \leqq \varepsilon. \tag{32}$$

Note that if $|\rho(a, b) - |\pi(a) - \pi(b)|| \leqq \varepsilon$ for some $\pi : A \to R$ and all $a, b \in A$, then all triples in A are $3\cdot\varepsilon$-collinear.

THEOREM 5. *Let $\rho : A \times A \to R$ be a function which satisfies conditions* (i) *and* (ii) *above, and suppose* (*rescaling if necessary*) *that $\rho(a, b) \geqq 1$ for all $a, b \in A$ with $a \neq b$. Then if for some $0 < \varepsilon < \frac{1}{3}$ all triples in A are ε-collinear, there exists a function $\pi : A \to R$ such that $|\rho(a, b) - |\pi(a) - \pi(b)|| \leqq \varepsilon$, provided that $\#A \neq 4$.*

Remark 5.A. The map $\rho_0 : \{1, 2, 3, 4\}^2 \to R$ given by

$$i,j \mapsto \begin{cases} 0 & \text{if } i=j, \\ 1 & \text{if } i \equiv j+1 \pmod 4, \\ 2 & \text{if } i \equiv j+2 \pmod 4 \end{cases} \tag{33}$$

in which every triple is 0-collinear shows conclusively that the condition $\#A \neq 4$ cannot be eliminated.

Proof: We define the *interval* of any $a, b \in A$ by

$$[\![a,b]\!] := \{c \in A \mid \rho(a,b) \geqq \max(\rho(a,c), \rho(b,c))\}. \tag{34}$$

We claim the map $A \times A \to 2^A : [a, b] \mapsto [\![a, b]\!]$ satisfies the following properties for all $a, b, c \in A$:

(I1) $a, b \in [\![a, b]\!] = [\![b, a]\!]$;

(I2) $a \in [\![b, c]\!]$ or $b \in [\![a, c]\!]$ or $c \in [\![a, b]\!]$;

(I3) $a \in [\![b, c]\!]$ and $b \in [\![a, c]\!] \Rightarrow a = b$;

(I4) $c \in [\![a, b]\!] \Rightarrow [\![a, c]\!] \subseteq [\![a, b]\!]$, and $c \in [\![b, d]\!]$ for all $d \in [\![a, c]\!]$.

Properties (I1) and (I2) are obvious from the definition. Property (I3) follows since the assumptions imply that $\rho(b, c) = \rho(a, c) \geqq \rho(a, b)$, which together with ε-collinearity implies $\rho(a, b) \leqq \varepsilon < 1$ and hence $\rho(a, b) = 0$, that is, $a = b$.

To prove (I4), we observe that the assumptions imply

$$-\varepsilon \leqq -\rho(a,b) + \rho(a,c) + \rho(b,c) \leqq \varepsilon \tag{35}$$

and

$$-\varepsilon \leqq -\rho(a,c) + \rho(a,d) + \rho(c,d) \leqq \varepsilon. \tag{36}$$

Therefore

$$-2\varepsilon \leqq -\rho(a,b) + \rho(a,d) + \rho(c,d) + \rho(b,c) \leqq 2\varepsilon. \tag{37}$$

In view of the fact that

$$2\rho(b,d) \leqq 2\max(\rho(b,d), \rho(b,c), \rho(c,d)) \leqq \rho(b,d) + \rho(b,c) + \rho(c,d) + \varepsilon, \tag{38}$$

i.e.,

$$\rho(b,d) \leqq \rho(b,c) + \rho(c,d) + \varepsilon, \tag{39}$$

this implies that

$$-\rho(a,b) + \rho(a,d) + \rho(b,d) \leqq 3\varepsilon.$$

It follows that

$$\rho(b,d) \leqq \rho(a,b) + (3\varepsilon - \rho(a,d)) \leqq \rho(a,b) \tag{40}$$

as well as

$$\rho(a,d) \leqq \rho(a,b) + (3\varepsilon - \rho(b,d)) \leqq \rho(a,b), \tag{41}$$

i.e., $d \in [\![a, b]\!]$. So indeed we have $[\![a, c]\!] \subseteq [\![a, b]\!]$, as claimed. In addition, we get

$$-\varepsilon \leqq \rho(a,b) - \rho(a,d) - \rho(b,d) \leqq \varepsilon, \tag{42}$$

which together with (37) implies

$$-3\varepsilon \leqq \rho(c,d) + \rho(b,c) - \rho(b,d) \leqq 3\varepsilon. \tag{43}$$

Together with $\varepsilon < \frac{1}{3}$, we get $\rho(c, d), \rho(b, c) \leqq \rho(b, d)$, i.e., $c \in [\![b, d]\!]$ as desired.

Hence we may invoke the following lemma, whose proof we shall outline after finishing the proof of the theorem (it may well be folklore).

LEMMA 5.A. *A map* $A \times A \to 2^A : [x, y] \mapsto [\![x, y]\!]$ *satisfies conditions* (I1)–(I4) *if and only if either*

(i) $\#A = 4$ *and there exists an ordering of* A, *say* $A = [a, b, c, d]$, *such that* $[\![x, y]\!] = \{x, y\}$ *if either* $x = y$ *or* $\{x, y\} \in \{\{a, b\}, \{b, c\}, \{c, d\}, \{a, d\}\}$, and $[\![a, b]\!] = [\![b, d]\!] = \{a, b, c, d\}$;

(ii) *There exists a linear ordering* "$\prec$" *of* A, *uniquely determined by* "$[\![\cdot]\!]$" *up to inversion, such that for* $x \prec y$ *we have* $[\![x, y]\!] = \{z \mid x \preceq z \preceq y\}$.

Because of this lemma, we may henceforth assume that there exists such a linear order "$\prec$" on A such that $a \prec b \prec c$ implies $\rho(a, c) \geqq \rho(a, b), \rho(b, c)$ and hence

$$-\varepsilon \leqq \rho(a,c) - \rho(a,b) - \rho(b,c) \leqq \varepsilon. \tag{44}$$

Let us therefore define a (nonsymmetric!) map $\omega : A \times A \to R$ by

$$\omega(a,b) := \begin{cases} \rho(a,b) + \varepsilon & \text{if } a \prec b, \\ 0 & \text{if } a = b, \\ -\rho(a,b) + \varepsilon & \text{if } a \succ b. \end{cases} \tag{45}$$

We claim that for all $a, b, c \in A$ we have

$$\omega(a,c) \leqq \omega(a,b) + \omega(b,c). \tag{46}$$

It is sufficient to check this only for the cases $a \prec b \prec c$ and $a \prec c \prec b$. In the first case we have indeed

$$\begin{aligned} \omega(a,c) &= \rho(a,c) + \varepsilon \leqq \rho(a,b) + \rho(b,c) + 2\varepsilon \\ &= \omega(a,b) + \omega(b,c) \end{aligned} \tag{47}$$

while in the second case we have

$$\begin{aligned} \omega(a,c) &= \rho(a,c) + \varepsilon \leqq \rho(a,b) - \rho(b,c) + 2\varepsilon \\ &= \omega(a,b) + \omega(b,c) \end{aligned} \tag{48}$$

as claimed.

It now follows from Theorem 1 that there exists some $\pi : A \to R$ such that $\pi(b) - \pi(a) \leqq \omega(a, b)$ for all $a, b \in A$, i.e., such that for $a \prec b$ we have

$$0 < \rho(a,b) - \varepsilon \leqq \pi(b) - \pi(a) \leqq \rho(a,b) + \varepsilon \tag{49}$$

and hence $\pi(b) - \pi(a) = |\pi(a) - \pi(b)|$ as well as

$$|\rho(a,b) - |\pi(a) - \pi(b)|| \leqq \varepsilon. \tag{50}$$ □

Proof of Lemma 5.A. The lemma is obvious if $\#A \leqq 3$; for $\#A = 4, 5$ it can be proven by a straightforward case by case analysis. To prove the lemma for $\#A > 5$ we use induction with respect to $\#A$ together with the fact that two linear orders define the same "interval map" $[\![\cdot]\!] : A \times A \to R$ if and only if they are either the same or are inversions of one another.

Hence take some $a \in A$ and let "$\prec_a$" be the linear order on $A \backslash a$ such that $[\![x, y]\!] = \{z \mid x \prec_a z \prec_a y\}$ which exists by the induction hypothesis. Similarly let "$\prec_b$" be the linear order which exists on $A \backslash b$ by induction ($a \neq b$). Without loss of generality we may assume that "$\prec_a$" and "$\prec_b$" agree on $A \backslash \{a, b\}$. We now define a relation "$\prec$" on A by

$$x \prec y \Leftrightarrow x \prec_a y \quad \text{or} \quad x \prec_b y \tag{51}$$

if $\{x, y\} \neq \{a, b\}$, while we put $a \prec b$ if there exists some $z \in A \backslash \{a, b\}$ such that either (1) $z \prec_b a$ and $a \in [\![b, z]\!]$, or (2) $a \prec_b z \prec_a b$, or (3) $b \prec_a z$ and $b \in [\![a, z]\!]$; otherwise we put $b \prec a$. We verify that "$\prec$" is a well-defined linear order on A and that for $x \prec y$ we have $[\![x, y]\!] = \{z \mid x \preceq z \preceq y\}$, as claimed. □

Acknowledgments. We thank the referees for useful suggestions regarding this work, and for bringing several important references to our attention.

REFERENCES

[1] L. Blumenthal, *Theory and Applications of Distance Geometry*, Cambridge University Press, Cambridge, UK, 1953 (reprinted by Chelsea, Bronx, NY, 1970).

[2] B. W. Char, G. J. Fee, K. O. Geddes, G. N. Gonnet, and M. B. Monagan, *A tutorial introduction to maple*, J. Symb. Comput., 2 (1986), pp. 179–200.

[3] M. B. Cozzens and F. S. Roberts, *Double semiorders and double indifference graphs*, SIAM J. Algebraic Discrete Methods, 3 (1982), pp. 566–583.

[4] G. M. Crippen and T. F. Havel, *Distance Geometry and Molecular Conformation*, Research Studies Press, Letchworth, UK, 1988.

[5] J.-P. Doignon, *Threshold representations of multiple semiorders*, SIAM J. Algebraic Discrete Methods, 8 (1987), pp. 77–84.

[6] A. W. M. Dress, A. S. Dreiding, and H. R. Haegi, *Classification of mobile molecules by category theory*, in Symmetries and Properties of Nonrigid Molecules: A Comprehensive Survey, Studies in Physical and Theoretical Chemistry, Vol. 23, J. Maruani and J. Serre, eds., Elsevier Scientific, Amsterdam, 1983, pp. 39–58.

[7] A. W. M. Dress, *Chirotopes and oriented matroids*, in Diskrete Strukturen, algebraische Methoden und Anwendungen, Bayreuther Math. Schriften, Vol. 21, A. Kerber, ed., 1986.

[8] A. W. M. Dress and T. F. Havel, *Shortest path problems and molecular conformation*, Discrete Appl. Math., 19 (1988), pp. 129–144.

[9] ———, *Criteria for the global consistency of two-threshold preference relations in terms of forbidden subconfigurations*, Adv. in Appl. Math., 10 (1989), pp. 379–395.

[10] S. E. Dreyfus and A. M. Law, *The Art and Theory of Dynamic Programming*, Mathematics in Science and Engineering, Vol. 130, R. Bellman, ed., Academic Press, New York, 1979.

[11] P. C. Fishburn, *Utility Theory for Decision Making*, John Wiley, New York, 1970.

[12] ———, *Interval Orders and Interval Graphs—A Study in Partially Ordered Sets*, John Wiley, New York, 1985.

[13] D. Yu. Grigor'ev and N. N. Vorobjov, *Solving systems of polynomial inequalities in subexponential time*, J. Symb. Comput., 5 (1988), pp. 37–64.

[14] T. F. Havel, I. D. Kuntz, and G. M. Crippen, *The theory and practice of distance geometry*, Bull. Math. Biol., 45 (1983), pp. 665–720.

[15] J. Heintz, P. Solernó, and M.-F. Roy, *On the complexity of semialgebraic sets*, in Proc. Information Processing 89, G. X. Ritter, ed., Elsevier Science Publishers B.V., North-Holland, 1989, pp. 293–298.

[16] D. E. Knuth and J. L. Szwarcfiter, *A structured program to generate all topological sorting arrangements*, Inform. Process. Lett., 2 (1974), pp. 153–157.

[17] F. S. Roberts, *On the compatibility between a graph and a simple order*, J. Combin. Theory, 11 (1971), pp. 28–38.

[18] ———, *Tolerance geometry*, Notre Dame J. Formal Logic, 14 (1973), pp. 68–76.

[19] ———, *Measurement Theory*, Encyclopedia of Mathematics, Vol. 7, Addison-Wesley, Reading, MA, 1979.

[20] M. Roubens and Ph. Vincke, *Preference Modelling*, Lecture Notes in Economics and Mathematical Systems, Vol. 250, Springer-Verlag, Berlin, New York, 1985.

[21] B. Roy and Ph. Vincke, *Pseudo-orders: definition, properties and numerical representation*, Math. Soc. Sci., 14 (1987), pp. 263–274.

[22] J. B. Saxe, *Embeddability of graphs in k-space is strongly* NP-*hard*, in Proc. 17th Allerton Conference in Communication, Control and Computing, 1979, pp. 480–489.

[23] D. Scott and P. Suppes, *Foundational aspects of theories of measurement*, J. Symbolic Logic, 23 (1958), pp. 113–128.

[24] A. Tarski, *A Decision Method for Elementary Algebra and Geometry*, 2nd ed., University of California, Berkeley, CA, 1951.

[25] ———, *What is elementary geometry?*, in Symposium on the Axiomatic Method, L. Henkin, P. Suppes, and A. Tarski, eds., North-Holland, Amsterdam, 1959, pp. 16–29.

[26] H. Walther, *Ten Applications of Graph Theory*, D. Reidel, Dordrecht, the Netherlands, 1984.

SIAM J. DISC. MATH.
Vol. 4, No. 4, pp. 550-567, November 1991

OBNOXIOUS FACILITY LOCATION ON GRAPHS*

ARIE TAMIR†

Abstract. This paper discusses new complexity results for several models dealing with the location of obnoxious or undesirable facilities on graphs. The focus is mainly on the continuous p-Maximin and p-Maxisum dispersion models, where the facilities can be established at the nodes or in the interiors of the edges. For the general (nonhomogeneous) case it is shown that both models are strongly NP-hard even when the underlying graph consists of a single edge.

For the homogeneous p-Maximin model it is proven that even the problem of finding a $\frac{2}{3}$-approximation solution is NP-hard, and a polynomial heuristic which provides a $\frac{1}{2}$-approximation to the model is presented. Tree graphs are considered, and new algorithms with lower complexity bounds for several versions of the model are presented.

For the p-Maxisum problem we show that the homogeneous case is NP-hard on general graphs. Turning to the homogeneous case on trees, a certain concavity property is identified and then utilized to improve upon the best known methods to solve this model.

Key words. location theory, obnoxious facilities, network center problems

AMS(MOS) subject classifications. 05C35, 90C27, 68A20, 90B05

1. Introduction. Let $G = (V, E)$ be an undirected graph with node set $V = \{v_1, \cdots, v_n\}$ and edge set E. Let $|E| = m$. Each edge has a positive length and is assumed to be rectifiable. We refer to interior points on an edge by their distances (along the edge) from the two nodes of the edge. Let $A(G)$ denote the continuum set of points on the edges of G. The edge lengths induce a distance function on $A(G)$; for any x, y in $A(G)$, $d(x, y)$ will denote the length of a shortest path connecting x and y. Also, for any subset $Y \subseteq A(G)$, $d(x, Y) = \text{Infimum } \{d(x, y) | y \in Y\}$.

Let $X = \{x_1, \cdots, x_p\}$ be a finite set of points in $A(G)$. Define the following matrices $D(X, X)$ and $\bar{D}(V, X)$:

$$D(X,X) = \{d(x_i,x_j)\}, \qquad 1 \leqq i,j \leqq p$$

$$\bar{D}(V,X) = \{d(v_i,x_j)\}, \qquad 1 \leqq i \leqq n, \quad 1 \leqq j \leqq p.$$

Let $f(X) = f(\bar{D}(V, X), D(X, X))$ be a real function which is isotone in the components of the matrices $\bar{D}(V, X)$ and $D(X, X)$. (A real function g defined on R^k is isotone if for any w and z in R^k, $w \leqq z$ implies $g(w) \leqq g(z)$.) A variety of location models in the literature are defined by optimizing various forms of f over classes of subsets $X \subseteq A(G)$, $|X| = p$. For example, the unweighted 1-center of the graph G is obtained by setting $p = 1$ and minimizing the function

$$f(\bar{D}(V,X), D(X,X)) = \text{Maximum } \{d(v_i, x_1) | 1 \leqq i \leqq n\}$$

over all points x_1 in $A(G)$.

Using location theory terminology, the set X is referred to as the set of new facilities, e.g., suppliers, that must be set up, and the set V is identified as the set of existing facilities, e.g., customers. Conventional or ordinary location models are frequently defined by minimizing an isotone objective f, since the goal is to minimize some function of the distances between all facilities.

In this paper we consider the location of obnoxious or undesirable facilities, e.g., garbage depots and nuclear reactors. Thus, our interest is in studying maximization

* Received by the editors June 26, 1989; accepted for publication (in revised form) July 9, 1990.

† School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel.

models with isotone criteria. The reader is referred to the recent papers by Moon and Chaudhry [25] and Erkut and Neuman [11], which survey analytical models and approaches as well as various applications for the location of obnoxious facilities.

In this study we focus on maximizing the following two particular objective functions which seem to be the most popular among researchers interested in obnoxious facility location. The last section of this paper will be devoted to two other related optimization criteria.

Let α_{ij}, $1 \leqq i \leqq n$, $1 \leqq j \leqq p$, and β_{ij}, $1 \leqq i, j \leqq p$ be nonnegative weights.

The p-Maximin problem. This model, which has also been labeled as the p-Dispersion model, has the following objective function to be maximized:

$$(1.1)\qquad \begin{aligned} f_1(X) = \text{Minimum}\ \{&\text{Minimum}\ \{\alpha_{ij} d(v_i, x_j) \mid 1 \leqq i \leqq n, 1 \leqq j \leqq p\},\\ &\text{Minimum}\ \{\beta_{ij}(d(x_i, x_j) \mid 1 \leqq i \neq j \leqq p\}\}.\end{aligned}$$

The p-Maxisum problem. This maximization model is defined by the objective

$$(1.2)\qquad f_2(X) = \left\{ \sum_{i=1}^{n} \sum_{j=1}^{p} \alpha_{ij} d(v_i, x_j) + \sum_{i=1}^{p} \sum_{j=1}^{p} \beta_{ij} d(x_i, x_j) \right\}.$$

By the homogeneous versions of the above models we refer to the case where the new facilities $\{x_1, \cdots, x_p\}$ provide and receive identical services. Formally, in the homogeneous case $\beta_{ij} = 1$, $1 \leqq i, j \leqq p$, and $\alpha_{ij} = \alpha_i$, for $1 \leqq i \leqq n$ and $1 \leqq j \leqq p$.

2. The p-Maximin problem.

The p-Maximin problem is NP-hard when defined on a general graph, even for the homogeneous case in which $\beta_{ij} = 1$, $1 \leqq i \neq j \leqq p$, $\alpha_{ij} = \infty$, $1 \leqq i \leqq n$, $1 \leqq j \leqq p$; that is, we want to maximize the minimum distance between new facilities. Note that if each x_i, $1 \leqq i \leqq p$, is further restricted to a node, then the p-Maximin model generalizes the independent set problem [14]. Therefore, the discrete version is also NP-hard. For the continuous case we show that if there exists a polynomial procedure to find an ε-approximation solution with $\varepsilon > \frac{2}{3}$, then $P = \text{NP}$.

PROPOSITION 2.1. *Let ε be a real number satisfying $\varepsilon > \frac{2}{3}$. Let z^* denote the optimal solution value of the p-*Maximin *problem. The problem of finding a set $X = \{x_1, \cdots, x_p\}$ in $A(G)$ such that $f_1(X) \geqq \varepsilon z^*$ is* NP-*hard.* (*X is called an ε-approximation solution.*)

Proof. Consider the case where $\alpha_{ij} = \infty$, $1 \leqq i \leqq n$, $1 \leqq j \leqq p$, and $\beta_{ij} = 1$, $1 \leqq i \neq j \leqq p$. We reduce the independent set problem on a graph G [14] to the p-Maximin problem. Given an undirected graph $G = (V, E)$, $V = \{v_1, \cdots, v_n\}$, suppose that each edge is of unit length. Consider the graph $G^1 = (V^1, E^1)$, where $V^1 = V \cup \{u_1, \cdots, u_n\}$, $E^1 = E \cup \{(v_1, u_1), \cdots, (v_n, u_n)\}$. For $1 \leqq i \leqq n$ let the length of the edge (v_i, u_i) be $\frac{1}{2}$.

To complete the proof we show that G has an independent set of cardinality p if and only if every ε-approximation solution X for the p-Maximin problem on G^1 satisfies $f_1(X) > 2$. Suppose first that G has an independent set of cardinality p, i.e., there exist $\bar{V} \subseteq V$, $|\bar{V}| = p$, and for each pair of distinct nodes, $v_i, v_j \in \bar{V}$, $d(v_i, v_j) \geqq 2$. Define $V^* \subseteq V^1$ by $V^* = \{u_i \mid v_i \in \bar{V}\}$. Then, if u_i and u_j are two distinct nodes in V^* we have $d(u_i, u_j) \geqq 3$. In particular, the optimal solution to the p-Maximin problem on G^1 is at least 3. Therefore, if X is an ε-approximation with $\varepsilon > \frac{2}{3}, f_1(X) \geqq 3\varepsilon > 2$.

Next suppose that $X = \{x_1, \cdots, x_p\}$ is some ε-approximation solution with $f_1(X) > 2$. If x_i, $1 \leqq i \leqq p$, is on some edge (v_t, u_t), $1 \leqq t \leqq n$, replace x_i by u_t. If x_i is on some edge (v_t, v_q), where $d(x_i, v_t) \leqq d(x_i, v_q)$, replace x_i by u_t. Let X^1 be the solution obtained in this process. (Note that by the construction all the elements in X^1 are distinct.)

It is easily seen that $f_1(X^1) > 2$. Therefore, $f_1(X^1) \geqq 3$. Define $V \subseteq V$ by

$$\bar{V} = \{v_i \mid u_i \in X^1\}.$$

Then $\bar{V}$ is an independent set of G of cardinality p. □

The above proposition suggests that unless P = NP there is no polynomial ε-approximation algorithm for the homogeneous p-Maximin problem on a general graph with $\varepsilon > \frac{2}{3}$. However, there exists a simple polynomial greedy algorithm which generates a $\frac{1}{2}$-approximation solution. We will consider a slightly more general model for this homogeneous version of the p-Maximin problem.

Let D and S be two nonempty compact subsets of $A(G)$. The homogeneous p-Maximin and p-Minimax problems are defined as follows.

The homogeneous p-Maximin model on D. Find $X_p = \{x_1, \cdots, x_p\}$, a set of p points in D, such that $\text{Minimum}_{1 \leqq i \neq j \leqq p} \{d(x_i, x_j)\}$ is maximized.

The homogeneous p-Minimax model on D and S. Find $X_p = \{x_1, \cdots, x_p\}$, a set of p points in S, such that $\text{Maximum}_{x \in D} \{d(x, X_p)\}$ is minimized.

Let R_p and r_p denote the optimal solution values of the above two models, respectively.

We also introduce the following related problems. Let $r > 0$.

The r-cover problem on D and S. Find $p = p(r)$, the smallest integer value, and points $x_1, \cdots, x_p$ in S, such that $d(x, X_p) \leqq r$, for every x in D. ($X_p = \{x_1, \cdots, x_p\}$.)

The r-anticover problem on D. Find $q = q(r)$, the largest integer, and points $x_1, \cdots, x_q$ in D, such that $d(x_i, x_j) \geqq r$, $1 \leqq i \neq j \leqq q$.

The open r-anticover problem on D. Find $q = q^+(r)$, the largest integer and points $x_1, \cdots, x_q$ in D, such that $d(x_i, x_j) > r$, $1 \leqq i \neq j \leqq q$.

The above types of cover problems generalize and unify several models cited and discussed in the literature, e.g., [11], [20], [25].

LEMMA 2.2. *Let D and S be two compact sets of $A(G)$, and let $p \geqq 1$. Then the solution values to the homogeneous* p-Maximin *and* p-Minimax *models satisfy $R_{p+1} \leqq 2r_p$.*

Proof. Let $r > 0$. Let $X_q = \{x_1, \cdots, x_q\}$, $q = q^+(2r)$, be a solution to the open $2r$-anticover problem, and let $Y_p = \{y_1, \cdots, y_p\}$, $p = p(r)$, be a solution to the r-cover problem. Since $d(x_i, x_j) > 2r$, $1 \leqq i \neq j \leqq q^+(2r)$, we need at least $q^+(2r)$ points in S to ensure a covering of a distance not exceeding r to each point in X_q. Thus, $p(r) \geqq q^+(2r)$, for every $r > 0$.

Let $r = r_p$. Since $p(r_p) \leqq p$ we obtain $q^+(2r_p) \leqq p$. There exist $p + 1$ points in D such that the distance between each pair of distinct points is at least R_{p+1}. If R_{p+1} were strictly greater than $2r_p$, we would have $q^+(2r_p) \geqq p + 1$, contradicting $q^+(2r_p) \leqq p$. Hence, $R_{p+1} \leqq 2r_p$. □

We now introduce a $\frac{1}{2}$-approximation heuristic to the p-Maximin problem on D. This heuristic is motivated by the 2-approximation procedure for the p-Minimax model given in Dyer and Frieze [9]. The idea of the procedure is to construct a sequence of p points such that each point is as far apart from the preceding set of points as possible.

ALGORITHM 2.3

```
Step 0. Choose an arbitrary point $x_1$ in $D$. Let $X_1 = \{x_1\}$.
Step 1. While $j < p$ do
        Determine $x_{i+1}$ in $D$ by $d(x_{i+1}, X_i) = \text{Max}_{x \in D} \{d(x, X_i)\}$.
        Let $\bar{R}_{i+1} = d(x_{i+1}, X_i)$ and set $X_{i+1} = X_i \cup \{x_{i+1}\}$.
```

THEOREM 2.4. *Let $X_p = \{x_1, \cdots, x_p\}$ be the set of points generated by Algorithm 2.3. Then X_p is a $\frac{1}{2}$-approximation solution to the p-*Maximin *problem on D, i.e.*,

$$d(x_i, x_j) \geqq \tfrac{1}{2}R_p, \qquad 1 \leqq i \neq j \leqq p.$$

Proof. Let $X_p = \{x_1, \cdots, x_p\}$ be the set of points generated by the algorithm. Consider the subset $X_{p-1} = \{x_1, \cdots, x_{p-1}\}$, and view it as a feasible solution to the $(p-1)$-Minimax problem with $S = D$. By construction, for each x in D, $d(x, X_{p-1}) \leqq d(x_p, X_{p-1}) = \bar{R}_p$. Therefore, $\bar{R}_p \geqq r_{p-1}$.

From Lemma 2.2 we obtain

$$\bar{R}_p \geqq r_{p-1} \geqq \tfrac{1}{2}R_p.$$

Again, by construction, $d(x_i, x_j) \geqq \bar{R}_p$, $1 \leqq i \neq j \leqq p$, and the result follows. □

It should be noted that when G is a tree and $S = A(G)$, the homogeneous p-Maximin problem is equivalent (dual) to the homogeneous $(p-1)$-Minimax problem, [2], [20], [27], more commonly known as the $(p-1)$-center problem. The inequality in Lemma 2.2 holds as an equality. There exist several polynomial algorithms to solve the p-Minimax problem on a tree network for various cases of D. Focusing on the case studied in this paper, i.e., $D = A(G)$, and using the above duality result to solve the $(p+1)$-Maximin model, we can use any of the known algorithms that solve the p-Minimax model, [1], [2], [13], [22], [32]. The algorithms with the known lowest complexity bounds appear in [13], [22]. The algorithm in [22] has an $O(n \log^2 n)$ bound when we implement the improvement in [7], while that of [13] is $O(n \min(p, n) \log(\max(p/n, n/p)))$. Note that the latter bound dominates the former only when $p = O(\log n)$.

The only published algorithm that solves the p-Maximin problem on a tree graph directly appears in [1]. While the algorithms for the p-Minimax problem use a simple $O(n)$ procedure to solve the main subroutine, the r-cover problem, the algorithm in [1] for the p-Maximin relies on an $O(n \log n)$ scheme to solve the r-anticover problem.

Due to the importance of the r-anticover problem, we next present a very simple linear time algorithm to solve this problem on tree graphs. We will consider the following generalization of the r-anticover problem studied also by Moon and Goldman [26].

The generalized anticover problem. Let r and r_i, $1 \leqq i \leqq n$, be a collection of $n + 1$ positive numbers. Find $q = q(r)$, the largest number, and points $x_1, \cdots, x_q$ in $A(G)$, such that

$$d(v_i, x_j) \geqq r_i \quad \text{for } 1 \leqq i \leqq n \text{ and } 1 \leqq j \leqq q, \tag{2.1}$$

$$d(x_i, x_j) \geqq r \quad \text{for } 1 \leqq i \neq j \leqq q. \tag{2.2}$$

Moon and Goldman [26] have presented a complicated algorithm to solve the above problem on a tree. However, the exact complexity of their algorithm is not specified. In contrast, our algorithm is quite simple and has a linear complexity.

The first phase of our algorithm identifies the feasible set for the points $\{x_i\}$ induced by the constraints (2.1). It is easy to see that the intersection of this feasible set with any edge of the tree is a segment (subedge) of the edge. Therefore, we characterize in linear time the endpoints in $A(G)$ of all those subedges. We augment all these endpoints to the node set of the tree (note that at most $2(n-1)$ nodes are added), and update the edge set accordingly.

In the second phase we solve an r-anticover problem with the additional supposition that the points $\{x_i\}$ can only be located on a distinguished specified subset of edges.

Tree terminology and notation. Let $T = (V, E)$, $V = \{v_1, \cdots, v_n\}$, be an undirected tree with $n \geqq 3$. We define the following partial ordering on the nodes and edges of T. Suppose that T is rooted at some node which is not a tip (leaf), i.e., its degree is at least two. Without loss of generality let v_1 be the root. Let v_i and v_j be a pair of distinct nodes. v_i is a descendant of v_j if v_j is on the unique path connecting v_i to v_1. Furthermore, if v_i and v_j are also connected by an edge in E we say that v_i is a son of v_j, and v_j is the father of v_i. v_i is the son endpoint of this edge, and v_j is its father endpoint. Note that v_1 has no father and every tip node of T has no sons. For each node v_i, which is not the root, we use $e(i)$ to denote the edge of T connecting v_i to its father.

For each node v_i which is not a tip we let C_i denote its set of sons. If all nodes in C_i are tips then C_i is called a cluster.

If (v_i, v_j) and (v_j, v_k) are two distinct edges in E, and v_j is a father of v_i and a son of v_k, then we say that (v_i, v_j) is a son of (v_j, v_k), and (v_j, v_k) is a father of (v_i, v_j). An edge (v_i, v_j) is called a tip edge if its son endpoint v_i is a tip of the tree.

Phase I. *Identify the feasibility subedges*. Suppose that the tree $T = (V, E)$ is rooted at some node, say v_1. Given the constraints (2.1) our task is to identify for each edge in E, the subedge which is consistent with (2.1). Such a subedge, if it is not empty, will be identified by its two endpoints. Using the father-son relationship induced by the rooted tree, we label these two endpoints as the son and father endpoints, respectively.

The procedure to identify the subedges is based on scanning the tree twice. First, starting with the tips of the tree we recursively compute the son endpoints for all edges. The father endpoints are also computed recursively while starting at the root, scanning all its descendants according to the partial ordering and terminating at the tips.

```
Step 1. Let $v_i$ be a node in $V$.
        If $v_i$ is a tip of the rooted tree set $\delta_i = r_i$.
        If $v_i$ is not a tip set
        $\delta_i = \text{Maximum}\{r_i, \text{Maximum}\{\delta_j - d(v_i, v_j) \mid v_j \in C_i\}\}$.
Step 2. Let $v_i$ be a node in $V$.
        If $v_i$ is the root of the tree set $\varepsilon_i = \delta_i$.
        If $v_i$ is not a root let $v_j$ be the father of $v_i$.
        Set $\varepsilon_i = \text{Maximum}\{\delta_i, \varepsilon_j - d(v_i, v_j)\}$.
```

We are now ready to compute the endpoints of all the feasibility subedges. Let (v_i, v_j) be an edge of the tree, where v_j is the father of v_i. If $\delta_i + \varepsilon_j > d(v_i, v_j)$ the feasibility subedge is empty. Otherwise, the son endpoint of the subedge is the point x_i on (v_i, v_j) satisfying $d(v_i, x_i) = \delta_i$, and its father endpoint is the point y_j on (v_i, v_j) satisfying $d(y_j, v_j) = \varepsilon_j$. If $x_i = y_j$ the feasibility subedge is reduced to a point.

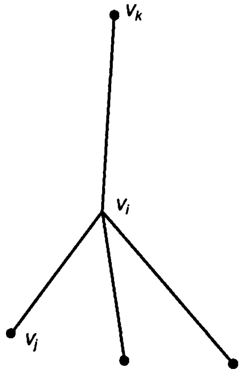
It is clear that the above procedure yields the endpoints of all subedges in $O(n)$ total time. We augment the endpoints to the node set of the tree and update the edge set accordingly. In particular, if an original edge has a nonempty subedge it will be replaced by either two or three new edges in the augmented tree. We then solve a modified r-anticover problem on the augmented tree.

Phase II. *The modified r-anticover problem*. This phase corresponds to constraints (2.2). Let $T = (V, E)$ be a rooted tree. Let V^1 be a subset of V and let E^1 be a subset of E. Given a positive real r, find $q = q(r)$, the largest integer, and points $x_1, \cdots, x_q$, where x_i, $1 \leqq i \leqq q$, is in V^1 or is on some edge in E^1, such that $d(x_i, x_j) \geqq r$, for $1 \leqq i \neq j \leqq q$.

ALGORITHM 2.5. The algorithm starts with the rooted tree and recursively processes and eliminates its clusters. (Without loss of generality assume that no node in V^1 is on an

edge in E^1.) The main step, the elimination of a cluster, is an improved and modified version of the $O(n \log n)$ procedure presented by Chandrasekaran and Daughety [1] for the standard case, i.e., $E^1 = E$. Our algorithm will run in $O(n)$ time.

Consider a cluster with a father v_i, and let C_i be the set of its sons. (Note that each son of v_i is a tip of the current updated tree.) Let v_k be the father of v_i.



The cluster elimination procedure.

Step 1. For each v_j in C_i apply the following: Suppose first that v_j is in V^1. If $d(v_j, v_i) \geqq r$, add v_j to the set of selected points and remove the edge (v_j, v_i) from the current tree. If $d(v_j, v_i) < r$ set $\lambda_j = d(v_j, v_i)$. Next, suppose that v_j is not in V^1. If the edge (v_j, v_i) is not in E^1, remove it from the current tree. Otherwise, compute

$$t_j = \left[\frac{d(v_j, v_i)}{r} \right].$$

(As usual, $\lfloor a \rfloor$ is the largest integer that is smaller or equal to a.) If $t_j \geqq 1$, select t_j points on the edge (v_i, v_j) as follows: The first point is v_j itself and the others are selected such that the distance between consecutive ones is r. Update the tree by reducing the length of (v_j, v_i) by rt_j. If the modified length λ_j is zero, remove the edge (v_i, v_j) from the current tree.

Step 2. Define $J_1 = \{j \mid \lambda_j < r/2\}$ and $J_2 = \{j \mid r > \lambda_j \geqq r/2\}$. (If both J_1 and J_2 are empty the cluster is eliminated. Stop.) If J_1 is nonempty let $j(1)$ in J_1 be an index satisfying $\lambda_{j(1)} = \text{Maximum } \{\lambda_j \mid j \in J_1\}$, and for each $j \in J_1, j \neq j(1)$, remove the edge (v_j, v_i) from the current tree. Set $J_1 = \{j(1)\}$. If J_2 is nonempty let $j(2)$ in J_2 be an index satisfying $\lambda_{j(2)} = \text{Minimum } \{\lambda_j \mid j \in J_2\}$.

Step 3. If J_1 is empty, then for each $j \in J_2, j \neq j(2)$, add the tip of the modified edge (v_j, v_i), obtained after the above length reduction, to the set of points already selected and remove that edge from the current tree. Go to Step 5 with $u = v_{j(2)}$ as the only son of v_i, and $d(u, v_i) = \lambda_{j(2)}$. If J_1 is nonempty and J_2 is empty go to Step 5 with $u = v_{j(1)}$ as the only son of v_i, and $d(u, v_i) = \lambda_{j(1)}$. Otherwise, go to Step 4.

Step 4. (J_1 and J_2 are nonempty.) If $\lambda_{j(1)} + \lambda_{j(2)} < r$, remove the edge $(v_{j(1)}, v_i)$ from the current tree, set $J_1 = \phi$ and go to Step 3. If $\lambda_{j(1)} + \lambda_{j(2)} \geqq r$, then for each $j \in J_2$, add the tip of the (modified) edge (v_j, v_i) to the set of points already selected and remove that edge from the current tree. Go to Step 5 with $u = v_{j(1)}$ as the only son of v_i, and $d(u, v_i) = \lambda_{j(1)}$.

Step 5. (u is the only son of v_i, and $d(u, v_i)$ is the length of the edge (u, v_i).) Consider the edge (v_i, v_k), connecting v_i to its father. If (v_i, v_k) is in E^1 or if $d(u, v_i) + d(v_i, v_k) < r$, replace the pair of edges (v_i, v_k), (u, v_i) by a single edge (u, v_i) having the length $d(u, v_i) + d(v_i, v_k)$, and augment this new edge to E^1 if it is not already there. Stop.

If (v_i, v_k) is not in E^1 and $d(u, v_i) + d(u, v_k) \geqq r$, add the point u (the tip of the edge (u, v_i)) to the set of points already selected, and remove the edges (u, v_i) and (v_i, v_k) from the current tree. Stop.

The complexity of the cluster elimination procedure is linear in its size. Therefore, when we apply this procedure recursively to the original tree until we reach its root, it terminates in $O(n)$ time. $q(r)$ is given by the cardinality of the set of points selected in this process.

The validity of the above algorithm follows from the arguments given in Chandrasekaran and Daughety [1] who provided an $O(n \log n)$ algorithm to solve the standard r-anticover problem, i.e., $E^1 = E$.

We note that the above $O(n)$ procedure for solving the generalized r-anticover problem can be implemented, as in [2] and [22], to yield a polynomial algorithm to maximize the following homogeneous p-Maximin problem on a tree $T = (V, E)$.

Find a set of points $X_p = \{x_1, \cdots, x_p\}$ in $A(T)$ maximizing the objective

$$(2.3) \quad f_1(x_1, \cdots, x_p) = \text{Minimum}\left\{\underset{1 \leqq i \leqq n}{\text{Minimum}}\ \{\alpha_i d(v_i, X_p)\}, \underset{1 \leqq i \neq j \leqq p}{\text{Minimum}}\ \{d(x_i, x_j)\}\right\}.$$

We demonstrate the approach with the case $p = 1$. We show how to use Phase I of the above algorithm to improve the complexity of the best known algorithm to solve the 1-Maximin problem on a tree. Drezner and Wesolowsky [8] have solved this model on a path graph in $O(n^3)$ time. Tamir [30] has presented an $O(n \log n)$ algorithm for paths, and an $O(H(T) \log^2 n)$ algorithm for a general tree graph T, where $H(T)$ is a parameter depending on the topology of the tree. ($H(T)$ is always bounded between n and n^2.)

We now provide an $O(n \log^2 n)$ algorithm, using the above. Let x^* be an optimal solution to the 1-Maximin problem. Then there exists a pair of distinct nodes v_i and v_j of the tree such that x^* is on the unique path connecting v_i and v_j and $\alpha_i d(v_i, x^*) = \alpha_j d(v_j, x^*)$. (Since $p = 1$, we write x^* for x_1^*, and α_i for α_{i1}, $1 \leqq i \leqq n$.) Thus, we have the following proposition.

PROPOSITION 2.6. *Let z^* be the optimal solution value to the* 1-Maximin *problem on a tree. Then z^* is an element in the set R,*

$$(2.4) \qquad R = \left\{\frac{d(v_i, v_j)}{\alpha_i^{-1} + \alpha_j^{-1}} \,\middle|\, 1 \leqq i \neq j \leqq n\right\}.$$

z^* is fully characterized by the following property. Let z be a positive real and consider the problem of finding whether there exists an x in $A(T)$ such that

$$(2.5) \qquad d(v_i, x) \geqq z/\alpha_i, \qquad 1 \leqq i \leqq n.$$

Then, z^* is the largest element in the set R, defined by (2.4), *such that the system* (2.5) *is feasible.*

Given a positive real z, the feasibility of (2.5) can be solved by Phase I above. Setting $r_i = z/\alpha_i$, $1 \leqq i \leqq n$, we note that (2.5) is feasible if and only if Phase I identifies at least one nonempty feasible subedge. Thus, the feasibility of (2.5) can be tested in $O(n)$ time.

With this linear time test we can implement the sophisticated search procedures of [7] and [22] and locate z^* in R in $O(n \log^2 n)$ total time.

We have presented above a polynomial time algorithm to solve the homogeneous p-Maximin problem on a tree. In contrast, the next result shows that the general (nonhomogeneous) case is NP-hard even on a single edge.

PROPOSITION 2.7. *The p-*Maximin *problem is strongly* NP-*hard even when the underlying tree consists of a single edge.*

Proof. Let $G = (V, E)$, $V = \{v_1, \cdots, v_n\}$, be an undirected graph with unit edge lengths. We reduce the Hamiltonian path problem on G [14] to the n-Maximin model on a single edge tree. Let $d(v_i, v_j)$, $1 \leqq i \neq j \leqq n$ be the distance between v_i and v_j on G. Consider the problem of locating a set of n points $\{x_1, \cdots, x_n\}$ on the (real) interval $[0, n-1]$, such that $|x_i - x_j| \geqq d(v_i, v_j)$, $1 \leqq i \neq j \leqq n$.

We claim that G has a Hamiltonian path if and only if the above location problem is feasible. Suppose first that $(v_{i(1)}, \cdots, v_{i(n)})$ indicates the node permutation on a Hamiltonian path. Define the n points on the interval by setting $x_{i(k)} = k - 1$, $k = 1, \cdots, n$. Using the triangle inequality for the distance function on G, we obtain $|x_i - x_j| \geqq d(v_i, v_j)$, for all $1 \leqq i \neq j \leqq n$. Conversely, suppose that $x_{i(1)} < x_{i(2)} \cdots < x_{i(n)}$ indicates the locations of the n points on $[0, n-1]$. By the constraints $x_{i(k)} - x_{i(k-1)} \geqq d(v_{i(k)}, v_{i(k-1)}) \geqq 1$, $k = 2, \cdots, n$, and $x_{i(n)} - x_{i(1)} \leqq n - 1$, we must have $x_{i(k)} - x_{i(k-1)} = 1$, $k = 2, \cdots, n$. Therefore, $(v_{i(1)}, \cdots, v_{i(n)})$ indicates a node ordering of some Hamiltonian path on G. □

Remark 2.8. We have not dealt here with exact algorithms to solve the p-Maximin problem on general graphs. We briefly note several related references and results.

Consider first the homogeneous case of (1.1), where $\alpha_{ij} = \infty$, $1 \leqq i \leqq n$, $1 \leqq j \leqq p$, $\beta_{ij} = 1$, $1 \leqq i, j \leqq p$. We can use the approaches in Tamir [28] and [29] for the related homogeneous p-Minimax problem, and derive similar results for this case of the p-Maximin model. In particular, we have obtained a result, similar to [29, Thm. 5], identifying a finite set containing the optimal objective value. Exact algorithms to solve the discrete version of this case, where the points selected must be nodes, appear in Erkut [10] and Kuby [21].

Referring next to the general form of (1.1) we note that the model can be solved in polynomial time when p is fixed. We have derived such an algorithm by adopting the approach used in [28] for the p-Minimax problem. We skip the details since the algorithm is practically inefficient due to its high complexity bound, which is exponential in p. For small values of p the algorithm can be significantly accelerated by using recent developments in linear programming. For example, it is shown in Tamir [30] that for $p = 1$, the optimal solution can be obtained in $O(mn)$ time.

Remark 2.9. We have mentioned above that Algorithm 2.3 is based on the heuristic of Dyer and Frieze [9] which generates a 2-approximation solution to the continuous symmetric p-Minimax model. We note in passing that the proof of Proposition 2.1 can be used to show that if there exists a polynomial heuristic to find an ε-approximation solution, with $\varepsilon < \frac{4}{3}$, to the p-Minimax problem, then $P = \text{NP}$. We conjecture that this result actually holds for any $\varepsilon < 2$. Similarly, we conjecture that the result in Proposition 2.1 holds for any $\varepsilon > \frac{1}{2}$.

3. The p-Maxisum problem.

We start by showing that the general model is NP-hard even for the trivial case where the graph consists of a single edge. We need the following lemma.

LEMMA 3.1. *Let $T = (V, E)$ be a tree graph. Then there is an optimal solution X^* to the p-*Maxisum *problem, where each $x \in X^*$ is a tip of T.*

Proof. Let X be an optimal solution to the p-Maxisum model. Let $u(X)$ denote the number of points in X that are not tips of T. Among all optimal solutions to the model let X^* have the additional property that $u(X)$ is minimized.

Suppose $\bar{x} \in X^*$ is not a tip of X^*. Fix all points in X^* but $\bar{x}$, and view the objective f_2 as a single variable $(\bar{x})$ function. For every fixed point y on the tree, the function $d(y, \bar{x})$ is convex on every path in T. Thus, $f_2(\bar{x})$ is convex on every path in T. Therefore, its maximum is attained at a tip of T. This contradicts the minimality property of X^*. □

PROPOSITION 3.2. *The* p-Maxisum *problem is* NP-*hard even when G is a graph consisting of a single edge.*

Proof. Consider the model with $\alpha_{ij} = 0$, $1 \leqq i \leqq n$, $1 \leqq j \leqq p$. From Lemma 3.1 there is an optimal solution X^*, where each $x_i^* \in X^*$ coincides with v_1 or v_2, the two nodes of G. Therefore, the p-Maxisum model reduces to the following Maximum cut problem, which is known to be NP-hard [14]:

$$\text{Find a subset } S \subseteq \{1, 2, \cdots, p\}, \text{ such that } \sum_{i \in S} \sum_{j \notin S} \beta_{ij} \text{ is maximized.} \qquad \square$$

For comparison purposes, it is interesting to note that the minimization of (1.2) on a tree graph can be performed in polynomial time by solving a sequence of $O(n)$ minimum cut problems on a graph with $O(p)$ nodes [19].

We will later show that the homogeneous model can be solved in $O(np)$ time on tree networks. However, on general graphs even the homogeneous model is NP-hard. Hansen and Moon [16] have considered the discrete version of the homogeneous case (with the additional supposition that $\alpha_{ij} = 0$, $1 \leqq i \leqq n$, $1 \leqq j \leqq p$), where the points $x_1, \cdots, x_p$ must be selected among the nodes of G. They have shown that the independent set problem [14] on a general graph is reducible to their model. Combining their reduction with the construction in the proof of Proposition 2.1, we obtain the following result for the (continuous) p-Maxisum problem.

PROPOSITION 3.3. *The* p-Maxisum *problem* (1.2) *is* NP-*hard on a general graph even when* $\alpha_{ij} = 0$, $1 \leqq i \leqq n$, $1 \leqq j \leqq p$, $\beta_{ij} = 1$, $1 \leqq i, j \leqq p$.

Proof. Let $G = (V, E)$, $V = \{v_1, \cdots, v_n\}$, be a graph with unit edge lengths. Let $G' = (V, E')$, be a complete graph with V as its node set. If e is in E let its length be one, otherwise set it equal to two. Extend G' to $G'' = (V'', E'')$ as follows: $V'' = V \cup \{u_1, \cdots, u_n\}$, $E'' = E' \cup \{(v_1, u_1), \cdots, (v_n, u_n)\}$. For $1 \leqq i \leqq n$, let the length of the edge (v_i, u_i) be one. Consider the homogeneous p-Maxisum problem on G'', with all the α-coefficients being equal to zero. It is easy to verify that the graph G has an independent set of cardinality p if and only if the solution value to the above p-Maxisum problem on G'' is equal to $4p(p-1)$. □

When p is fixed (1.2) can be solved in polynomial time on general graphs. Consider first the single facility case, i.e., $p = 1$. Church and Garfinkel [6] have studied this model and provided an $O(n^3)$ algorithm to find the optimal location of the new center x_1. We can improve this bound by using the following observation. Suppose that we restrict the new center to be located on a given edge. Then for each node v_i, $\alpha_{i1} d(v_i, x_1)$ is a concave piecewise linear function on this edge, and it has at most one breakpoint there. Thus, the objective $f_2(x_1)$ is piecewise linear and concave. The maximum point of f_2 on a given edge can be obtained in $O(n)$ time using the recent general algorithms developed by Zemel [35]. Therefore, we conclude that the 1-Maxisum problem on a general graph can be solved in $O(mn)$ time, provided that the distances between all the nodes are given.

The case $p = 2$ can also be solved by using a similar approach. In this case the objective takes on the following form:

$$f_2(x_1, x_2) = \sum_{i=1}^{n} \sum_{j=1}^{2} \alpha_{ij} d(v_i, x_j) + (\beta_{12} + \beta_{21}) d(x_1, x_2). \tag{3.1}$$

When $\alpha_{i1} = \alpha_{i2} = 0$ for $1 \leqq i \leqq n$, the problem reduces to the problem of finding the generalized diameter of a graph. The latter has been solved by Chen and Garfinkel [5]. They have identified a discrete finite set of points that will include at least one optimal solution. Specifically, there is an optimal solution where x_j, $j = 1, 2$, is either a node, or else there exists a node v_i and a simple cycle containing v_i and x_j whose length is $2d(v_i, x_j)$. To maximize (3.1) we solve $O(m^2)$ restricted subproblems. A subproblem is obtained by restricting x_1 and x_2 to be located either on the same edge or on a given pair of edges. In either case we can easily verify that $f_2(x_1, x_2)$ is piecewise linear and concave over the restricted domain. Furthermore, due to the nature of $f_2(x_1, x_2)$, the algorithms in Zemel [35] are applicable in this case and the optimal solution to a subproblem can be obtained in $O(n)$ time. Therefore the global maximizer of (3.1) can be computed in $O(m^2n)$ time.

The above approach can be generalized to yield an $O(m^pn)$ algorithm to solve the p-Maxisum problem (1.2) for every fixed p.

In light of Propositions 3.2 and 3.3 we focus now on the homogeneous case when the graph is a tree. The objective takes on the following form:

$$f_2(X) = \sum_{i=1}^{n} \sum_{j=1}^{p} \alpha_i d(v_i, x_j) + \sum_{i=1}^{p} \sum_{j=1}^{p} d(x_i, x_j). \tag{3.2}$$

Ting [34] has presented an $O(np^2)$ algorithm for this model. Hansen and Moon [16] have studied the case where $\alpha_i = 0$, $1 \leqq i \leqq n$, and restricted the p new facilities to the nodes, allowing no two to be located at the same node. Their algorithm also has the $O(np^2)$ complexity bound.

We improve the above complexity bound by reformulating the objective (3.2). The new formulation will identify useful concavity properties.

Using Lemma 3.1, we may assume that each of the p points $x_1, \cdots, x_p$ is a node of the given tree. For $1 \leqq i \leqq n$, let y_i denote the number of new points (facilities) established at node v_i. The optimization model is now formulated as:

$$\text{Maximize}\, f_2(y_1, \cdots, y_n)$$

$$\text{subject to}$$

$$f_2(y_1, \cdots, y_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i d(v_i, v_j) y_j + \sum_{i=1}^{n} \sum_{j=1}^{n} d(v_i, v_j) y_i y_j \tag{3.3}$$

$$\sum_{j=1}^{n} y_j = p, y_j \text{ nonnegative and integer, } j = 1, \cdots, n.$$

LEMMA 3.4. *Let $T = (V, E)$ be a tree with nonnegative edge lengths, $\{w_e\}$, $e \in E$. Then the quadratic $f_2(y_1, \cdots, y_n)$ defined by* (3.3) *is concave when restricted to the hyperplane*

$$\left\{ y \mid y \in R^n, \sum_{j=1}^{n} y_j = p \right\}.$$

Proof. It is sufficient to establish the concavity for the quadratic portion of f_2.

Suppose that the tree is rooted at node v_1 and v_1 is not a tip. Consider an edge $e \in E$, and let V^e be the subset of V consisting of all nodes that are disconnected from v_1 by the removal of e. It is easy to verify that w_e, the length of e, will appear exactly

$2(\sum_{\{i|v_i\in V^e\}}\sum_{\{j|v_j\in V-V^e\}} y_iy_j)$ times in the quadratic expression. Therefore,

$$\sum_{i=1}^{n}\sum_{j=1}^{n} d(v_i,v_j)y_iy_j=\sum_{e\in E}2w_e\left(\sum_{\{i|v_i\in V^e\}}\ \sum_{\{j|v_j\in V-V^e\}} y_iy_j\right)$$

$$=\sum_{e\in E}2w_e\left(\sum_{\{i|v_i\in V^e\}} y_i\right)\left(\sum_{\{j|v_j\in V-V^e\}} y_j\right).$$

Using the constraint

$$\sum_{j=1}^{n} y_j=p,$$

we obtain

$$\sum_{i=1}^{n}\sum_{j=1}^{n} d(v_i,v_j)y_iy_j=\sum_{e\in E}2w_e\left(p\sum_{\{i|v_i\in V^e\}} y_i-\left(\sum_{\{i|v_i\in V^e\}} y_i\right)^2\right),$$

which is a concave function. □

The concavity result of Lemma 3.3 can be utilized to yield an $O(np)$ algorithm to solve (3.3). The objective in (3.3) is now formulated as:

$$f_2(y_1,\cdots,y_n)=\sum_{e\in E} w_e\left[\left(\sum_{\{j|v_j\in V-V^e\}}\alpha_j\right)\sum_{\{i|v_i\in V^e\}} y_i+\left(\sum_{\{j|v_j\in V^e\}}\alpha_j\right)\left(p-\sum_{\{i|v_i\in V^e\}} y_i\right)\right]$$

$$+\sum_{e\in E}2pw_e\sum_{\{i|v_i\in V^e\}} y_i-\sum_{e\in E}2w_e\left(\sum_{\{i|v_i\in V^e\}} y_i\right)^2.$$

Letting

$$\bar{w}_e=w_e\left(\sum_{\{j|v_j\in V-V^e\}}\alpha_j-\sum_{\{j|v_j\in V^e\}}\alpha_j\right)+2pw_e,\qquad e\in E,$$

$$f_2(y_1,\cdots,y_n)=\sum_{e\in E}\bar{w}_e\left(\sum_{\{i|v_i\in V^e\}} y_i\right)-\sum_{e\in E}2w_e\left(\sum_{\{i|v_i\in V^e\}} y_i\right)^2 \tag{3.4}$$

$$+p\sum_{e\in E} w_e\left(\sum_{\{i|v_i\in V^e\}}\alpha_i\right).$$

Note that the coefficients $\{\bar{w}_e\}$, $e\in E$, can easily be computed in $O(n)$ total time.

To simplify the presentation we augment a node v_0 to the given tree $T=(V,E)$ and connect v_0 to v_1, the current root of T, by an edge which is consistently labeled $e(1)$. We view v_0 as the super root of the augmented tree, $T^1=(V\cup\{v_0\},E\cup\{e(1)\})$.

For each e in E, let $T_e=(V^e,E^e)$ be the subtree induced by V^e. In particular, we note that $T_{e(1)}=T=(V,E)=(V^{e(1)},E^{e(1)})$. We maximize (3.4) recursively by considering its restriction to the subtrees $\{T_e\}$. We will start at the tips of the tree and recursively proceed until we reach the super root of the tree. (To simplify the notation we delete and ignore the constant term in (3.4) while maximizing $f_2(y_1,\cdots,y_n)$.)

For each edge z of the augmented tree T^1 that is not a tip edge define

$$f_2(z{:}y)=\sum_{e\in E^z}\bar{w}_e\left(\sum_{\{i|v_i\in V^e\}} y_i\right)-\sum_{e\in E^z}2w_e\left(\sum_{\{i|v_i\in V^e\}} y_i\right)^2, \tag{3.5}$$

and

(3.6)

$$F_2(z{:}k) = \text{Maximum}\left\{f_2(z{:}y) \mid \sum_{\{i \mid v_i \in V^z\}} y_i = k, y_i \text{ nonnegative and integer } 1 \leqq i \leqq n\right\}.$$

The solution value to our model is given by $F_2(e(1){:}p)$.

PROPOSITION 3.5. *For every edge z, $F_2(z{:}k)$ is concave in k, i.e., the difference function $F_2(z{:}k+1) - F_2(z{:}k)$ is monotone nonincreasing.*

Proof. The result follows directly from the concavity of the function $f_2(z{:}y)$. □

ALGORITHM 3.6. To compute the optimal solution value to our model, $F_2(e(1){:}p)$, we give the recursive equations for computing $F_2(z{:}k)$ for all edges z and integer $1 \leqq k \leqq p$.

We use the tree terminology presented above. We start the recursion by defining F_2 for edges z having the property that their son endpoint, say v_i, is such that C_i is a cluster. Consider such an edge $z = e(i)$, and let v_i be its son endpoint. (Recall that for each son v_j of v_i $e(j)$ denotes the edge connecting v_j to v_i.) Then

$$(3.7)\quad F_2(z{:}k) = \text{Maximum}\left\{\sum_{\{j \mid v_j \in C_i\}} \bar{w}_{e(j)} y_j - \sum_{\{j \mid v_j \in C_i\}} 2w_{e(j)} y_j^2 \mid \sum_{\{j \mid v_j \in C_i\}} y_j = k,\right.$$
$$\left. y_j \text{ nonnegative integer, } 1 \leqq j \leqq n\right\}.$$

In general, when v_i is a node and C_i is not a cluster, we have the following:

Suppose that v_i is the son endpoint of an edge $e(i)$. Then

$$(3.8)\quad F_2(e(i){:}k) = \text{Maximum}\left\{\sum_{\{j \mid v_j \in C_i\}} (F_2(e(j){:}k_j) + \bar{w}_{e(j)} k_j - 2w_{e(j)} k_j^2) \mid \sum_{\{j \mid v_j \in C_i\}} k_j = k,\right.$$
$$\left. k_j \text{ nonnegative integer } 1 \leqq j \leqq n\right\}.$$

Using Proposition 3.5 we note that the maximization defining $F_2(e(i){:}k)$ is a special case of the standard discrete resource allocation model with a separable concave objective function. Using known algorithms for the latter model (see, e.g., [18, Chap. 4]), we conclude that the complexity of computing $F_2(e(i){:}k)$ for all values of $k = 1, \cdots, p$ combined is $O(|C_i|p)$. Therefore, the total effort for computing the optimal solution to the homogeneous p-Maxisum problem on a tree is

$$O\left(p \sum_{i=1}^{n} |C_i|\right) = O(np).$$

The above $O(np)$ algorithm was motivated by the case where p is relatively small. For example, the complexity is linear when p is fixed. We now show that the model is polynomially solvable even when p is a variable integer given as part of the input.

Consider the representation of the objective function in (3.4). For each edge $e(j)$ of the rooted tree define

$$z_j = \sum_{\{i \mid v_i \in V^{e(j)}\}} y_i.$$

The homogeneous p-Maxisum problem is now formulated as an integer quadratic program with a separable objective.

$$\text{Maximize } \sum_{j=1}^{n} \bar{w}_{e(j)} z_j - \sum_{j=1}^{n} 2 w_{e(j)} z_j^2$$

subject to

$$z_j \geqq \sum_{\{i \mid v_i \in C_j\}} z_i, \ 1 \leqq j \leqq n, \text{ and } v_j \text{ is not a tip node,} \tag{3.9}$$

$$z_1 = p,$$

$$z_j \text{ is a nonnegative integer, } 1 \leqq j \leqq n.$$

We note in passing that Lemma 3.1 implies that there is an optimal solution $\{y_1^*, \cdots, y_n^*\}$ to (3.4) where $y_i^* = 0$ if node v_i is not a tip. Therefore, there is an optimal solution $\{z_1^*, \cdots, z_n^*\}$ to (3.9) with

$$z_j^* = \sum_{\{i \mid v_i \in C_j\}} z_i^*, \ 1 \leqq j \leqq n, \text{ and } v_j \text{ is not a tip node.}$$

It is now easy to observe that the constraints in (3.9) represent a flow problem on the tree where p units of a single commodity are to be transferred from the tips of the tree to its root v_1. Therefore, (3.9) is a special case of the flow model discussed in Minoux [24]. Applying his approach to our tree flow problem yields an $O(n^3 \log p)$ algorithm. By implementing more sophisticated data structures we were able to reduce the bound to $O(n^2 \log n \log p)$. For the sake of brevity we skip the details and present, instead, a polynomial algorithm whose bound is independent of p.

Consider first the fractional relaxation obtained from (3.9) by deleting the integrality constraints on the variables. Let $\bar{z}$ be an optimal solution to the fractional relaxation and let τ be the complexity bound to compute $\bar{z}$.

We apply the proximity results in the recent paper by Granot and Skorin-Kapov [15]. Specifically, given $\bar{z}$, the problem of finding the integer solution to (3.9) is now reduced to a flow problem of the same type, where p, the number of units that must flow into the root node, is replaced by some polynomial in n. Thus, the linear constraints, defined by a totally unimodular flow matrix, are independent of p. (Note that p appears only in the linear portion of the objective in (3.4) and (3.9). Therefore, p will appear only in the linear portion of the objective of the reduced problem.) The solution to the reduced integer quadratic program can be obtained by the algorithm in Minoux [24], mentioned above. The running time will depend (polynomially) on n only, i.e., $O(n^2 \log^2 n)$.

To conclude we now have an $O(\tau + n^2 \log^2 n)$ algorithm to solve (3.9), where τ is the complexity bound to compute a solution to the fractional relaxation of (3.9). To solve the latter we apply the algorithm in Chandrasekaran and Kabadi [4]. Since p does not appear in the quadratic portion of the objective, τ is independent of p. It depends (polynomially) on n and the sizes of the edge lengths.

The algorithm in [4] is a general quadratic programming algorithm. We have developed a special purpose, strongly polynomial algorithm to solve the fractional relaxation of (3.9) with $\tau = O(n^2)$. This algorithm is presented in [31].

Remark 3.7. The above algorithm for the (continuous) p-Maxisum problem is based on Lemma 3.1, which limits the search for the optimal solution to the set of tip nodes. This algorithm can easily be modified to the case where the solution is originally confined to any other discrete set of points on the tree. We can even allow upper bounds on the total number of points in $\{x_1, \cdots, x_p\}$ that can be established at each one of the points in the given discrete set. The version of the problem discussed by Hansen and Moon [16] is of that nature. They have confined the p new facilities to the node set, but allow no pair of these to be located at the same node. Therefore, their model can also be solved in $O(np)$ time.

Remark 3.8. The $O(np)$ bound applies to a general tree graph. Improvements are possible for several special cases. For example, suppose that $T = (V, E)$ is a star tree, i.e., T is a cluster where v_1, its root, is the only node which is not a tip. In this case the solution to the problem is given by $F_2(e(1){:}p)$ where $F_2(e(1){:}p)$ is defined by (3.7) for $z = e(1)$ and $k = p$. Therefore, the p-Maxisum problem is reduced to the standard discrete effort allocation problem with a separable concave quadratic objective. The latter problem can be solved in $O(n)$ time [18].

Remark 3.9. For exact algorithms to solve the discrete version of the homogeneous p-Maxisum problem with $\alpha_{ij} = 0$, $1 \leqq i \leqq n$, $1 \leqq j \leqq p$, on a general graph the reader is referred to Erkut, Baptie, and Von Holenbalken [12], Hansen and Moon [16], and Kuby [21]. There are also several related papers cited in [16].

4. Related optimization models. We have studied above the two most common objectives used for locating obnoxious facilities, the p-Maximin and the p-Maxisum criteria. There are several other models mentioned in the literature (see the surveys in [11] and [25]). In this section we discuss briefly two models that we find to be more challenging combinatorially. We report on some of our results and pose a few open problems.

The first model that we consider has been introduced and motivated by Moon and Chaudhry [25]. They have labeled it as the p-Defense problem.

Find points $\{x_1, \cdots, x_p\}$ in $A(G)$ that will maximize the following objective:

$$(4.1) \qquad f_3(x_1, \cdots, x_p) = \sum_{i=1}^{p} \text{Minimum } \{d(x_i, x_j) \mid 1 \leqq j \leqq p, j \neq i\}.$$

The second model has been recently suggested by Ting in his Ph.D. dissertation [34].

Find points $\{x_1, \cdots, x_p\}$ in $A(G)$ that will maximize the following objective:

$$(4.2) \quad f_4(x_1, \cdots, x_p) = \sum_{i=1}^{n} \alpha_i \text{Minimum } \{d(v_i, x_j) \mid 1 \leqq j \leqq p\} + \sum_{i=1}^{p} \sum_{j=1}^{p} d(x_i, x_j).$$

We are not aware of any analytic or algorithmic results for these two models. However, few results and approaches discussed above in §§ 2 and 3 can be modified and applied for models (4.1) and (4.2). For example, both models are NP-hard when defined on general graphs. (The same result holds even when we consider the discrete versions and confine the points $\{x_1, \cdots, x_p\}$ to the node set of the underlying graph.)

Turning to tree graphs, the recursive solution approach of § 3 seems to be applicable for (4.1) and (4.2) as well. However, this recursive approach is discrete in nature, and therefore it requires the optimal points to belong to some prespecified discrete set of points. Moreover, this set must be of polynomial cardinality if we wish the solution procedure to be of polynomial complexity. If such a set is identified we reduce the model to its discrete version by augmenting the points in this set to the node set of the tree. Indeed, we have used the recursive approach and constructed polynomial algorithms of complexity $O(p^2n^3)$ and $O(p^2n^2)$ for the discrete versions of (4.1) and (4.2), respectively.

Therefore, to obtain polynomial procedures for solving the continuous version we need to identify a discrete set that includes at least one optimal solution. Such a set is called a finite dominating set (FDS).

There are several continuous network location models for which an FDS of polynomial cardinality has been found. More recently, Hooker, Garfinkel, and Chen [17], have unified most of these models by identifying common convexity-concavity properties. However, we could not see how to apply their framework to (4.1) and (4.2). Even for tree graphs the objectives in (4.1) and (4.2) do not seem to possess the convexity conditions needed in the general framework of [17]. By using a direct approach we have been able to prove that the set of tip nodes of a tree constitutes an FDS for the following generalization of (4.2):

$$(4.3)\quad f_5(x_1, \cdots, x_p) = \sum_{i=1}^{n} \alpha_i \,\text{Minimum}\, \{d(v_i, x_j) \mid 1 \leqq j \leqq p\} + \sum_{i=1}^{p} \sum_{j=1}^{p} \beta_{ij} d(x_i, x_j).$$

LEMMA 4.1. *Let α_i, $1 \leqq i \leqq n$, β_{ij}, $1 \leqq i, j \leqq p$, be nonnegative numbers. Suppose that $G = (V, E)$ is a tree graph. Then there exists an optimal solution $\{x_1^*, \cdots, x_p^*\}$ maximizing $f_5(x_1, \cdots, x_p)$ such that x_j^* is a tip of the tree, $1 \leqq j \leqq p$.*

For the sake of brevity we skip the details of the proof. Unlike the proof of Lemma 3.1, which exhibits a similar result, our proof of Lemma 4.1 is fairly involved. In fact, we have not been able to identify any convexity property, which usually suffices for the existence of a maximum solution at the extreme points. Surprisingly, (4.2) might have isolated maximum solutions which contain some nontip nodes even for the case where the tree is a path connecting a pair of nodes.

Example 4.2. Consider the four node path tree depicted in Fig. 4.1, with $d(v_1, v_2) = d(v_3, v_4) = 1$, and $d(v_2, v_3) = 2$. Let $p = 3$. Let $\alpha_1 = \alpha_4 = 0$, $\alpha_2 = \alpha_3 = 1$, $\beta_{ij} = 1$, $1 \leqq i, j \leqq 3$. An optimal solution satisfying the property in Lemma 4.1 is $x_1 = v_1$, $x_2 = x_3 = v_4$. An isolated optimal solution that contains a nontip node is $x_1 = v_1$, $x_2 = v_4$, and x_3 is the midpoint of the path connecting v_1 and v_4.

Next we turn to the p-Defense problem, defined by (4.1), on tree graphs. As mentioned above, we have constructed a polynomial recursive algorithm to solve the discrete version of the model when the p points are restricted to the node set of the tree. So far we have not been successful in our attempt to obtain an FDS of polynomial cardinality (in n and p) for the continuous problem on a general tree. For a star tree we have identified an FDS of $O(n^3p^2)$ cardinality.

When the graph is a path connecting two tip nodes the solution to the p-Defense model coincides with the unique solution to the homogeneous p-Maximin problem. It has been conjectured that this result holds in general. However, this is not the case even for star trees. First, we observe the following simple result for the p-Defense model.

PROPOSITION 4.3. *Let $\{x_1^*, \cdots, x_p^*\}$ be an optimal solution to* (4.1). *Define $I = \{i \mid x_i^* = x_j^*, \text{for some } 1 \leqq j \leqq p, j \neq i\}$. Suppose that I is nonempty. Then there exists a point x^* in $A(G)$, called a barrier, such that the set $\{y_1, \cdots, y_p\}$, defined by*

$$y_i = \begin{cases} x^* & \text{if } i \in I, \\ x_i^* & \text{otherwise} \end{cases}$$

is optimal for (4.1).

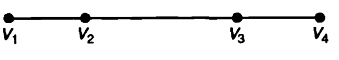


FIG. 4.1

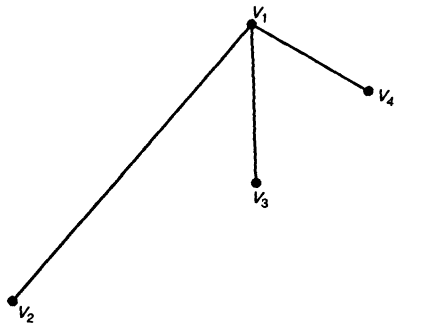


FIG. 4.2

If G is not a path and p is sufficiently large, every optimal solution to the p-Defense problem has a barrier point. No optimal solution to the p-Maximin problem has a pair of points x_i, x_j, $i \neq j$, with $d(x_i, x_j) = 0$. The following example demonstrates that the two problems might have different optimal solutions even if the solution to the p-Defense problem has no barrier point.

Example 4.4. Consider the star tree in Fig. 4.2. Let the edge lengths be $d(v_1, v_2) = 6$, $d(v_1, v_3) = 2$, $d(v_1, v_4) = 1$. The unique solution to the 3-Defense problem consists of the nodes v_2, v_3, and v_4. The unique solution to the 3-Maximin problem consists of v_2, v_3 and the midpoint of the path connecting v_2 and v_3.

5. Summary. We have considered obnoxious facility location on graphs using the p-Maximin and p-Maxisum criteria. These criteria are defined by (1.1) and (1.2), respectively.

For the general (nonhomogeneous) case we have shown that both models are strongly NP-hard even when the underlying graph consists of a single edge.

The other main results for the homogeneous p-Maximin problem are as follows. Unless $P = \text{NP}$ there is no polynomial ε-approximation algorithm for the problem on a general graph with $\varepsilon > \frac{2}{3}$. A $\frac{1}{2}$-approximation for the same model is given by the following greedy heuristic. Construct a sequence of p points such that each point is as far apart as possible from the set of points selected before.

Turning to tree graphs we present a linear time algorithm for the (homogeneous) r-anticover problem and apply it to get polynomial time algorithms for the homogeneous p-Maximin problem. For example, we improve upon previous results and solve the single facility case in $O(n \log^2 n)$ time.

For the p-Maxisum problem we have shown that the homogeneous case is strongly NP-hard on general graphs. Focusing on the homogeneous case on tree graphs we have presented an $O(np)$ dynamic programming scheme for its solution. We have then identified useful concavity properties and reformulated the model as a maximum concave separable quadratic flow problem. This formulation has led to polynomial and strongly polynomial algorithms for this homogeneous p-Maxisum problem.

In § 4 we briefly discuss two other models dealing with the location of obnoxious facilities.

Acknowledgment. The current simple and elegant proof of Proposition 2.7 is by Brandon. It improves significantly upon the original proof that was based on a complicated reduction.

REFERENCES

[1] R. Chandrasekaran and A. Daughety, *Location on tree networks: p-Center and n-Dispersion problems*, Math. Oper. Res., 6 (1981), pp. 50–56.

[2] R. Chandrasekaran and A. Tamir, *An* $O((n \log p)^2)$ *algorithm for the continuous p-Center on a tree*, SIAM J. Algebraic Discrete Meth., 1 (1980), pp. 370–375.

[3] ———, *Locating obnoxious facilities*, unpublished report, Tel Aviv University, 1979.

[4] R. Chandrasekaran and S. Kabadi, *Strongly polynomial algorithm for a class of combinatorial LCPs*, Oper. Res. Lett., 6 (1987), pp. 91–92.

[5] C. K. Chen and R. S. Garfinkel, *The generalized diameter of a graph*, Networks, 12 (1982), pp. 335–340.

[6] R. L. Church and R. S. Garfinkel, *Locating an obnoxious facility on a network*, Transportation Sci., 12 (1978), pp. 107–118.

[7] R. Cole, *Slowing down sorting networks to obtain faster sorting algorithms*, J. Assoc. Comput. Mach., 34 (1987), pp. 200–208.

[8] Z. Drezner and G. O. Wesolowsky, *Location of multiple obnoxious facilities*, Transportation Sci., 19 (1985), pp. 193–202.

[9] M. E. Dyer and A. M. Frieze, *A simple heuristic for the p-Center problem*, Oper. Res. Lett., 3 (1985), pp. 285–288.

[10] E. Erkut, *The discrete p-Dispersion problem*, European J. Oper. Res., 46 (1990), pp. 48–60.

[11] E. Erkut and S. Neuman, *Analytical models for locating undesirable facilities*, European J. Oper. Res., 40 (1989), pp. 275–291.

[12] E. Erkut, T. Baptie, and B. Von Holenbalken, *The discrete p-Maxisum location problem*, Res. Report 88-4, Dept. of Finance and Management Science, University of Alberta, Canada, 1988.

[13] G. N. Frederickson and D. B. Johnson, *Finding k-th paths and p-Centers by generating and searching good data structures*, J. Algorithms, 4 (1983), pp. 61–80.

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

[15] F. Granot and J. Skorin-Kapov, *Some proximity and sensitivity results in quadratic integer programming*, Math. Programming, 47 (1990), pp. 259–268.

[16] P. Hansen and D. Moon, *Dispersing facilities on a network*, RRR #52-88, RUTCOR, Rutgers University, 1988.

[17] J. N. Hooker, R. S. Garfinkel, and C. K. Chen, *Finite dominating sets for network location problems*, Working Paper 19-87-88, GSIA, Carnegie Mellon University, 1988.

[18] T. Ibaraki and N. Katoh, *Resource Allocation Problems. Algorithmic Approaches*, MIT Press, Cambridge, MA, 1988.

[19] A. W. J. Kolen, *Location Problems on Trees and in the Rectilinear Plane*, Stitchting Mathematisch Centrum, Amsterdam, 1982.

[20] A. W. J. Kolen and A. Tamir, *Covering problems*, in Discrete Location Theory, R. L. Francis and P. B. Mirchandani, eds., John Wiley, New York, 1990.

[21] M. J. Kuby, *Programming models for facility dispersion: the p-Dispersion and Maxisum dispersion problems*, Geographical Analysis, 19 (1987), pp. 315–329.

[22] N. Megiddo and A. Tamir, *New results on the complexity of p-Center problems*, SIAM J. Comput., 12 (1983), pp. 751–758.

[23] N. Megiddo, A. Tamir, E. Zemel, and R. Chandrasekaran, *An* $O(n \log^2 n)$ *algorithm for the k-th longest path in a tree with applications to location problems*, SIAM J. Comput., 10 (1981), pp. 328–337.

[24] M. Minoux, *Solving integer minimum cost flows with separable convex cost objective polynomially*, Math. Programming Stud., 26 (1986), pp. 237–239.

[25] D. Moon and S. S. Chaudhry, *An analysis of network location problems with distance constraints*, Management Science, 30 (1984), pp. 290–397.

[26] D. Moon and A. J. Goldman, *Tree network location problems with minimum separations*, IIE Trans., to appear.

[27] D. R. Shier, *A* Min-Max *theorem for p-Center problems on a tree*, Transportation Sci., 11 (1977), pp. 243–252.

[28] A. Tamir, *A finite algorithm for the continuous p-Center location problem on a graph*, Math. Programming, 31 (1985), pp. 298–306.

[29] ———, *On the solution value of the continuous p-Center location problem on a graph*, Math. Oper. Res., 12 (1987), pp. 340–349.

[30] ———, *Improved complexity bounds for center location problems on networks by using dynamic data structures*, SIAM J. Discrete Math., 1 (1988), pp. 377–396.

[31] ———, *A strongly polynomial algorithm for minimum convex separable quadratic cost flow problems on series-parallel graphs*, Technical Report, Dept. of Statistics and Operations Research, New York University, 1989.

[32] A. TAMIR AND E. ZEMEL, *Locating centers on a tree with discontinuous supply and demand regions*, Math. Oper. Res., 7 (1982), pp. 183–192.

[33] S. S. TING, *A linear time algorithm for maxisum facility location on tree networks*, Transportation Sci., 18 (1984), pp. 76–84.

[34] ———, *Obnoxious facility location problems on networks*, Ph.D. dissertation, The Johns Hopkins University, 1988.

[35] E. ZEMEL, *An $O(n)$ algorithm for the linear multiple choice knapsack and related problems*, Inform. Process. Lett., 18 (1984), pp. 119–121.

SIAM J. DISC. MATH.
Vol. 4, No. 4, pp. 568–574, November 1991

009

PLANE TREES AND H-VECTORS OF SHELLABLE CUBICAL COMPLEXES*

CLARA CHAN†

Abstract. Stanley first defined the generalized toric h-vector, a fundamental combinatorial invariant of polyhedral complexes (and more general objects). In the case where the complex is simplicial, this invariant can be computed by shelling, or taking apart the complex in a certain order. This paper shows how any shellable complex with cubical facets can be dealt with analogously. Based on a result of Shapiro the h-vector of any shellable cubical complex is formulated in terms of certain classes of plane trees.

Key words. cell complex, face poset, generalized h-vector, preorder, shelling

AMS(MOS) subject classifications. 05C05, 06A10

1. Background.

For general background and terminology on posets, see [2, Chap. 3].

Let $\hat{P}$ be a finite graded poset with $\hat{0}$ and $\hat{1}$ such that for all $x, y \in \hat{P}$, we have $\mu(x,y) = (-1)^{r(y)-r(x)}$, where μ is the Möbius function of $\hat{P}$, and r is its rank function. Then $\hat{P}$ is called *Eulerian*. Let $P = \hat{P} - \{\hat{1}\}$. For all $t \in P$, let $P_t = \{s \in P : \hat{0} \leq s < t\}$. Let d be the rank of P. Now define $f(P,x)$ and $g(P,x)$ inductively by

1. $f(\phi, x) = g(\phi, x) = 1$
2. $f(P,x) = \sum_{t \in P} g(P_t, x)(x-1)^{d-r(t)}$
3. $g(P,x) = \sum_{i=0}^{\lfloor d/2 \rfloor}(k_i - k_{i-1})x^i$, where k_i is the coefficient of x^i in $f(P,x)$

Then $\deg f(P,x) = d$, and the h-vector of P is given by $h(P) = (h_0, h_1, \cdots, h_d)$, with $h_i = k_{d-i}$ from above.

Fact. (See [1] for details and proofs.) When P is simplicial, i.e., P_t is Boolean for all $t \in P$, we have

$$\sum_{i=0}^{d} h_i x^{d-i} = \sum_{i=0}^{d} f_{i-1}(x-1)^{d-i},$$

where f_i is the number of elements of rank $i+1$ in P. Moreover, for any Eulerian $\hat{P}$ we have the *Dehn–Sommerville equations* $h_i = h_{d-i}$ for all $0 \leq i \leq d$, so that $g(P,x)$ completely determines $f(P,x)$.

Example. The face poset $\hat{L}_d$ of a d-dimensional cube is Eulerian (see [2, (3.8)]).

The following result was proved by Gessel (see [1]).

PROPOSITION 1. *We have*

$$g(L_d, x) = \sum_{k=0}^{\lfloor d/2 \rfloor} \frac{1}{d-k+1} \begin{pmatrix} d \\ k \end{pmatrix} \begin{pmatrix} 2d-2k \\ d \end{pmatrix} (x-1)^k.$$

Based on this result, Shapiro gave the following description of $g(L_d, x)$ in terms of plane trees (see [2, Ex. 3.71g]). If two vertices in a plane tree share an edge, we call the lower vertex a *child* of the upper, and write $a_n(i)$ for the number of n-vertex

* Received by the editors July 9, 1990; accepted for publication (in revised form) December 13, 1991.

† Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139. This work was supported by Office of Naval Research grant N00014-85-G-0084.

plane trees in which exactly i vertices have more than one child.

PROPOSITION 2. *We have*

$$g(L_d, x) = \sum_{i=0}^{\lfloor d/2 \rfloor} a_{d+1}(i) x^i.$$

Proof. Since there is only one tree on one vertex, we have $a_1(i) = \delta_{i0}$. By removing the root, we see that a plane tree on $n > 1$ vertices is determined by the ordered set of trees rooted at children of the original root. See Fig. 1.

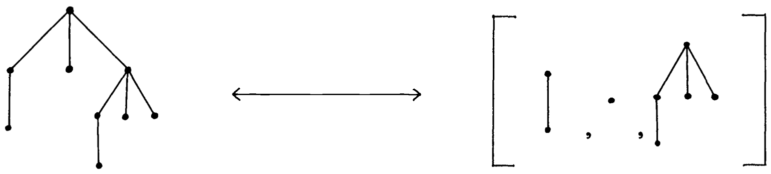


FIG. 1.

Let $\mathbf{N}$ denote the natural numbers, and let $\mathbf{P}$ denote the positive integers. For all $u \in \mathbf{N}$, $v \in \mathbf{P}$, let $[u]_v$ be the set of all $(u_1, \cdots, u_v) \in \mathbf{P}^v$ such that $\sum_{i=1}^{v} u_i = u$, and $[[u]]_v$ the set of all $(u_1, \cdots, u_v) \in \mathbf{N}^v$ such that $\sum_{i=1}^{v} u_i = u$. We have

$$a_n(i) = a_{n-1}(i) + \sum_{j=2}^{n-1} \sum_{b \in [n-1]_j} \sum_{t \in [[i-1]]_j} a_{b_1}(t_1) \cdots a_{b_j}(t_j) \quad \text{for } n > 1.$$

Let $z = \sum_{n \geq 1} \sum_{i \geq 0} a_n(i) y^i x^n$. Then $z = x + xz + xyz^2/(1-z)$, so

$$\begin{aligned}
(1+xy-x)z &= \frac{1}{2}\left(1 - \sqrt{1 + 4(x^2 - x^2y - x)}\right) = -\frac{1}{2}\sum_{k \geq 1} \tbinom{1/2}{k} 4^k (x^2 - x^2y - x)^k \\
&= \sum_{k \geq 1} \sum_{s=0}^{k} \frac{1}{k} \tbinom{2k-2}{k-1} x^{2k} \tbinom{k}{s} x^{s-k} (y-1)^s \\
&= \sum_{n \geq 1} \sum_{s=0}^{\lfloor n/2 \rfloor} \frac{1}{n-s} \tbinom{2n-2s-2}{n-s-1} \tbinom{n-s}{s} (y-1)^s x^n \\
&= \sum_{n \geq 1} \sum_{s=0}^{\lfloor n/2 \rfloor} \frac{1}{n-s} \left\{ \tbinom{2n-2s-2}{n-1} \tbinom{n-1}{s} + \tbinom{2n-2s-2}{n-2} \tbinom{n-2}{s-1} \right\} (y-1)^s x^n \\
&= (1+xy-x) \sum_{n \geq 1} g(L_{n-1}, y) x^n,
\end{aligned}$$

by Proposition 1. □

DEFINITION. A finite graded poset P with $\hat{0}$ is *lower Eulerian* if P_t is Eulerian for all $t \in P$. So for all lower Eulerian P, we can define $f(P,x)$ as in the Eulerian case.

Example. Let P be a finite graded poset with $\hat{0}$ such that for all $t \in P$ we have P_t isomorphic to L_r, for some r. In this paper, a *cubical* $(d-1)$*-complex* Q is a geometric realization (see [2, (3.8)]) of such a poset P of rank d. Thus, the face poset P_Q of a cubical complex Q is lower Eulerian, and we can define the h-vector of Q by $h(Q) = h(P_Q)$.

DEFINITION. Given any cubical $(d-1)$-complex Q, a *shelling* of Q is an ordering $(F_1, \cdots, F_r)$ of its facets such that for all $i > 1$ we have that $F_i \cap (F_1 \cup \cdots \cup F_{i-1})$ is a union of $(d-2)$-faces homeomorphic to a ball or sphere. If such an ordering exists, then Q is called *shellable.*

2. H-vectors of shellable cubical complexes.

Let Q be a shellable cubical $(d-1)$-complex with a given shelling. Let F be a facet of Q, and I the intersection of F with previous facets in the shelling. If I is a union of $0 \leq i \leq d-1$ antipodally unpaired $(d-2)$-faces and $0 \leq j \leq d-1-i$ pairs of antipodal $(d-2)$-faces, the *h-vector contribution by* F is $\sum_{t \in P_F \setminus P_I} g((P_Q)_t, x)(x-1)^{d-r(t)}$, where r is the rank function of P_Q. We will call F an (i,j)*-facet* (with respect to the given shelling) and denote its h-vector contribution by $f_d(i,j,x) = \sum_{k=0}^{d} b_d(i,j,k)x^k$. (So $f_d(0,d-1,x) = g(L_{d-1},x)$, for example.) Let $b_d(i,j,-1) = 0$ for all i,j. See Fig. 2.

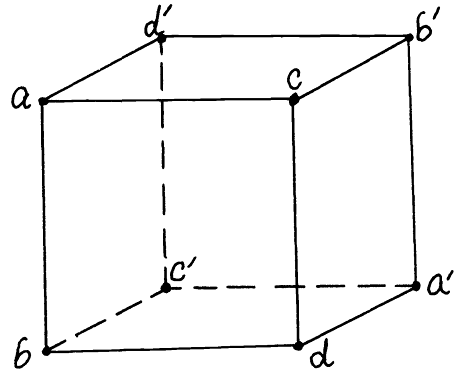


FIG. 2. *Let Q be the boundary of the three-dimensional cube with shelling $(abcd, ab'cd', a'b'cd, abc'd, a'bc'd, a'b'c'd)$. $(0,0)$-facet: $abcd$; $(1,0)$-facet: $ab'cd'$; $(2,0)$-facet: $a'b'cd, abc'd$; $(1,1)$-facet: $a'bc'd$; $(0,2)$-facet: $a'b'c'd'$.*

Note. Given a shelling of Q, if we let $s_{i,j}$ be the number of (i,j)-facets in the shelling, then it is clear that $f(P_Q,x) = \sum_{i,j} s_{i,j} f_d(i,j,x)$. Thus $h(Q)$ is given by the sum of the h-vector contributions by facets of Q.

LEMMA 1. *For all $0 \leq k \leq d$ we have that $b_d(0,0,k)$ is the number of d-vertex trees such that exactly k vertices have at most one child.*

Proof. It is clear that $f_d(0,0,x) = (x-1)f(L_{d-1},x) + g(L_{d-1},x)$. We also have $(x-1)f(L_r,x) = x^{r+1}g(L_r,x^{-1}) - g(L_r,x)$ (see [1]). Thus by Proposition 2 we have

$$f_d(0,0,x) = x^d g(L_{d-1},x^{-1}) = \sum_{k=0}^{\lfloor (d-1)/2 \rfloor} a_d(k)x^{d-k},$$

with $a_d(k)$ as defined earlier. The lemma follows. □

LEMMA 2. *For all* $1 \leq i \leq d-1$ *and* $0 \leq k \leq d$, *we have*

$$b_d(i,0,k) = b_d(i-1,0,k) + b_{d-1}(i-1,0,k) - b_{d-1}(i-1,0,k-1).$$

Proof. The lemma is true if and only if for all $i \geq 1$ we have

(1) $$f_d(i,0,x) = f_d(i-1,0,x) - (x-1)f_{d-1}(i-1,0,x).$$

It is easy to see that an $\llcorner i,0)$-facet contributes everything that an (i–1,0)-facet contributes to the h-vector, except for the contribution by a (d–2)-face that intersects the previous facets in i–1 antipodally unpaired (d–3)-faces. From this we deduce (1). □

LEMMA 3. *For* $0 \leq k \leq d$, $1 \leq i \leq d-2$, *and* $1 \leq j \leq d\text{–}1\text{–}i$, *we have*

$$b_d(i,j,k) = b_d(i,j-1,k) + 2b_{d-1}(i,j-1,k) - 2b_{d-1}(i,j-1,k-1).$$

Proof. Equivalently, we need to show that for all $1 \leq i \leq d\text{–}2$, $1 \leq j \leq d\text{–}1\text{–}i$, we have

(2) $$f_d(i,j,x) = f_d(i,j-1,x) - 2(x-1)f_{d-1}(i,j-1,x).$$

Similarly to the proof above, we deduce (2) by comparing the h-vector contribution by an (i,j)-facet with that contributed by an $(i,j\text{–}1)$-facet. □

3. The connection to plane trees.

At this point we introduce some plane tree terminology.

DEFINITIONS. An *n-tree* is a plane tree on n vertices. Two children of the same vertex are *siblings.* A vertex is a *fork* if it has more than one child; otherwise it is a *nonfork.* A vertex with no siblings is an *only child.* A child of the root vertex is a *root child.* If a vertex has a sibling to its left and right, it is an *inner child.* In this paper, the vertices of all plane trees are ordered recursively by root first, and then subtrees of the root, from left to right. This is called *preorder.* See Fig. 3.

If the ith vertex in an n-tree has exactly one child, we will call this vertex an i'. For $1 \leq j \leq n\text{–}2$, if the $(n-j)$th vertex is followed (in preorder) by an inner, only, or root child, we will call this vertex a j''. For all $j \leq n-2$, let $c_n(i,j,k)$ be the number of n-trees with exactly k nonforks which are neither $1', \cdots, i'$ nor $1'', \cdots, j''$. Let $c_n(0,n-1,k) = a_n(k)$ as defined earlier, and $c_n(i,j,-1) = 0$ for all i,j.

We now can state our main result.

THEOREM 1. *Let F be a facet of a cubical $(d-1)$-complex with given shelling. If F is an (i,j)-facet, then the h-vector contribution by F is* $\sum_{k=0}^{d} c_d(i,j,k)x^k$.

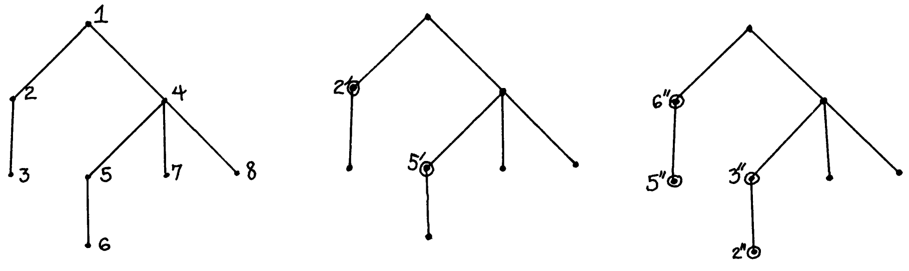


FIG. 3. (a) *Preorder*; (b) *i'-vertices*; (c) *j''-vertices.*

Proof. We must show that $b_d(i,j,k) = c_d(i,j,k)$ for all $0 \leq k \leq d$. First we consider the case $j = 0$. Since there is no such thing as $0'$ or $0''$, we have $c_d(0,0,k) = b_d(0,0,k)$ for all $-1 \leq k \leq d$ by Lemma 1. So by Lemma 2, it suffices to show that for all $1 \leq i \leq d-1$, $0 \leq k \leq d$ we have

$$(3) \qquad c_d(i,0,k) = c_d(i-1,0,k) + c_{d-1}(i-1,0,k) - c_{d-1}(i-1,0,k-1).$$

Now given any $(d-1)$-tree with s nonforks which are not $1', \cdots, t'$, we can get a d-tree with $s+1$ nonforks not $1', \cdots, t'$ by inserting a vertex between the $(t+1)$th vertex and its parent. (In Fig. 4, circled vertices are nonforks that are not $1', \cdots, t'$.) This map

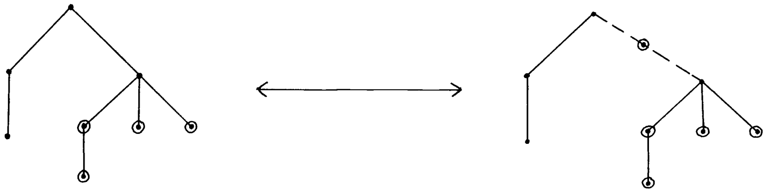


FIG. 4. $d = 9, t = 3, s = 4$.

is injective. From this observation (3) easily follows.

Now consider $j > 0$. As noted in the last section, $f_d(0, d-1, x) = g(L_d, x)$, so the theorem holds for $j = d-1$. By the definition of shelling, $1 \leq j \leq d-2$ $\Rightarrow$ $1 \leq i \leq d-2$. Fix such i. Now $c_d(i,0,k) = b_d(i,0,k)$ for all $-1 \leq k \leq d$ from above, so by Lemma 3, it suffices to show that for all $1 \leq j \leq d-1-i$, $0 \leq k \leq d$ we have

$$(4) \qquad c_d(i,j,k) = c_d(i,j-1,k) + 2c_{d-1}(i,j-1,k) - 2c_{d-1}(i,j-1,k-1).$$

Given any $(d-1)$-tree with exactly s nonforks not $1', \cdots, i'$ nor $1'', \cdots, j''$ we can get a d-tree with exactly $s+1$ nonforks not $1', \ldots, i'$ nor $1'', \cdots, j''$ in two ways: (In Figs.

5 and 6, circled vertices are nonforks that are neither $1', \cdots, i'$ nor $1'', \cdots, j''$.)

1. Insert a vertex between the $(d-1-j)$th vertex and its parent.

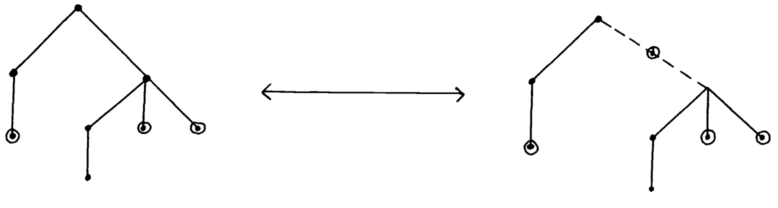


FIG. 5. $d = 9, i = 2, j = 4, s = 3$.

2. Replace the $(d-1-j)$th vertex and its offspring by a single leaf. If there is a $(d-j)$th vertex in the cropped tree, call this vertex v, and reinsert the removed subtree so that its root has v as a sibling on its immediate right. If no such v exists, insert the removed subtree so that it is the rightmost subtree directly under the root.

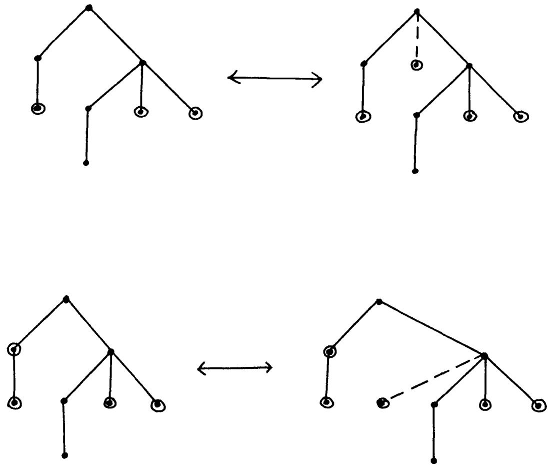


FIG. 6. (a) $d = 9, i = 2, j = 4, s = 3$; (b) $d = 9, i = 1, j = 3, s = 4$.

These two maps are injective and have disjoint images. The identity (4) follows. □

Example. Let Q be the rhombic dodecahedron shown in Fig. 7, with shelling

$$(afeg, cged, eda'b', b'c'ef, afc'd', abcg, a'b'c'g, a'f'cd, bce'f', e'd'ab, c'g'e'd', a'f'e'g').$$

We have $s_{0,0} = 1$, $s_{1,0} = 2$, $s_{2,0} = 6$, $s_{1,1} = 2$, $s_{0,2} = 1$. So $f(Q, x) = (x^2 + x^3) = 2(2x^2) + 6(x + x^2) + 2(2x) + (1 + x) = 1 + 11x + 11x^2 = x^3$.

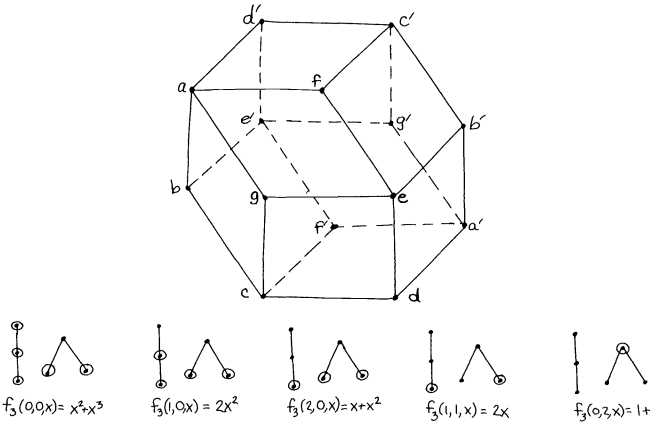


FIG. 7. (a) $f_3(0,0,x) = x^2+x^3$; (b) $f_3(1,0,x) = 2x^2$; (c) $f_3(2,0,x) = x+x^2$; (d) $f_3(1,1,x) = 2x$; (e) $f_3(0,2,x) = 1+x$.

Acknowledgment. I thank Richard Stanley for his helpful advice.

REFERENCES

[1] R. STANLEY, *Generalized H-vectors, intersection cohomology of toric varieties, and related results,* Adv. Stud. Pure Math., 11 (1987), Commutative Algebra and Combinatorics, pp. 187–213.

[2] ———, *Enumerative Combinatorics,* vol. 1, Brooks–Cole, Pacific Grove, CA, 1986.

SIAM J. DISC. MATH.
Vol. 4, No. 4, pp. 575–588, November, 1991

010

A LARGE DEVIATION RATE AND CENTRAL LIMIT THEOREM FOR HORTON RATIOS*

STANLEY XI WANG† AND EDWARD C. WAYMIRE‡

Abstract. Although originating in hydrology, the classical Horton analysis is based on a geometric progression that is widely used in the empirical analysis of branching patterns found in biology, atmospheric science, plant pathology, etc., and more recently in tree register allocation in computer science. The main results of this paper are a large deviation rate and a central limit theorem for Horton bifurcation ratios in a standard network model. The methods are largely self-contained. In particular, derivations of some previously known results of the theory are indicated along the way.

Key words. random tree, bifurcation ratio, large deviation, central limit theorem

AMS(MOS) subject classifications. primary 60F10,60F05,60C05; secondary 86A05,60J85

1. Introduction.

River systems around the world are known to hydrologists largely with the aid of maps. This has led to a number of interesting statistical/geometrical observations about rivers, which are understood to varying degrees of empiricism and mathematical rigor. In the present paper we consider a statistic introduced by Horton [19], called the *stream order statistic*, to measure the bifurcation complexity in river networks. Among other things, this statistic has been used to provide an estimate on the total length of rivers in the United States at roughly three million miles; see Leopold [21]. In addition, applications to other naturally occurring branching patterns can be found in Horsfield [17], [18], Berry and Bradley [3], Borchert and Slade [4], Steingraeber, Kascht, and Frank [25], Aho, Sethi, and Ullman [1], Flajolet and Odlyzko [11], Flajolet and Prodinger [10], Flajolet, Raoult, and Vuillemin [12], and Vauchaussade and Viennot [26], to name a few. Some of these and other references can also be found in Jarvis and Woldenberg [20].

For a theoretical formulation of Horton's order analysis, geomorphologists and hydrologists consider an idealized river network represented by a rooted binary tree digraph having n degree-one vertices representing sources; consult Shreve [24] and Chartrand and Lesniak [6] for some of the graph theory terminology. The number n of sources is called the *network magnitude.* Edges of the graph connecting sources to adjacent *junctions* (degree three vertices) are called *external links*, and those between two junctions are called *internal links.* The edge incident to the root is called the *stem.* The stem is regarded as an external link if and only if $n = 1$; otherwise the stem is internal. Each external link is said to have order one. An edge incident to two order-one links is then defined to have order two. We now proceed inductively as follows. An edge has order $k > 1$ if and only if it is incident to two edges of order i and j such that either $i = j = k - 1$ or $i \neq j$ and $\max\{i, j\} = k$. A *stream* (and its order) is defined as a maximal connected path of incident edges of the same order (which

*Received by the editors April 24, 1989; accepted for publication (in revised form) November 16, 1990.

†Department of Mechanical Engineering, California Institute of Technology, Pasadena, California 91125. This author was partially supported as a Graduate Research Assistant under grant 26220-GS from the Army Research Office.

‡Department of Mathematics, Oregon State University, Corvallis, Oregon 97331. This research was supported in part by grant 26220-GS from the Army Research Office and the Mathematical Sciences Institute of Cornell University and National Science Foundation grant DMS-8801466.

is then taken to be the *stream order*). Thus the order-one streams are precisely the (order-one) external links. The maximal order taken over streams in the network is called the *network order*.

Fluctuations in network statistics are represented by assuming all networks of the same magnitude n to be equiprobable, called the *random model*. Because of an inherent imprecision in the small scale details of large river networks, we typically would like robust (large source number) asymptotics for the network variables. Precise asymptotics on the expected (i.e., phase average) network order (maximum stream order) have been obtained by Meir, Moon, and Pounder [22], who show that for fixed n, the expected order is $\frac{1}{2}\log_2 n + O(1)$. Shreve has conjectured on the basis of computer simulation that the location of the mode should also roughly coincide with this value, the details of which are clarified by the results of Meir, Moon, and Pounder [22]. Note that for a given value of n, the largest network order possible is $1 + \log_2 n$. Our focus here is on the asymptotics for sample averages of the lower-order streams and links. Specifically, let $L_{i,n}$ and $S_{i,n}$ denote the sample numbers of links and streams, respectively, of order i in a network of magnitude n. These represent the sample values we compute from a network map. The ratios $L_{2,n}/L_{1,n} = L_{2,n}/n$ and $S_{2,n}/S_{1,n} = S_{2,n}/n$ are called Horton link and stream number *bifurcation ratios*, respectively. Empirical forms of *Horton's laws* refer to the asymptotic stability of these ratios for basins of large magnitude.

Noting a simple mean and variance computation by Werner [27], Gupta and Waymire [15] provide the theoretical counterpart in the form of a law of large numbers for the stream number bifurcation ratios and Mesa [23] for the link numbers; see also Gupta and Waymire [14] for a related overview. The purpose of the present paper is to provide a description of the fluctuations in the form of central limit theorems and large deviation rates where possible. We consider two statistics, the stream number bifurcation ratio and the link number bifurcation ratio. We first obtain a large deviation rate for the former ratio, from which a central limit theorem will also follow. Although we have not obtained the corresponding descriptions of the fluctuations for the link number bifurcation ratio, the computations given in § 4 suggest that similar results should hold for this ratio.

Some very interesting results have already appeared in the literature, which provide Gaussian asymptotic approximations to combinatorial enumerations, e.g., see Carlitz et al. [5], Harper [16], Flajolet and Odlyzko [11], and Bender [2]. In particular, Bender [2] provides a general condition for the asymptotic normality of a doubly indexed sequence of positive numbers that essentially requires a pole in the bivariate generating function. Bender [2] also gives a number of interesting example applications. However, these results do not seem to be applicable to the present problem since the singularity in the bivariate generating function is not a pole; see the remark at the end of the next section.

The precise statements of main results are given in § 2. Various other results are obtained along the way, including the laws of large numbers, which serve to unify the previously known asymptotics and exact formulae for stream and link number probabilities and their expected values. In addition, a few new exact formulae are also provided in this connection. The proofs of the main results are given in § 5. Both the calculation of large deviation rates and the central limit theorem rest on the natural recursive structure of random model described in § 3. An analysis of the asymptotic form of the factorial moments of the distribution of stream and link numbers is given in § 4, which may be read independently of the proofs of the main results. In particular, we obtain the correct *factorial moments* of a Poisson distribution with parameter

$\lambda \propto n$ for the asymptotic (in n) factorial moments of the bifurcation ratios, but the (correct) asymptotic Gaussian approximation to this Poisson distribution is wrong for the bifurcation ratios (i.e., right mean, wrong variance parameter). These factorial moment calculations illustrate the delicacy of the problems due to the occurence of $\infty - \infty$ effects.

2. Statement of results. Let Ω_n denote the collection of rooted binary tree graphs of magnitude n. Then, according to a classic (1858) formula of Cayley, its cardinality $|\Omega_n|$ is given by

$$|\Omega_n| = \frac{1}{2n-1}\binom{2n-1}{n} \sim \frac{2^{2n-2}}{\sqrt{\pi}} n^{-3/2}, \quad n \to \infty. \tag{2.1}$$

The random model is defined by the probability measure P_n, which assigns probability $|\Omega_n|^{-1}$ to each $\tau \in \Omega_n$. The random variables $L_{i,n}, S_{i,n}$, denoting link numbers and stream numbers of order i, respectively, are defined on Ω_n as in the Introduction. The random variables $R_{L,2}^{(n)} = L_{2,n}/n$ and $R_{S,2}^{(n)} = S_{2,n}/n$ are referred to as *link* and *stream number bifurcation ratios*, respectively.

It is now well known that the value of each of the bifurcation ratios $S_{2,n}/n$ and $L_{2,n}/n$ stabilizes according to the following law of large numbers.

THEOREM 2.1. (Law of large numbers). *For the random model,* (i)$S_{2,n}/n \to \frac{1}{4}$ *in probability as* $n \to \infty$; (ii) $L_{2,n}/n \to \frac{1}{2}$ *in probability as* $n \to \infty$.

In fact, this may be obtained by the methods of the present paper according to which one has the following properties.

PROPOSITION 2.2. *For* $n \geq 3$, (i)

$$E\{S_{2,n}\} = \frac{n(n-1)}{2(2n-3)} \sim \frac{n}{4}$$

$$E\{S_{2,n}(S_{2,n}-1)\} = \frac{n(n-1)(n-2)(n-3)}{4(2n-5)(2n-3)} \sim \frac{n^2}{16};$$

(ii)

$$E\{L_{2,n}\} = \frac{2n-1}{\binom{2n-1}{n}}\{2^{2n-3} - 2^{n-2} - \sum_{m=1}^{n-2}\binom{2m}{m}\frac{2^{n-m-2}}{2m-1}(2^{n-m-1}-1)\} \sim \frac{n}{2}$$

$$E\{L_{2,n}(L_{2,n}-1)\} = \frac{2n-1}{\binom{2n-1}{n}}\{4(2n-5)\binom{2n-6}{n-3} - \sum_{k=0}^{n-4}(3n-3k-13)(2k+1)\binom{2k}{k}2^{n-k-3}\} \sim \frac{n^2}{4}.$$

The exact expressions in (ii) are new, but the exact forms in (i) have been obtained previously by other methods. It is important to note from the exact calculations that the asymptotic formulae for the first two factorial moments do *not* provide the asymptotic variance; i.e., there is an $\infty - \infty$ contribution. For example,

$$\begin{aligned} \mathrm{Var} S_{2,n} &= ES_{2,n}(S_{2,n}-1) + ES_{2,n} - (ES_{2,n})^2 \\ &= \frac{n(n-1)(n-2)(n-3)}{2(2n-3)(2n-3)(2n-5)} \sim \frac{n}{16} \quad , \end{aligned} \tag{2.2}$$

while

$$\frac{n^2}{16}+\frac{n}{4}-(\frac{n}{4})^2=\frac{n}{4}. \tag{2.3}$$

This point is significant in a theory that rests largely on asymptotics, as is further demonstrated in §4. This aside, note that Theorem 2.1 follows from the linearity of the mean and variance in n (Proposition 2.2) by an application of Chebyshev's inequality.

Proposition 2.2 is proved in § 4 together with a generalization to the precise asymptotic forms of the higher-order factorial moments. The moment analysis is based on the following identity which may be of independent interest.

LEMMA 2.1. *Let $\Gamma(x)$ denote the gamma function. Then, for each $n = 1, 2, \cdots$, we have*

$$\sum_{i=1}^{n}\binom{n+1}{i}\Gamma(i-\frac{1}{2})\Gamma(n-i+\frac{1}{2})=4\sqrt{\pi}\Gamma(n+\frac{1}{2}). \tag{2.4}$$

The identity (2.4) has several interesting variants. For example,

$$\sum_{i=1}^{n}\binom{n+1}{i}\binom{n-1}{i-1}\binom{2n-2}{2i-2}^{-1}=2(2n-1). \tag{2.5}$$

Remark. This lemma is used in the moment computations given in § 4. Its verification is quite amenable to "proof" by symbolic algebra software; e.g., Macsyma or Maple. This was, in fact, our first approach (after calculator tests) to checking the assertion. However, the result can also be obtained as a case of Gauss's theorem for ${}_2F_1$ hypergeometric functions or by induction, as indicated below. For us, the identity (2.4) was uncovered by the process of "matching asymptotics," i.e., in trying to identify the slowly varying part of the asymptotic Tauberian expansion (4.11) below.

Proof of Lemma 2.1.

Method 1 (Classical identities). We have

$${}_2F_1\left(\begin{matrix}a,b & \\ & ;1\\ c & \end{matrix}\right)=\frac{\Gamma(c)\Gamma(c-a-b)}{\Gamma(c-a)\Gamma(c-b)}. \tag{2.6}$$

From this we note in the cases where $i = 0$ and $i = n + 1$ that

$$\binom{n+1}{i}\Gamma(i-\frac{1}{2})\Gamma(n-i+\frac{1}{2})=-2\sqrt{\pi}\Gamma(n+\frac{1}{2}), \tag{2.7}$$

so that the identity of Lemma 2.1 is reduced to

$$\Gamma(-\frac{1}{2})\Gamma(n+\frac{1}{2})\ {}_2F_1\left(\begin{matrix}-n-1 & , & -\frac{1}{2} & \\ & & & ;1\\ & \frac{1}{2}-n & & \end{matrix}\right)=0. \tag{2.8}$$

Likewise, the equivalent variant (2.5) may be obained as a specialization of the (Hagan/Rothe) identity (3.146) given in Gould [13]; take $y = 2n+1, p = n+1, x = 1, q = -1$, and $z = 2$, there.

Method 2 (Induction). Let $B(u,v)$ denote a Beta function. It is well known that

$$B(u,v) = \frac{\Gamma(u)\Gamma(v)}{\Gamma(u+v)}.$$

Thus, it suffices to show that

$$\frac{1}{2\sqrt{\pi}}(n-1)!\sum_{i=1}^{n}\binom{n+1}{i}\int_0^{\frac{\pi}{2}}(\cos\theta)^{2(i-1)}(\sin\theta)^{2(n-i)}d\theta = \Gamma(n+\frac{1}{2}). \tag{2.9}$$

By induction, suppose for $k \le n$ (2.9) is true; then for $k = n+1$, we have

$$\begin{aligned}
\text{lhs} &= \frac{1}{2\sqrt{\pi}}n!\sum_{i=1}^{n+1}\binom{n+2}{i}\int_0^{\pi/2}(\cos\theta)^{2(i-1)}(\sin\theta)^{2(n+1-i)}d\theta \\
&= \frac{1}{2\sqrt{\pi}}n!\sum_{i=1}^{n}[\binom{n+1}{i}+\binom{n+1}{i-1}]\int_0^{\pi/2}(\cos\theta)^{2(i-1)}(\sin\theta)^{2(n+1-i)}d\theta \\
&\quad + \frac{1}{2\sqrt{\pi}}n!\binom{n+2}{n+1}\int_0^{\pi/2}(\cos\theta)^{2n}d\theta \\
&= n\Gamma(n+\frac{1}{2}) - \frac{1}{2\sqrt{\pi}}n\sum_{i=1}^{n}(n-1)!\binom{n+1}{i} + \int_0^{\pi/2}(\cos\theta)^{2i}(\sin\theta)^{2(n-i)}d\theta \\
&\quad + \frac{n!}{2\sqrt{\pi}}\sum_{i=0}^{n-1}\binom{n+1}{i} + \int_0^{\pi/2}(\cos\theta)^{2i}(\sin\theta)^{2(n-i)}d\theta + \frac{n!(n+2)}{2\sqrt{\pi}}\int_0^{\pi/2}(\cos\theta)^{2n}d\theta \\
&= n\Gamma(n+\frac{1}{2}) + \frac{n!}{2\sqrt{\pi}}\frac{\Gamma(n+\frac{1}{2})\Gamma(\frac{1}{2})}{\Gamma(n+1)} = \Gamma(n+\frac{3}{2}) = \text{rhs}.
\end{aligned}$$

To accompany the law of large numbers it is important to have some measure of the fluctuations from the average. The idea behind the large deviation rate is that the probability of a deviation from the mean by some prescribed amount goes to zero at an exponentially fast rate, which we may try to calculate. The following results describe probabilities of fluctuations from this point of view.

THEOREM 2.3. (Large deviation rate). *For the random model,*

$$\lim_{n\to\infty}\frac{1}{n}\log P(\frac{S_{2,n}}{n} > y) = -I(y), y \in (\frac{1}{4},\frac{1}{2}) \tag{2.10}$$

and

$$\lim_{n\to\infty}\frac{1}{n}\log P(\frac{S_{2,n}}{n} < y) = -I(y), y \in (0,\frac{1}{4}), \tag{2.11}$$

where

$$I(y) = (4y-1)\tanh^{-1}(4y-1) - \log(\cosh(\tanh^{-1}(4y-1))). \tag{2.12}$$

The rate function $I(y)$ is called an *entropy function* in the theory of large deviations; see Ellis [9]. The graph of the entropy function (2.12) is a ∪-shape on the interval $(0,\frac{1}{2})$ with a minimum at $(\frac{1}{4},0)$.

THEOREM 2.4. (Central Limit Theorem). *For the random model, we have*

$$\sqrt{n}(\frac{S_{2,n}}{n} - \frac{1}{4}) \Rightarrow N(0, \frac{1}{16}), \quad n \to \infty, \tag{2.13}$$

where $\Rightarrow$ denotes convergence in distribution, and $N(\mu, \sigma^2)$ denotes a limit law that is Gaussian having mean μ and variance σ^2.

Remark. Let $l_i(n,k), s_i(n,k)$ denote the number of trees in Ω_n having k links of order i and k streams of order i, respectively, as defined in the Introduction, and define

$$\hat{l}_2(x,y) = \sum_{n,k} l_2(n,k)x^n y^k, \tag{2.14}$$

$$\hat{s}_2(x,y) = \sum_{n,k} s_2(n,k)x^n y^k. \tag{2.15}$$

Also, let

$$\hat{l}'_2(x,y) = \sum_{n,k} l'_2(n,k)x^n y^k, \tag{2.16}$$

where $l'_2(n,k)$ denotes the number of trees of network order 2 in Ω_n having k links of order 2. Then considerations of the recursive structure give the following relations:

$$\hat{s}_2 = \hat{s}_2^2 + 2x\hat{s}_2 + x^2 y, \tag{2.17}$$

$$\hat{l}'_2 = x^2 y + 2xy\hat{l}'_2, \tag{2.18}$$

$$\hat{l}_2 = \hat{l}'_2 + 2x(\hat{l}_2 - \hat{l}'_2) + \hat{l}_2^2. \tag{2.19}$$

Solving for these generating functions, we find singularities other than poles. This does not seem to be covered by general theory; cf. Bender [2].

3. Some preliminaries.

We continue to let $l_i(n,k), s_i(n,k)$ denote the number of trees in Ω_n having k links of order i and k streams of order i, respectively. Then for the random model

$$P_n(L_{i,n} = k) = \frac{l_i(n,k)}{|\Omega_n|}, \tag{3.1}$$

$$P_n(S_{i,n} = k) = \frac{s_i(n,k)}{|\Omega_n|}, \quad n \geq 1, k \geq 0. \tag{3.2}$$

In view of the recursive structure of the trees as described precisely in Meir, Moon, and Pounder [22], we obtain convolution identities among the $s_2(n,k)$'s and $l_2(n,k)$'s of the following forms.

LEMMA 3.1. *For the random model,*

$$s_2(1,k) = \delta_{0,k},\ s_2(2,k) = \delta_{1,k},\ s_2(4,1) = 4,\ s_2(4,2) = 1, \tag{3.3a}$$

(i)

$$s_2(n,0) = 0, \qquad s_2(n,1) = 2^{n-2},\ n \geq 2, \tag{3.3b}$$

$$s_2(n,k) = \sum_{m=1}^{n-1} \sum_{j=0}^{k} s_2(m,j) s_2(n-m,k-j),\ n \geq 5, k \neq 2; \tag{3.3c}$$

(ii)

$$l_2(1,k) = \delta_{0,k},\ l_2(2,k) = \delta_{1,k},\ l_2(3,k) = 2\delta_{2,k}, l_2(n,0) = 0, \tag{3.4a}$$

$$l_2(n,n-1) = 2^{n-2}, l_2(n,n-2) = \sum_{m=2}^{n-2} l_2(m,m-1) l_2(n-m,n-m-1), \tag{3.4b}$$

$$l_2(n,k) = \sum_{m=1}^{n-1} \sum_{j=0}^{k} l_2(m,j) s_2(n-m,k-j),\ n \geq 5, k \leq n-3. \tag{3.4c}$$

The proof of these recursions are fairly straightforward and will be left to the reader. It is to be noted that, in either case, special provision must be made for the *order-three networks* (i.e., $n = 4$, $k = 2$ for stream count, $n = 4, k = n-2$ for link count).

The convolution forms in the identities of Lemma 3.1 transform under

$$\hat{s}_2(n,x) = \sum_{k=1}^{n} s_2(n,k) x^k, \tag{3.5}$$

$$\hat{l}_2(n,x) = \sum_{k=0}^{n} l_2(n,k) x^k, \tag{3.6}$$

according to the following lemma.

LEMMA 3.2.

(i)

$$\hat{s}_2(1,x) = 1, \hat{s}_2(2,x) = x, \hat{s}_2(3,x) = 2x, \hat{s}_2(4,x) = x^2 + 4x, \tag{3.7a}$$

$$\hat{s}_2(n,x) = \sum_{m=1}^{n-1} \hat{s}_2(m,x) \hat{s}_2(n-m,x), n \geq 3, \tag{3.7b}$$

and

(ii)

$$\hat{l}_2(1,x) = 1,\ \hat{l}_2(2,x) = x, \tag{3.8a}$$

$$\hat{l}_2(n,x) = (2x)^{n-2}(x-1) + \sum_{m=1}^{n-1} \hat{l}_2(m,x)\hat{l}_2(n-m,x), n \geq 3. \tag{3.9b}$$

Although somewhat lengthy, Lemma 3.2 is easily verified as a consequence of Lemma 3.1 and its proof is omitted; the details for (ii) can be found in Mesa [23]. These recursions represent the starting point for our proofs. Define moment generating functions

$$\psi_n(\xi) = Ee^{\xi S_{2,n}} = \sum_k e^{k\xi} P_n(S_{2,n} = k) = |\Omega_n|^{-1}\hat{s}_2(n, e^{\xi}) \tag{3.10}$$

and

$$\lambda_n(\xi) = Ee^{\xi L_{2,n}} = \sum_k e^{k\xi} P_n(L_{2,n} = k) = |\Omega_n|^{-1}\hat{l}_2(n, e^{\xi}). \tag{3.11}$$

Then Lemma 3.2 provides recursive equations of convolution type for $|\Omega_n|\psi_n(\xi)$ and $|\Omega_n|\lambda_n(\xi)$ which will be analyzed in §§4 and 5.

For ease in reference, we close this section with the statements of the theorem to be used in the proofs in § 5. Theorem 3.3 seems to have a somewhat fragmented history and has been useful in diverse contexts; see Cox and Griffeath [7] and references therein. A systematic treatment of the elements of large deviation theory can be found in Ellis [9] and Deuschel and Stroock [8].

THEOREM 3.3. (Sievers,Plachky and Steinbach, Ellis, Cox and Griffeath). *Let $\{X_n : n = 0, 1 \cdots\}$ be a sequence of random variables and let*

$$\varphi_n(\xi) = a_n^{-1}\log Ee^{\xi X_n}, \tag{3.12}$$

where $\{a_n\}$ is a sequence of positive numbers such that $a_n \to \infty$ as $n \to \infty$. Assume that on the interval $(\xi_-, \xi_+) \ni 0$, we have

$$\lim_{n\to\infty} \varphi_n(\xi) = \varphi_\infty(\xi) < \infty, \tag{3.13}$$

where $\varphi_\infty(\xi)$ is strictly convex and C^2 on (ξ_-, ξ_+). If φ_n' is convex on $[0, \xi_+)$, and $\lim_{n\to\infty} \varphi_n''(0) = \sigma^2 = \varphi_\infty''(0)$, then

$$\lim_{n\to\infty} a_n^{-1}\log P(\frac{X_n}{a_n} > y) = -I(y), y \in (\mu, \alpha_+) \tag{3.14}$$

and

$$\lim_{n\to\infty} a_n^{-1}\log P(\frac{X_n}{a_n} < y) = -I(y), y \in (\alpha_-, \mu), \tag{3.15}$$

where $\mu = \varphi_\infty'(0), \alpha_- = \varphi_\infty'(\xi_-+), \alpha_+ = \varphi_\infty'(\xi_+-)$, and $I(y)$ is the Legendre transform of $\varphi_\infty(\xi)$. In addition,

$$\frac{X_n - EX_n}{\sqrt{a_n}} \Rightarrow N(0, \sigma^2), \quad n \to \infty. \tag{3.16}$$

4. A moment analysis. This section may be read independently of the proofs of the main results of the paper given in the next section.

The factorial moments are defined by

$$\nu_n^{[r]} = E_n[S_{2,n}]_r = |\Omega_n|^{-1}\frac{d^r}{dx^r}\hat{s}_2(n,x)|_{x=1}, \tag{4.1}$$

$$\mu_n^{[r]} = E_n[L_{2,n}]_r = |\Omega_n|^{-1}\frac{d^r}{dx^r}\hat{l}_2(n,x)|_{x=1}, \tag{4.2}$$

where $[x]_r = x(x-1)\cdots(x-r+1), x \in R$, $r = 1, 2\cdots$. In view of Lemma 3.2, we have accordingly, for $\overline{\nu}_n^{[r]} = |\Omega_n|\nu_n^{[r]}$ and $\overline{\mu}_n^{[r]} = |\Omega_n|\mu_n^{[r]}$, that for each $r = 1, 2, \cdots, n \geq 4$,

$$\overline{\nu}_n^{[r]} = \sum_{m=1}^{n-1}\sum_{j=0}^{r}\binom{r}{j}\overline{\nu}_n^{[r-j]}\overline{\nu}_{n-m}^{[j]} \tag{4.3}$$

(4.4)

$$\overline{\mu}_n^{[r]} = \{\frac{1}{n-1}\sum_{m-1}^{n}\delta_{r,m}\}r(n-1)(n-2)\cdots(n-r)2^{n-2} + \sum_{m=1}^{n-1}\sum_{j=0}^{r}\binom{r}{j}\overline{\mu}_n^{[r-j]}\overline{\mu}_{n-m}^{[j]}.$$

To obtain these, simply note that

$$\frac{d^r}{dx^r}\hat{s}_2(n,x) = \sum_{m=1}^{n-1}\sum_{j=0}^{r}\binom{r}{j}\frac{d^{r-j}}{dx^{r-j}}\hat{s}_2(m,x)\frac{d^j}{dx^j}\hat{s}_2(n-m,x),$$

$$\frac{d^r}{dx^r}\hat{l}_2(n,x) = \{2^{n-2}(n-1)(n-2)\cdots(n-r)[(n-1)x-n+r+1]x^{n-r-2}\}\frac{1}{n-1}\sum_{m-1}^{n}\delta_{r,m}$$
$$+\sum_{m=1}^{n-1}\sum_{j=0}^{r}\binom{r}{j}\frac{d^{r-j}}{dx^{r-j}}\hat{l}_2(m,x)\frac{d^j}{dx^j}\hat{l}_2(n-m,x),$$

Proof of Proposition. 2.2. To verify (ii), for example, use (4.4) with $\overline{\mu}_1^{[1]} = 0$ and $\overline{\mu}_2^{[1]} = 1$. Then, $\hat{\mu}_1(t) = \sum_{n=1}^{\infty}\overline{\mu}_n^{[1]}t^n$ satisfies

$$\begin{aligned}\hat{\mu}_1(t) &= t^2 + \sum_{n=3}^{\infty}2^{n-2}t^n + 2\sum_{n=3}^{\infty}\sum_{m=1}^{n-1}\overline{\mu}_m^{[1]}|\Omega_{n-m}|t^n\\ &= \frac{t^2}{1-2t} + 2\hat{\mu}_1(t)\hat{\Omega}(t),\end{aligned}$$

where $\hat{\Omega}(t) = \sum_{n=1}^{\infty}|\Omega_n|t^n = (1-\sqrt{1-4t})/2$. Thus,

$$\begin{aligned}\hat{\mu}_1(t) &= t^2\{\frac{-1}{1-2t} + \frac{2}{1-4t}\}(1-4t)^{1/2}\\ &= \sum_{n=0}^{\infty}\{2\cdot 4^n - 2^n\}t^{n+2}\sum_{m=0}^{\infty}\binom{\frac{1}{2}}{m}(-4t)^m\\ &= \sum_{n=2}^{\infty}\{\sum_{\{(m,k):m+k=n,k\geq 2,m\geq 0\}}\{2\cdot 4^{k-2} - 2^{k-2}\}\binom{\frac{1}{2}}{m}(-4)^m\}t^n.\end{aligned}$$

Thus,

$$\overline{\mu}^{[1]}_n = \sum_{m=0}^{n-2} \{2 \cdot 4^{n-m-2} - 2^{n-m-2}\} \binom{\frac{1}{2}}{m} (-4)^m, n \geq 2. \tag{4.5}$$

Since

$$\binom{\frac{1}{2}}{m} = 2^{-2m}(-1)^{m-1}\frac{1}{2m-1}\binom{2m}{m}, m \geq 1,$$

we have

$$\overline{\mu}^{[1]}_n = 2 \cdot 4^{n-2} - 2^{n-2} + \sum_{m=1}^{n-2} (-1)^{2m-1}\frac{1}{2m-1}\binom{2m}{m}\{2^{2n-2m-3} - 2^{n-m-2}\}. \tag{4.6}$$

Also, $\overline{\mu}^{[2]}_1 = 0$ and $\overline{\mu}^{[2]}_2 = 0$, Then, $\hat{\mu}_2(t) = \sum_{n=1}^{\infty} \overline{\mu}^{[2]}_n t^n$ satisfies

$$\begin{aligned}
\hat{\mu}_2(t) &= \sum_{n=3}^{\infty} \sum_{m=1}^{n-1} \sum_{j=0}^{2} \binom{2}{j} \overline{\mu}^{[2-j]}_m \overline{\mu}^{[j]}_{n-m} t^n + 2\sum_{n=3}^{\infty} (n-2)2^{n-2}t^n \\
&= \frac{4t^3}{(1-2t)^2} + 2\sum_{n=3}^{\infty}\sum_{m=1}^{n-1} \overline{\mu}^{[2]}_{n-m}|\Omega_m| t^n + 2\sum_{n=3}^{\infty}\sum_{m=1}^{n-1} \overline{\mu}^{[1]}_m \overline{\mu}^{[1]}_{n-m} t^n \\
&= \frac{4t^3}{(1-2t)^2} + 2\hat{\mu}_2(t)\hat{\Omega}(t) + 2\hat{\mu}_1^2(t) \\
&= \frac{4t^3 - 14t^4}{(1-4t)\tfrac{3}{2}(1-2t)^2}. \qquad (4.7)
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\sum_{j=0}^{\infty} \overline{\mu}^{[2]}_{j+3} t^j &= \frac{\hat{\mu}_2(t)}{t^3} \\
&= 4(1-4t)^{-3/2}(1-2t)^{-2} - 14t(1-4t)^{-3/2}(1-2t)^{-2} \\
&= \sum_{k=0}^{\infty} \binom{\frac{-3}{2}}{k} (-4)^k t^k \sum_{m=1}^{\infty} m(2t)^{m-1} - 14t\sum_{k=0}^{\infty} \binom{\frac{-3}{2}}{k} (-4)^k t^k \sum_{m=1}^{\infty} m(2t)^{m-1} \\
&= 4\sum_{n=0}^{\infty} \sum_{m+k=n}^{\infty} \binom{\frac{-3}{2}}{k} (-4)^k 2^{n-k}(n-k+1)t^n \\
&\quad - 14\sum_{n=0}^{\infty} \sum_{m+k=n}^{\infty} \binom{\frac{-3}{2}}{k} (-4)^k 2^{n-k}(n-k+1)t^{n+1} \\
&= 4 + \sum_{j=1}^{\infty} \{4\binom{\frac{-3}{2}}{j}(-4)^j + \sum_{k=0}^{j-1} \binom{\frac{-3}{2}}{k} (-4)^k 2^{j-k}(3k-3j+4)\}t^j.
\end{aligned}$$

Thus, $\overline{\mu}^{[2]}_3 = 4$, and

$$\overline{\mu}^{[2]}_{j+3} = 4\binom{\frac{-3}{2}}{j}(-4)^j + \sum_{k=0}^{j-1} \binom{\frac{-3}{2}}{k}(-4)^k 2^{j-k}(3(k-j)+4), \qquad j \geq 1,$$

i.e.,

$$\overline{\mu}_n^{[2]} = 4\binom{\frac{-3}{2}}{n-3}(-4)^{n-3} + \sum_{k=0}^{n-4}\binom{\frac{-3}{2}}{k}(-4)^k 2^{n-k-3}(3k-3n+13), n \geq 4.$$

Since

$$\binom{\frac{-3}{2}}{k} = 2^{-2k}(-1)^k(2k+1)\binom{2k}{k},$$

we obtain,

$$E\{L_{2,n}(L_{2,n}-1)\} = 4(2n-5)\binom{2n-6}{n-3}$$

$$-\sum_{k=0}^{n-4}(3n-3k-13)(2k+1)\binom{2k}{k}2^{n-k-3}. \tag{4.8}$$

The proof of part (i) of Proposition 2.2 is similar and left to the reader.

For the higher-order moments, the Tauberian theorem can be used with Lemma 2.1 to show for each $r \geq 1$,

$$E[S_{2,n}]_r \sim (\frac{n}{4})^r, \qquad n \to \infty \tag{4.9}$$

$$E[L_{2,n}]_r \sim (\frac{n}{2})^r, \qquad n \to \infty. \tag{4.10}$$

Following is how we obtain (4.10) from Lemma 2.1 and the above. The case of (4.9) is similar. When the Tauberian theorem is applied, we need only consider the terms having highest power of $(1-4t)$ in the dominators of (4.4) and, by an induction argument applied to (4.4), we may check that $\hat{\mu}_r(t) = \sum_{n=1}^{\infty}\overline{\mu}_n^{[r]}t^n$ is $O((1-4t)^{-(2r-1)/2})$ as $t \to \frac{1}{4}$. With this observation, we are ready to show (4.10). Again by induction, suppose that (4.10) is true for $r \leq k$; so by the Tauberian theorem, we have

$$\overline{\mu}_n^{[k]} \sim \frac{1}{\Gamma(k-\frac{1}{2})}n^{k-(3/2)}\Lambda_k(n)2^{2n}, \tag{4.11}$$

where the last term on the right come from a change of variable of the form $u = 4t$.

Since $\overline{\mu}_n^{[r]} = |\Omega_n|\mu_n^{[r]}$, and by (2.1), we have

$$\Lambda_k(n) = \frac{1}{4\sqrt{\pi}}\Gamma(k-\frac{1}{2})(\frac{1}{2})^k. \tag{4.12}$$

Now for $r = k+1$, we have

$$\overline{\mu}_n^{[k+1]} \sim \frac{1}{\Gamma(k+\frac{1}{2)})}k^{n-1/2}\Lambda_{k+1}(n)2^{2n}. \tag{4.13}$$

By (4.4), we have

$$\Lambda_{k+1}(n) = \sum_{j=0}^{k+1}\binom{k+1}{j}\Lambda_{k+1-j}(n)\Lambda_j(n). \tag{4.14}$$

By (4.12) and Lemma 2.1, (4.14) is readily simplified. Plug this into (4.13) and the result follows.

5. Proof of main result. Consider first $\psi_n(\xi)$. Note that

$$\psi_1(\xi) = 1, \qquad \psi_2(\xi) = \psi_3(\xi) = e^{\xi}. \tag{5.1a}$$

We will show that

$$\psi_n(\xi) = \frac{(2n-1)}{\binom{2n-1}{n}} \sum_{k=0}^{[\frac{n}{2}]} \binom{n-k}{k} \frac{1}{2(n-k)-1} \binom{2n-2k-1}{n-k} (e^{\xi}-1)^k. \tag{5.1b}$$

Let $a_n(\xi) = |\Omega_n|\psi_n(\xi)$; then $a_1(\xi) = 1, a_2(\xi) = e^{\xi}, a_3(\xi) = 2e^{\xi}$. Also, by (3.7b),

$$a_n(\xi) = \sum_{m=1}^{n-1} a_m(\xi) a_{n-m}(\xi).$$

Then, $\hat{a}(s,\xi) = \sum_{n=1}^{\infty} a_n(\xi)s^n$ satisfies

$$\hat{a}(s,\xi) = s + s^2(e^{\xi}-1) + \sum_{m=1}^{\infty}\sum_{n=m+1}^{\infty} a_m(\xi)s^m a_{n-m}(\xi)s^{n-m}.$$

Thus,

$$\begin{aligned}
\hat{a}(s,\xi) &= \frac{1}{2}\{1 - \sqrt{1-4s-4s^2(e^{\xi}-1)}\} \\
&= \frac{1}{2}\sum_{n=1}^{\infty} \binom{\frac{1}{2}}{n} (-1)^{n+1}(4s)^n(1+s(e^{\xi}-1))^n \\
&= \frac{1}{2}\sum_{n=1}^{\infty}\sum_{k=0}^{\infty} \binom{\frac{1}{2}}{n} (-1)^{n+1}(4)^n \binom{n}{k} (e^{\xi}-1)^k s^{n+k} \\
&= \frac{1}{2}\sum_{m=1}^{\infty} \sum_{\{(n,k):n+k=m, n\geq k\wedge 1\}} \binom{\frac{1}{2}}{n} (-1)^{n+1}(4)^n \binom{n}{k} (e^{\xi}-1)^k s^m. \qquad (5.2)
\end{aligned}$$

Then we have

$$a_m(\xi) = \frac{1}{2} \sum_{\{(n,k):n+k=m, n\geq k\wedge 1\}} \binom{\frac{1}{2}}{n}\binom{n}{k} (-1)^{n+1}(4)^n(e^{\xi}-1)^k. \tag{5.3}$$

Thus,

$$\psi_n(\xi) = \frac{1}{2|\Omega_n|} \sum_{\{(j,k):j+k=n, j\geq k\wedge 1\}} \binom{\frac{1}{2}}{j}\binom{j}{k} (-1)^{j+1}(4)^j(e^{\xi}-1)^k.$$

Note that

$$\binom{\frac{1}{2}}{j} = \frac{2^{-2j+1}(-1)^{j-1}(2j-2)!(2j-1)}{j!(j-1)!(2j-1)}.$$

Thus,

$$\psi_n(\xi) = \frac{1}{|\Omega_n|} \sum_{\{(j,k):j+k=n,j\geq k\wedge 1\}} \binom{j}{k} \frac{1}{2j-1} \binom{2j-1}{j} (e^{\xi}-1)^k. \tag{5.4}$$

By change of variable again, we get (5.1b).

Now using (2.1) and (5.1b), we have

$$\psi_n(\xi) = \sum_{k=0}^{[n/2]} \frac{(n-k)!|\Omega_{n-k}|(e^{\xi}-1)^k}{(n-2k)!|\Omega_n|k!}, \tag{5.5}$$

with

$$\frac{|\Omega_{n-k}|}{|\Omega_n|} \sim 4^{-k}(1-\frac{k}{n})^{-3/2}, \quad k \leq \frac{n}{2}, n \to \infty. \tag{5.6}$$

In particular, (5.5) may be expressed as

$$\psi_n(\xi) = \sum_{k=0}^{[n/2]} \frac{(1-\frac{k}{n})^{-\frac{3}{2}}(1-\frac{k}{n})\cdots(1-\frac{(2k-1)}{n})(\frac{n}{4}e^{\xi}-1)^k}{k!}. \tag{5.7}$$

The appropriate choice of scaling and the computation of the asymptotic variance as

$$a_n = n \quad \text{and} \quad \sigma^2 = \varphi''_{\infty}(0) \tag{5.8}$$

can be determined very simply from (5.7) using (2.2). In particular,

$$\sigma^2 = \lim_{n\to\infty} \varphi''_{\infty}(0) = \frac{1}{16}. \tag{5.9}$$

Taking $a_n = n$ in the application of Theorem 3.3, we may check that

$$\lim_{n\to\infty} \varphi_n(\xi) = \frac{\xi}{4} + \log(\cosh(\frac{\xi}{4})) = \varphi_{\infty}(\xi). \tag{5.10}$$

This computation in a neighborhood of zero can be made by a saddle point method. In particular, for $\xi \geq 0$ we need only to calculate the maximum term of the sum for k ranging between 0 and $\frac{n}{2}$, and for $\xi < 0$ we calculate the maximum difference between pairwise successive terms $2k$ and $2k+1$ with k ranging from 1 to $\frac{n}{4}$ (which nicely factors). The parameters $\alpha_- = 0$ and $\alpha_+ = \frac{1}{2}$ and the computation of the Legendre transform as

$$I(y) = (4y-1)\tanh^{-1}(4y-1) - \log(\cosh(\tanh^{-1}(4y-1))) \tag{5.11}$$

follow. We now apply Theorem 3.3 to get both the large deviation probabilities and the central limit theorem.

Acknowledgements. We learned about the connections with classical combinatorial identities and the first method of proof of Lemma 2.1 from Otto G. Ruehr. We also thank Rabi Bhattacharya for some helpful discussions.

REFERENCES

[1] A.V. Aho, R. Sethi, and J. D. Ullman, *Compilers*, Addison–Wesley, Reading, MA, 1986.

[2] E. A. Bender, *Central and local limit theorems applied to asymptotic enumeration,* J. Combin. Theory (A), 15 (1973), pp. 91–111.

[3] M. Berry and P. M. Bradley, *The application of network analysis to the study of brannching patterns of large dendritic fields,* Brain Research, 109 (1976), pp. 111–132.

[4] R. Borchert and N. A. Slade, *Bifurcation ratios and adaptive geometry of trees,* Bot. Gaz.,142 (1981), pp. 394–401.

[5] L. Carlitz, D. C. Kurtz, R. Scoville, and O. P. Stackelberg, *Asymptotic properties of Eulerian numbers,* Z. Wahr. Verw. Geb., 23(1972), pp. 47–54.

[6] G. Chartrand and L. Lesniak, *Graphs and DiGraphs*, Second edition, Wadsworth, Monterey, CA, 1986.

[7] T. Cox and D. Griffeath, *Large deviations for Poisson systems of independent random walks,* Z. Wahr. Verw. Geb., 66 (1984), pp. 543–558.

[8] J.D. Deuschel and D. W. Stroock, *Large Deviations*, Academic Press, Boston, 1989.

[9] R. S. Ellis, *Entropy, Large Deviations, and Statistical Mechanics* Springer-Verlag, Berlin, 1985.

[10] P. Flajolet and H. Prodinger, *Register allocation for unary-binary trees,* SIAM J. Comput., 13 (1984), pp. 629–640.

[11] P. Flajolet and A.M. Odlyzko, *Limit distributions for coefficients of iterates of polynomials with applications to combinatorial enumerations,* Math. Proc. Cambridge Philos. Soc., 96 (1984), pp. 237–253.

[12] P. Flajolet, J. C. Raoult, and J. Vuillemin, *The number of registers recquired for evaluating arithmetic expressions,* Theoret. Comput. Sci., 9 (1979), pp. 99–125.

[13] H. W. Gould, *Combinatorial Identities*, Morgantown Press, Morgantown, WV, 1972.

[14] V. K. Gupta and E. Waymire, *The spatial geometry of random networks and a problem in river basin hydrology,* AMS-IMS-SIAM Conference on Spatial Statistics and Imaging, 1990.

[15] ———, *On the formulation of an analytical approach to hydrologic response and similarity at the basin scale,* J. Hydrol., 65 (1983) pp. 95–123.

[16] L.H. Harper, *Stirling behavior is asymptotically normal,* Ann. Math. Stat., 38 (1967) pp. 410–414.

[17] K. Horsfield, *Are diameter, length, and branching ratios meaningful in the lung?*, J. Theoret. Biol., 87 (1980), pp. 773–784.

[18] ———, *The structure of the tracheobronchial tree,* in Scientific Foundations of Respitory Medicine, J. G. Scadding, G. Cumming, and W. M. Thurlbeck, eds., Heinemann, London, 1981, pp. 54–77.

[19] R. Horton, *Erosional development of streams and their drainage basins: Hydrophysical approach to quantitative morphology,* Bull. Geol. Soc. Amer., 56, (1945), pp. 275–370.

[20] R.S. Jarvis and M.J. Woldenberg, *River Networks* Benchmark Papers in Geology, 80, Hutchinson Ross, Stroudsburg, PA, 1984.

[21] L. B. Leopold, *Rivers,* American Scientist, 50 (1962), pp. 511–537.

[22] A. Meir, J. W. Moon, and J. R. Pounder, *On the order of random channel networks,* SIAM J. Alg. Disc. Meth., 1 (1980), pp. 25–33.

[23] O. Mesa, *Analysis of channel networks parameterized by elevation,* Ph.D. Dissertation, Dept. of Civil Engineering, University of Mississippi, University, MS, 1986.

[24] R. L. Shreve, *Statistical law of stream numbers,* J. Geol., 74 (1966), pp. 17–37.

[25] D. A. Steingraeber, L. J. Kascht, and D. H. Franck, *Variation of shoot morphology and bifurcation ratio in sugar maple (Acer Saccarum) saplings,* Amer. J. Botany, 66 (1979), pp. 441–445.

[26] M. Vauchaussade and G. Viennot, *Enumeration of* RNA*'s secondary structures by complexity,* in Proc. Inter. Conference Math. in Medicine and Biology, Bary, Italy, 1983.

[27] C. Werner, *Two models for Horton's law of stream numbers,* Canadian Geographer, 16(1), pp. 50–68.

SIAM J. DISC. MATH.
Vol. 4, No. 4, p. 589, November 1991

011

ERRATUM: AN INTERGER PROGRAM FOR CODES*

MARTIN DOWD†

Version 3 of the conjecture is false, by Fisher's inequality, since b_0 can be small compared to v. Indeed, choose an integer $a > 1$, and let $v = (a^2 - 1)(a - 1)$ and $k = a(a-1)$; then $b_0 = v/(a-1)$. It might be true that $b_1 \leq K_1 b_0$ for $b_0 \geq K_2 v$, else $b_1 \leq K_3 v$.

* Received by the editors May 14, 1991; accepted for publication June 12, 1991. SIAM J. Disc. Math., 4(1991), pp. 293–303.

† 2532 Orange Avenue, Costa Mesa, California 92627.